Brian Aitken, April 2024

Books and Borrowing Developer Blog Posts

The following text relating to the development of the systems for the Books and Borrowing project has been extracted from more general blog posts published at https://digital-humanities.glasgow.ac.uk/blog/

13th January 2020

On Friday afternoon I met with Matt Sangster and Katie Halsey to discuss their Books and Borrowers project. This is a major AHRC project that I helped write the proposal for. We heard before Christmas that the project has been funded, which is excellent news, so we met this week to discuss our next steps. The project doesn't actually start until June, but I'm going to try and get some of the technical aspects in place before then in order to allow the project's RAs to get started straight away. It's all very exciting and hopefully it will be a great project to work on when the time comes.

3rd February 2020

I spent some time this week going through the sample data that Katie Halsey had sent me from a variety of locations for the Books and Borrowing project. I went through all of the sample data and compiled a list of all of the fields found in each. This is a first step towards identifying a core set of fields and of mapping the analogous fields across different datasets. I also included the GU students and professors from Matthew's pilot project but I have not included anything from the images from Inverness as deciphering the handwriting in the images is not something I can spend time doing. With this mapping document in place I can now think about how best to store the different data recorded at the various locations in a way that will allow certain fields to be cross-searched.

23rd March 2020

I had a conference call with Katie Halsey and Matt Sangster about the Books and Borrowers project, which is due to start at the beginning of June. It was my first time using the Zoom videoconferencing software and it worked very well, other than my cat trying to participate several times. We had a good call and made some plans for the coming weeks and months. I'm going to try and get an initial version of the content management system and database for the project in place before the official start of the project so that the RAs will be able to use this straight away. This is of even greater importance now as they are likely to be limited in the kinds of research activities they can do at the start of the project because of travel restrictions and will need to work with digital materials.

30th March 2020

I had a useful conference call with the PI Katie Halsey and Co-I Matt Sangster last week, and the main outcome of that meeting for me was that I'd further expand upon the data design document I'd previously started in order to bring it into line with our understanding of the project's requirements. This involved some major reworking of the entity-relationship diagram I had previously designed based on my work with the sample datasets, with the database structure increasing from 11 related tables to 21, incorporating a new system to trace books

and their authors across different libraries, to include borrower cross-references and to greatly increase the data recorded about libraries. I engaged in many email conversations with Katie and Matt over the course of the week as I worked on the document, and on Friday I sent them a finalised version consisting of 34 pages and more than 7,000 words. This is still in 'in progress' version and will no doubt need further tweaks based on feedback and also as I build the system, but I'd say it's a pretty solid starting point. My next step will be to add a new section to the document that describes the various features of the content management system that will connect to the database and enable to project's RAs to add and edit data in a streamlined and efficient way.

13th April 2020

For the Books and Borrowers project I spent some time downloading and looking through the digitised and transcribed borrowing registers of St. Andrews. They have made three registers from the second half of the 18th century available via a Wiki interface (see https://arts.st-andrews.ac.uk/transcribe/index.php?title=Main_Page) and we were given access to all of these materials that had been extracted and processed by Patrick McCann, who I used to work very closely with back when we were both based at HATII and worked for the Digital Curation Centre. Having looked through the materials it's clear that we will be able to use the transcriptions, which will be a big help. The dates will probably need to be manually normalised, though, and we will need access to higher resolution images than the ones we have been given in order to make a zoom and pan interface using them.

20th April 2020

I spent the bulk of the week working on the Books and Borrowing project. Katie and Matt got back with some feedback on the data description document that I completed and sent to them before Easter and I spent some time going through this feedback and making an updated version of the document. After sending the document off I started working on a description of the content management system. This required a lot of thought and planning as I needed to consider how all of the data as defined in the document would be added, edited and deleted in the most efficient and easy to use manner. By the end of the week I'd written some 2,500 words about the various features of the CMS, but there is still a lot to do. I'm hoping to have a version completed and sent off to Katie and Matt early next week.

27th April 2020

I spent the majority of the week continuing to work on the requirements document for the Books and Borrowing project. As I worked through this I returned to the database design and made some changes as my understanding of the system increased. This included adding in a new field for the original transcription and a new 'order on page' field to the borrowing table as I realised that without such a column it wouldn't be possible for an RA to add a new record anywhere other than after the last record on a page. It's quite likely that an RA will accidentally skip a record and might need to reinstate it, or an RA might intentionally want to leave out some records to return to later. The 'order on page' column (which will be automatically generated but can be manually edited) will ensure these situations can be handled.

As I worked through the requirements I began to realise that the amount of data the RAs may have to compile for each borrowing record is possibly going to be somewhat overwhelming.

Much of it is optional, but completing all the information could take a long time: creating a new Book Holding and Item record, linking it to an Edition and Work or creating new records for these, associating authors or creating new authors, adding in genre information, creating a new borrower record or associating an existing one, adding in occupations, adding in cross references to other borrowers, writing out a diplomatic transcription, filling in all of the core fields and additional fields. That's a huge amount to do for each record and we may need to consider what is going to be possible for the RAs to do in the available time.

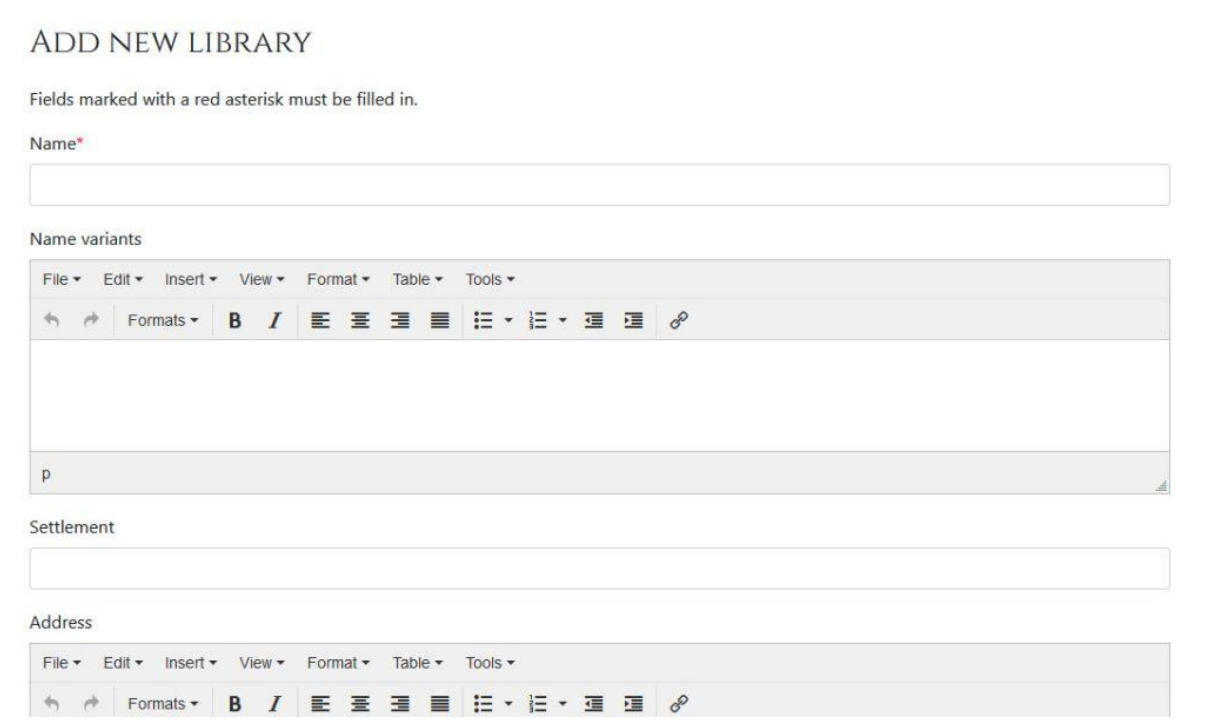
By the end of Tuesday I had finished working on a first version of the requirements document, weighing in at more than 11,000 words, and sent it on to Katie and Matt for feedback. We have agreed to meet (via Zoom) next Tuesday to discuss any changes to the document. During the rest of the week I began to develop the systems for the project. This included implementing the database (creating each of the 23 tables that will be needed to store the project's data) and installing and configuring WordPress, which will be used to power the simple parts of the project website.

4th May 2020

I spent much of the four working days on the development of the content management system for the Books and Borrowing project. The project RAs will start using the system in June and I'm aiming to get everything up and running before then so this is my main focus at the moment. I also had a Zoom meeting with project PI Katie Halsey and Co-I Matt Sangster on Tuesday to discuss the requirements document I'd completed last week and the underlying data structures I'd defined in the weeks before. Both Katie and Matt were very happy with the document, although Matt had a few changes he wanted made to the underlying data structures and the CMS. I made the necessary changes to the data design / requirements document and the project's database that I'd set up last week. The changes were:

Borrowing spans have now been removed from libraries and these will instead be automatically inferred based on the start and end dates of ledger records held in these libraries. Ledgers now have a new 'ledger type' field which currently allows the choice of 'Professorial', 'Student' or 'Town'. This field will allow borrowing spans for libraries to be altered based on a selected ledger type. The way occupations for borrowers is recorded has been updated to enable both original occupations from the records and a normalised list of occupations to be recorded. Borrowers may not have an original occupation but still might have a standardised occupation so I've decided to use the occupations table as previously designed to hold information about standardised occupations. A borrower may have multiple standardised occupations. I have also added a new 'original occupation' field to the borrower record where any number of occupations found for the borrower in the original documentation (e.g. river watcher) can be added if necessary. The book edition table now has an 'other authority URL' field and an 'other authority type' field which can be used if ESTC is not appropriate. The 'type' currently features 'Worldcat', 'CERL' and 'Other' and 'Language' has been moved from Holding to Edition. Finally, in Book Holding the short title is now original title and long title is now standardised title while the place and date of publication fields have been removed as the comparable fields at Edition level will be sufficient.

In terms of the development of the CMS, I created a Bootstrap-based interface for the system, which currently just uses the colour scheme I used for Matt's pilot 18th Century Borrowing project. I created the user authentication scripts and the menu structure and then started to create the actual pages. So far I've created a page to add a new library record and all of the information associated with a library, such as any number of sources. I then created the facility to browse and delete libraries and the main 'view library' page, which will act as a hub through which all book and borrowing records associated with the library will be managed. This page has a further tab-based menu with options to allow the RA to view / add ledgers, additional fields, books and borrowers, plus the option to edit the main library information. So far I've completed the page to edit the library information and have started work on the page to add a ledger. I'm making pretty good progress with the CMS, but there is still a lot left to do. Here's a screenshot of the CMS if you're interested in how it looks:



11th May 2020

I spent the majority of it working on the content management system for the Books and Borrowing project. The project is due to begin at the start of June and I'm hoping to have the CMS completed and ready to use by the project team by then, although there is an awful lot to try and get into place. I can't really go into too much detail about the CMS, but I have completed the pages to add a library and to browse a list of libraries with the option of deleting a library if it doesn't have any ledgers. I've also done quite a lot with the 'View library' page. It's

possible to edit a library record, add a ledger and add / edit / delete additional fields for a library. You can also list all of the ledgers in a library with options to edit the ledger, delete it (if it contains no pages) and add a new page to it. You can also display a list of pages in a ledger, with options to edit the page or delete it (if it contains no records). You can also open a page in the ledger and browse through the next and previous pages.

I've been trying a new approach with the CMS for this project, involving more in-page editing. For example, the list of ledgers is tabular based with fields for things like the number of pages, the ledger name and its start and end dates. When the 'edit' button is pressed on rather than taking the user away from this page to a separate page, the row in the table becomes editable. This approach is rather more complicated to develop and relies a lot more on JavaScript, but it seems to be working pretty well. It was further complicated by having textareas that use the TinyMCE text editing tool, which then needs to be reinitiated when the editable boxes load in. Also, you can't have multiple forms within a table in HTML, meaning there can be only one form wrapped around the whole table. Initially I was thinking that when the row became editable the JavaScript would add in form tags in the row too, but this approach doesn't work properly so instead I've just had to implement a single form with its type controlled by hidden inputs that change when a row is selected. The situation is complicated as it's not just the ledger record that needs to be edited from within the table, but there are also facilities to add and edit ledger pages, which also need to use the same form.

At the moment I'm in the middle of creating the facility to add a new borrowing record to the page. This is the most complex part of the system as a record may have multiple borrowers, each of which may have multiple occupations, and multiple books, each of which may be associated with higher level book records. Plus the additional fields for the library need to be taken into consideration too. By the end of the week I was at the point of adding in an auto-complete to select an existing borrower record and I'll continue with this on Monday.

18th May 2020

I spent week 9 of Lockdown continuing to implement the content management system for the Books and Borrowing project. I was originally hoping to have completed an initial version of the system by the end of this week, but this was unfortunately not possible due to having to juggle work and home-schooling, commitments to other projects and the complexity of the project's data. It took several days to complete the scripts for uploading a new borrowing record due to the interrelated nature of the data structure. A borrowing record can be associated with one or more borrowers, and each of these may be new borrower records or existing ones, meaning data needs to be pulled in via an autocomplete to prepopulate the section of the form. Books can also be new or existing records but can also have one or more new or existing book item records (as a book may have multiple volumes) and may be linked to one or more project-wide book edition records which may already exist or may need to be created as part of the upload process, and each of these may be associated with a new or existing top-level book work record. Therefore the script for uploading a new borrowing record needs to incorporate the 'add' and 'edit' functionality for a lot of associated data as well. However, as I have implemented all of these aspects of the system now it will make it quicker and easier to develop the dedicated pages for adding and editing borrowers and the various book levels once

I move onto this. I still haven't working on the facilities to add in book authors, genres or borrower occupations, which I intend to move onto once the main parts of the system are in place.

After completing the scripts for processing the display of the 'add borrowing' form and the storing of all of the uploaded data I moved onto the script for viewing all of the borrowing records on a page. Due to the huge number of potential fields I've had to experiment with various layouts, but I think I've got one that works pretty well, which displays all of the data about each record in a table split into four main columns (Borrowing, Borrower, Book Holding / Items, Book Edition / Works). I've also added in a facility to delete a record from the page. I then moved on to the facility to edit a borrowing record, which I've added to the 'view' page rather than linking out to a separate page. When the 'edit' button is pressed on for a record its row in the table is replace with the 'edit' form, which is identical in style and functionality to the 'add' form, but is prepopulated with all of the record's data. As with the 'add' form, it's possible to associated multiple borrowers and book items and editions, and also to manage the existing associations using this script. The processing of the form uses the same logic as the 'add' script so thankfully didn't require much time to implement.

What I still need to do is add authors and borrower occupations to the 'view page', 'add record' and 'edit record' facilities, add the options to view / edit / add / delete a library's book holdings and borrowers independently of the borrowing records, plus facilities to manage book editions / works, authors, genres and occupations at the top level as opposed to when working on a record. I also still need to add in the facilities to view / zoom / pan a page image and add in facilities to manage borrower cross-references. This is clearly quite a lot, but the core facilities of adding, editing and deleting borrowing, borrower and book records is now in place, which I'm happy about. Next week I'll continue to work on the system ahead of the project's official start date at the beginning on June.

25th May 2020

This week I continued to focus on the development of the content management system for the Books and Borrowing project. On Tuesday I had a Zoom meeting to demonstrate the system as it currently stands to the project PI Katie Halsey and Co-I Matt Sangster. Monday was a bank holiday but I decided to work it and take the day off at a later date in order to prepare a walkthrough and undertake a detailed testing of the system, which uncovered a number of bugs that I then tracked down and fixed. My walkthrough went through all of the features that are so far in place: creating, editing and deleting libraries, viewing libraries, adding ledgers and additional fields to libraries, viewing, editing and deleting these ledgers and additional fields, adding pages to ledgers, editing and deleting them, viewing a page, the automated approach to constructing navigation between pages, viewing records on pages and then the big thing: adding and editing borrowing records. This latter process can involve adding data about the borrowing (e.g. lending date), one or more borrowers (which may be new borrowers or ones already in the system), a new or existing book holding, which may consist of one or more book items (e.g. volumes 1 and 3 of a book) and may be connected to one or more new or existing project-wide book edition records which may have a new or existing top-level book work record.

The walkthrough via Zoom went well, with me sharing my screen with Katie and Matt so they could follow my actions as I used the CMS. I was a bit worried they would think the add / edit borrowing record form would be too complicated but although it does look rather intimidating, most of the information is optional and many parts of it will be automatically populated by linking to existing records via autocomplete drop-downs, so once there is a critical mass of existing data in the system (e.g. existing book and borrower records) the process of adding new borrowing records will be much quicker and easier.

The only major change that I needed to make following the walkthrough was to add a new 'publication end date' field to book edition and book work records as some books are published in parts over multiple years (especially books comprised of multiple volumes). I implemented this after the meeting and then spent most of the remainder of the week continuing to implement further aspects of the CMS. I made a start on the facility to view a list of all book holding records that have been created for a library, through which the project team will be able to bring up a list of all borrowing records that involve the book. I got as far as getting a table listing the book holdings in place, but as the project team will be started next week I figured it would make more sense to try and tackle the last major part of the system that still needed to be implemented: creating and associating author records with the four levels of book record.

A book may have any number of authors and their associations with a book record cascades down through the levels. For example, if an author is associated with a book via its top-level 'book work' record then the author will automatically be associated with a related 'book edition' record, any 'book holding' records this edition is connected to and any 'book item' records belonging to the book holding. But we need to be able to associate an author not just with 'book works' but with any level of book record, as a book may have a different author at one of these levels (e.g. a particular volume may be attributed to a different author) or the same author may be referred to by a different alias in a particular edition. Therefore I had to update the already complicated add / edit borrowing record form to enable authors to be created, associated and disassociated with any book level. Plus I needed to add in an autocomplete facility to enable authors already in the system to be attached to records and to ensure that the author sections clear and reset themselves if the user removes the book from the borrowing record. It took a long time to implement this system, but by the end of the week I'd got an initial version working. It will need a lot of testing and no doubt some fixing next week, but it's a relief to get this major part of the system in place. I also added in a little feature that keeps the user's CMS session going for as long as the browser is on a page of the CMS, which is very important as the complicated forms may take a long time to complete and it would be horrible if the sessions timed out before the user was able to submit the form.

1st June 2020

On Tuesday we had a Zoom call with all six members of the core project team, during which I demonstrated the CMS as it currently stands. This gave me an opportunity to demonstrate the new Author association facilities I had created last week. The demonstration all went very smoothly and I think the team are happy with how the system works, although no doubt once they actually begin to use it there will be bugs to fix and workflows to tweak. I also spent some

time before the meeting testing the system again, and fixing some issues that were not quite right with the author system.

I spent the remainder of my time on the project completing work on the facility to add, edit and view book holding records directly via the library page, as opposed to doing so whilst adding / editing a borrowing record. I also implemented a similar facility for borrowers as well. Next week I will begin to import some of the sample data from various libraries into the system and will allow the team to access the system to test it out.

8th June 2020

My main task was importing sample data into the content management system. Matt had sent me the latest copy of the Glasgow Student data over the weekend, and once I had the data processing scripts from the PC at work I could then process his spreadsheet and upload it to the pilot project database. Processing the Glasgow Student data was not entirely straightforward as the transcriber had used Microsoft Office formatting in the spreadsheet cells to replicate features such as superscript text and strikethroughs. It is a bit of a pain to export an Excel spreadsheet as plain text while retaining such formatting, but thankfully I'd solved that issue previously and my script was able to take an Excel file that had been saved as HTML and then pick out the formatting to keep whilst ditching all of the horrible HTML formatting that Microsoft adds in to Office files that are saved in that format.

Once the Glasgow Student data had been uploaded to the pilot project website I could then migrate it to the Books and Borrowing data structure. It took the best part of a day to write a script that processed the data, dealing with issues like multiple book levels, additional fields and generating ledgers and pages. After the migration there were 3 ledgers, 403 pages and 8191 borrowing records, with associations to 832 borrowers and 1080 books. With this in place I then began to import sample data from a previous study of Innerpeffray library. This was also in a spreadsheet, but was structured very differently and I needed to write a separate data import script to process it. There were some additional complications due to the character encoding the spreadsheet uses, that resulting in lots of hidden special characters being embedded in the text when the spreadsheet was converted to a plain text file for upload. This really messed up the upload process and took some time to get to the bottom of. Also, there is variation in page numbering (e.g. sometimes '3r', sometimes '3 r') and this resulted in multiple pages being created for each variation before I spotted the issue. Also, the spreadsheet is not always listed in page order – there were records from earlier pages added in amongst later pages. This also messed up the upload process before I spotted the issue and updated my script to take this into consideration. There were also some issues of data failing to upload when it contained accented characters, but I think I got to the bottom of that.

As with the Glasgow data, I created editions from holdings. I did add in a check to see whether any of the Glasgow editions matched the titles of the Innerpeffray titles, and used the existing Glasgow edition if this situation arose, but due to the differences in transcription I don't think any existing editions have been used. This will need some manual correction at some point. Similarly, there may be some existing Glasgow authors that might be used rather than repeating

the same information from Innerpeffray but due to differences in transcription I don't think this will have happened either. As before, author data has for now just been uploaded into the 'surname' field and will need to be manually split up further and some Glasgow and Innerpeffray authors will need to be merged. For example, in the Glasgow data we have 'Cave, William, 1637-1713.' Whereas in Innerpeffray we have 'Cave, William, 1637-1713'. Because of the full stop at the end of the Glasgow author these have ended up being inserted as separate authors. After the upload process was complete there were 6550 borrowing records for Innerpeffray, split over 340 pages in one ledger. A total of 1017 unique borrowers and 840 unique book holdings were added to the library.

I created user accounts for the rest of the team to access the CMS and test things out once the sample data for these two libraries was in place. The project PI, Katie Halsey spotted an issue with the autocomplete for selecting an existing edition not working, so I spent some time investigating this. It turns out that there are more character encoding issues with the data that are resulting in the JSON file that is generated for use in the autocomplete failing to be valid. This is also happening with the AJAX script that populates the fields once an autocomplete option is selected. I only investigated this on Friday afternoon and didn't have time to fix it, but I'm hoping that next week if I fix the character encoding issues and ensure all line break characters are removed from the data then things will be ok.

15th June 2020

I spent most of my time on the Books and Borrowing project, as there is still a huge amount to do to get the project's systems set up. Last week I'd imported several thousand records into the database and had given the team access to the Content Management System to test things out. One thing that cropped up was that the autocomplete that is used for selecting existing books, borrowers and authors was sometimes not working, or if it did work on selection of an item the script that then populates all of the fields about the book, borrower or author was not working. I'd realised that this was because there were invisible line break characters (\n or \r) in the imported data and the data is passed to the autocomplete via a JSON file. Line break characters are not allowed in a JSON file and therefore the autocomplete couldn't access the data. I spent some time writing a script that would clean the data of all offending characters and after running this the autocomplete and pre-population scripts worked fine. However, a further issue cropped up with the text editors in the various forms in the CMS. These use the TinyMCE widget to allow formatting to be added to the text area, which works great. However, whenever a new line is created this adds in HTML paragraphs (' ', which is good) but the editor also adds a hidden line break character ('\r' or '\n' which is bad). When this field is then used to populate a form via the selection of an autocomplete value the line break makes the data invalid and the form fails to populate. After identifying this issue I managed ensured all such characters are stripped out of any uploaded data and that fixed the issue.

I had to spend some time fixing a few more bugs that the team had uncovered during the week. The 'delete borrower' option was not appearing, even when a borrower was associated with no records, and I fixed this. There was also an issue with autocompletes not working in certain situations (e.g. when trying to add an existing borrower to a borrowing record that was initially created without a borrower). I tracked down and fixed these. Another issue involved the record

page order incrementing whenever the record was edited, even when this had not been manually changed, while another involved book edition data not getting saved in some cases when a borrowing record was created. I tracked down and fixed these issues too.

With these fixes in place I then moved on to adding new features to the CMS, specifically facilities to add and browse the book works, editions and authors that are used across the project. Pressing on the 'Add Book' menu item nowloads a page through which you can choose to add a Book Work or a Book Edition (with associated Work, if required). You can also associate authors with the Works and Editions too. Pressing on the 'Browse Books' option now loads a page that lists all of the Book Works in a table, with counts of the number of editions and borrowing records associated with each. There's also a row for all editions that don't currently have a work. There are currently 1925 such editions so most of the data appears in this section, but this will change.

Through the page you can edit a work (including associating authors) by pressing on the 'edit' button. You can delete a work so long as it isn't associated with an Edition. You can bring up a list of all editions in the work by pressing on the eye icon. Once loaded, the editions are displayed in a table. I may need to change this as there are so many fields relating to editions that the table is very wide. It's usable if I make my browser take up the full width of my widescreen monitor, but for people using a smaller screen it's probably going to be a bit unwieldy. From the list of editions you can press the 'edit' button to edit one of them – for example assigning one of the 'no work' editions to a work (existing or newly created via the edit form). You can also delete an edition if it's not associated with anything. The Edition table includes a list of borrowing records, but I'll also need to find a way to add in an option to display a list of all of the associated records for each, as I imagine this will be useful.

Pressing on the 'Add Author' menu item brings up a form allowing a new author to be added, which will then be available to associate with books throughout the CMS, while pressing on the 'Browse Authors' menu item brings up a list of authors. At the moment this table (and the book tables) can't be reordered by their various columns. This is something else I still need to implement. You can delete an author if it's not associated with anything and also edit the author details. As with the book tables I also need to add in a facility to bring up a list of all records the author is associated with, in addition to just displaying counts. I also noticed that there seems to be a bug somewhere that is resulting in blank authors occasionally being generated, and I'll need to look into this.

I then spent some time setting up the project's server, which is hosted at Stirling University. I was given access details by Stirling's IT Support people and managed to sign into the Stirling VPN and get access to the server and the database. There was an issue getting write access to the server, but after that was resolved I was able to upload all of the CMS files, set up the WordPress instance that will be the main project website and migrate the database.

I was hoping I'd be able to get the CMS up and running on the new server without issue, but unfortunately this did not prove to be the case. It turns out that the Stirling server uses a different (and newer) version of the PHP scripting language than the Glasgow server and some of the functionality is different, for example on the Glasgow server you can call a function with less parameters than it is set up to require (e.g. addAuthor(1) when the function is set up to take 2 parameters (e.g.addAuthor(1,2)). The version on the Stirling server doesn't allow this and instead the script breaks and a blank page is displayed. It took a bit of time to figure out what was going on, and now I know what the issue is I'm going to have to go through every script and check how every function is called, and this is going to be my priority next week.

22nd June 2020

Last week I'd planned to migrate the CMS from my test server at Glasgow to the official project server at Stirling, but during the process some discrepancies between PHP versions on the servers meant that the code which worked fine at Glasgow was giving errors at Stirling. As mentioned in last week's post, on the Stirling server calling a function while passing less than the required number of variables resulted in a fatal error, plus database 'warnings' (e.g. an empty string rather than a numeric zero being inserted into an integer field) were being treated as fatal errors too. It took most of Monday to go through my scripts and identify all the places such issues cropped up, but by the end of the day I had the CMS set up and fully usable at Stirling and had asked the team to start using it.

I then spent some further time working on the public website for the project, installing a theme, working with fonts and colour schemes, selecting header images, adding logos to the footer and other such matters. I made six different versions of the interface and emailed screenshots to the team for comment. We all agreed on the interface and I then made some further tweaks to it, during which time team member Kit Baston was adding content to the pages. On Thursday the website went live and you can access it here: https://borrowing.stir.ac.uk/. Here's a screenshot too:

I also continued to make improvements to the CMS this week, adding new functionality to the pages for browsing book editions, book works and authors. The table of Book Works now includes a column listing the number of Holdings each Work is associated with and now includes the options of ordering the listed Works by any of the columns in the table. When a book work row is expanded and its associated editions loads in, this table also now features the number of holdings an edition is associated with and allows the table to be ordered by any of the columns. I then made the number of holdings and records listed for each Work and Edition a link (so long as the number is greater than 0). Pressing on the link brings up a popup that lists the holdings and records. Each item in the list features an 'eye' icon and pressing on this will take you to the record in question (either in the library's list of holdings or the page that the borrowing record appears on) with the page opening at the item in question.

I updated the 'browse authors' page in a similar way: added in the option of ordering the table by any of the columns and adding in counts of associated works, editions, holdings and items that are also now links that open up a popup containing all related items. Each of these feature an 'eye' icon and you can press on one of these to be taken to the record in question. Holdings and Items will open in the corresponding library's list of book holdings while works and editions will load the 'Browse books' page. Linking to an edition was a bit tricky as editions are dynamically loaded into the page via JavaScript when a book work row is expanded. I had to

pass variables to the page that flagged that one work should be open on page load, triggered the loading in of the editions and then scrolled the page to the correct location once the editions had loaded. If the edition has no work then the 'no work specified' section needs to open, which currently takes a long time due to there being 1911 such editions at present. There isn't currently a 'loading' icon or anything but things do load in the background and the page will eventually jump down to the correct place. I also fixed a bug whereby if you disassociated a book holding from a record the edition and work autocompletes stopped working for that record.

On Friday I had a Zoom call wit Project PI Katie Halsey and Co-I Matt Sangster to discuss my work on the project and to decide where I should focus my attention next. We agreed that it would be good to get all of the sample data into the system now, so that the team can see what's already there and begin the process of merging records and rationalising the data. Therefore I'll be spending a lot of next week writing import scripts for the remaining datasets.

29th June 2020

I continued to work on the Books and Borrowing project for much of this week, this time focussing on importing some of the existing datasets from previous transcription projects. I had previously written scripts to import data from Glasgow University library and Innerpeffray library, which gave us 14,738 borrowing records. This week I began by focussing on the data from St Andrews University library.

The St Andrews data is pretty messy, reflecting the layout and language of the original documents, so I haven't been able to fully extract everything and it will require a lot of manual correcting. However, I did manage to migrate all of the data to a test version of the database running on my local PC and then updated the online database to incorporate this data.

The data I've got are CSV and HTML representations of transcribed pages that come from an existing website with pages that look like this: https://arts.st-andrews.ac.uk/transcribe/index.php?title=Page:UYLY205_2_Receipt_Book_1748-1753.djvu/100. The links in the pages (e.g. Locks Works) lead through to further pages with information about books or borrowers. Unfortunately the CSV version of the data doesn't include the links or the linked to data, and as I wanted to try and pull in the data found on the linked pages I therefore needed to process the HTML instead.

I wrote a script that pulled in all of the files in the 'HTML' directory and processed each in turn. From the filenames my script could ascertain the ledger volume, its dates and the page number. For example 'Page_UYLY205_2_Receipt_Book_1748-1753.djvu_10.html' is ledger 2 (1748-1753) page 10. The script creates ledgers and pages, and adds in the 'next' and 'previous' page links to join all the pages in a ledger together.

The actual data in the file posed further problems. As you can see from the linked page above, dates are just too messy to automatically extract into our strongly structured borrowed and returned date system. Often a record is split over multiple rows as well (e.g. the borrowing record for 'Rollins belles Lettres' is actually split over 3 rows). I could have just grabbed each row and inserted it as a separate borrowing record, which would then need to be manually merged, but I figured out a way to do this automatically. The first row of a record always appears to have a code (the shelf number) in the second column (e.g. J.5.2 for 'Rollins') whereas subsequent rows that appear to belong to the same record don't (e.g. 'on profr Shaws order by' and 'James Key'). I therefore set up my script to insert new borrowing records for rows that have codes, and to append any subsequent rows that don't have codes to this record until a row with a code is reached again.

I also used this approach to set up books and borrowers too. If you look at the page linked to above again you'll see that the links through to things are not categorised – some are links to books and others to borrowers, with no obvious way to know which is which. However, it's pretty much always the case that it's a book that appears in the row with the code and it's people that are linked to in the other rows. I could therefore create or link to existing book holding records for links in the row with a code and create or link to existing borrower records for links in rows without a code. There are bound to be situations where this system doesn't quite work correctly, but I think the majority of rows do fit this pattern.

The next thing I needed to do was to figure out which data from the St Andrews files should be stored as what in our system. I created four new 'Additional Fields' for St Andrews as follows:

Original Borrowed date: This contains the full text of the first column (e.g. Decr 16)

Code: This contains the full text of the second column (e.g. J.5.2)

Original Returned date: This contains the full text of the fourth column (e.g. Jan. 5)

Original returned text: This contains the full date of the fifth column (e.g. 'Rollins belles Lettres V. 2d')

In the borrowing table the 'transcription' field is set to contain the full text of the 'borrowed' column, but without links. Where subsequent rows contain data in this column but no code, this data is then appended to the transcription. E.g. the complete transcription for the third item on the page linked to above is 'Rollins belles Lettres Vol 2^d on profr Shaws order by James Key'.

The contents of all pages linked to in the transcriptions are added to the 'editors notes' field for future use if required. Both the page URL and the page content are included, separated by a bar (|) and if there are multiple links these are separated by five dashes. E.g. for the above the notes field contains:

‘Rollins_belles_Lettres| Possibly: De la maniere d’enseigner et d’etuder les belles-lettres, Par raport à l’esprit & au coeur, by Charles Rollin. (A Amsterdam : Chez Pierre Mortier, M. DCC. XLV. [1745]) http://library.st-andrews.ac.uk/record=b2447402~S1

—— profr_Shaws| https://arts.st-andrews.ac.uk/biographical-register/data/documents/1409683484

—— James_Key| Possibly James Kay: https://arts.st-andrews.ac.uk/biographical-register/data/documents/1389455860

___‘

As mentioned earlier, the script also generates book and borrower records based on the linked pages too. I’ve chosen to set up book holding rather than book edition records as the details are all very vague and specific to St Andrews. In the holdings table I’ve set the ‘standardised title’ to be the page link with underscores replaced with dashes (e.g. ‘Rollins belles Lettres’) and the page content is stored in the ‘editors notes’ field. One book item is created for each holding to be used to link to the corresponding borrowing records.

For borrowers a similar process is followed, with the link added to the surname column (e.g. Thos Duncan) and the page content added to the ‘editors notes’ field (e.g. Possibly Thomas Duncan: https://arts.st-andrews.ac.uk/biographical-register/data/documents/1377913372 ’). All borrowers are linked to records as ‘Main’ borrowers.

During the processing I noticed that the fourth ledger had a slightly different structure to the others, with entire pages devoted to a particular borrower, whose name then appeared in a heading row in the table. I therefore updated my script to check for the existence of this heading row, and if it exists my script then grabs the borrower name, creates the borrower record if it doesn’t already exist and then links this borrower to every borrowing item found on the page. After my script had finished running we had 11147 borrowing records, 996 borrowers and 6395 book holding records for St Andrew in the system.

I then moved onto looking at the data for Selkirk library. This data was more nicely structured than the St Andrews data, with separate spreadsheets for borrowings, borrowers and books and borrowers and books connected to borrowings via unique identifiers. Unfortunately the dates were still transcribed as they were written rather than being normalised in any way, which meant it was not possible to straightforwardly generate structured dates for the records and these will need to be manually generated. The script I wrote to import the data took about a day to write, and after running it we had a further 11,431 borrowing records across two registers and 415 pages entered into our database.

As with St Andrews, I created book records as Holding records only (i.e. associated specifically with the library rather than being project-wide 'Edition' records. There are 612 Holding records for Selkirk. I also processed the borrower records, resulting in 86 borrower records being added. I added the dates as originally transcribed to an additional field named 'Original Borrowed Date' and the only other additional field is in the Holding records for 'Subject', that will eventually be merged with our 'Genre' when this feature becomes available.

Also this week I advised Katie on a file naming convention for the digitised images of pages that will be created for the project. I recommended that the filenames shouldn't have spaces in them as these can be troublesome on some operating systems and that we'd want a character to use as a delimiter between the parts of the filename that wouldn't appear elsewhere in the filename so it's easy to split up the filename. I suggested that the page number should be included in the filename and that it should reflect the page number as it will be written into the database – e.g. if we're going to use 'r' and 'v' these would be included. Each page in the database will be automatically assigned an auto-incrementing ID, and the only means of linking a specific page record in the database with a specific image will be via the page number entered when the page is created, so if this is something like '23r' then ideally this should be represented in the image filename.

Katie had wondered about using characters to denote ledgers and pages in the filename (e.g. 'L' and 'P') but if we're using a specific delimiting character to separate parts of the filename then using these characters wouldn't be necessary and I suggested it would be better to not use 'L' as a lower case 'l' is very easy to confuse with a '1' or a capital 'I' which might confuse future human users.

Instead I suggested using a '-' instead of spaces and a '_' as a delimiter and pointed out that we should ensure that no other non-alphanumeric characters are ever used in the filename – no apostrophes, commas, colons, semi-colons, ampersands etc and to make sure the '-' is really a minus sign and not one of the fancy dashes (–) that get created by MS Office. This shouldn't be an issue when entering a filename, but might be if a list of filenames is created in Word and then pasted into the 'save as' box, for example.

Finally, I suggested that it might be best to make the filenames entirely lower case, as some operating systems are case sensitive and if we don't specify all lower case then there may be variation in the use of case. Following these guidelines the filenames would look something like this:

jpg

dumfries-presbytery_2_3v.jpg

standrews-ul_9_300r.jpg

6th July 2020

I wrote a script to import data from Haddington, which took some time due to the large number of additional fields in the data (15 across Borrowers, Holdings and Borrowings), but are executing it resulted in a further 5,163 borrowing records across 2 ledgers and 494 pages being added, including 1399 book holding records and 717 borrowers.

I then moved onto the datasets from Leighton and Wigtown. Leighton was a much smaller dataset, with just 193 borrowing records over 18 pages in one ledger and involving 18 borrowers and 71 books. As before, I have just created book holding records for these (rather than project-wide edition records), although in this case there are authors for books too, which I have also created. Wigtown was another smaller dataset. The spreadsheet has three sheets, the first is a list of borrowers, the second a list of borrowings and the third a list of books. However, no unique identifiers are used to connect the borrowers and books to the information in the borrowings sheet and there's no other field that matches across the sheets to allow the data to be automatically connected up. For example, in the Books sheet there is the book 'History of Edinburgh' by author 'Arnot, Hugo' but in the borrowings tab author surname and forename are split into different columns (so 'Arnot' and 'Hugo' and book titles don't match (in this case the book appears as simply 'Edinburgh' in the borrowings). Therefore I've not been able to automatically pull in the information from the books sheet. However, as there are only 59 books in the books sheet it shouldn't take too much time to manually add the necessary data when created Edition records. It's a similar issue with Borrowers in the first sheet – they appear with name in one column (e.g. 'Douglas, Andrew') but in the Borrowings sheet the names are split into separate forename and surname columns. There are also instances of people with the same name (e.g. 'Stewart, John') but without unique identifiers there's no way to differentiate these. There are only 110 people listed in the Borrowers sheet, and only 43 in the actual borrowing data, so again, it's probably better if any details that are required are added in manually.

I imported a total of 898 borrowing records for Wigtown. As there is no page or ledger information in the data I just added these all to one page in a made-up ledger. It does however mean that the page can take quite a while to load in the CMS. There are 43 associated borrowers and 53 associated books, which again have been created as Holding records only

and have associated authors. However, there are multiple Book Items created for many of these 53 books – there are actually 224 book items. This is because the spreadsheet contains a separate 'Volume' column and a book may be listed with the same title but a different volume. In such cases a Holding record is made for the book (e.g. 'Decline and Fall of Rome') and an Item is made for each Volume that appears (in this case 12 items for the listed volumes 1-12 across the dataset). With these datasets imported I have now processed all of the existing data I have access to, other than the Glasgow Professors borrowing records, but these are still being worked on.

I did some other tasks for the project this week as well, including reviewing the digitisation policy document for the project, which lists guidelines for the team to follow when they have to take photos of ledger pages themselves in libraries where no professional digitisation service is available. I also discussed how borrower occupations will be handled in the system with Katie.

13th July 2020

I returned to adding features to the content management system, after spending recent weeks importing datasets. I added a number of indexes to the underlying database which should speed up the loading of certain pages considerably. E.g. the browse books, borrowers and author pages. I then updated the 'Books' tab when viewing a library (i.e. the page that lists all of the book holdings in the library) so that it now lists the number of book holdings in the library above the table. The table itself now has separate columns for all additional fields that have been created for book holdings in the library and it is now possible to order the table by any of the headings (pressing on a heading a second time reverses the ordering). The count of 'Borrowing records' for each book in the table is now a button and pressing on it brings up a popup listing all of the borrowing records that are associated with the book holding record, and from this pop-up you can then follow a link to view the borrowing record you're interested in. I then made similar changes to the 'Borrowers' tab when viewing a library (i.e. the page that lists all of the borrowers the library has). It also now displays the total number of borrowers at the top. This table already allowed the reordering by any column, so that's not new, but as above, the 'Borrowing records' count is now a link that when clicked on opens a list of all of the borrowing records the borrower is associated with.

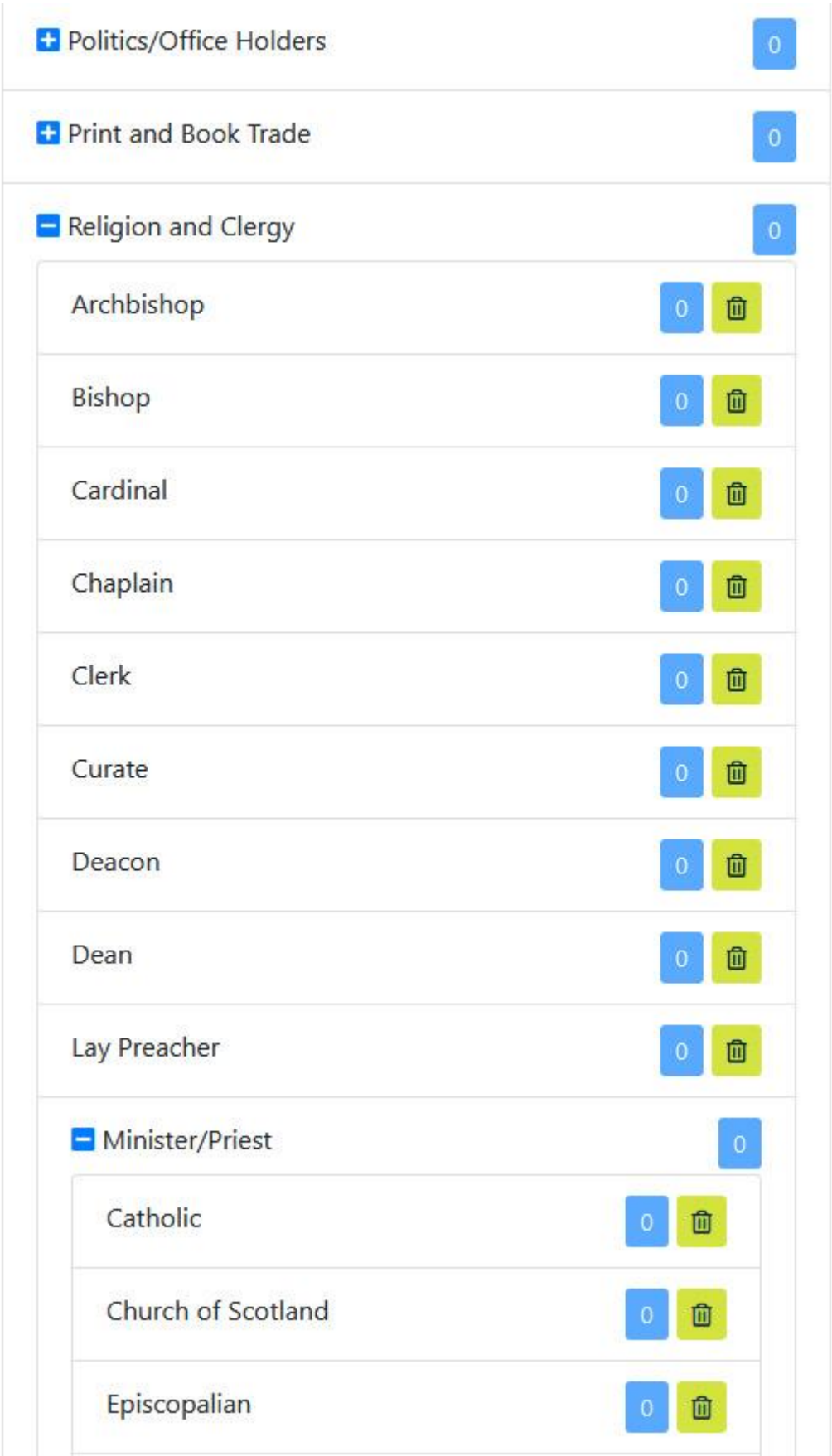
The big new feature I implemented this week was borrower cross references. These can be added via the 'Borrowers' tab within a library when adding or editing a borrower on this page. When adding or editing a borrower there is now a section of the form labelled 'Cross-references to other borrowers'. If there are any existing cross references these will appear here, with a checkbox beside each that you can tick if you want to delete the cross reference (the user can tick the box then press 'Edit' to edit the borrower and the reference will be deleted). Any number of new cross references can be added by pressing on the 'Add a cross-reference' button (multiple times, if required). Doing so adds two fields to the form, one for a 'description', which is the text that shows how the current borrower links to the referenced borrowing record, and one for 'referenced borrower', which is an auto-complete. Type in a name or part of a name and any borrower that matches in any library will be listed. The library appears in brackets after the borrower's name to help differentiate records. Select a borrower and then when the 'Add' or 'Edit' button is pressed for the borrower the cross reference will be made.

Cross-references work in both directions – if you add a cross reference from Borrower A to Borrower B you don't then need to load up the record for Borrower B to add a reference back to Borrower A. The description text will sit between the borrower whose form you make the cross reference on and the referenced borrower you select, so if you're on the edit form for Borrower A and link to Borrower B and the description is 'is the son of' then the cross reference will appear as 'Borrower A is the son of Borrower B'. If you then view Borrower B the cross reference will still be written in this order. I also updated the table of borrowers to add in a new 'X-Refs' column that lists all cross-references for a borrower.

20th July 2020

I spent a couple of days this week continuing to add features to the content management system for the Books and Borrowing project. I have now implemented the 'normalised occupations' part of the CMS. Originally occupations were just going to be a set of keywords, allowing one or more keyword to be associated with a borrower. However, we have been liaising with another project that has already produced a list of occupations and we have agreed to share their list. This is slightly different as it is hierarchical, with a top-level 'parent' containing multiple main occupations. E.g. 'Religion and Clergy' features 'Bishop'. However, for our project we needed a third hierarchical level do differentiate types of minister/priest, so I've had to add this in too. I've achieved this by means of a parent occupation ID in the database, which is 'null' for top-level occupations and contains the ID of the parent category for all other occupations.

I completed work on the page to browse occupations, arranging the hierarchical occupations in a nested structure that features a count of the number of borrowers associated with the occupation to the right of the occupation name. These are all currently zero, but once some associations are made the numbers will go up and you'll be able to click on the count to bring up a list of all associated borrowers, with links through to each borrower. If an occupation has any child occupations a '+' icon appears beside it. Press on this to view the child occupations, which also have counts. The counts for 'parent' occupations tally up all of the totals for the child occupations, and clicking on one of these counts will display all borrowers assigned to all child occupations. If an occupation is empty there is a 'delete' button beside it. As the list of occupations is going to be fairly fixed I didn't add in an 'edit' facility – if an occupation needs editing I can do it directly through the database, or it can be deleted and a new version created. Here's a screenshot showing some of the occupations in the 'browse' page:



I also created facilities to add new occupations. You can enter an occupation name and optionally specify a parent occupation from a drop-down list. Doing so will add the new

occupation as a child of the selected category, either at the second level if a top level parent is selected (e.g. 'Agriculture') or at the third level if a second level parent is selected (e.g. 'Farmer'). If you don't include a parent the occupation will become a new top-level grouping. I used this feature to upload all of the occupations, and it worked very well.

I then updated the 'Borrowers' tab in the 'Browse Libraries' page to add 'Normalised Occupation' to the list of columns in the table. The 'Add' and 'Edit' borrower facilities also now feature 'Normalised Occupation', which replicates the nested structure from the 'browse occupations' page, only features checkboxes beside each main occupation. You can select any number of occupations for a borrower and when you press the 'Upload' or 'Edit' button your choice will be saved. Deselecting all ticked checkboxes will clear all occupations for the borrower. If you edit a borrower who has one or more occupations selected, in addition to the relevant checkboxes being ticked, the occupations with their full hierarchies also appear above the list of occupations, so you can easily see what is already selected. I also updated the 'Add' and 'Edit' borrowing record pages so that whenever a borrower appears in the forms the normalised occupations feature also appears.

I also added in the option to view page images. Currently the only ledgers that have page images are the three Glasgow ones, but more will be added in due course. When viewing a page in a ledger that includes a page image you will see the 'Page Image' button above the table of records. Press on this and a new browser tab will open. It includes a link through to the full-size image of the page if you want to open this in your browser or download it to open in a graphics package. It also features the 'zoom and pan' interface that allows you to look at the image in the same manner as you'd look at a Google Map. You can also view this full screen by pressing on the button in the top right of the image.

10th August 2020

I spent some time working with some of the digitised images of the register pages. We now have access to the images from Westerkirk library and in these records appear in a table that spreads across both recto and verso pages but we have images of the individual pages. The project RA who is transcribing the records is treating both recto and verso as a single 'page' in the system, which makes sense. We therefore need to stitch the r and v images together into on single image to be associated with this 'page'. I downloaded all of the images and have found a way to automatically join two page images together. However, there is rather a lot of overlap in the images, meaning the book appears to have two joins and some columns are repeated. I could possibly try to automatically crop the images before joining them, but there is quite a bit of variation in the size of the overlap so this is never going to be perfect and may result in some information getting lost. The other alternative would be to manually crop and join the images, which I did some experimentation with. It's still not perfect due to the angle of the page changing between shots, but it's a lot better. The downside with this approach is that someone would have to do the task. There are about 230 images, so about 115 joins, each one taking 2-3 minutes to create, so maybe about 5 or so hours of effort. I've left it with the PI and Co-I to decide what to do about this. I also downloaded the images for Volume 1 of the register for Innerpeffray library and created tilesets for these that will allow the images to be zoomed

and panned. I also fixed a bug relating to adding new book items to a record and responded to some feedback about the CMS.

17th August 2020

I made some tweaks to the CMS to make is easier to edit records. When a borrowing record is edited the page automatically scrolls down to the record that was edited. This also happens for books and borrowers when accessed and edited from the 'Books' and 'Borrowers' tabs in a library. I also wrote an initial script that will help to merge some of the duplicate author records we have in the system due to existing data with different formats being uploaded from different libraries. What it does is strip all of the non-alpha characters from the forename and surname fields, makes them lower case then joins them together. So for example, author ID (AID) 111 has 'Arthur' as forename and 'Bedford' as surname while AID 1896 has nothing for forename and 'Bedford, Arthur, 1668-1745' as surname. When stripped and joined together these both become 'bedfordarthur' and we have a match.

There are 162 matches that have been identified, some consisting of more than two matched author records. I exported these as a spreadsheet. Each row includes the author's AID, title, forename, surname, othername, born and died (each containing 'c' where given), a count of the number of books the record is associated with and the AID of the record that is set to be retained for the match. This defaults to the first record, which also appears in bold, to make it easier to see where a new batch of duplicates begins.

The editors can then go through this spreadsheet and reassign the 'AID to keep' field to a different row. E.g. for Francis Bacon the AID to keep is given as 1460. If the second record for Francis Bacon should be kept instead the editor would just need to change the value in this column for all three Francis Bacons to the AID for this row, which is 163. Similarly, if something has been marked as a duplicate and it's wrong, then set the 'AID to keep' accordingly. E.g. There are four 'David Hume' records, but looking at the dates at least one of these is a different person. To keep the record with AID 1610 separate, replace the AID 1623 in the 'AID to keep' column with 1610. It is likely that this spreadsheet will be used to manually split up the imported authors that just have all their data in the surname column. Someone could, for example take the record that has 'Hume, David, 1560?-1630?' in the surname column and split this into the correct columns.

I also generated a spreadsheet containing all of the authors that appear to be unique. This will also need checking for other duplicates that haven't been picked up as there are a few. For example AID 1956 'Heywood, Thomas, d. 1641' and 1570 'Heywood, Thomas, -1641.' Haven't been matched because of that 'd'. Similarly, AID 1598 'Buffon, George Louis Leclerc, comte de, 1707-1788' and 2274 'Buffon, Georges Louis Leclerc, comte de, 1707-1788.' Haven't been matched up because one is 'George' and the other 'Georges'. Accented characters have also not been properly matched, e.g. AID 1457 'Beze, Theodore de, 1519-1605' and 397 'Bèze, Théodore de, 1519-1605.'. I could add in a Levenshtein test that matches up things that are one character different and update the script to properly take into account accented characters for matching purposes, or these are things that could just be sorted manually.

24th August 2020

I processed the images of a register from Westerkirk library. For this register I needed to stitch together the images of the left and right pages to make a single image, as each spread features a table that covers both pages. As we didn't want to have to manually join hundreds of images I wrote a script that did this, leaving a margin between the two images as they don't line up perfectly. I used the command-line tool Imagemagick to achieve this – firstly adding the margin to the left-hand image and secondly joining this to the right-hand image. I then needed to generate tilesets of the images using Zoomify, but when I came to do so the converter processed the images the wrong way round – treating them as portrait rather than landscape and resulting in tilesets that were all wrong. I realised that when joining the page images together the image metadata hadn't been updated: two portrait images were joined together to make one landscape image, but the metadata still suggested that the image was portrait, which confused the Zoomify converter. I therefore had to run the images through Imagemagick again to strip out all of the metadata and then rotate the images 90 degrees clockwise, which resulted in a set of images I could then upload to the server.

31st August 2020

I spent a bit of time getting the page images for the Westerkirk library uploaded to the server and the page records created for each corresponding page image.

14th September 2020

I wrote an import script to process the Glasgow Professors borrowing records, comprising of more than 7,000 rows in a spreadsheet. It was tricky to integrate this with the rest of the project's data and it took about a day to write the necessary processing scripts. I can only run the scripts on the real data in the evening as I need to take the CMS offline to do so, otherwise changes made to the database whilst I'm integrating the data will be lost and unfortunately it took three attempts to get the import to work properly. There are a few reasons why this data has been particularly tricky. Firstly, it needs to be integrated with existing Glasgow data, rather than being a 'fresh' upload to a new library. This caused some problems as my scripts that match up borrowing records and borrowers were getting confused with the existing Student borrowers. Secondly, the spreadsheet order was not in page order for each register – the order appears to have been '10r', '10v', then '11r' etc then after '19v' came '1r'. This is presumably to do with Excel ordering numbers as text. I tried reordering on the 'sort order' column but this also ordered things weirdly (all the numbers beginning with 1, then all the numbers beginning with 2 etc). I tried changing the data type of this field to a number rather than text but that just resulted in Excel giving errors in all of the fields. What this meant was I needed to sort the data in my own script before I could use it (otherwise the 'next' and 'previous' page links would all have been wrong), and it took time to implement this. However, I got there in the end.

11th January 2021

I made some improvements and fixes to the Content Management System. I'd been meaning to enhance the CMS for some time, but due to other commitments to other projects I didn't have the time to delve into it. It felt good to find the time to return to the project this week.

I updated the 'Books' and 'Borrowers' tabs when viewing a library in the CMS. I added in pagination to speed up the loading of the pages. Pages are now split into 500 record blocks and you can navigate between pages using the links above and below the tables. For some reason the loading of the page is still a bit slow on the Stirling server whereas it was fine on the Glasgow server I was using for test purposes. I'm not entirely sure why as I'd copied the database over too – presumably the Stirling server is slower. However, it is still a massive improvement on the speed of the page previously.

I also changed the way tables scroll horizontally. Previously if a table was wider than the page a scrollbar appeared above and below the table, but this was rather awkward to use if you were looking at the middle of the table (you had to scroll up or down to the beginning or end of the table, then use the horizontal scrollbar to move the table along a bit, then navigate back to the section of the page you were interested in). Now the scrollbar just appears at the bottom of the browser window and can always be accessed no matter where in the table you are.

I also removed the editorial notes from tables by default to reduce clutter, and added in a button for showing / hiding the editors' notes near the top of each page. I also added a limit option in the 'Books' and 'Borrowers' pages within a library to limit the displayed records to only those found in a specific ledger. I added in a further option to display those records that are not currently associated with any ledgers too.

I then deleted the 'original borrowed date' and 'original returned date' fields in St Andrews data as these were no longer required. I deleted these additional fields from the system and all data that were contained in these fields.

It had been noted that the book part numbers were not being listed numerically. As part numbers can contain text as well as numbers (e.g. 'Vol. II'), this field in the database needed to be set as text rather than an integer. Unfortunately the database doesn't order numbers correctly when they are contained in a non-numerical field – instead all the ones come first (1, 10, 11) then all the twos (2, 20, 22) etc. However, I managed to find a way to ensure that the numbers are ordered correctly.

I also fixed the 'Add another Edition/Work to this holding' button that was not working. This was caused by the Stirling server running a different version of PHP that doesn't allow functions to have variable numbers of arguments. The autocomplete function was also not working at edition level and I investigated this. The issue was being caused by tab characters appearing in edition titles, and I updated my script to ensure these characters are stripped out before the data is formatted as JSON.

18th January 2021

I continued to make refinements to the CMS, namely reducing the number of books and borrowers from 500 to 200 to speed up page loads, adding in the day of the week that books

were borrowed and returned, based on the date information already in the system, removing tab characters for edition titles as these were causing some issues for the system, replacing the editor's notes rich text box with a plain text area to save space on the edit page and adding a new field to the borrowing record that allows the editor to note when certain items appear for display only and should otherwise be overlooked, for example when generating stats. This is to be used for duplicate lines and lines that are crossed out. I also had a look through the new sample data from Craigston that was sent to us this week.

25 January 2021

For the Books and Borrowing project I participated in the project's Zoom call on Monday to discuss the project's CMS and how to amalgamate the various duplicate author records that resulted from data uploads from different libraries. After the call I made some required changes to the CMS, such as making the editor's notes fields visible by default again, and worked on the duplicate authors matching script to add in further outputs when comparing the author names with Levenshtein ratings of 1 and 2. I also reviewed some content that was sent to us from another library.

8th February 2021

I spent some time on the Books and Borrowing project on two data-related tasks. First was to look through the new set of digitised images from Edinburgh University Library and decide what we should do with them. Each image is of an open book, featuring both recto and verso pages in one image. We may need to split these up into individual images, or we may just create page records that cover both pages. I alerted the project PI Katie Halsey to the issue and the team will make a decision about which approach to take next week. The second task was to look through the data from Selkirk library that another project had generated. We had previously imported data for Selkirk that another researcher had compiled a few years before our project began, but recently discovered that this data did not include several thousand borrowing records of French prisoners of war, as the focus of the researcher was on Scottish borrowers. We need these missing records and another project has agreed to let us use their data. I had intended to completely replace the database I'd previously ingested with this new data, but on closer inspection of the new data I have a number of reservations about doing so.

The data from the other project has been compiled in an Excel spreadsheet and as far as I can tell there is no record of the ledger volume or page that each borrowing record was originally located on. In the data we already have there is a column for 'source ref', containing the ledger volume (e.g. 'volume 1') and a column for 'page number', containing a unique ID for each page in the spreadsheet (e.g. '1010159r'). Looking through the various sheets in the new spreadsheet there is nothing comparable to this, which is vital for our project, as borrowing records must be associated with page records, which in turn must be associated with a ledger. It also would make it extremely difficult to trace a record back to the original physical record.

Another issue is that in our existing data the researcher has very handily used unique identifiers for readers (e.g. 'brodie_james'), borrowing records (e.g. '1') and books (e.g. 'adam_view_religion') that tie the various records together very nicely. The new project's data does not appear to use any unique identifiers to connect bits of data together. For example,

there are three 'John Anderson' borrowers and in the data we're currently using these are differentiated by their IDs as 'anderson_john', 'anderson_john2' and 'anderson_john3'. This means it's easy to tell which borrower appears in the borrowing records. In the new project's data three different fields are required to identify the borrower: surname, forename and residence. This data is stored in separate columns in the 'All loans' sheet (e.g. 'Anderson', 'John', 'Cramalt'), but in the 'Members' sheet everything is joined together in one 'Name' field, e.g. 'Anderson, John (Cramalt)'. This lack of unique identifiers combined with the inconsistent manner of recording name and place will make it very difficult to automatically join up records and I've flagged this up with Katie for further discussion with the team. It's looking like we may want to try and identify the POW records from the new project's data and amalgamate these with the data we already have, rather than replacing everything.

15th February 2021

I spent a bit of time working for the Books and Borrowing project, processing images for a ledger from Edinburgh University Library, uploading these to the server and generating page records and links between pages for the ledger.

22nd February 2021

I continued to work on the Books and Borrowing project, generating image tilesets for the scans of several volumes of ledgers from Edinburgh University Library and writing scripts to generate pages in the Content Management System, creating 'next' and 'previous' links as required and associating the relevant images. I also had an email correspondence about some of the querying methods we will develop for the data, such as collocation information.

1st March 2021

There was quite a bit of work to be done for the Books and Borrowing project this week. Several more ledgers had been digitised and needed tilesets and page records generated for them. The former requires the processing and upload of many gigabytes of images files which takes quite some time to complete, especially as the upload speed from my home computer to the server never gets beyond about 56Kb per second. However, I just end up leaving my PC on overnight and generally the upload has completed by the morning. Generating page records generally involves me updating a script to change image filename parts and page numbers, and to specify the first and last page and the script does the rest, but there are some quirks that need to be sorted out manually. For the Wigtown data some of the images were not sequentially numbered, which meant I couldn't rely on my script to generate the correct page structure. For one of the Edinburgh ledgers the RA has already manually created some pages and had added more than a hundred borrowing records to them so I had to figure out a way to incorporate these. The page images are a double spread (so two pages per image) but the pages the RA had made were individual, so what I needed to do was to remove the manual pages, generate a new set and then update the page references for each of the borrowing records so they appeared on the correct new page.

8th March 2021

I spent a bit of time working on the Books and Borrowing project, generating more page image tilesets and their corresponding pages for two more of the Edinburgh ledgers and adding an 'Events' page to the project website and giving more members of the project team permission to edit the site.

15th March 2021

I also spent some further time on the Books and Borrowing project, creating tilesets and page records for several new volumes. In fact, we ran out of space on the server. The project is digitising around 20,000 pages of library records from 1750-1830 and we're approaching 5,000 pages so far. I'd originally suggested that we'd need about 60GB of server space for the images (3MB per image x 20,000). However, the JPEGS we've been receiving from the digitisation units have been generated at maximum quality / minimum compression and are around 9MB each, so my estimates were out. Dropping the JPEG quality setting down from 12 to 10 would result in 3MB files so I could do this to save space if required. However, there is another issue. The tilesets I'm generating for each image so that they can be zoomed and panned like a Google Map are taking up as much as 18MB per image. So we may need a minimum of 540GB of space (possibly 600GB to be safe): 9×20,000 for the JPEGs plus 18×20,000 for the tilesets. This is an awful lot of space, and storing image tilesets isn't actually necessary these days of an IIIF server (https://iiif.io/about/) could be set up. IIIF is now well established as the best means of hosting images online and it would be hugely useful to use. Rather than generating and hosting thousands of tilesets at different zoom levels we could store just one image per page on the server and it would serve up the necessary subsection at the required zoom level based on the request from the client. This issue is that people in charge of servers don't' like having to support new software. I entered into discussions with Stirling's IT people about the possibility of setting up an IIIF server, and these talks are currently ongoing, so in the meantime I still need to generate the tilesets.

22nd March 2021

I continued with the processing of library registers for the Books and Borrowing project. These are coming in rather quickly now and I'm getting a bit of a backlog. This is because I have to download the image files, then process then to generate tilesets, and then upload all of the images and their tilesets to the server. It's the tilesets that are the real sticking point, as these consist of thousands of small files. I'm only getting an upload speed of about 70KB/s and I'm having to upload many gigabytes of data. I did a test where I zipped up some of the images and uploaded this zip file instead and was getting a speed of around 900KB/s and as it looks like I can get command-line access to the server I'm going to investigate whether zipping up the files, then uploading them then unzipping them will be a quicker process. I also had to spend some time sorting out connection issues to the server as the Stirling VPN wasn't letting me connect. It turned out that they had switched to multi-factor authentication and I needed to set this up before I could continue.

5th April 2021

I spent some of this time continuing to download and process images of library register books for the Books and Borrowing project, including 14 from St Andrews and several further books from Edinburgh.

12 April 2021

I'd been in discussion with the Stirling University IT people about setting up a IIIF server for the project, and I heard this week that they have agreed to this, which is really great news. Previously in order to allow page images to be zoomed and panned like a Google Map we had to generate and store tilesets of each page image at each zoom level. It was taking hours to

generate the tilesets for each book and days to upload the images to the server, and was requiring a phenomenal amount of storage space on the server. For example, the tilesets for one of the Edinburgh volumes consisted of around 600,000 files and took up around 14GB of space. This was in addition to the actual full-size images of the pages (about 250 at around 12MB each).

An IIIF server means we only need to store the full-size images of each page and the server dynamically chops up and serves sections of the image at the desired zoom level whenever anyone uses the zoom and pan image viewer. It's a much more efficient system. However, it does mean I needed to update the 'Page image' page of the CMS to use the IIIF server, and it took a little time to get this working. I'd decided to use the OpenLayers library to access the images, as this is what I'd previously been using for the image tilesets, and it has the ability to work with a IIIF server (see https://openlayers.org/en/latest/examples/iiif.html). However, it did take some time to get this working, as the example and all of the documentation is fully dependent on the node.js environment, even though the library itself really doesn't need to be. I didn't want to convert my CMS to using node.js and have yet another library to maintain when all I needed was a simple image viewer, so I had to rework the code example linked to above to strip out all of the node dependencies, module syntax and import statements. For example 'var options = new IIIFInfo(imageInfo).getTileSourceOptions()' needed to be changed to 'var options = new ol.format.IIIFInfo(imageInfo).getTileSourceOptions()'. As none of this is documented anywhere on the OpenLayers website it took some time to get right, but I got there in the end and the CMS now has an OpenLayers based IIIF image viewer working successfully.

19th April 2021

I continued to upload the page images of the Edinburgh registers, finishing the upload of 16 registers and then generating the page records for all of the pages in the content management system. I then started on the St Andrews registers.

26th April 2021

I received the last of the digitised images of borrowing registers from Edinburgh (other than one register which needs conservation work), and I uploaded these to the project's content management system, creating all of the necessary page records. We have a total of 9,992 page images as JPEG files from Edinburgh, totalling 105GB. Thank goodness we managed to set up an IIIF server for the image files rather than having to generate and store image tilesets for each of these page images. Also this week I uploaded the images for 14 borrowing registers from St Andrews and generated page records for each of these.

14th June 2021

I made some updates to the Books and Borrowing CMS. This took some time to get started on as my access to the Stirling VPN had been cancelled, and without such access I couldn't access the project's server. Thankfully with the help of Stirling's Information Services people my access was reinstated on Monday and I could start working on the updates. After familiarising myself with the systems again I had some further questions about the updates suggested by Matt Sangster, resulting in an email conversation and a suggestion by him that he discusses things further with the team next Monday. Gerry McKeever had suggested some further updates, though, and I worked on these.

The first issue was the ordering of the 'Books' tab when viewing a library. This list of books (of which there can be thousands) is paginated with 200 books per page, with options to order the table by a variety of columns (e.g. book name and number of associated borrowings). However, the ordering was only ordering the subset of 200 books rather than the whole set.

I updated the page so that the complete dataset is reordered rather than just the 200 records that are displayed per page. However, this has a massive performance hit that wipes out the page loading speed increase that was gained from paginating the list in the first place. To reorder the data the page needs to load the entire dataset and then reorder it. In the case of St Andrews this means that more than 7,200 book records need to be loaded, with multiple sub-queries for each of these records required to bring back the counts of borrowing records and information about book items, book editions and authors.

With the previous paginated way of viewing the data the CMS was taking a couple of seconds to load the 'Books' page for St Andrews. With the new update in place it was taking more than 1 minute and 20 seconds for the page to load. When running the exact same code and database on my local PC it was taking 10 seconds to load, so presumably the spec of my local PC is considerably better than the server (either that or it's having to handle a lot of other database requests at the same time, which is affecting performance).

I had considered storing the data in a session variable, which would mean after the first horrendous load time the data would be ready and waiting in the server's memory to be used until you closed your browser, however, as the data is continuously being worked on this would mean the information displayed would possibly not accurately reflect the current state of the data, which may be confusing. What I am planning on doing when I develop the front-end is to create a cached version of the data, so counts of borrowing records etc won't need to be recalculated each time a user queries something, but creating such a cached version wouldn't really work whilst the data is still being worked on. I could set the system up to refresh the cache every night, but that would mean the CMS would again not reflect the current state of the data, which isn't good. I also updated the 'Borrowers' page to allow full reordering of data here too. This isn't quite as slow as the books page.

I spoke to the server admin people to see if they could think of a reason why the server loading speed was so much worse that on my local PC. They reckoned it was because the database is stored on a different server to the code, and the sheer number of individual queries being sent meant that small delays in connecting between servers were mounting up. I reworked the code somewhat to try and streamline the number of database queries that need to be made. Only two of the columns can now be selected to order the data by: Book Holding title and number of borrowing records. I'm hoping these are the most important anyway. I have updated the queries so that the bulk of the data is only retrieved for the 200 records that are on the visible page (as used to be the case) with only a single query of the holding table and then a further query for each relevant holding record to bring back a count of its borrowing records now being

made on the full dataset (e.g. for St Andrews for each of the 7,391 books). This has made a huge difference and has brought the page loading times back down to a more acceptable few seconds.

Gerry's second request was that when the book list is limited to a specific register the counts of borrowings updated to reflect this. I updated the code so that counts of borrowing records on both the 'Books' and 'Borrowers' tabs get limited to just the selected register and thankfully there was no performance hit associated with this update.

21st June 2021

I continued to work on the Books and Borrowings project, having an email conversation with the digitisers at the NLS about file formats and methods of transferring files, and making further updates to the CMS to add features and make things run quicker. I managed to reduce the number of database calls on the 'Books' tab in the library view again, which should mean the page loads faster. Previously all book holding records were returned and then a separate query was executed for each to count the number of borrowings whereas I've now nested the count query in the initial query. So for St Andrews with its 7471 books this has cut out 7471 individual queries.

I'd realised that the 'borrowing records' count column in this 'Books' table isn't actually a count of borrowing records at all, but a count of the number of book items that have been borrowed for the book holding. I've figured out a way to return a count of borrowing records instead, and I replaced the old way with the new way, so the 'Borrowing Records' column now does what it should do. This means the numbers listed have changed, e.g. 'Universal History' now has 177 borrowing records rather than 269 and is no longer the most borrowed book holding at St Andrews. I also changed the popup so that each borrowing record only appears once (e.g. David Gregory on 1748-6-7 now only has one borrowing record listed). I added a further 'Total borrowed items' column in as well, to hold the information that was previously in the 'Borrowing Records' column, and it's possible to order the table by this column too. I also noticed that I'd accidentally removed columns displaying additional fields from the table, so I have reinstated these. For St Andrews this means the 'Classmark' column is now back in the table. I also realised that my new nested count queries were not limiting their counts when a specific register was selected so updated them to take this into consideration too.

5th July 2021

I made some further updates to the content management system for the Books and Borrowing project. There was a bug in the 'clear borrower' feature that resulted in the normalised occupation fields not getting clears. This meant that unless a researcher noticed and manually removed the selected occupations it would be very easy to end up with occupations assigned to the wrong borrower. I implemented a fix for this bug, so all is well now. I had also been alerted to an issue with the library's 'books' tab. When limiting the listed books to only those mentioned in a specific register the list of associated borrowing records that appears in a popup was not limiting the records to those in the specified register. I fixed this as well, and also made a comparable fix to the 'borrowers' tab as well.

19th July 2021

I created a script to batch process some of the Edinburgh registers for the Books and Borrowing project. The page images are double spreads and had been given a number for both the recto and the verso (e.g. 1-2, 3-4), but the student registers only ever use the verso page. I was therefore asked to write a script to renumber all of these (e.g. 1-2 becomes 1, 3-4 becomes 2), which I created and executed on a test version of the site before applying to the live data.

9th August 2021

I spoke to Katie Halsey, PI of the Books and Borrowing project about the development of the API for the project and the data export facilities.

30th August 2021

I created a new user account for someone who will be working for the project and I also received the digitised images for another library register, this time from the NLS. I downloaded these and then uploaded them to the server, associating the images with the page records that were already in the system. The process was a little more complicated and time consuming than I'd anticipated as the register has several blank pages in it that are not in our records but have been digitised. Therefore the number of page images didn't match up with the number of pages, plus page images were getting associated with the wrong page. I had to manually look through the page images and delete the blanks, but I was still off by one image. I then had to manually check through the contents of the images to compare them with the transcribed text to see where the missing image should have gone. Thankfully I managed to track it down and reinstate it (it had one very faint record on it, which I hadn't noticed when viewing and deleting blank thumbnails). With that in place all images and page records aligned and I could made the associations in the database. I also sent Gerry McKeever the zipped up images (several gigabytes) for a couple of the St Andrews registers as he prefers to have the complete set when working on the transcriptions.

20th September 2021

I also received a spreadsheet of borrowing records covering five registers for the Books and Borrowing project and went through it to figure out how the data might be integrated with our system. The biggest issue is figuring out which page each record is on. In the B&B system each borrowing record must 'belong' to a page, which in turn 'belongs' to a register. If a borrowing record has no page it can't exist in the system. In this new data only three registers have a 'Page No.' column and not every record in these registers has a value in this column. We'll need to figure out what can be done about this, because as I say, having a page is mandatory in the B&B system. We could use the 'photo' column as this is present across all registers and every row. However, I noticed that there are multiple photos per page, e.g. for SL137144 page 2 has 2 photos (4538 and 4539) so photo IDs don't have a 1:1 relationship with pages. If we can think of a way to address the page issue then I should be able to import the data.

27th September 2021

I had a Zoom call for the Books and Borrowing project to discuss adding data from a new source to the database. The call gave us an opportunity to discuss the issues with the data that I'd highlighted last week. It was good to catch up with the team again and to discuss the issues with the researcher who had originally prepared the spreadsheet containing the data. We

managed to address all of the issues and the researcher is going to spend a bit of time adapting the spreadsheet before sending it to me to be batch uploaded into our system.

4th October 2021

I was sent a batch of images of a register from Dumfries Presbytery Library and I needed to batch process them in order to fix the lighting levels and rename them prior to upload. It took me a little time to figure out how to run a batch process in the ancient version of Photoshop I have. After much hopeless Googling I found some pages from 'Photoshop CS2 For Dummies' on Google Books that discussed Photoshop Actions (see https://books.google.co.uk/books?id=RLOmw2omLwgC&lpg=PA374&dq=&pg=PA332#v=onepage&q&f=false) which made me realise the 'Actions', which I'd failed to find in any of the menus, were available via the tabs on the right of the screen, and I could 'record' and action via this. After running the images through the batch I uploaded them to the server and generated the page records for each corresponding page in the register.

18th October 2021

I was given access to a large number of images of registers from the Advocates Library that had been digitised by the NLS. I downloaded these, batch processed them to add in the register numbers as a prefix to the filenames, uploaded the images to our server, created register records for each register and page records for each page. The registers, pages and associated images can all now be accessed via our CMS.

1st November 2021

I worked on the Books and Borrowers project, running a script to remove blank pages from all of the Advocates registers and discussing some issues with the Innerpeffray data and how we might deal with this.

13th December 2021

I began to investigate replacing the Innerpeffray data for Books and Borrowing with a new dataset that Kit has worked on. This is going to be quite a large and complicated undertaking and after working through the data I had a set of questions to ask Kit before I proceeded to delete any of the existing data. Unfortunately she is currently on jury duty so I'll need to wait until she's available again before I can do anything further. Also this week a huge batch of images became available to us from the NLS and I spent some time downloading these and moving them to an external hard drive as they'd completely filled up the hard drive of my PC.

20th December 2021

My major task of the week was to deal with the new Innerpeffray data for the Books and Borrowing project. I'd previously uploaded data from an existing spreadsheet in the early days of the project, but it turns out that there were quite a lot of issues with the data and therefore one of the RAs has been creating a new spreadsheet containing reworked data. The RA Kit got back to me this week after I'd checked some issues with her last week and I therefore began the process of deleting the existing data and importing the new data.

I was a pretty torturous process but I managed to finish deleting the existing Innerpeffray data and imported the new data. This required a pretty complex amount of processing and checking via a script I wrote this week. I managed to retain superscript characters in the transcriptions, something that proved to be very tricky as there is no way to find and replace superscript characters in Excel. Eventually I ended up copying the transcription column into Word, then saving the table as HTML, stripping out all of the rubbish Word adds in when it generates an HTML file and then using this resulting file alongside the main spreadsheet file that I saved as a CSV. After several attempts at running the script on my local PC, then fixing issues, then rerunning, I eventually reckoned the script was working as it should – adding page, borrowing, borrower, borrower occupation, book holding and book item records as required. I then ran the script on the server and the data is now available via the CMS.

There were a few normalised occupations that weren't right and I updated these. There were also 287 standardised titles that didn't match any existing book holding records in Innerpeffray. For these I created a new holding record and (if there's an ESTC number) linked to a corresponding edition.

3rd January 2022

We had received a further batch of images for 23 library registers from the NLS, which I needed to download from the NLS's server and process. This involved renaming many thousands of images via a little script I'd written in order to give the images more meaningful filenames and stripping out several thousand images of blank pages that had been included but are not needed by the project. I then needed to upload the images to the project's web server and then generate all of the necessary register and page records in the CMS for each page image.

I also needed up update the way folio numbers were generated for the registers. For the previous batch of images from the NLS I had just assigned the numerical part of the image's filename as the folio number, but it turns out that most of the images have a hand-written page number in the top-right which starts at 1 for the first actual page of borrowing records. There are usually a few pages before this, and these need to be given Roman numerals as folio numbers. I therefore had to write another script that would take into consideration the number of front-matter pages in each register, assign Roman numerals as folio numbers to them and then begin the numbering of borrowing record pages from 1 after that, incrementing through the rest of the volume.

I guess it was inevitable with data of this sort, but I ran into some difficulties whilst processing it. Firstly, there were some problems with the Jpeg images the NLS had sent for two of the volumes. These didn't match the Tiff images for the volumes, with each volume having an incorrect number of files. Thankfully the NLS were able to quickly figure out what had gone wrong and were able to supply updated images.

The next issue to crop up occurred when I began to upload the images to the server. After uploading about 5Gb of images the upload terminated, and soon after that I received emails

from the project team saying they were unable to log into the CMS. It turns out that the server had run out of storage. Each time someone logs into the CMS the server needs a tiny amount of space to store a session variable, but there wasn't enough space to store this, meaning it was impossible to log in successfully. I emailed the IT people at Stirling (Where the project server is located) to enquire about getting some further space allocated but I haven't heard anything back yet. In the meantime I deleted the images from the partially uploaded volume which freed up enough space to enable the CMS to function again. I also figured out a way to free up some further space: The first batch of images from the NLS also included images of blank pages across 13 volumes – several thousand images. It was only after uploading these and generating page records that we had decided to remove the blank pages, but I only removed the CMS records for these pages – the image files were still stored on the server. I therefore wrote another script to identify and delete all of the blank page images from the first batch that was uploaded, which freed up 4-5Gb of space from the server, which was enough to complete the upload of the second batch of registers from the NLS. We will still need more space, though, as there are still many thousands of images left to add.

I also took the opportunity to update the folio numbers of the first batch of NLS registers to bring them into line with the updated method we'd decided on for the second batch (Roman numerals for front-matter and then incrementing page numbers from the first page of borrowing records). I wrote a script to renumber all of the required volumes, which was mostly a success.

However, I also noticed that the automatically generated folio numbers often became out of step with the hand-written folio numbers found in the top-right corner of the images. I decided to go through each of the volumes to identify all that became unaligned and to pinpoint on exactly which page or pages the misalignment occurred. This took some time as there were 32 volumes that needed checked, and each time an issue was spotted I needed to look back through the pages and associated images from the last page until I found the point where the page numbers correctly aligned. I discovered that there were numbering issues with 14 of the 32 volumes, mainly due to whoever wrote the numbers in getting muddled. There are occasions where a number is missed, or a number is repeated. In once volume the page numbers advance by 100 from one page to the next. It should be possible for me to write a script that will update the folio numbers to bring them into alignment with the erroneous handwritten numbers (for example where a number is repeated these will be given 'a' and 'b' suffixes). I didn't have time to write the script this week but will do so next week.

Also for the project this week I looked through the spreadsheet of borrowing records from the Royal High School of Edinburgh that one of the RAs has been preparing. I had a couple of questions about the spreadsheet, and I'm hoping to be able to process it next week. I also exported the records from one register for Gerry McKeever to work on, as these records now need to be split across two volumes rather than one.

10th January 2022

I continued to work on the Books and Borrowing project for a lot of this week, completing some of the tasks I began last week and working on some others. We ran out of server space for

digitised page images last week, and although I freed up some space by deleting a bunch of images that were no longer required we still have a lot of images to come. The team estimates that a further 11,575 images will be required. If the images we receive for these pages are comparable to the ones from the NLS, which average around 1.5Mb each, then 30Gb should give us plenty of space. However, after checking through the images we've received from other digitisation units it turns out that the NLS images are a vit of an outlier in term of file size and generally 8-10Mb is more usual. If we use this as an estimate then we would maybe require 120Gb-130Gb of additional space. I did some experiments with resizing and changing the image quality of one of the larger images, managing to bring an 8.4Mb image down to 2.4Mb while still retaining its legibility. If we apply this approach to the tens of thousands of larger images we have then this would result in a considerable saving of storage. However, Stirling's IT people very kindly offered to give us a further 150Gb of space for the images so this resampling process shouldn't be needed for now at least.

Another task for the project this week was to write a script to renumber the folio numbers for the 14 volumes from the Advocates Library that I noticed had irregular numbering. Each of the 14 volumes had different issues with their handwritten numbering, so I had to tailor my script to each volume in turn, and once the process was complete the folio numbers used to identify page images in the CMS (and eventually in the front-end) entirely matched the handwritten numbers for each volume.

My next task for the project was to import the records for several volumes from the Royal High School of Edinburgh but I ran into a bit of an issue. I had previously been intending to extract the 'item' column and create a book holding record and a single book item record for each distinct entry in the column. This would then be associated with all borrowing records in RHS that also feature this exact 'item'. However, this is going to result in a lot of duplicate holding records due to the contents of the 'item' column including information about different volumes of a book and/or sometimes using different spellings.

For example, in SL137142 the book 'Banier's Mythology' appears four times as follows (assuming 'Banier' and 'Bannier' are the same):

Banier's Mythology v. 1, 2

Banier's Mythology v. 1, 2

Bannier's Myth 4 vols

Bannier's Myth. Vol 3 & 4

My script would create one holding and item record for 'Banier's Mythology v. 1, 2' and associate it with the first two borrowing records but the 3rd and 4th items above would end up generating two additional holding / item records which would then be associated with the 3rd and 4th borrowing records.

No script I can write (at least not without a huge amount of work) would be able to figure out that all four of these books are actually the same, or that there are actually 4 volumes for the one book, each requiring its own book item record, and that volumes 1 & 2 need to be associated with borrowing records 1&2 while all 4 volumes need to be associated with borrowing record 3 and volumes 3&4 need to be associated with borrowing record 4. I did wonder whether I might be able to automatically extract volume data from the 'item' column but there is just too much variation.

We're going to have to tackle the normalisation of book holding names and the generation of all required book items for volumes at some point and this either needs to be done prior to ingest via the spreadsheets or after ingest via the CMS.

My feeling is that it might be simpler to do it via the spreadsheets before I import the data. If we were to do this then the 'Item' column would become the 'original title' and we'd need two further columns, one for the 'standardised title' and one listing the volumes, consisting of a number of each volume separated with a comma. With the above examples we would end up with the following (with a | representing a column division):

Banier's Mythology v. 1, 2 | Banier's Mythology | 1,2

Banier's Mythology v. 1, 2 | Banier's Mythology | 1,2

Bannier's Myth 4 vols | Banier's Mythology | 1,2,3,4

Bannier's Myth. Vol 3 & 4 | Banier's Mythology | 3,4

If each sheet of the spreadsheet is ordered alphabetically by the 'item' column it might not take too long to add in this information. The additional fields could also be omitted where the 'item' column has no volumes or different spellings. E.g. 'Hederici Lexicon' may be fine as it is. If the 'standardised title' and 'volumes' columns are left blank in this case then when my script reaches the record it will know to use 'Hederici Lexicon' as both original and standardised titles and to generate one single unnumbered book item record for it. We agreed that normalising the data prior to ingest would be the best approach and I will therefore wait until I receive updated data before I proceed further with this.

Also this week I generated a new version of a spreadsheet containing the records for one register for Gerry McKeever, who wanted borrowers, book items and book holding details to be included in addition to the main borrowing record. I also made a pretty major update to the CMS to enable books and borrower listings for a library to be filtered by year of borrowing in addition to filtering by register. Users can either limit the data by register or year (not both). They need to ensure the register drop-down is empty for the year filter to work, otherwise the selected register will be used as the filter. On either the 'books' or 'borrowers' tab in the year

box they can add either a single year (e.g. 1774) or a range (e.g. 1770-1779). Then when 'Go' is pressed the data displayed is limited to the year or years entered. This also includes the figures in the 'borrowing records' and 'Total borrowed items' columns. Also, the borrowing records listed when a related pop-up is opened will only feature those in the selected years.

24th January 2022

For the Books and Borrowing project I participated in the team Zoom call on Monday to discuss the upcoming development of the front-end and API for the project, which will include many different search and browse facilities, graphs and visualisations. I followed this up with a lengthy email to the PI and Co-I where I listed some previous work I've done and discussed some visualisation libraries we could use. In the coming weeks I'll need to work with them to write a requirements document for the front-end. I also downloaded images from Orkney library, uploaded all of them to the server and generated the necessary register and page records. One register with 7 pages already existed in the system and I ensured that page images were associated with these and the remaining pages of the register fit in with the existing ones. I also processed the Wigtown data that Gerry McKeever had been working on, splitting the data associated with one register into two distinct registers, uploading page images and generating the necessary page records. This was a pretty complicated process, and I still need to complete the work on it next week, as there are several borrowing records listed as separate rows when in actual fact they are merely another volume of the same book borrowed at the same time. These records will need to be amalgamated.

31st January 2022

For the Books and Borrowing project I completed the work I started last week on processing the Wigtown data, writing a little script that amalgamated borrowing records that had the same page order number on any page. These occurrences arose when multiple volumes of a book were borrowed by a person at the same time and each volume was recorded separately. My script worked perfectly and many such records were amalgamated.

I then moved onto incorporating images of register pages from Leighton into the CMS. This proved to be a rather complicated process for one of the four registers as around 30 pages for the register had already been manually created in the CMS and had borrowing records associated with them. However, these pages had been created in a somewhat random order, starting at folio number 25 and mostly being in order down to 43, at which point the numbers are all over the place, presumably because the pages were created in the order that they were transcribed. As it stands the CMS relies on the 'page ID' order when generating lists of pages as 'Folio Number' isn't necessarily in numerical order (e.g. front / back matter with Roman numerals). If out of sequence pages crop up a lot we may have to think about adding a new 'page order' column, or possibly use the 'previous' and 'next' IDs to ascertain the order pages should be displayed. After some discussion with the team it looks like pages are usually created in page order and Leighton is an unusual case, so we can keep using the auto-incrementing page ID for listing pages in the contents page. I therefore generated a fresh batch of pages for the Leighton register then moved the borrowing records from the existing mixed up pages to the appropriate new page, then deleted the existing pages so everything is all in order.

7th February 2022

I had a call with the PI and Co-I of the Books and Borrowing project about the requirements for the front-end and the various search and browse functionality it would need to have. I'd started writing a requirements document before the meeting and we discussed this, plus their suggestions and input from others. It was a very productive meeting and I continued with the requirements document after the call. There's still a lot to put into it, and the project's data and requirements are awfully complicated, but I feel like we're making good progress and things are beginning to make sense.

21st February 2022

I downloaded a new batch of images for five more registers that had been digitised for us by the NLS. I then processed these, uploaded them to our server and generated register and page records for each page image. I also processed the data from the Royal High School of Edinburgh that had been sent to me in a spreadsheet. There were records from five different registers and it took quite some time to write a script that would process all of the data, including splitting up borrower and book data, generating book items where required and linking everything together so that a borrower and a book only exist once in the system even if they are associated with many borrowing records. Thankfully I'd done this all before for previous external datasets, but the process is always different for each dataset so there was still much in the way of reworking to be done.

I completed my scripts and ran them on a test instance of the database running on my local PC to start with. When all was checked and looking good I ran the scripts on the live server to incorporate the new register data with the main project dataset. After completing the task there were 19,994 borrowing records across 1,438 register pages, involving 1,932 books and 2,397 borrowers. Some tweaking of the data may be required (e.g. I noticed there are two 'Alexander Adam' borrowers, which seems to have occurred because there was a space character before the forename sometimes) but on the whole it's all looking good to me.

7th March 2022

I also spent about a day continuing to work on the requirements document for the Books and Borrowing project. I haven't quite finished this initial version of the document but I've made good progress and I aim to have it completed next week. Also for the project I participated in a Zoom call with RA Alex Deans and NLS Maps expert Chris Fleet about a subproject we're going to develop for B&B for the Chambers Library in Edinburgh. This will feature a map-based interface showing where the borrowers lived and will use a historical map layer for the centre of Edinburgh.

14th March 2022

I completed a first version of the requirements for the public website, which has taken a lot of time and a lot of thought to put together, resulting in a document that's more than 5,000 words long. On Friday I had a meeting with PI Katie and Co-I Matt to discuss the document. We spent an hour going through it and a list of questions I'd compiled whilst writing it, and I'll need to make some modifications to the document based on our discussions. I also downloaded

images of more library registers from St Andrews and one further register from Glasgow that I will need to process when I'm back at work too.

28th March 2022

I updated the front-end requirements document based on my discussions with the PI and Co-I and set it on for the rest of the team to give feedback on. I also uploaded a new batch of register images from St Andrews (more than 2,500 page images taking up about 50Gb) and created all of the necessary register and page records. I also did the same for a couple of smaller registers from Glasgow. I also exported spreadsheets of authors, edition formats and edition languages for the team to edit too.

11th April 2022

I created a new version of the requirements document for the front-end for the Books and Borrowing project following feedback form the project team on the previous version.

16th May 2022

I regenerated a list of possible duplicate authors in the Books and Borrowing system after the team had carried out some work to remove duplicates. I will be able to use the spreadsheet I have now to amalgamate duplicate authors, a task which I will tackle next week.

23rd May 2022

I wrote a script to strip out duplicate author records from the data and reassign any books associated with the duplicates to the genuine author records. The script iterated through each author in the 'duplicates' spreadsheet, found all rows where the 'AID' did not match the 'AID to keep' column, reassigned any book author records from the former to the latter and then deleted the author record. The script deleted 310 duplicate authors and reassigned 735 books to other authors, making the data in the content management system a lot cleaner.

30th May 2022

I had a chat with RA Alex Deans about the data for the Chambers Library sub-project that we're hoping to launch in July. Although this data is already in the system it needs additional latitude and longitude data so we can position borrowers on an interactive map. We decided to add this data and some other data using the 'additional fields' system in the CMS and Alex is hopefully going to get this done by next week.

I'd made a start on the API for the project last week, and this week I completed the endpoint that displays all of the data that will be needed for the 'Browse Libraries' page, which can be accessed as JSON or CSV data. This includes counts of registers, borrowing records, books and borrowers plus a breakdown of the number of borrowings per year at each library that will be used for the stacked column chart. The systems reside on servers at Stirling University, and their setup has the database on a different server to the code. This means there is an overhead when sending queries to the database as each one needs to be sent as an HTTP request rather than dealt with locally. This has led me to be a bit more efficient when constructing queries. For example, rather than running individual 'count' queries for each library after running an initial query to retrieve all library details I've instead used subqueries as part of the initial query

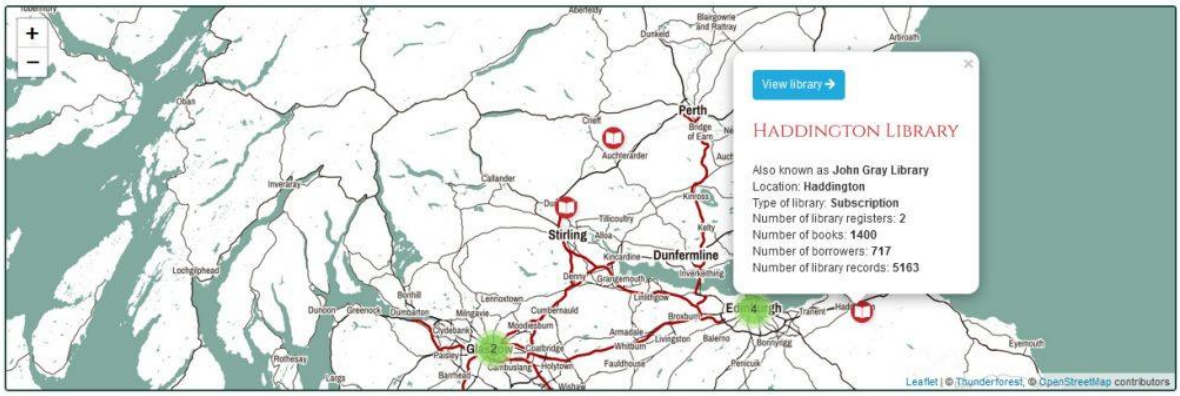
so all the data including the counts gets processed and returned by the database via one HTTP request.

With the data retrieval aspects of the 'browse libraries' page completed I then moved on to developing the page itself. It has an introductory section (with placeholder text for now) then a map showing the locations of the libraries. Any libraries that currently have lat/lng data appear on this map. The markers are clustered when zoomed out, with the number referring to the number of libraries in the cluster. I selected a map design that I thought fitted in with the site, but this might change, and I used an open book icon for the library map marker on a red background (to match the site's header text colour) and again this may change. You can hover over a marker to see the library name and press on a marker to open a popup containing a link to the library, the library name and alternative names, location, foundation date, type and statistics about registers, books, borrowers and records.

Beneath the map is a tabular view of the data. This is the exact same data as is found on the map. Library names are buttons leading to the library's page. You can change the order of the table by pressing on a heading (e.g. to see which library has the most books). Pressing a second time reverses the order. Below is a screenshot showing the map and the table, with the table ordered by number of borrowing records:

LIBRARIES

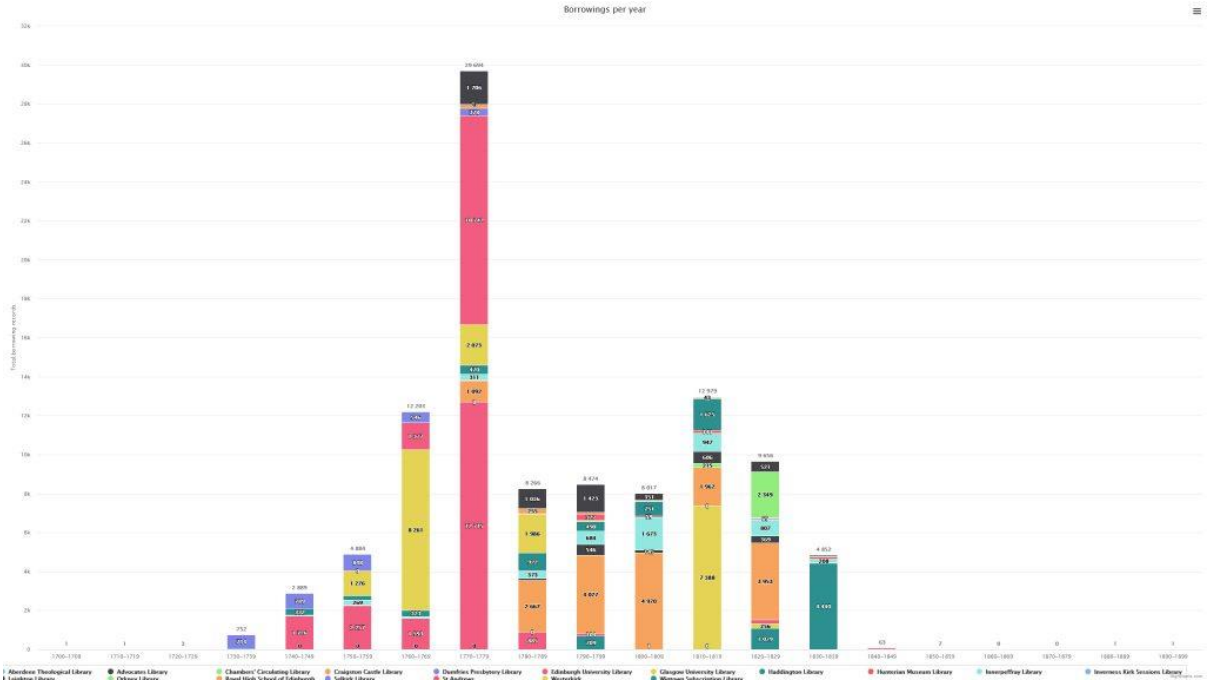
Introductory text will go here.



Name	Location	Foundation date	Type	Registers	Books	Borrowers	Borrowing records
St Andrews →	St Andrews		Institutional	25	8232	1509	26596
Royal High School of Edinburgh →			Institutional	5	1932	2403	20012
Glasgow University	Glasgow	1451	Institutional	8	3460	1012	15369

Beneath the table is a stacked column chart showing borrowings at the libraries over time that I created using the extremely useful HighCharts JavaScript library (See https://www.highcharts.com/demo). At the moment the borrowing records start somewhere between 1700 and 1710 and end somewhere between 1890 and 1899. Actually, there are some borrowing records beyond even this but are presumably mistakes (e.g. one had a year of '179'

or something like that). As generating a graph with a bar for each year would result in about 200 bars I decided this wasn't feasible and instead grouped borrowings into decades. This sort of works, but we still have many decades at the start and end that only have a few records, but we may limit the decades we focus on. We're also visualising the data from 18 libraries in the chart, which is a lot. This takes up a lot of space under the chart (where you can hover over a name to highlight the data in the bars). However, you can open the menu to view the chart full screen, which makes it more legible. You can also view the year data in a table by selecting the 'data table' option. Below is a screenshot of the bar chart:



There are a couple of things I could do to make this more legible if required. Firstly, we could use a stacked bar chart instead (https://www.highcharts.com/demo/bar-stacked). The years would then be on the y-axis and we could have a very long chart with all of the years in place rather than aggregating to decades. This would make it more difficult to view the legend and the x-axis tick marks, as you would need to scroll down to see them. Secondly, we could stick with the decade view but then give the user the option of selecting a decade to view a new chart featuring the individual years in that decade. This would make it harder for users to get the big picture all at once, although I guess the decade view would give that.

6th June 2022

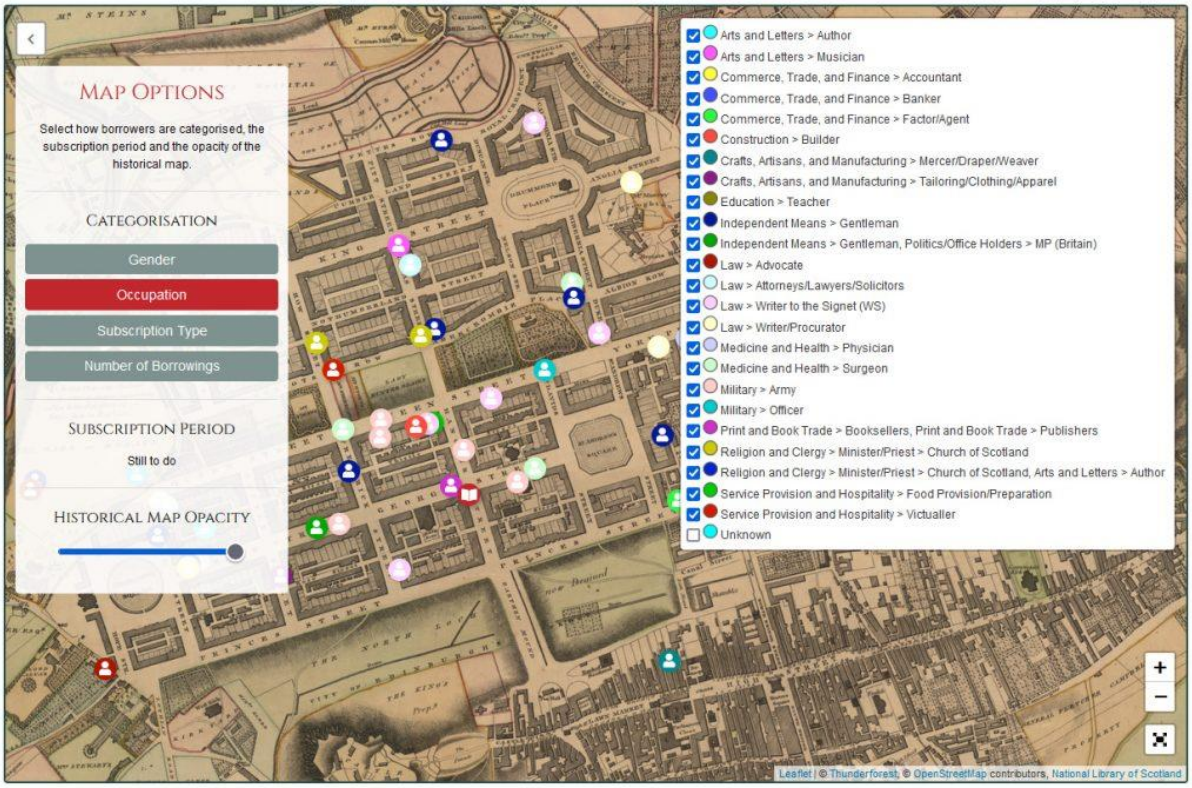
We'd been sent a new library register from the NLS and I spent a bit of time downloading the 700 or so images, processing them and uploading them into our system. As usual, page numbers go a bit weird. Page 632 is written as 634 and then after page 669 comes not 670 but 700! I ran my script to bring the page numbers in the system into line with the oddities of the written numbers. On Friday I downloaded a further library register which I'll need to process next week.

My main focus for the project was the Chambers Library interactive map sub-site. The map features the John Ainslie 1804 map from the NLS, and currently it uses the same modern map as I've used elsewhere in the front-end for consistency, although this may change. The map

defaults to having a ‘Map options’ pane open on the left, and you can open and close this using the button above it. I also added a ‘Full screen’ button beneath the zoom buttons in the bottom right. I also added this to the other maps in the front-end too. Borrower markers have a ‘person’ icon and the library itself has the ‘open book’ icon as found on other maps.

By default the data is categorised by borrower gender, with somewhat stereotypical (but possibly helpful) blue and pink colours differentiating the two. There is one borrower with an ‘unknown’ gender and this is set to green. The map legend in the top right allows you to turn on and off specific data groups. The screenshot below shows this categorisation:

The next categorisation option is occupation, and this has some problems. The first is there are almost 30 different occupations, meaning the legend is awfully long and so many different marker colours are needed that some of them are difficult to differentiate. Secondly, most occupations only have a handful of people. Thirdly, some people have multiple occupations, and if so these are treated as one long occupation, so we have both ‘Independent Means > Gentleman’ and then ‘Independent Means > Gentleman, Politics/Office Holders > MP (Britain)’. It would be tricky to separate these out as the marker would then need to belong to two sets with two colours, plus what happens if you hide one set? I wonder if we should just use the top-level categorisation for the groupings instead? This would result in 12 groupings plus ‘unknown’, meaning the legend would be both shorter and narrower. Below is a screenshot of the occupation categorisation as it currently stands:



The next categorisation is subscription type, which I don't think needs any explanation. I then decided to add in a further categorisation for number of borrowings, which wasn't originally discussed but as I used the page I found myself looking for an option to see who borrowed the most, or didn't borrow anything. I added the following groupings, but these may change: 0, 1-10, 11-20, 21-50, 51-70, 70+ and have used a sequential colour scale (darker = more borrowings). We might want to tweak this, though, as some of the colours are a bit too similar. I haven't added in the filter to select subscription period yet, but will look into this next week.

At the bottom of the map options is a facility to change the opacity of the historical map so you can see the modern street layout. This is handy for example for figuring out why there is a cluster of markers in a field where 'Ainslie Place' was presumably built after the historical map was produced.

I decided to not include the marker clustering option in this map for now as clustering would make it more difficult to analyse the categorisation as markers from multiple groupings would end up clustered together and lose their individual colours until the cluster is split. Marker hover-overs display the borrower name and the pop-ups contain information about the borrower. I still need to add in the borrowing period data, and also figure out how best to link out to information about the borrowings or page images. The Chambers Library pin displays the same information as found in the 'libraries' page you've previously seen.

13th June 2022

I processed and imported a further register for the Advocates library that had been digitised by the NLS. I also continued with the interactive map of Chambers library borrowers, although I

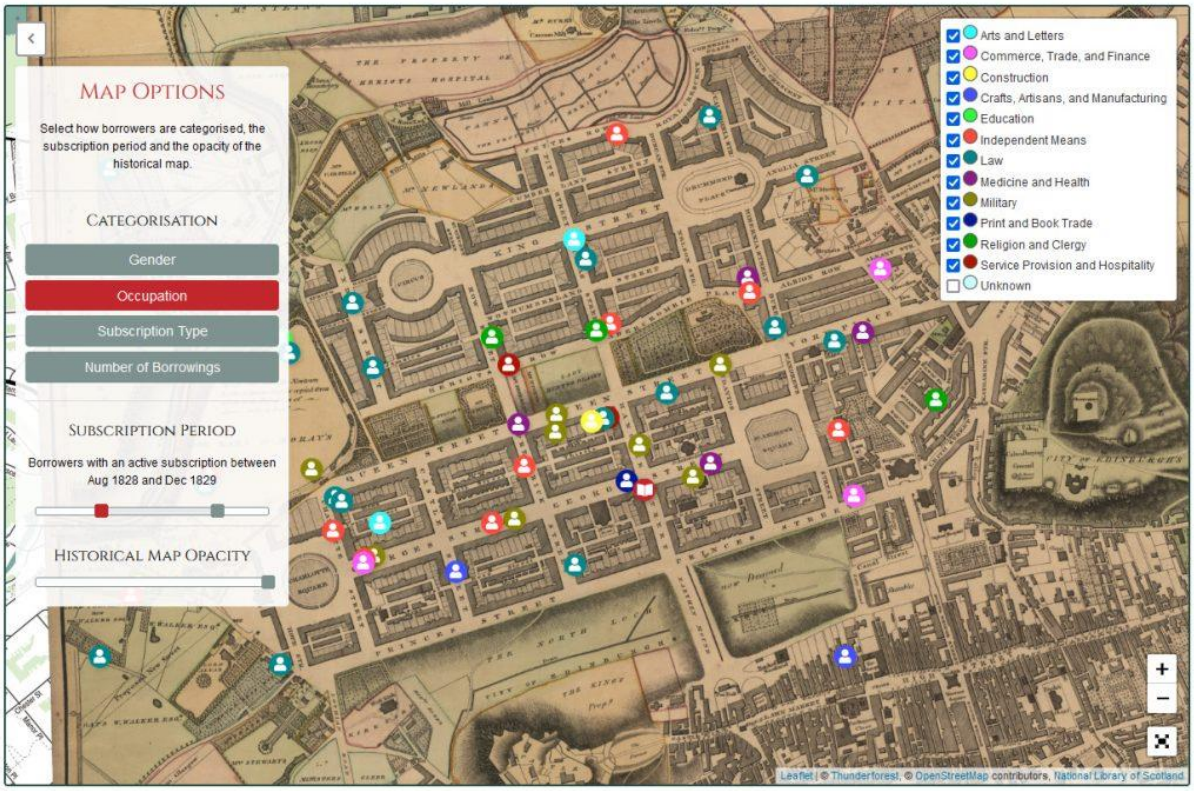
couldn't spend as much time on this as I'd hoped as my access to Stirling University's VPN had stopped working and without VPN access I can't connect to the database and the project server. It took a while to resolve the issue as access needs to be approved by some manager or other, but once it was sorted I got to work on some updates.

One thing I'd noticed last week was that when zooming and panning the historical map layer was throwing out hundreds of 403 Forbidden errors to the browser console. This was not having any impact on the user experience, but was still a bit messy and I wanted to get to the bottom of the issue. I had a very helpful (as always) chat with Chris Fleet at NLS Maps, who provided the historical map layer and he reckoned it was because the historical map only covers a certain area and moving beyond this was still sending requests for map tiles that didn't exist. Thankfully an option exists in Leaflet that allows you to set the boundaries for a map layer (https://leafletjs.com/reference.html#latlngbounds) and I updated the code to do just that, which seems to have stopped the errors.

I then returned to the occupations categorisation, which was including far too many options. I therefore streamlined the occupations, displaying the top-level occupation only. I think this works a lot better (although I need to change the icon colour for 'unknown'). Full occupation information is still available for each borrower via the popup.

I also had to change the range slider for opacity as standard HTML range sliders don't allow for double-ended ranges. We require a double-ended range for the subscription period and I didn't want to have two range sliders that looked different on one page. I therefore switched to a range slider offered by the jQuery UI interface library (https://jqueryui.com/slider/#range). The opacity slider still works as before, it just looks a little different. Actually, it works better than before, as the opacity now changes as you slide rather than only updating after you mouse-up.

I then began to implement the subscription period slider. This does not yet update the data. It's been pretty tricky to implement this. The range needs to be dynamically generated based on the earliest and latest dates in the data, and dates are both year and month, which need to be converted into plain integers for the slider and then reinterpreted as years and months when the user updates the end positions. I think I've got this working as it should, though. When you update the ends of the slider the text above that lists the months and years updates to reflect this. The next step will be to actually filter the data based on the chosen period. Here's a screenshot of the map featuring data categorised by the new streamlined occupations and the new sliders displayed:



20th June 2022

I completed an initial version of the Chambers Library map for the Books and Borrowing project this week. It took quite a lot of time and effort to implement the subscription period range slider. Searching for a range when the data also has a range of dates rather than a single date means we needed to make a decision about what data gets returned and what doesn't. This is because the two ranges (the one chosen as a filter by the user and the one denoting the start and end periods of subscription for each borrower) can overlap in many different ways. For example, the period chosen by the user is **05 1828 to 06 1829**. Which of the following borrowers should therefore be returned?

1. Borrowers range is **06 1828 to 02 1829**: Borrower's range is fully within the period so should definitely be included
2. Borrowers range is **01 1828 to 07 1828**: Borrower's range extends beyond the selected period at the start and ends within the selected period. Presumably should be included.
3. Borrowers range is **01 1828 to 09 1829**: Borrower's range extends beyond the selected period in both directions. Presumably should be included.
4. Borrowers range is **05 1829 to 09 1829**: Borrower's range begins during the selected period and ends beyond the selected period. Presumably should be included.
5. Borrowers range is **01 1828 to 04 1828**: Borrower's range is entirely before the selected period. Should not be included
6. Borrowers range is **07 1829 to 10 1829**: Borrower's range is entirely after the selected period. Should not be included.

Basically if there is any overlap between the selected period and the borrower's subscription period the borrower will be returned. But this means most borrowers will always be returned a lot of the time. It's a very different sort of filter to one that purely focuses on a single date – e.g. filtering the data to only those borrowers whose subscription periods *begins* between 05 1828 and 06 1829.

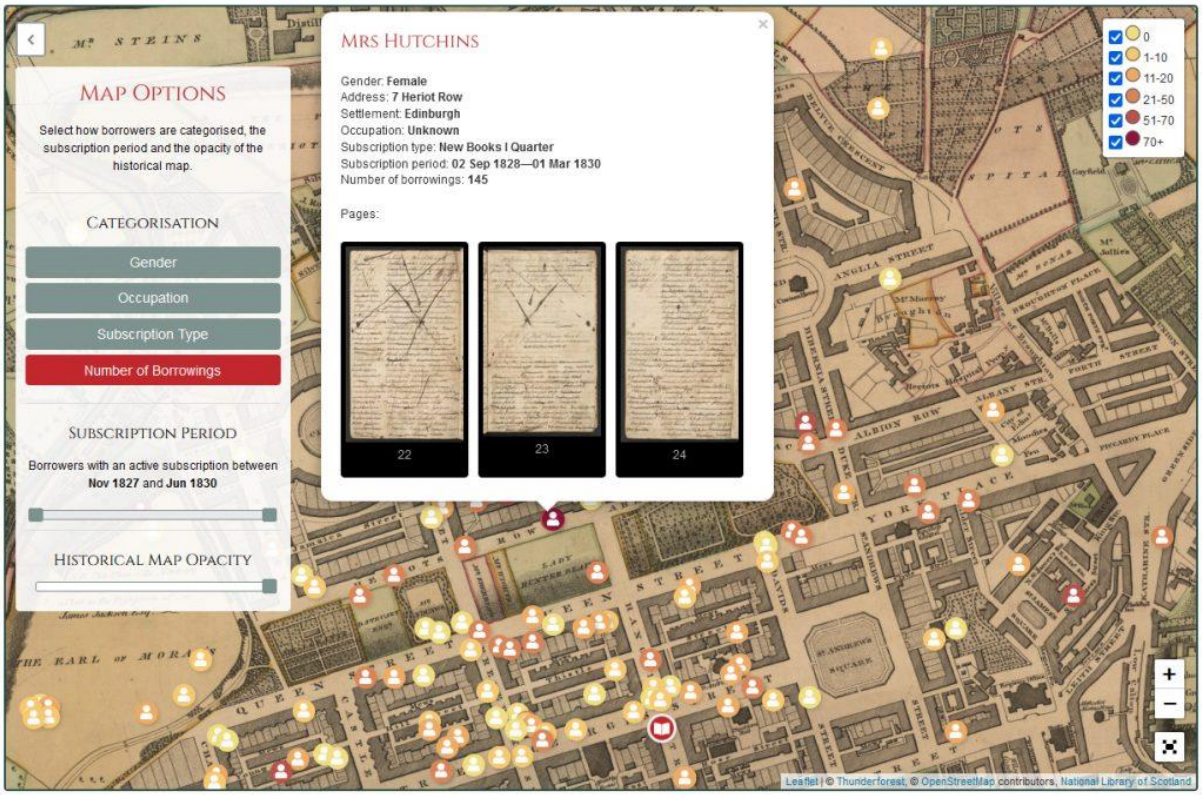
Based on the above assumptions I began to write the logic that would decide which borrowers to include when the range slider is altered. It was further complicated by having to deal with months as well as years. Here's the logic in full if you fancy getting a headache:

```
if(((mapData[i].sYear>startYear || (mapData[i].sYear==startYear &&
mapData[i].sMonth>=startMonth)) && ((mapData[i].eYear==endYear && mapData[i].eMonth
<=endMonth) || mapData[i].eYear<endYear)) || ((mapData[i].sYear<startYear
||(mapData[i].sYear==startYear && mapData[i].sMonth<=startMonth)) &&
((mapData[i].eYear==endYear && mapData[i].eMonth >=endMonth) ||
mapData[i].eYear>endYear)) || ((mapData[i].sYear==startYear &&
mapData[i].sMonth<=startMonth || mapData[i].sYear>startYear) &&
((mapData[i].eYear==endYear && mapData[i].eMonth <=endMonth) ||
mapData[i].eYear<endYear) && ((mapData[i].eYear==startYear && mapData[i].eMonth
>=startMonth) || mapData[i].eYear>startYear)) || (((mapData[i].sYear==startYear &&
mapData[i].sMonth>=startMonth) || mapData[i].sYear>startYear) &&
((mapData[i].sYear==endYear && mapData[i].sMonth <=endMonth) ||
mapData[i].sYear<endYear) && ((mapData[i].eYear==endYear && mapData[i].eMonth
>=endMonth) || mapData[i].eYear>endYear)) || ((mapData[i].sYear<startYear
||(mapData[i].sYear==startYear && mapData[i].sMonth<=startMonth)) &&
((mapData[i].eYear==startYear && mapData[i].eMonth >=startMonth) ||
mapData[i].eYear>startYear)))
```

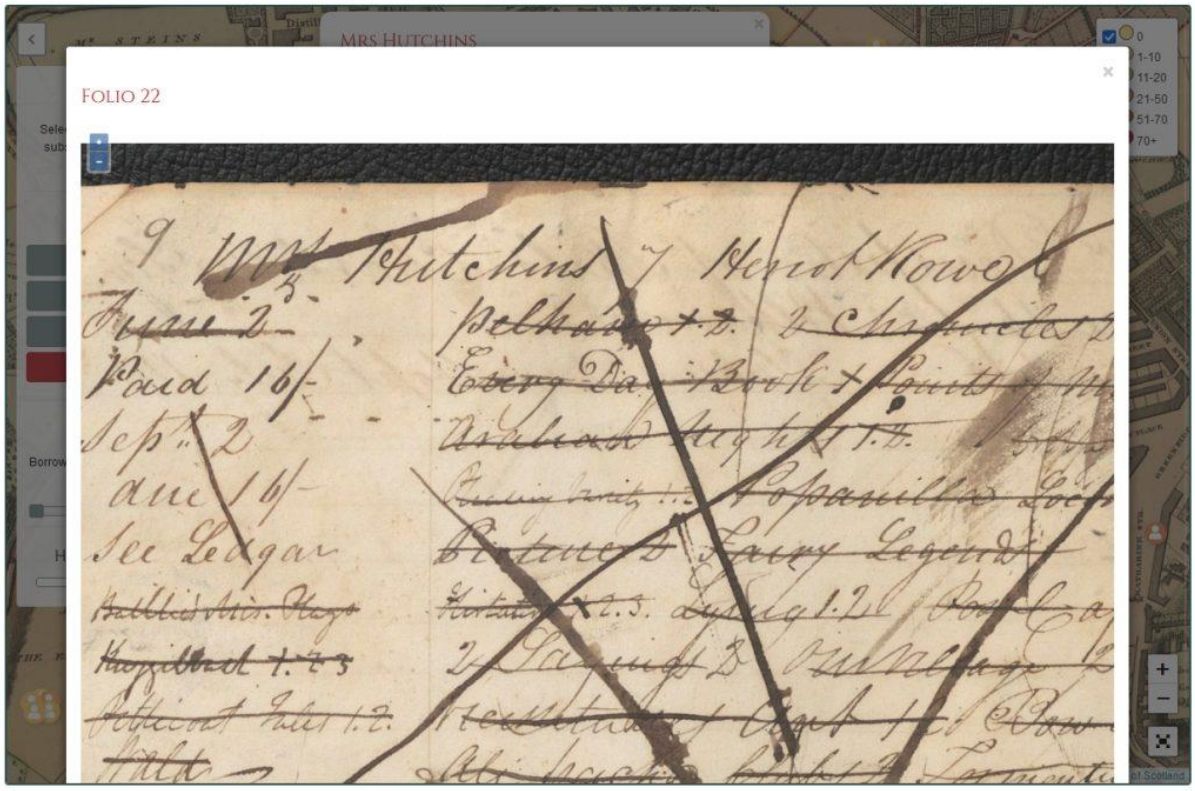
I also added the subscription period to the popups. The only downside to the range slider is that the occupation marker colours change depending on how many occupations are present during a period, so you can't always tell an occupation by its colour. I might see if I can fix the colours in place, but it might not be possible.

I also noticed that the jQuery UI sliders weren't working very well on touchscreens so installed the jQuery TouchPunch library to fix that (https://github.com/furf/jquery-ui-touch-punch). I also made the library marker bigger and gave it a white border to more easily differentiate it from the borrower markers.

I then moved onto incorporating page images in the resource too. Where a borrower has borrower records the relevant pages where these borrowing records are found now appear as thumbnails in the borrower popup. These are generated by the IIIF server based on dimensions passed to it, which is much nicer than having to generate and store thumbnails directly. I also updated the popup to make it wider when required to give more space for the thumbnails. Here's a screenshot of the new thumbnails in action:



Clicking on a thumbnail opens a further popup containing a zoomable / pannable image of the page. This proved to be rather tricky to implement. Initially I was going to open a popup in the page (outside of the map container) using a jQuery UI Dialog. However, I realised that this wouldn't work when the map was being viewed in full-screen mode, as nothing beyond the map container is visible in such circumstances. I then considered opening the image in the borrower popup but this wasn't really big enough. I then wondered about extending the 'Map options' section and replacing the contents of this with the image, but this then caused issues for the contents of the 'Map options' section, which didn't reinitialise properly when the contents were reinstated. I then found a plugin for the Leaflet mapping library that provides a popup within the map interface (https://github.com/w8r/Leaflet.Modal) and decided to use this. However, it's all a little complex as the popup then has to include another mapping library called OpenLayers that enables the zooming and panning of the page image, all within the framework of the overall interactive map. It is all working and I think it works pretty well, although I guess the map interface is a little cluttered, what with the 'Map Options' section, the map legend, the borrower popup and then the page image popup as well. Here's a screenshot with the page image open:



All that's left to do now is add in the introductory text once Alex has prepared it and then make the map live. We might need to rearrange the site's menu to add in a link to the Chambers Map as it's already a bit cluttered.

Also for the project I downloaded images for two further library registers for St Andrews that had previously been missed. However, there are already records for the registers and pages in the CMS so we're going to have to figure out a way to work out which image corresponds to which page in the CMS. One register has a different number of pages in the CMS compared to the image files so we need to work out how to align the start and end and if there are any gaps or issues in the middle. The other register is more complicated because the images are double pages whereas it looks like the page records in the CMS are for individual pages. I'm not sure how best to handle this. I could either try and batch process the images to chop them up or batch process the page records to join them together. I'll need to discuss this further with Gerry, who is dealing with the data for St Andrews.

4th July 2022

Whilst I'd been on holiday I'd been sent page images for a further ten library registers and I needed to process these. This can be something of a time-consuming process as each set of images needs to be processed in a different way, such as renaming images, removing unnecessary images at the start and end, uploading the images to the server, generating the page images for each register and then bringing the automatically generated page numbers into line with any handwritten page numbers on the images, which may not always be sequentially numbered. I processed two registers for the Advocates library from the NLS and three registers from Aberdeen library. I looked into processing the images for a register from the High School of Edinburgh, but I had some questions about the images and didn't hear back from the

researcher before the end of the week, so I needed to leave these. The remaining registers were from St Andrews and I had further questions about these, as the images are double-page spreads but existing page records in the CMS treat each page separately. As the researcher dealing with St Andrews was on holiday I'll need to wait until I'm back to deal with these too.

25th July 2022

I published the Chambers Library Map (https://borrowing.stir.ac.uk/chambers-library-map/) and reorganised the site menu to make space for the new page link. The resource has been very well received and I'm pretty pleased with how it's turned out.

8th August 2022

I uploaded images for one more Advocates library register from the NLS, including generating pages, associating images and fixing the page numbering to align with the handwritten numbers. I also received images for a second register for Haddington library from the NLS, and I needed some help with this as we already have existing pages for this register in the CMS, but the number of images received didn't match. Thankfully the RA Kit Baston was able to look over the images and figure out what needed to be done, which included inserting new pages in the CMS and then me writing a script to associate images with records. I also added two missing pages to the register for Dumfries Presbytery and added in a missing image for Westerkirk library.

15th August 2022

I added two further library registers from the NLS to our system. This means there should now only be one further register to come from the NLS, which is quite a relief as each register takes some time to process. I also finally got round to processing the four registers for St Andrews, which had been on my 'to do' list since late July. It was very tricky to rename the images into a format that we can use on the server because the lack of trailing zeros meant a script to batch process the images loaded them in the wrong order. The was made worse because rather than just being numbered sequentially the image filenames were further split into 'parts'. For example, the images beginning 'UYLY 207 11 Receipt book part 11' were being processed before images beginning 'UYLY 207 11 Receipt book part 2' as programming languages when ordering strings consider 11, 12 etc to come before 2. This was also then happening within each 'part', e.g. 'UYLY207 15 part 43_11.jpg' was coming before 'UYLY207 15 part 43_2.jpg'. It took most of the morning to sort this out, but I was then able to upload the images to the server and I create new registers, generate pages and associate images for the two new registers (207-11 and 207-15).

However, the other two registers already exist in the CMS as page records with associated borrowing records. Each image of the register is an open spread showing two borrowing pages and we had previously decided that I should run a script to merge pages in the CMS and then associate the merged record with one of the page images. However, I'm afraid this is going to need some manual intervention. Looking at the images for 206-1 and comparing them to the existing page records for this register, it's clear that there are many blank pages in the two-page spreads that have not been replicated in the CMS. For example, page 164 in the CMS is for 'Profr Spens'. The corresponding image (in my renamed images) is 'UYLY206-1_00000084.jpg'. The data is on the right-hand page and the left-hand page is blank. But in the CMS the preceding page is for 'Prof. Brown', which is on the left-hand page of the preceding

image. If I attempted to automatically merge these two page records into one this would therefore result in an error.

I'm afraid what I need is for someone who is familiar with the data to look through the images and the pages and create a spreadsheet noting which pages correspond to which image. Where multiple pages correspond to one page I can then merge the records. So for example: Pages 159 (id 1087) and 160 (ID 1088) are found on image UYLY206-1_00000082.jpg. Page 161 (1089) corresponds to UYLY206-1_00000083.jpg. The next page in the CMS is 164 (1090) and this corresponds to UYLY206-1_00000084.jpg. So a spreadsheet could have two columns:

Page ID	Image
1087	UYLY206-1_00000082.jpg
1088	UYLY206-1_00000082.jpg
1089	UYLY206-1_00000083.jpg
1900	UYLY206-1_00000084.jpg

Also, the page numbers in the CMS don't tally with the handwritten page numbers in the images (e.g. the page record 1089 mentioned above has page 161 but the image has page number 162 written on it). And actually, the page numbers would need to include two pages, e.g. 162-163. Ideally whoever is going to manually create the spreadsheet could add new page numbers as a further column and I could then fix these when I process the spreadsheet too. This task is still very much in progress.

Also for the project this week I created a 'full screen' version of the Chambers map that will be pulled into an iframe on the Edinburgh University Library website when they create an online exhibition based on our resource.

22nd August 2022

I had to make a few tweaks to the Chambers map of borrowers to make the map work better on smaller screens. I ensured that both the 'Map options' section on the left and the 'map legend' on the right are given a fixed height that is shorter than the map and the areas become scrollable, as I'd noticed that on short screens both these areas could end up longer than the map and therefore their lower parts were inaccessible. I've also added a 'show/hide' button to the map legend, enabling people to hide the area if it obscures their view of the map.

I also sent on some renamed library register files from St Andrews to Gerry for him to align with existing pages in the CMS, replaced some of the page images for the Dumfries register and renamed and uploaded images for a further St Andrews register that already existed in the CMS, ensuring the images became associated with the existing pages.

I started to work on the images for another St Andrews register that already exists in the system, but for this one the images are a double page spread so I need to merge two pages into one in the CMS. The script needs to find all odd numbered pages then move the records on these to the preceding even numbered page, and at the same time regenerate the 'page order' for each

record so they follow on from the existing records. Then the even page needs its folio number updated to add in the odd number (e.g. so folio number 2 becomes '2-3'. Then I need to delete the odd page record and after all that is done I need to regenerate the 'next' and 'previous' page links for all pages. I completed everything except the final task, but I really need to test the script out on a version of the database running on my local PC first, as if anything goes wrong data could very easily be lost. I'll need to tackle this next week as I ran out of time this week.

29th August 2022

On Friday last week I didn't quite get to finish a script to merge page records for one of the St Andrews registers as it needed further testing on my local PC before I ran it on the live data. I tackled this issue first thing on Monday and it was a task I had hoped would only take half an hour or so. Unfortunately things did not go well and it took most of the morning to sort out. I initially attempted to run things on my local PC to test everything out, but I forgot to update the database connection details. Usually this wouldn't be an issue as generally the databases I work with use 'localhost' as a connection URL, so the Stirling credentials would have been wrong for my local DB and the script would have just quit, but Stirling (where the system is hosted) uses a full URL instead of 'localhost'. This meant that even though I had a local copy of the database on my PC and the scripts were running on a local server set up on my PC the scripts were in fact connecting to the real database at Stirling. This meant the live data was being changed. I didn't realise this as the script was running and as it was taking some time I cancelled it, meaning the update quit halfway through changing borrowing records and deleting page records in the CMS.

I then had to write a further script to delete all of the page and borrowing records for this register from the Stirling server and reinstate the data from my local database. Thankfully this worked ok. I then ran my test script on the actual local database on my PC and the script did exactly what I wanted it to do, namely:

Iterate through the pages and for each odd numbered page move the records on these to the preceding even numbered page, and at the same time regenerate the 'page order' for each record so they follow on from the existing records. Then the even page needs its folio number updated to add in the odd number (e.g. so folio number 2 becomes '2-3') and generate an image reference based on this (e.g. UYLY207-2_2-3). Then delete the odd page record and after all that is done regenerate the 'next' and 'previous' page links for all pages.

This all worked so I ran the script on the server and updated the live data. However, I then noticed that there are gaps in the folio numbers and this has messed everything up. For example, folio number 314 isn't followed by 315 but 320. 320 isn't an odd number so it doesn't get joined to 314. All subsequent page joins are then messed up. There are also two '350' pages in the CMS and two images that reference 350. We have UYLY207-2_349-350 and also UYLY207-2_350-351. There might be other situations where the data isn't uniform too.

I therefore had to use my 'delete and reinsert' script again to revert to the data prior to the update as my script wasn't set up to work with pages that don't just increment their folio number by 1 each time. After some discussion with the RA I updated the script again so that it would work with the non-uniform data and thankfully all worked fine after that. Later in the

week I also found some time to process two further St Andrews registers that needed their pages and records merged, and thankfully these went much smoother.

5th September 2022

I also continued to process library registers and their images for the Books and Borrowing project. I processed three registers from the Royal High School, each of which required different amounts of processing and different methods. This included renaming images, adding in missing page records, creating entire new runs of page records, uploading hundreds of images and changing the order of certain page records. I also wrote a script to identify which page records still did not have associated image files after the upload, as each of the registers is missing some images.

12th September 2022

I processed the images for another library register. This didn't go entirely smoothly. I had been sent 73 images and these were all upside down so needed rotating. It then transpired that I should have been sent 273 images so needed to chase up the missing ones. Once I'd been sent the full set I was then able to generate the page images for the register, upload the images and associate them with the records.

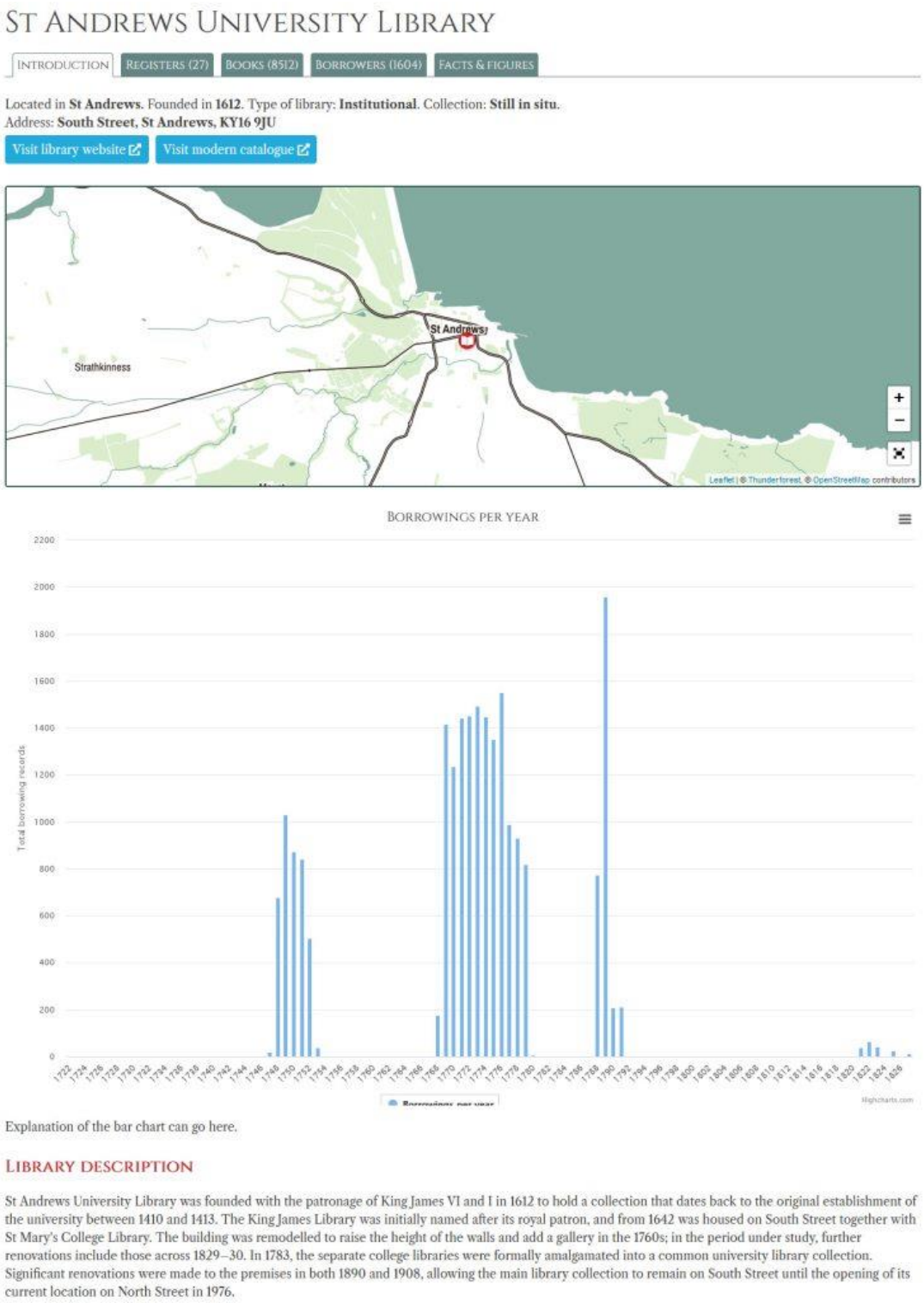
26th September 2022

I spent most of my time this week getting back into the development of the front-end for the Books and Borrowing project. It's been a long time since I was able to work on this due to commitments to other projects and also due to there being a lot more for me to do than I was expecting regarding processing images and generating associated data in the project's content management system over the summer. However, I have been able to get back into the development of the front-end this week and managed to make some pretty good progress. The first thing I did was to make some changes to the 'libraries' page based on feedback I received ages ago from the project's Co-I Matt Sangster. The map of libraries used clustering to group libraries that are close together when the map is zoomed out, but Matt didn't like this. I therefore removed the clusters and turned the library locations back into regular individual markers. However, it is now rather difficult to distinguish the markers for a number of libraries. For example, the markers for Glasgow and the Hunterian libraries (back when the University was still on the High Street) are on top of each other and you have to zoom in a very long way before you can even tell there are two markers there.

I also updated the tabular view of libraries. Previously the library name was a button that when clicked on opened the library's page. Now the name is text and there are two buttons underneath. The first one opens the library page while the second pans and zooms the map to the selected library, whilst also scrolling the page to the top of the map. This uses Leaflet's 'flyTo' function which works pretty well, although the map tiles don't quite load in fast enough for the automatic 'zoom out, pan and zoom in' to proceed as smoothly as it ought to.

After that I moved onto the library page, which previously just displayed the map and the library name. I updated the tabs for the various sections to display the number of registers, books and borrowers that are associated with the library. The Introduction page also now features the

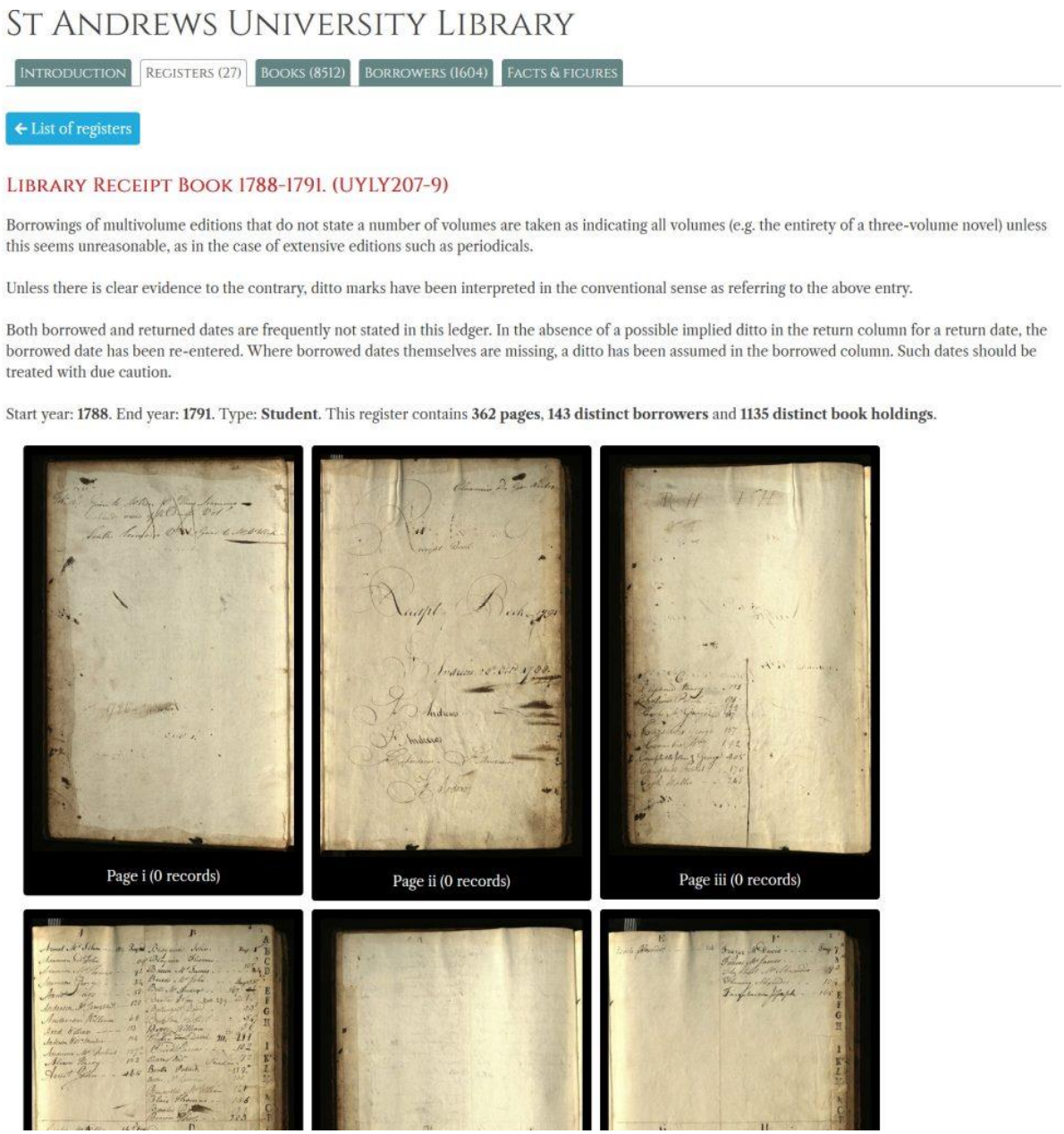
information recorded about the library that has been entered into the CMS. This includes location information, dates, links to the library etc. Beneath the summary info there is the map, and beneath this is a bar chart showing the number of borrowings per year at the library. Beneath the bar chart you can find the longer textual fields about the library such as descriptions and sources. Here's a screenshot of the page for St Andrews:



I also worked on the 'Registers' tab, which now displays a tabular list of the selected library's registers, and I also ensured that when you select one of the tabs other than 'Introduction' the

page automatically scrolls down to the top of the tabs to avoid the need to manually scroll past the header image (but we still may make this narrower eventually). The tabular list of registers can be ordered by any of the columns and includes data on the number of pages, borrowers, books and borrowing records featured in each.

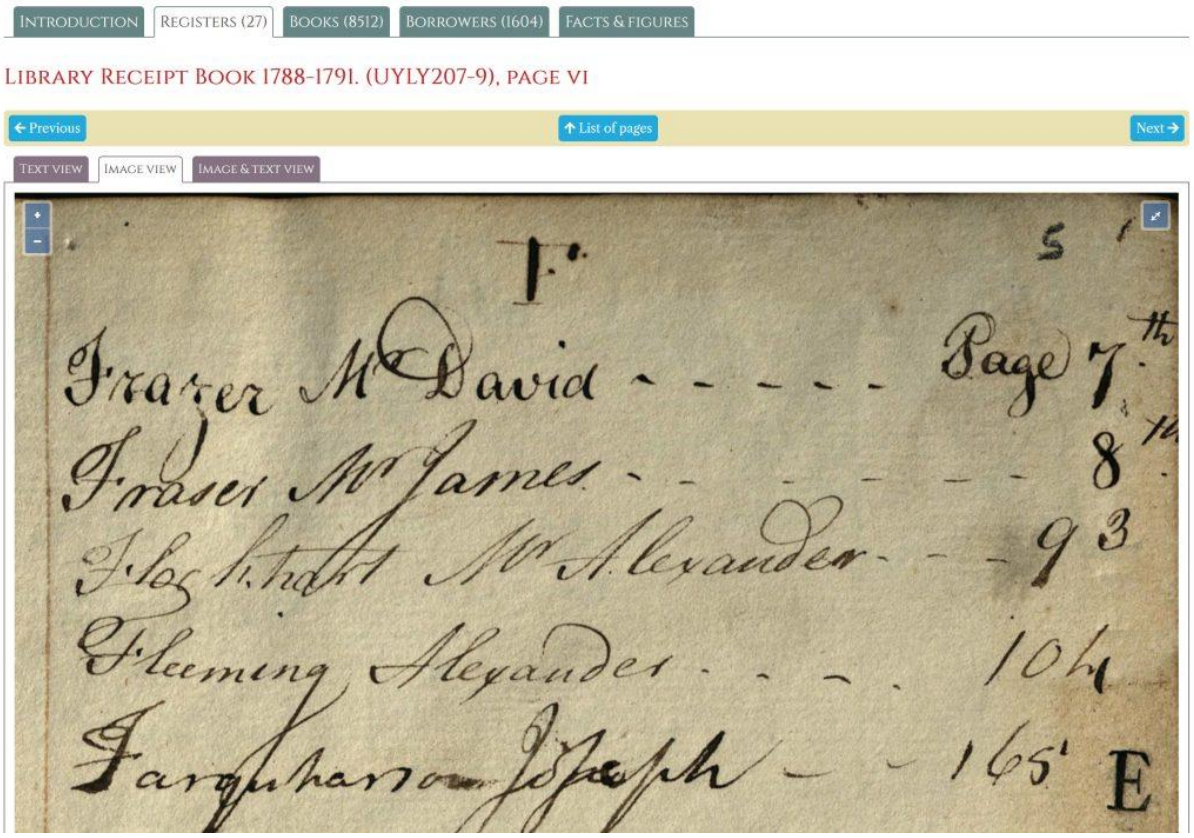
When you open a register the information about it is displayed (e.g. descriptions, dates, stats about the number of books etc referenced in the register) and large thumbnails of each page together with page numbers and the number of records on each page are displayed. The thumbnails are rather large and I could make them smaller, but doing so would mean that all the pages end up looking the same – beige rectangles. The thumbnails are generated on the fly by the IIIF server and the first time a register is loaded it can take a while for the thumbnails to load in. However, generated thumbnails are then cached on the server so subsequent page loads are a lot quicker. Here's a screenshot of a register page for St Andrews:



One thing I also did was write a script to add in a new 'pageorder' field to the 'page' database table. I then wrote a script that generated the page order for every page in every register in the system. This picks out the page that has no preceding page and iterates through pages based on the 'next page' ID. Previously pages in lists were ordered by their auto-incrementing ID, but this meant that if new pages needed to be inserted for a register they ended up stuck at the end of the list, even though the 'next' and 'previous' links worked successfully. This new 'pageorder' field ensures lists of pages are displayed in the proper order. I've updated the CMS to ensure this new field is used when viewing a register, although I haven't as of yet updated the CMS to regenerate the 'pageorder' for a register if new pages are added out of sequence. For now if this happens I'll need to manually run my script again to update things.

Anyway, back to the front-end: The new 'pageorder' is used in the list of pages mentioned above so the thumbnails get displaying in the correct order. I may add pagination to this page, as all of the thumbnails are currently on one page and it can take a while to load, although these days people seem to prefer having long pages rather than having data split over multiple pages.

The final section I worked on was the page for viewing an actual page of the register, and this is still very much in progress. You can open a register page by pressing on its thumbnail and currently you can navigate through the register using the 'next' and 'previous' buttons or return to the list of pages. I still need to add in a 'jump to page' feature here too. As discussed in the requirements document, there will be three views of the page: Text, Image and Text and Image side-by-side. Currently I have implemented the image view only. Pressing on the 'Image view' tab opens a zoomable / pannable interface through which the image of the register page can be viewed. You can also make this interface full screen by pressing on the button in the top right. Also, if you're viewing the image and you use the 'next' and 'previous' navigation links you will stay on the 'image' tab when other pages load. Here's a screenshot of the 'image view' of the page:



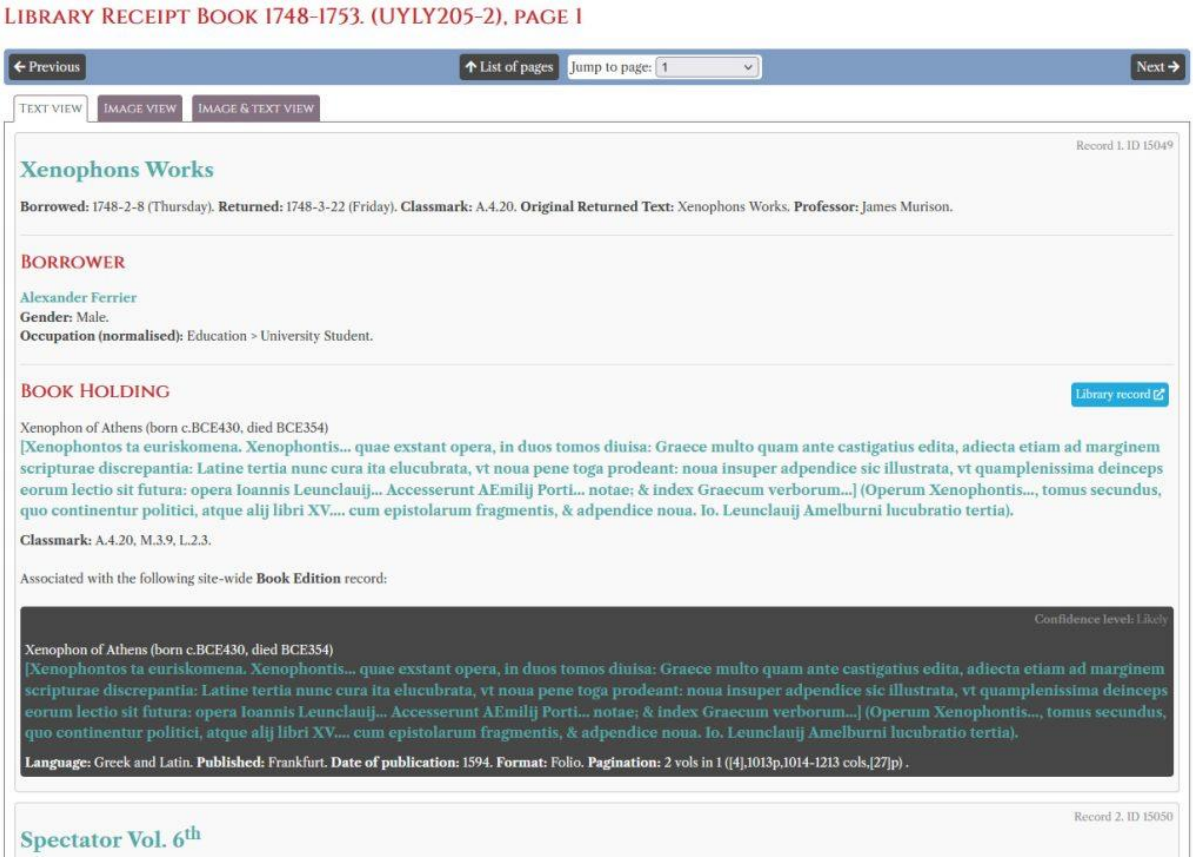
3rd October 2022

I completed an initial version of the 'page' view, including all three views (image, text and image and text). I added in a 'jump to page' feature, allowing you (as you might expect) to jump directly to any page in the register when viewing a page. I also completed the 'text' view of the page, which now features all of the publicly accessible data relating to the records – borrowing records, borrowers, book holding and item records and any associated book editions and book works, plus associated authors. There's an awful lot of data and it took quite a lot of time to think about how best to lay it all out (especially taking into consideration screens of different sizes), but I'm pretty happy with how this first version looks.

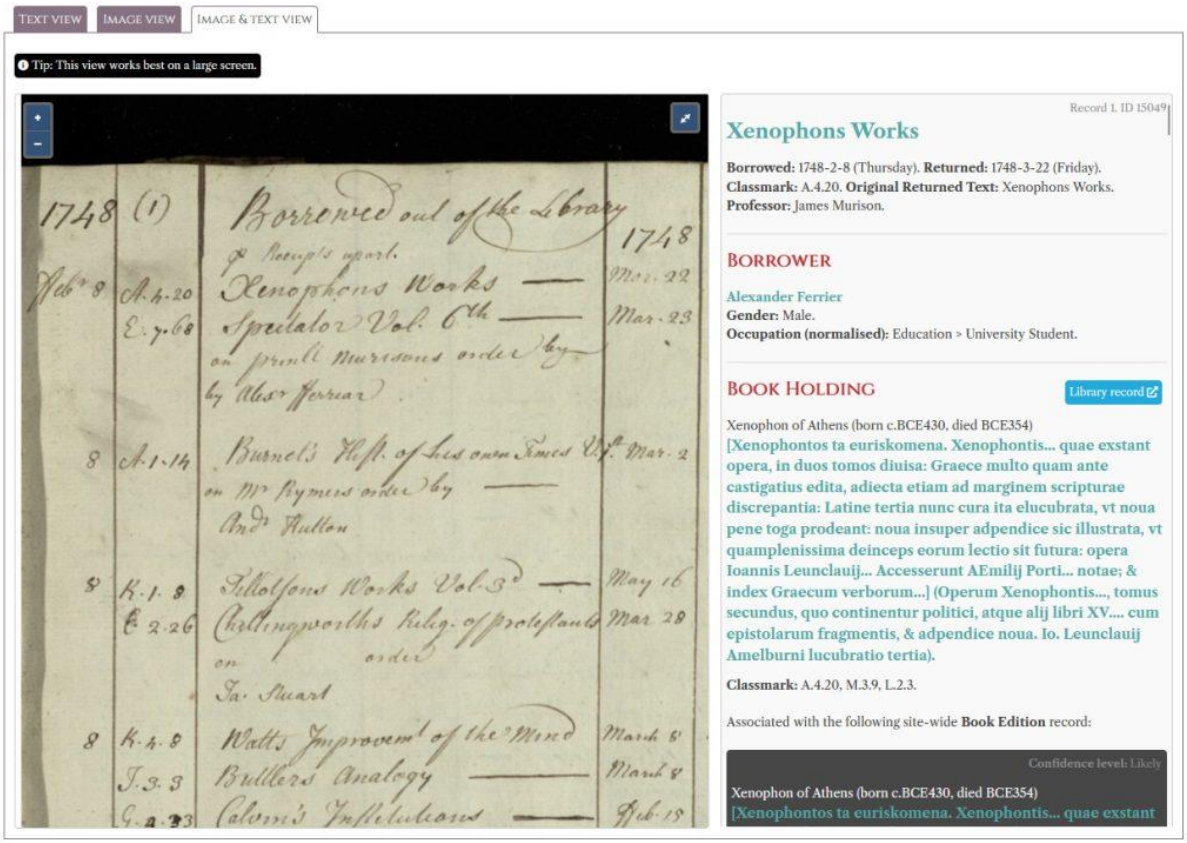
Currently the first thing you see for a record is the transcribed text, which is big and green. Then all fields relating to the borrowing appear under this. The record number as it appears on the page plus the record's unique ID are displayed in the top right for reference (and citation). Then follows a section about the borrower, with the borrower's name in green (I've used this green to make all of the most important bits of text stand out from the rest of the record but the colour may be changed in future). Then follows the information about the book holding and any specific volumes that were borrowed. If there is an associated site-wide book edition record (or records) these appear in a dark grey box, together with any associated book work record (although there aren't many of these associations yet). If there is a link to a library record this appears as a button on the right of the record. Similarly, if there's an ESTC and / or other authority link for the edition these appear to the right of the edition section.

Authors now cascade down through the data as we initially planned. If there's an author associated with a work it is automatically associated with and displayed alongside the edition and holding. If there's an author associated with an editon but not a work it is then associated with the holding. If a book at a specific level has an author specified then this replaces any cascading author from this point downwards in the sequence. Something that isn't in place yet are the links from information to search results, as I haven't developed the search yet. But eventually things like borrower name, author, book title etc will be links allowing you to search directly for the items.

One other thing I've added in is the option to highlight a record. Press anywhere in a record and it is highlighted in yellow. Press again to reset it. This can be quite useful as you're scrolling through a page with lots of records on if there are certain records you're interested in. You can highlight as many records as you want. It's possible that we may add other functionality to this, e.g. the option to download the data for selected records. Here's a screenshot of the text view of the page:



I also completed the 'image and text' view. This works best on a large screen (i.e. not a mobile phone, although it is just about possible to use it on one, as I did test this out). The image takes up about 60% of the screen width and the text takes up the remaining 40%. The height of the records section is fixed to the height of the image area and is scrollable, so you can scroll down the records whilst still viewing the image (rather than the whole page scrolling and the image disappearing off the screen). I think this view works really well and the records are still perfectly usable in the more confined area and it's great to be able to compare the image and the text side by side. Here's a screenshot of the same page when viewing both text and image:



I tested the new interface out with registers from all of our available libraries and everything is looking good to me. Some registers don't have images yet, so I added in a check for this to ensure that the image views and page thumbnails don't appear for such registers. After that I moved onto developing the interface to browse book holdings when viewing a library. I created an API endpoint for returning all of the data associated with holding records for a specified library. This includes all of the book holding data, information about each of the book items associated with the holding record (including the number of borrowing records for each), the total number of borrowing records for the holding, any associated book edition and book work records (and there may be multiple editions associated with each holding) plus any authors associated with the book. Authors cascade down through the record as they do when viewing borrowing records in the page. This is a gigantic amount of information, especially as libraries may have many thousands of book holding records. The API call loads pretty rapidly for smaller libraries (e.g. Chambers Library with 961 book holding records) but for larger ones (e.g. St Andrews with over 8,500 book holding records) the API call takes too long to return the data (in the latter case it takes about a minute and returns a JSON file that's over 6Mb in size). The problem is the data needs to be returned in full in order to do things like order it by largest number of borrowings. Clearly dynamically generating the data each time is going to be too slow so instead I am going to investigate caching the data. For example, that 6Mb JSON file can just site there as an actual file rather than being generated each time. Instead I will write a script to regenerate the cached files and I can run this whenever data gets updated (or maybe once a week whilst the project is still active). I'll continue to work on this next week.

10th October 2022

I worked on the 'books' tab in the library page. I'd started on the API endpoint for this last week, which returned all books for a library and then processed them. This was required as books

have two title fields (standardised and original title), either one of which may be blank so to order to books by title the records first need to be returned to see which 'title' field to use. Also ordering by number of borrowings and by author requires all books to be returned and processed. This works fine for smaller libraries (e.g. Chambers has 961 books) but returning all books for a large library like St Andrews that has more then 8,500 books was taking a long time, and resulting in a JSON file that was over 6MB in size.

I created an initial version of the 'books' page using this full dataset, with tabs across the top for each initial letter of the title (browsing by author and number of borrowings is still to do) and a count of the number of books in each tab also displayed. Book records are then displayed in a similar manner to how they appear in the 'page' view, but with some additional data, namely total counts of the number of borrowings for the book holding record and counts of borrowings of individual items (if applicable). These will eventually be linked to the search.

The page looked pretty good and worked pretty well, but was very inefficient as the full JSON file needed to be generated and passed to the browser every time a new letter was selected. Instead I updated the underlying database to add two new fields to the book holding table. The first stores the initial letter of the title (standardised if present, original if not) and the second stores a count of the total number of borrowings for the holding record. I wrote a couple of scripts to add this data in, and these will need to be run periodically to refresh these cached fields as the do not otherwise get updated when changes are made in the CMS. Having these fields in place means the scripts will be able to pinpoint and return subsets of the books in the library at the database query level rather than returning all data and then subsequently processing it. This makes things much more efficient as less data is being processed at any one time.

I still need to add in facilities to browse the books by initial letter of the author's surname and also facilities to list books by the number of borrowings, but for now you can at least browse books alphabetically by title. Unfortunately for large libraries there is still a lot of data to process even when only dealing with specific initial letters. For example, there are 1063 books beginning with 'T' in St Andrews so the returned data still takes quite a few seconds to load in.

24th October 2022

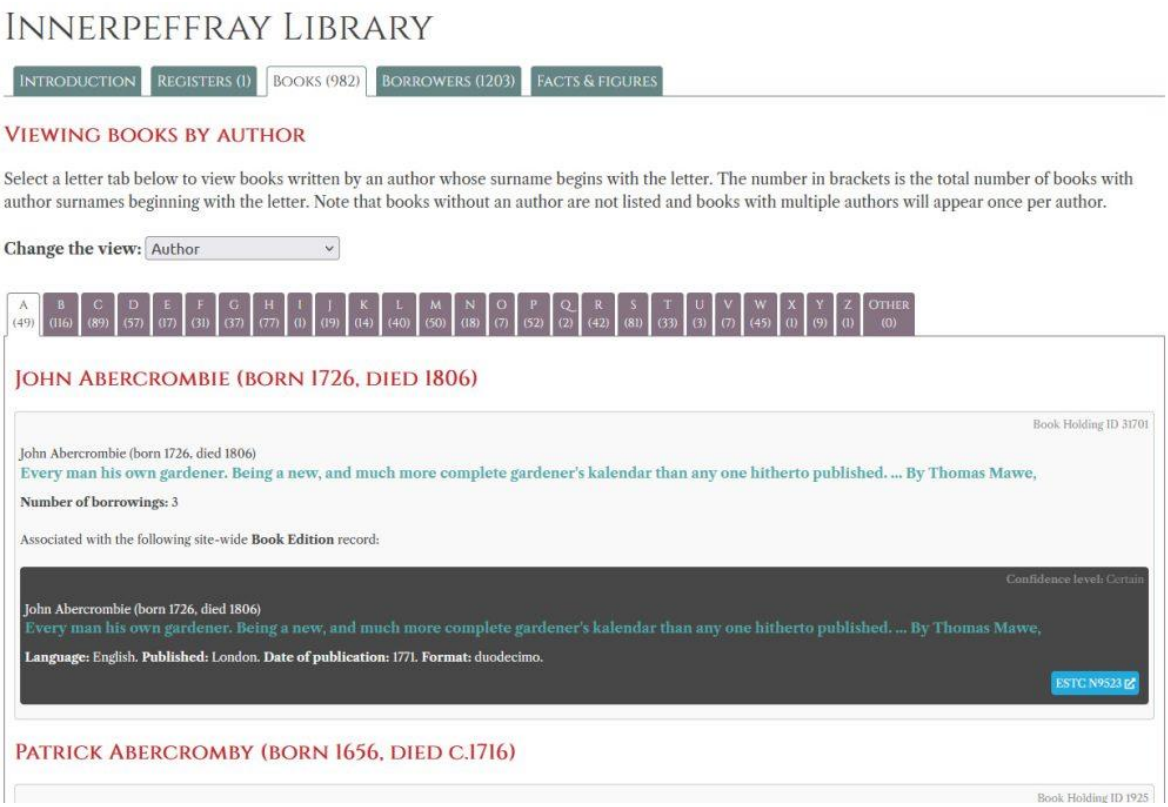
I'd received images for two library registers from Selkirk whilst I was away and I set about integrating them into our system. This required a bit of work to get the images matched up to the page records for the registers that already exist in the CMS. Most of the images are double-pages but the records in the CMS are of single pages marked as 'L' or 'R'. Not all of the double-page images have both 'L' and 'R' in the CMS and some images don't have any corresponding pages in the CMS. For example in Volume 1 we have '1010199l' followed by '1010203l´followed by '1010205l' and then '1010205r'. This seems to be quite correct as the missing pages don't contain borrowing records. However, I still needed to figure out how to match up images and page records. As with previous situations, the options were either slicing the images down the middle to create separate 'L' and 'R' images to match each page or joining the 'L' and 'R' page records in the CMS to make one single record that then matches the double-page image. There are several hundred images so manually chopping them up wasn't really an option, and automatically slicing them down the middle wouldn't work too well as the

page divide is often not in the centre of the image. This then left joining up the page records in the CMS as the best option and I wrote a script to join the page records, rename them to remove the 'L' and 'R' affixes, moving all borrowing records across and renumbering their page order and then deleting the now empty pages. Thankfully it all seemed to work well. I also uploaded the images for the final register from the Royal High School, which thankfully was a much more straightforward process as all image files matched references already stored in the CMS for each page.

I then returned to the development of the front-end for the project. When I looked at the library page I'd previously created I noticed that the interactive map of library locations was failing to load. After a bit of investigation I realised that this was caused by new line characters appearing in the JSON data for the map, which was invalidating the file structure. These had been added in via the library 'name variants' field in the CMS and were appearing in the data for the library popup on the map. I needed to update the script that generated the JSON data to ensure that new line characters were stripped out of the data, and after that the maps loaded again.

Before I went on holiday I'd created a browse page for library books that split the library's books up based on the initial letter of their titles. The approach I'd taken worked pretty well, but St Andrews was still a bit of an issue due to it containing many more books than the other libraries (more than 8,500). Project Co-I Matt Sangster suggested that we should omit some registers from the data as their contents (including book records) are not likely to be worked on during the course of the project. However, I decided to just leave the data in place for now, as excluding data for specific registers would require quite a lot of reworking of the code. The book data for a library is associated directly with the library record and not the specific registers and all the queries would need to be rewritten to check which registers a book appears in. I reckon that if these registers are not going to be tackled by the project it might be better to just delete them, not just to make the system work better but to avoid confusing users with messy data, but I decided to leave everything as it is for now.

This week I added in two further ways of browsing books in a selected library: By author and by most borrowed. A drop-down list featuring the three browse options appears at the top of the 'Books' page now, and I've added in a title and explanatory paragraph about the list type. The 'by author' browse works in a similar manner to the 'by title' browse, with a list of initial letter tabs featuring the initial letter of the author's surname and a count of the number of books that have an author with a surname beginning with this letter. Note that any books that don't have an associated author do not appear in this list. I did think about adding a 'no author' tab as well, but some libraries (e.g. St Andrews) have so many books without specified authors that the data for this tab would take far too long to load in. Note also that if a book has multiple authors then the book will appear multiple times – once for each author. Here's a screenshot of how the interface currently looks:



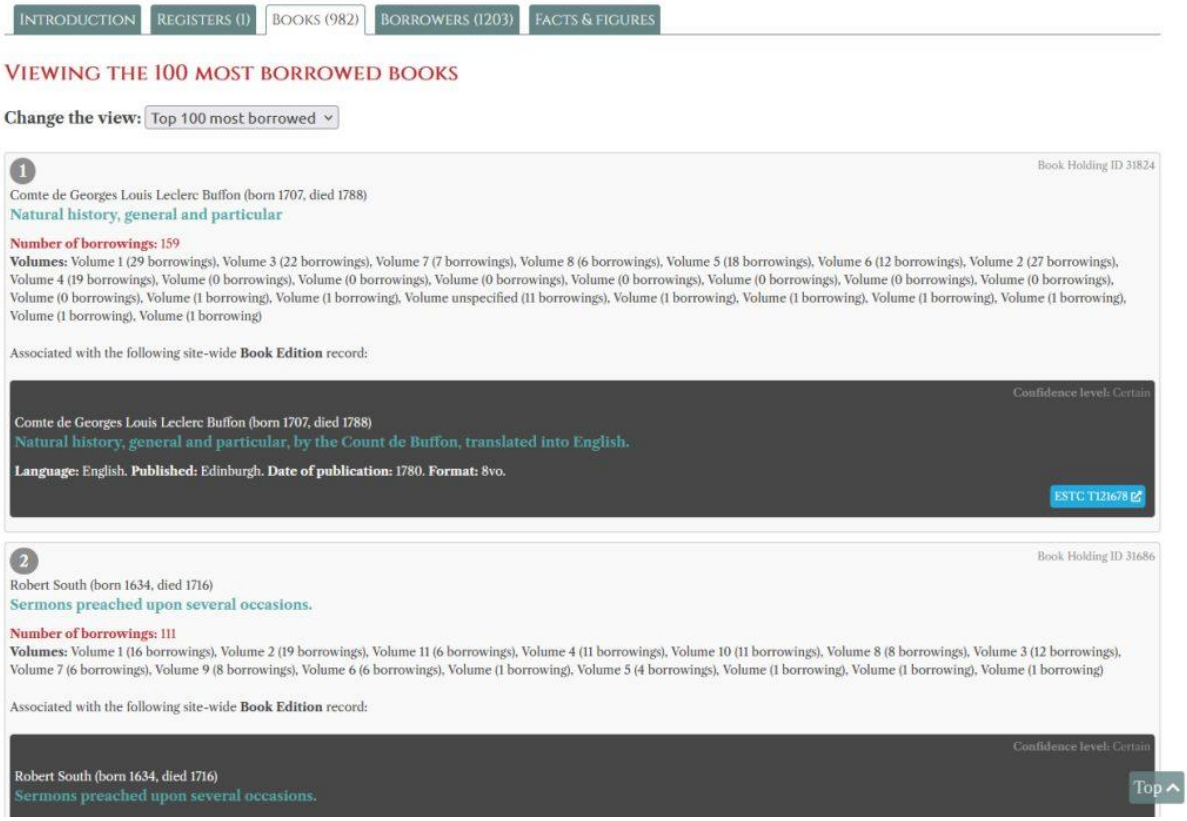
The actual list of books works in a similar way to the 'title' list but is divided by author, with authors appearing with their full name and dates in red above a list of their books. The ordering of the records is by author surname then forename then author ID then book title. This means two authors with the same name will still appear as separate headings with their books ordered alphabetically. However, this has also uncovered some issues with duplicate author records.

Getting this browse list working actually took a huge amount of effort due to the complex way we store authors. In our system an author can be associated with any one of four levels of book record (work / edition / holding / item) and an author associated at a higher level needs to cascade down to lower level book records. Running queries directly on this structure proved to be too resource intensive and slow so instead I wrote a script to generate cached data about authors. This script goes through every author connection at all levels and picks out the unique authors that should be associated with each book holding record. It then stores a reference to the ID of the author, the holding record and the initial letter of the author's surname in a new table that is much more efficient to reference. This then gets used to generate the letter tabs with the number of book counts and to work out which books to return when an author surname beginning with a letter is selected.

However, one thing we need to consider about using cached tables is that the data only gets updated when I run the script to refresh the cache, so any changes / additions to authors made in the CMS will not be directly reflected in the library books tab. This is also true of the 'browse books by title' lists I previously created too. I noticed when looking at the books beginning with 'V' for a library (I can't remember which) that one of the titles clearly didn't begin with a 'V',

which confused me for a while before I realised it's because the title must have been changed in the CMS since I last generated the cached data.

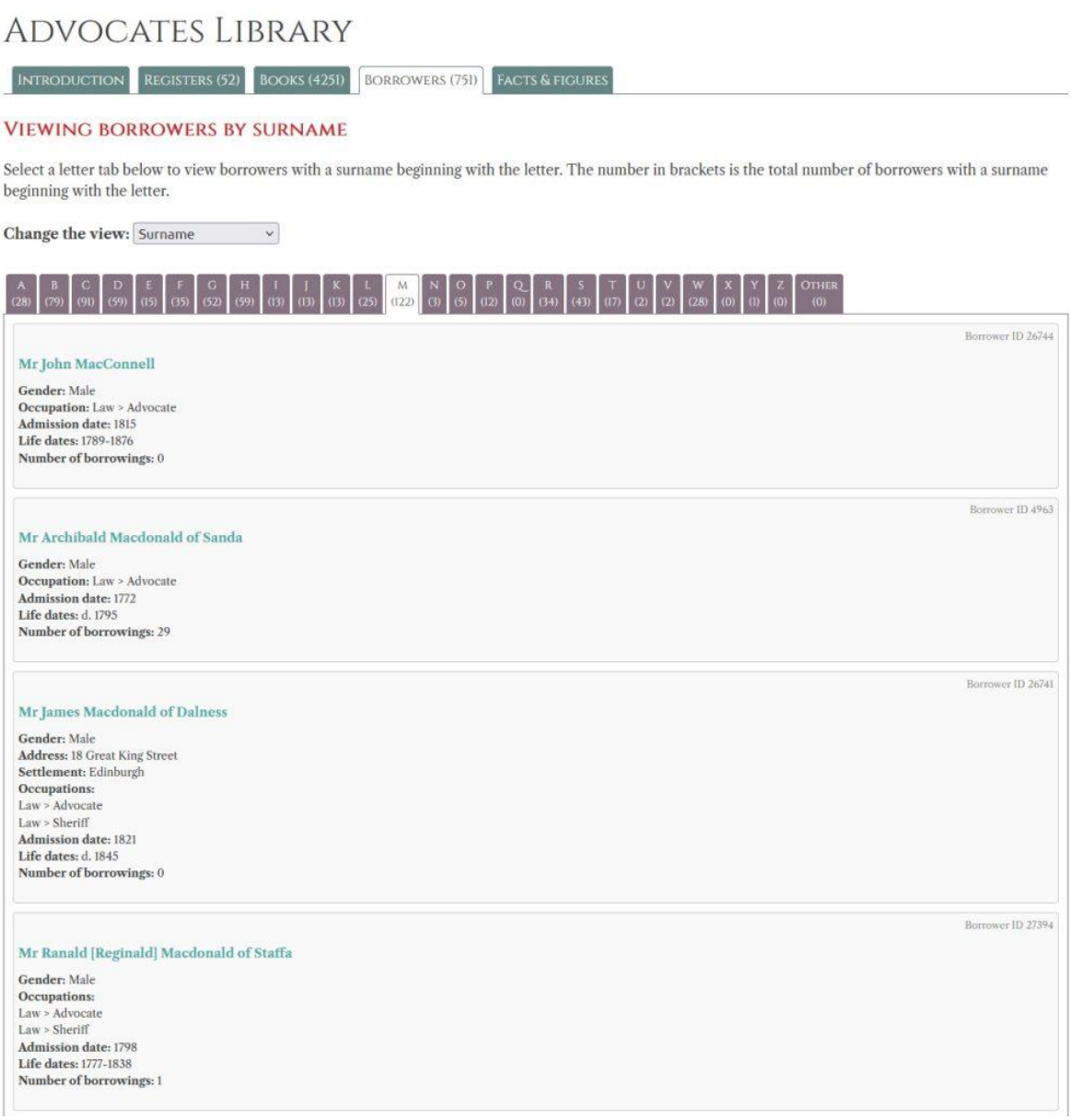
The 'most borrowed' page lists the top 100 most borrowed books for the library, from most to least borrowed. Thankfully this was rather more straightforward to implement as I had already created the cached fields for this view. I did consider whether to have tabs allowing you to view all of the books by number of borrowings, but I wasn't really sure how useful this would be. In terms of the display of the 'top 100' the books are listed in the same way as the other lists, but the number of borrowings is highlighted in red text to make it easier to see. I've also added in a number to the top-left of the book record so you can see which place a book has in the 'hitlist', as you can see in the following screenshot:



I also added in a 'to top' button that appears as you scroll down the page (it appears in the bottom right, as you can see in the above screenshot). Clicking on this scrolls to the page title, which should make the page easier to use – I've certainly been making good use of the button anyway.

31st October 2022

To begin with I worked on the 'borrowers' tab in the 'library' page and created an initial version of it. Here's an example of how it looks:



As with books, the page lists borrowers alphabetically, in this case by borrower surname. Letter tabs and counts of the number of borrowers with surnames beginning with the letter appear at the top and you can select a letter to view all borrowers with surnames beginning with the letter. I had to create a couple of new fields in the borrower table to speed the querying up, saving the initial letter of each borrower's surname and a count of their borrowings.

The display of borrowers is similar to the display of books, with each borrower given a box that you can press on to highlight. Borrower ID appears in the top right and each borrower's full name appears as a green title. The name is listed as it would be read, but this could be updated if required. I'm not sure where the 'other title' field would go if we did this, though – presumably something like 'Macdonald, Mr Archibald of Sanda'.

The full information about a borrower is listed in the box, including additional fields and normalised occupations. Cross references to other borrowers also appear. As with the 'Books' tab, much of this data will be linked to search results once I've created the search options (e.g. press on an occupation to view all borrowers with this occupation, press on the number of borrowings to view the borrowings) but this is not in place yet. You can also change the view from 'surname' to 'top 100 borrowers', which lists the top 100 most prolific borrowers (or less if there are less than 100 borrowers in the library). As with the book tab, a number appears at the top left of each record to show the borrower's place on the 'hitlist' and the number of borrowings is highlighted in red to make it easier to spot.

I also fixed some issues with the book and author caches that were being caused by spaces at the start of fields and author surnames beginning with a non-capitalised letter (e.g. 'von') which was messing things up as the cache generation script was previously only matching upper case, meaning 'v' wasn't getting added to 'V'. I've regenerated the cache to fix this.

I then decided to move onto the search rather than the 'Facts & figures' tab as I reckoned this should be prioritised. I began work on the quick search initially, and I'm still very much in the middle of this. The quick search has to search an awful lot of and to do this several different queries need to be run. I'll need to see how this works in terms of performance as I fear the 'quick' search risks being better named the 'slow' search.

We've stated that users will be able to search for dates in the quick search and these need to be handled differently. For now the API checks to see whether the passed search string is a date by running a pattern match on the string. This converts all numbers in the string into an 'X' character and then checks to see whether the resulting string matches a valid date form. For the API I'm using a bar character (|) to designate a ranged date and a dash to designate a division between day, month and year. I can't use a slash (/) as the search string is passed in the URL and slashes have meaning in URLs. For info, here are the valid date string patterns:

"XXXX","XXXX-XX","XXXX-XX-XX","XXXX|XXXX","XXXX|XXXX-XX","XXXX|XXXX-XX-XX","XXXX-XX|XXXX","XXXX-XX|XXXX-XX","XXXX-XX|XXXX-XX-XX","XXXX-XX-XX|XXXX","XXXX-XX-XX|XXXX-XX","XXXX-XX-XX|XXXX-XX-XX"

So for example, if someone searches for '1752' or '1752-03' or '1752-02|1755-07-22' the system will recognise these as a date search and process them accordingly. I should point out that I can and probably will get people to enter dates in a more typical way in the front-end, using slashes between day, month and year and a dash between ranged dates (e.g. '1752/02-1755/07/22') but I'll convert these before passing the string to the API in the URL.

I have the query running to search the dates, and this in itself was a bit complicated to generate as including a month or a month and a day in a ranged query changes the way the query needs

to work. E.g. if the user searches for '1752-1755' then we need to return all borrowing records with a borrowed year of '1752' or later and '1755' or earlier. However, if the query is '1752/06-1755-03' then the query can't just be 'all borrowed records with a borrowed year of '1752' or later and a borrowed month of '06' or later and a borrowed year of '1755' or earlier and a borrowed month of '03' or earlier as this would return no results. This is because the query is looking to return borrowings with a borrowed month of '06' or later and also '03' or earlier. Instead the query needs to find borrowing records that have a borrowed year of 1752 AND a borrowed month of '06' or later OR have a borrowed year later than 1752 AND have a borrowed year of 1755 AND a borrowed month of '03' or earlier OR have a borrowed year earlier than 1755.

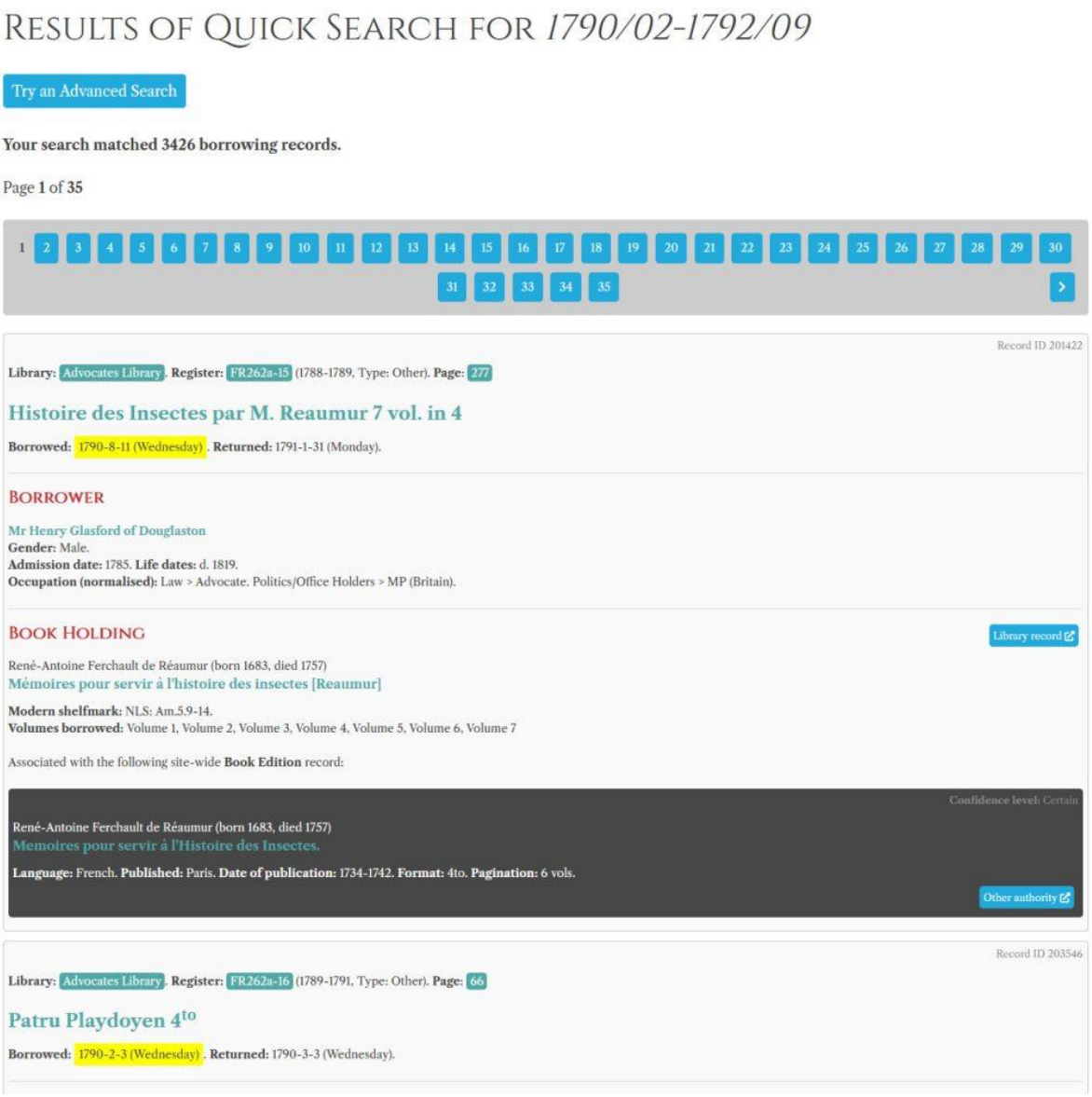
I also have the queries running that search for all necessary fields that aren't dates. This currently requires five separate queries to be run to check fields like author names, borrower occupations, book edition fields such as ESTC etc. The queries currently return a list of borrowing IDs, and this is as far as I've got. I'm wondering now whether I should create a cached table for the non-date data queried by the quick search, consisting of a field for the borrowing ID and a field for the term that needs to be searched, with each borrowing having many rows depending on the number of terms they have (e.g. a row for each occupation of every borrower associated with the borrowing, a row for each author surname, a row for each forename, a row for ESTC). This should make things much speedier to search, but will take some time to generate. I'll continue to investigate this next week.

7th November 2022

One of the project RAs had spotted an issue with a library register page appearing out of sequence so I spent a little time rectifying that. Other than that I continued to develop the front-end, working on the quick search that I had begun last week and by the end of the week I was still very much in the middle of working through the quick search and the presentation of the search results.

I have an initial version of the search working now and I created an index page on the test site I'm working on that features a quick search box. This is just a temporary page for test purposes – eventually the quick search box will appear in the header of every page. The quick search does now work for both dates using the pattern matching I discussed last week and for all other fields that the quick search needs to cover. For example, you can now view all of the borrowing records with a borrowed date between February 1790 and September 1792 (1790/02-1792/09) which returns 3426 borrowing records. Results are paginated with 100 records per page and options to navigate between pages appear at the top and bottom of each results page.

The search results currently display the complete borrowing record for each result, which is the same layout as you find for borrowing records on a page. The only difference is additional information about the library, register and page the borrowing record appears on can be found at the top of the record. These appear as links and if you press on the page link this will open the page centred on the selected borrowing record. For date searches the borrowing date for each record is highlighted in yellow, as you can see in the screenshot below:



The non-date search also works, but is currently a bit too slow. For example a search for all borrowing records that mention 'Xenophon' takes a few seconds to load, which is too long. Currently non-date quick searches do a very simple find and replace to highlight the matched text in all relevant fields. This currently makes the matched text upper case, but I don't intend to leave it like this. You can also search for things like the ESTC too.

However, there are several things I'm not especially happy about:

1. The speed issue: the current approach is just too slow
2. Ordering the results: currently there are no ordering options because the non-date quick search performs five different queries that return borrowing IDs and these are then just bundled together. To work out the ordering (such as by date borrowed, by borrower name) many more fields in addition to borrowing ID would need to be returned, potentially for thousands of records and this is going to be too slow with the current data structure

3. The search results themselves are a bit overwhelming for users, as you can see from the above screenshot. There is so much data it's a bit hard to figure out what you're interested in and I will need input from the project team as to what we should do about this. Should we have a more compact view of results? If so what data should be displayed? The difficulty is if we omit a field that is the only field that includes the user's search term it's potentially going to be very confusing

4. This wasn't mentioned in the requirements document I wrote for the front-end, but perhaps we should provide more options for filtering the search results. I'm thinking of facetted searching like you get in online stores: You see the search results and then there are checkboxes that allow you to narrow down the results. For example, we could have checkboxes containing all occupations in the results allowing the user to select one or more. Or we have checkboxes for 'place of publication' allowing the user to select 'London', or everywhere except 'London'.

5. Also not mentioned, but perhaps we should add some visualisations to the search results too. For example, a bar graph showing the distribution of all borrowing records in the search results over time, or another showing occupations or gender of the borrowings in the search results etc. I feel that we need some sort of summary information as the results themselves are just too detailed to easily get an overall picture of.

I came across the Universal Short Title Catalogue website this week (e.g. https://www.ustc.ac.uk/explore?q=xenophon) it does a lot of the things I'd like to implement (graphs, facetted search results) and it does it all very speedily with a pleasing interface and I think we could learn a lot from this.

Whilst thinking about the speed issues I began experimenting with Apache Solr (https://solr.apache.org/) which is a free search platform that is much faster than a traditional relational database and provides options for facetted searching. We use Solr for the advanced search on the DSL website so I've had a bit of experience with it. Next week I'm going to continue to investigate whether we might be better off using it, or whether creating cached tables in our database might be simpler and work just as well for our data. But if we are potentially going to use Solr then we would need to install it on a server at Stirling. Stirling's IT people might be ok with this (they did allow us to set up a IIIF server for our images, after all) but we'd need to check. I should have a better idea as to whether Solr is what we need by the end of next week, all being well.

14th November 2022

I spent almost all of this week working with a version of Apache Solr installed on my laptop, experimenting with data from the Books and Borrowing project and getting to grips with setting up a data core and customising a schema for the data, preparing data for ingest into Solr, importing the data and running queries on it, including facetted searching.

I started the week experimenting with our existing database, creating a cache table and writing a script to import a sample of 100 records. This cache table could hold all of the data that the quick search would need to query and would be very speedy to search, but I realised that other aspects related to the searching would still be slow. Facetted searching would still require several other database queries to be executed, as would extracting all of the fields that would

be necessary to display the search results and it seemed inadvisable to try and create all of this functionality myself when an existing package like Solr could already do it all.

Solr is considerably faster than using the database approach and its querying is much more flexible. It also offers facetted search options that are returned pretty much instantaneously which would be hopelessly slow if I attempted to create something comparable directly with the database. For example, I can query the Solr data to find all borrowing records that involve a book holding record with a standardised title that includes the word 'Roman', returning 3325 records, but Solr can then also return a breakdown of the number of records by other fields, for example publication place:

"London",2211,

"Edinburgh",119,

"Dublin",100,

"Paris",30,

"Edinburgh; London",16,

"Cambridge",4,

"Eton",3,

"Oxford",3,

"The Hague",3,

"Naples",2,

"Rome",2,

"Berlin",1,

"Glasgow",1,

"Lausanne",1,

"Venice",1,

"York",1

Format:

"8vo",842,

"octavo",577,

"4to",448,

"quarto",433,

"4to.",88,

"8vo., plates, port., maps.",88,

“folio”,76,

“duodecimo”,67,

“Folio”,33,

“12mo”,19,

“8vo.”,17,

“8vo., plates: maps.”,16

Borrower gender:

“Male”,3128,

“Unknown”,109,

“Female”,64,

“Unclear”,2

These would then allow me to build in the options to refine the search results further by one (or more) of the above criteria. Although it would be possible to build such a query mechanism myself using the database it is likely that such an approach would be much slower and would take me time to develop. It seems much more sensible to use an existing solution if this is going to be possible.

In my experiments with Solr on my laptop I Initially imported 100 borrowing records exported via the API call I created to generate the search results page. This gave me a good starting point to experiment with Solr’s search capabilities, but the structure of the JSON file returned from the API was rather more complicated than we’d need purely for search purposes and includes a lot of data that’s not really needed either, as the returned data contains everything that’s needed to display the full borrowing record. I therefore worked out a simpler JSON structure that would only contain the fields that we would either want to search or could be used in a simplified search results page. Here’s an example:

{

“bnid”: 1379,

“lid”: 6,

“slug”: “glasgow-university”,

“lname”: “Glasgow University Library”,

“rid”: 2,

“rname”: “3”,

“syear”: 1760,

“eyear”: 1765,

“rtype”: “Student”,

“pid”: 107,

“fnum”: “4r”,

“transcription”: “Euseb: Eclesiastical History”,

“bday”: 17,

“bmonth”: 9,

“byear”: 1760,

“rday”: 1,

“rmonth”: 10,

“ryear”: 1760,

“borrowed”: “1760-09-17”,

“returned”: “1760-10-01”,

“bdayofweek”: “Wednesday”,

“rdayofweek”: “Wednesday”,

“originaltitle”: “”,

“standardisedtitle”: “Ancient ecclesiasticall histories of the first six hundred years after Christ; written in the Greek tongue by three learned historiographers, Eusebius, Socrates, and Evagrius.”,

“brids”: [“1”],

“bfnames”: [“Charles”],

“bsnames”: [“Wilson”],

“bfullnames”: [“Charles Wilson”],

“boccs”: [“University Student”, “Education”],

“bgenders”: [“Male”],

“aids”: [“74”],

“asnames”: [“Eusebius of Caesarea”],

“afullnames”: [” Eusebius of Caesarea”],

“beids”: [“88”],

“edtitles”: [“Ancient ecclesiasticall histories of the first six hundred years after Christ; written in the Greek tongue by three learned historiographers, Eusebius, Socrates, and Evagrius.”],

“estcs”: [“R21513”],

“langs”: [“English”],

“pubplaces”: [“London”],

“formats”: [“folio”]

}

I wrote a script that would export individual JSON files like the above for each active borrowing record in our system (currently 141,335 records). I ran this on a version of the database stored on my laptop rather than running it on the server to avoid overloading the server. I then created a Solr Core for the data and specified an appropriate schema. This defines each of the above fields and the types of data the fields can hold (e.g. some fields can hold multiple values, such as borrower occupations, some fields are text strings, some are integers, some are dates). I then ran the Solr script that ingests the data.

It took a lot of time to get things working as I needed to experiment with the structure of the JSON files that my script generated in order to account for various complexities in the data. I also encountered some issues with the data that only became apparent at the point of ingest when records were rejected. These issues only affected a few records out of nearly 150,000 so I needed to tweak and re-run the data export many times until all issues were ironed out. As both the data export and the ingest scripts took quite a while to run the whole process took several days to get right.

Some issues encountered include:

1. Empty fields in the data resulting in no data for the corresponding JSON field (e.g. “bday”: <nothing here>) which invalidated the JSON file structure. I needed to update the data export script to ensure such empty fields were not included.

2. Solr’s date structure requiring a full date (e.g. 1792-02-16) and partial dates (e.g. 1792) therefore failing. I ended up reverting to an integer field for returned dates as these are generally much more vague and having to generate placeholder days and months where required for the borrowed date.

3. Solr’s default (and required) ID field having to be a string rather than an integer, which is what I’d set it to in order to match our BNID field. This was a bit of a strange one as I would have expected an integer ID to be allowed and it took some time to investigate why my nice integer ID was failing.

4. Realising more fields should be added to the JSON output as I went on and therefore having to regenerate the data each time (e.g. I added in borrower gender and IDs for borrowers, editions, works and authors)

5. Issues with certain characters appearing in the text fields causing the import to break. For example, double quotes needed to be converted to the entity ‘&quote;’ as their appearance in the JSON caused the structure to be invalid. I therefore updated the translation, original title and standardised title fields, but then the import still failed as a few borrowers also have double quotes in their names.

However, once all of these issues were addressed I managed to successfully import all 141,355 borrowing records into the Solr instance running on my laptop and was able to experiment with queries, all of which are running very quickly and will serve our needs very well. And now that the data export script is properly working I’ll be able to re-run this and ingest new data very easily in future.

The big issue now is whether we will be allowed to install an Apache Solr instance on a server at Stirling. We would need the latest release of Solr (v9 https://solr.apache.org/downloads.html) to be installed on a server. This requires Java JRE version 11 or higher (https://solr.apache.org/guide/solr/latest/deployment-guide/system-requirements.html). Solr uses the Apache Lucene search library and as far as I know it fires up a Java based server called Jetty when it runs. The deployment guide can be found here: https://solr.apache.org/guide/solr/latest/deployment-guide/solr-control-script-reference.html

When Solr runs a web-based admin interface is available through which the system can be managed and the data can be queried. This would need securing, and instructions about doing so can be found here: https://solr.apache.org/guide/solr/latest/deployment-guide/securing-solr.html

I think basic authentication would be sufficient, ideally with access limited to on-campus / VPN users. Other than for testing purposes there should only be one script that connects to the Solr URL (our API) so we could limit access to the IP address of this server, or if Solr is going to be installed on the same server then limiting access to localhost could work.

In terms of setting up the Solr instance, we would only need a single node installation (not SolrCloud). Once Solr is running we'd need a Core to be created. I have the schema file the core would require and can give instructions about setting this up. I'm assuming that I would not be given command-line access to the server, which would unfortunately mean that someone in Stirling's IT department would need to execute a few commands for me, including setting up the Core and ingesting the data each time we have a new update.

One downside to using Solr is it is a separate system to the B&B database and will not reflect changes made to the project's data until we run a new data export / ingest process. We won't want to do this too frequently as exporting the data takes at least an hour, then transferring the files to the server for ingest will take a long time (uploading hundreds of thousands of small files to a server can take hours. Zipping them up then uploading the zip file and extracting the file also takes a long time). Then someone with command-line access to the server will need to run the command to ingest the data. We'll need to see if Stirling are prepared to do this for us.

Until we hear more about the chances of using Solr I'll hold off doing any further work on B&B. I've got quite a lot to do for other projects that I've been putting off whilst I focus on this issue so I need to get back into that.

21st November 2022

I had a chat with IT Services at Stirling about setting up an Apache Solr system for the project. It's looking like we will be able to proceed with this option, which will be great.

9th January 2023

I spoke to the IT people at Stirling about their current progress with setting up a Solr instance for the Books and Borrowing project. I also replaced a selection of library register images with better versions for that project and arranged a meeting for next Monday with the project's PI and Co-I to discuss progress with the front-end.

16th January 2023

I had a Zoom call with project PI Katie and Co-I Matt on Monday to discuss the front-end developments and some of the outstanding tasks left to do. The main one is to implement a genre classification system for books, and we now have a plan for how to deal with these. Genres will be applied at work level and will then filter down to lower levels. I also spent some time speaking to Stirling's IT people about setting up a Solr instance for the project, as discussed in posts before Christmas. Thankfully it was possible to get this set up and by the end of the week we had a Solr instance set up that I was able to query from a script on our server. Next week I will begin to integrate Solr queries with the front-end that I'm working on. I also generated spreadsheets containing all of the book edition and book work data that Matt had requested and engaged in email discussions with Matt and Katie about how we might automatically generate Book Work records from editions and amalgamate some of the many duplicate book edition records that Matt had discovered whilst looking through the data.

23rd January 2023

I spent much of the week working on the Books and Borrowing project, working with the new Solr instance that the Stirling IT people set up for me last week. I spent some time creating a new version of the API that connects to Solr and then setting up the Solr queries necessary to search all fields of the Solr index for a regular quick search and the 'borrowed' fields for date searches. This included returning the data necessary to provide the facetted search options on the search results page (i.e. filters for the search results). I also set up a new development version of the front-end, leaving my existing pre-Solr version in place in case things go wrong and I need to revert to it.

As with the previous version of the site, you can perform a quick search, which searches the numerous fields that are specified in the requirements document. Dates can either be a single date or a range. Text searches can use wildcards * to match any characters (e.g. tes* will match all words beginning 'tes') and ? to match a single character (e.g. h?ll matching 'hill' and 'hell').

Currently the results still display the full records with 100 records per page. I did consider changing this to a more compact view with a link to open the full record, but I haven't implemented this as of yet. I might add an option to switch between a 'compact' and 'full' view of the records instead, as I think having to click through to the full record each time you're interested would get a bit annoying.

There have been a lot of changes to the back end, even if the front-end doesn't look that different. Behind the scenes the API now connects to the Solr instance and queries are formatted for and passed to Solr, which then returns the data. Solr is very fast, but the loading of 100 results which are full records does still take some time. Queries to Solr currently return the 100 relevant borrowing IDs and then another API call retrieves all of the data for these 100 IDs from the regular database. A compact view could potentially rely solely on data stored in Solr, which would be a lot quicker to load, if we want to pursue that option.

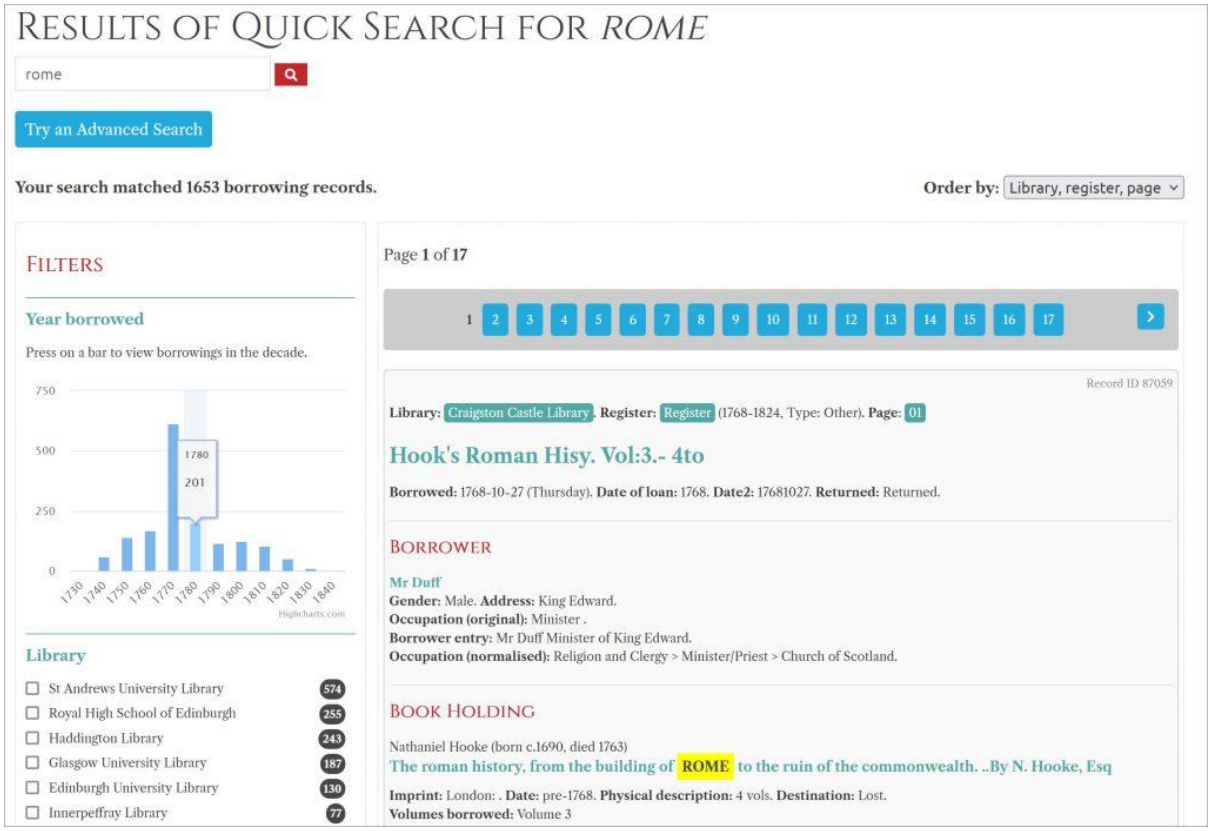
In addition to returning the IDs for the 100 borrowing records that are to be displayed on any one results page, Solr also returns the total number of matching borrowing records plus the facetted search information. For the moment the following information is included in the facetted data: Borrowing year, library name, borrower gender and occupation, author name and book language, place of publication and format. These appear as 'Filter' options down the left-hand side of the results page, currently as a series of checkboxes, the name of the item in question and the number of results in the overall results that the item is found in. Pressing on a checkbox filters the results in question and the results page reloads as soon as you press a checkbox. This causes both the results and the filters to narrow, displaying only those that continue to match. You can then click on other checkboxes to narrow things further, or deselect a checkbox to return to the non-filtered view.

I think the filters are going to be hugely useful, but they're not perfect yet. There are issues with the data for occupations and authors. This is because the data has been stemmed by Solr for search purposes, meaning the field is broken down into individual word stems (e.g. 'educ' for 'education'). I will fix this but it will require me to regenerate the data and get the Stirling IT people to replace the existing data. I've also noticed that data from test libraries is in Solr too and I'll need to ensure this gets removed.

With all of this in place I then moved on to providing different sorting options for the search results, for example ordering the results by borrowed date, library or author name. This required some tweaking of the Solr queries and the API and then some updates to the front-end to ensure the selected sorting option is dealt with and remembered. However, I did come across a limitation in Solr, in that it is not possible for Solr to order data by fields that contain multiple values. This means that for now sorting by things like author name and borrower occupation won't work as each of these can contain multiple values per record. I'll therefore have to make concatenated versions of these fields for sort purposes and will do this when I regenerate the data.

This initial version of the facetted search results page displayed years in the same way as other search filters: as a series of checkboxes, year labels and counts of results in each year. What I really wanted to do was to display this as a bar chart instead, using the HighCharts library that I use for other visualisations in the front-end. I wanted to group years into decades where the range of years is greater than a decade and enable the user to then press on a decade bar to view the results for individual years within the decade, with the bar chart then displaying the individual years. I managed to get the 'by decade' bar chart working this week. You can hover over a bar to view the exact total for the decade. You can also click on a decade to filter the search to that decade. This is the bit I'm still working on. Currently no bar chart is displayed and you need to use your browser's 'back' button to return, but the filter does actually work. Eventually a bar chart with borrowings for each year in the decade will be displayed, together with a button for returning back. In this view you will be able to further click on a year bar to filter the results to the selected year. I'll continue with this next week and the 'Year borrowed' checkboxes will be removed once the bar chart is fully working. It's took quite a while to get the

bar chart working as there was a lot of logic that needed to be worked out in order for borrowings to be grouped into decades and to accommodate gaps in the data (e.g. if there is no data for a decade we still need this decade to get displayed otherwise the graph looks odd). Below is a screenshot of the new front-end with facetted searching and 'year borrowed' bar chart:



Also for the Books and Borrowing project this week I had a Zoom call with Katie and Matt to discuss genre classification (which has now been decided upon) and batch editing the book edition records to fix duplicates and to auto-generate book work records for any editions that need them. I also sent on some data exports containing the distinct book formats and places of publication that are in the system as they will need some editorial work as well. I also responded to a few queries from one of the project RAs who wanted some queries run on the data for a library he has worked on.

30th January 2023

I spent almost all of this week working on the Books and Borrowing project. This first two days were mainly spent dealing with data related issues. This included writing a script to merge duplicate editions based on a spreadsheet of editions that I'd previously sent Matt to which he had added a column to denote which duplicate should be merged with which. It took quite some time to write the script due to having to deal with associated book works and authors. Some of the duplicates that were to be deleted had book work associations whilst the edition to keep didn't. These cases had to be checked for and the book work association had to be transferred over.

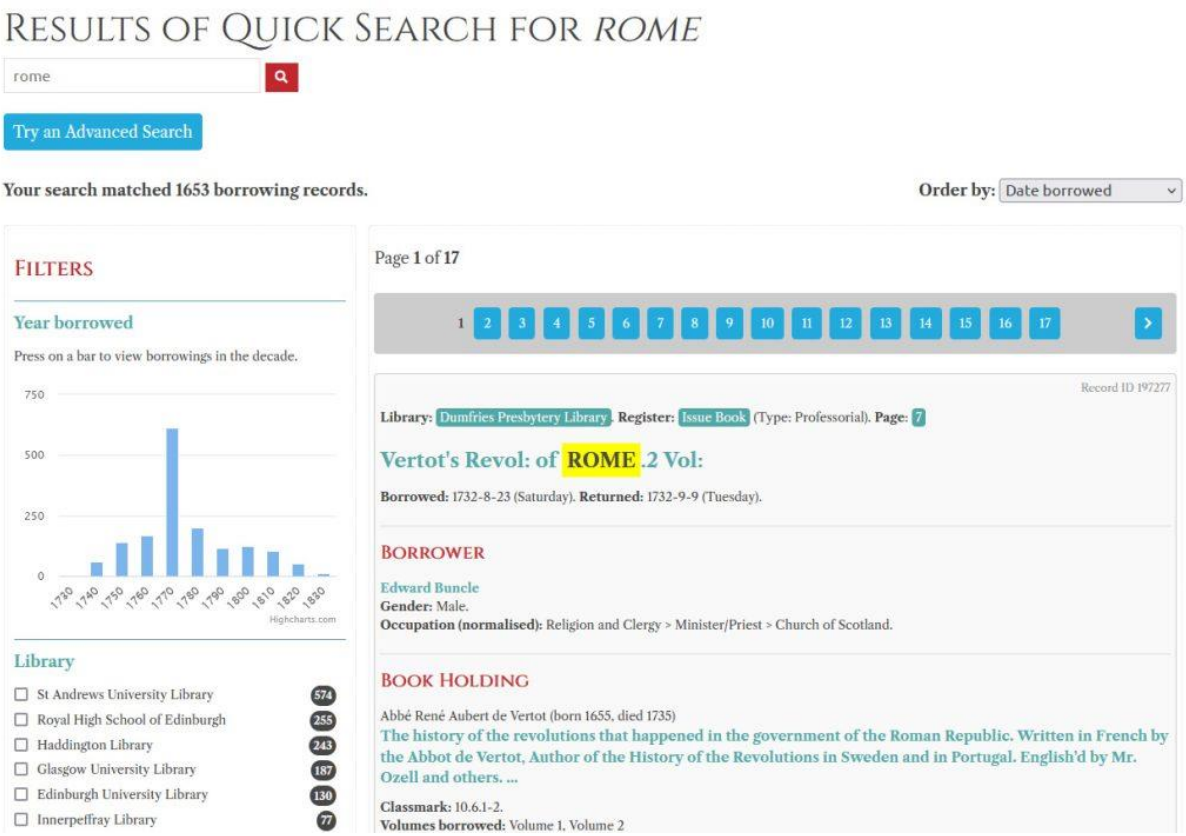
Authors were a little more complicated as both the duplicate to be deleted and the one to keep may have multiple associated authors. If the duplicate edition to keep had no authors but the one to be deleted did then each of these had to be associated with the edition to keep. But if both the edition to delete and the one to keep had authors only those authors from the 'to delete' edition that were not already represented in the 'to keep' edition's author list had to be associated. In such cases where an author did need to be associated with the 'to keep' edition I also added in a further check to ensure the author being associated didn't have the same name (but different ID) as one already associated, as there are duplicate authors in the system.

With all of this done the script then had to reassign the holding records from the 'to delete' edition to the 'to keep' one and then finally delete the relevant edition. As the script makes significant changes to the data I first ran it on a version of the data I had running on my laptop to check that the script worked as intended, which thankfully it did. After completing the test I then (after taking another backup of the database in case of problems) ran the script on the live data. The process resulted in 541 duplicate editions being deleted from the system and as far as I can tell all is well. We now have 13,086 editions in the system and 13,014 of these do not have an associated book work. We only have 75 book works in the system.

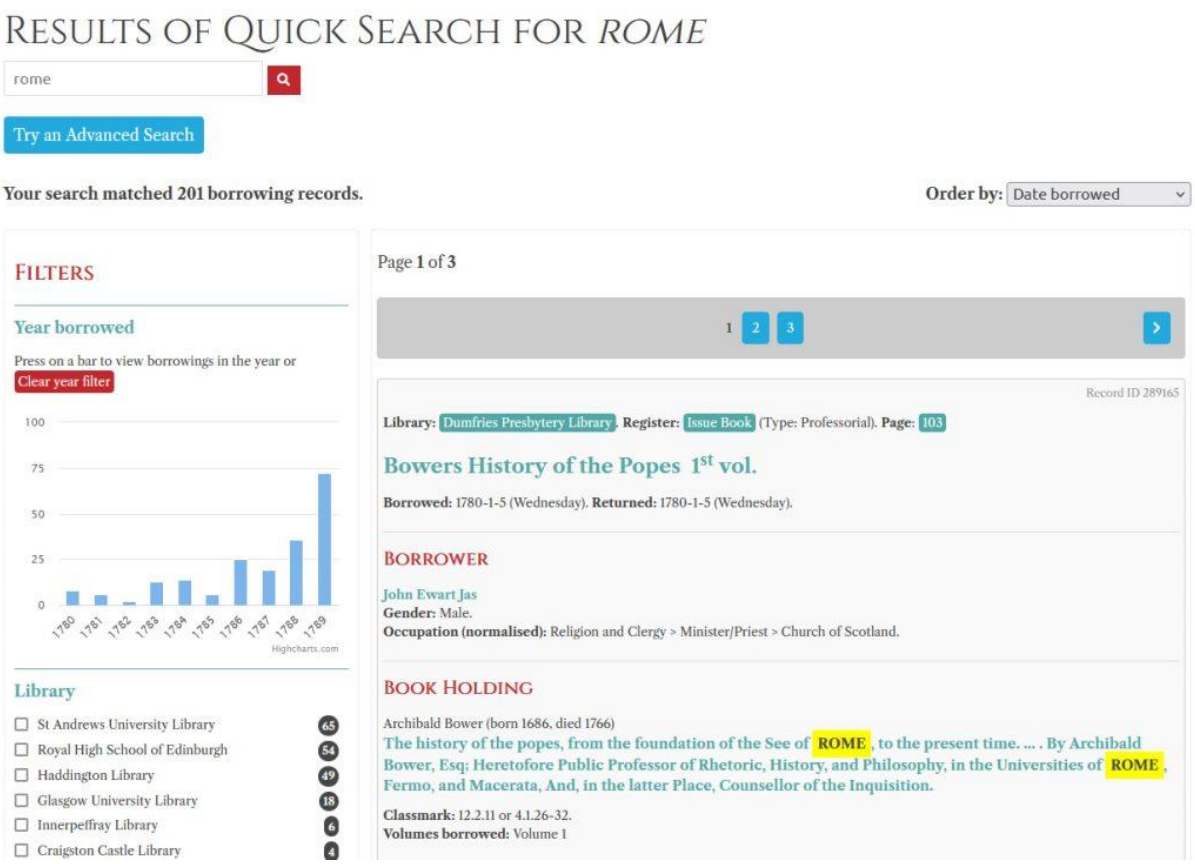
The next step is to assign book works to editions and add in book genres. In order to do this I created a further spreadsheet containing the editions with columns for book work, authors and three columns which can be used to record up to three genres. I also sent Matt and Katie a further spreadsheet containing the details of the 75 existing book works in our system. It's going to be rather complicated to fill in the spreadsheet as there's a lot going on and it took me quite a while to figure out a workflow for filling it in. Hopefully with that in place filling it in should be straightforward, if time-consuming.

I also ran some queries, did some checks and generated some spreadsheets for the Wigtown data for Gerry McKeever. With these data related issues out of the way I then returned to developing the front-end. Whilst working on an issue relating to ordering the results by date I noticed that we have quite a lot of borrowing records in the system that have no dates. There are almost 12,000 that don't have a 'borrowed year'. There's possibly a good reason for it, but of these 2,376 have a borrowed day and a borrowed month but no year, which seems more strange. I emailed Katie and Matt about this and they're going to investigate.

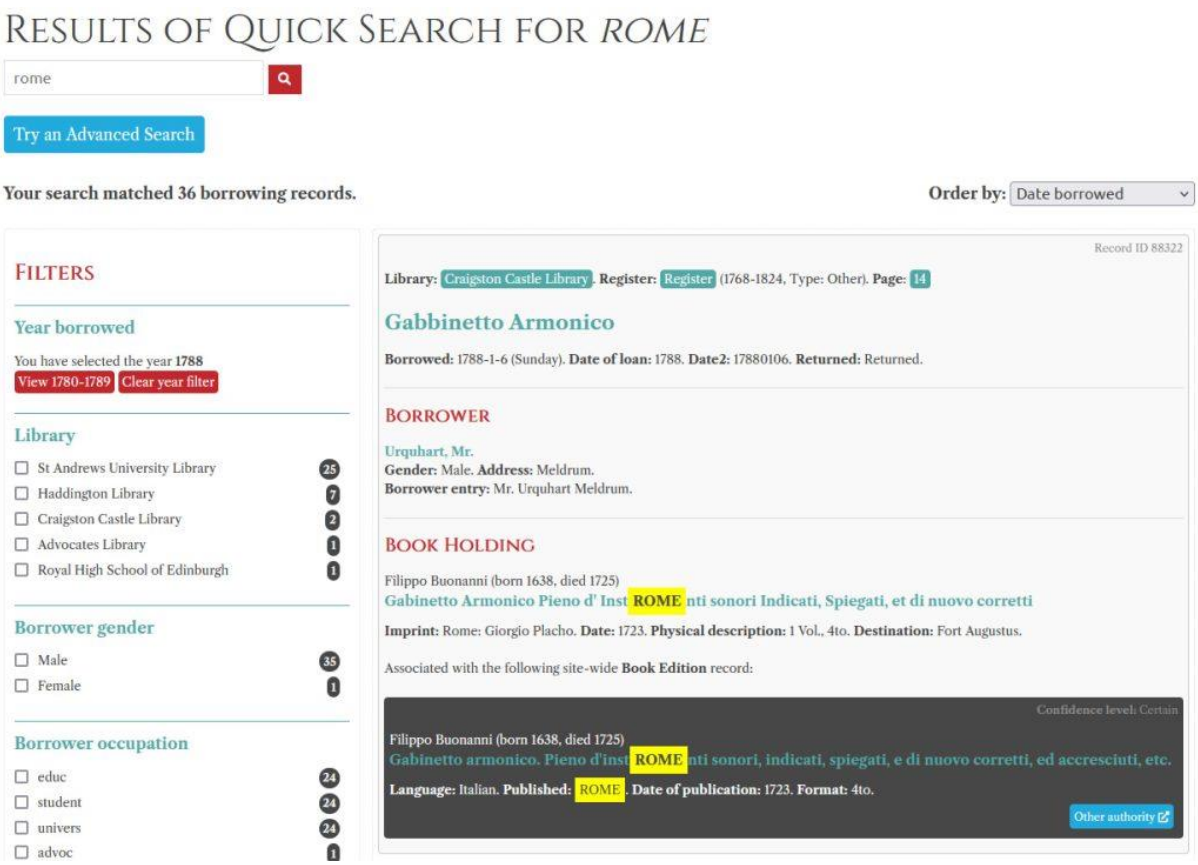
I managed to finish work on the 'Year borrowed' bar chart this week. Without providing a year filter the bar chart shows the distribution of borrowing records divided into decades, for example this search for 'rome', ordered by date borrowed:



You can then click on one of the decade bars to limit the results to just those in the chosen decade, for example clicking on the '1780' bar:



This then displays a bar chart showing a breakdown of borrowing records per year within the selected decade. You are given the option of clearing the year filter to return to the full view and you can also click on an individual year bar to limit the results to just that year, for example limiting to the year 1788:



When you reach this level no bar chart is displayed as year is the unit that's filtered and there is only one year selected. But options are given to return to the decade view or clear the year filter. You can of course combine the year filter with any of the other filter options. I guess at year level we could display a similar bar chart for borrowings per month, but this might be too fine-grained and confusing (plus would be a lot more work as everything is currently set up to work with year only). It's something to consider, though.

I did spot a problem with the bar chart: I realised that when you searched for an individual year or a range within an individual year the results were still showing the options to view the decade and clear the year filter, both of which then gave errors. This has now been sorted – no year filter options should be shown when the main search is only for a year.

For the remainder of the week I began working on the advanced search. As specified in the requirements document, currently the advanced search page features two tabs, one for a 'simple' advanced search and one for an 'advanced' advanced search. So far I've just been working on the forms, which in turn has necessitated making some changes to the API (to bring back a simple list of all libraries and to enable an entire list of registers to be returned). The forms allow you to select / deselect libraries and select / deselect all. In the 'Simple' tab there

are also textboxes for entering date of borrowing, author forename and surname, year of birth / death and book title, plus a placeholder for genre. The requirements document stated that date of borrowing would have boxes for entering years and days and a drop-down list for selecting month, with two sets to be used for range dates. I've decided that since the quick search already allows dates to be entered directly as text that it would make sense to just follow the same method for the advanced search.

Author dates as currently specified are going to be a bit messy for BC dates, where people need to enter a negative value. This is messy because a dash is used for date ranges so we may end up with something like '-1000–200' (that's two dashes in the middle). I'm not sure what we can do about this, though. I guess having different boxes for 'from' and 'to' for ranged dates would avoid the issue. For the 'advanced' advanced search lists of selectable registers will appear depending on the libraries that are selected. This is what I'm still in the middle of working on.

If I have the time I would like to create a new theme for the website that will look pretty similar but will use the Bootstrap front-end toolkit (https://getbootstrap.com/). The current WordPress theme doesn't use this which means creating complex layouts is more difficult and messy. I created a Bootstrap based WordPress theme for the Anglo-Norman Dictionary (e.g. this search form: https://anglo-norman.net/textbase-search/) but I'll just have to see how much time I have as I think it's better to get the essentials in place first. But what it means is in the meantime things like the search form layout will possibly not be finalised (but will be functional).

6th Febraury 2023

I spent the day continuing to work on the advanced search interface for the Books and Borrowing project. Whilst doing so I noticed that the borrower title data contained lots of variants that will probably need to be standardised (e.g. 'capt', 'capt.' and 'captain') so I emailed Katie and Matt a list of these. I also spotted a problem with the occupations that had occurred during batch import of data for Innerpeffray. There are 62 borrowers that have been assigned to the occupation category 'Minister/Priest' but this occupation is the only one where there are three hierarchical levels and 'Minister/Priest' is the second level and should therefore not be assignable. Only endpoints of the hierarchy, such as 'Catholic' and 'Church of Scotland' should be assignable. Hopefully this will be a fairly simple thing to fix, though.

For the Advanced Search form the requirements document stated that there will be an option for selecting libraries and a further one for selecting registers that will dynamically update depending on the libraries that are selected. As I worked on this I realised it would be simpler to use if I just amalgamated the two choices, so instead I created one area that lists libraries and the registers contained in each. From this you can select / deselect entire libraries and/or registers within libraries. It does mean the area is rather large and I may update the interface to hide the registers unless you manually choose to view them. But for now the listed registers are all displayed and include the number of borrowings in each. There are several that have no borrowings and if these continue to have no borrowings I should probably remove them from the list as they would never feature in the search results anyway.

There is also a section for fields relating to the borrowing and a further one for fields relating to borrower. This includes a list of borrower titles, with the option of selecting / deselecting any of these. Beside each one is a count of the number of borrowers that have each title. I'm currently still working on borrower occupation. This currently features another area with checkboxes for each level of occupation, with counts of the number of borrowers in each occupation. I'm still working on the select / deselect options so these are not all working at all levels yet.

13th February 2023

I managed to complete an initial version of the advanced search form. In my previous update I said I hadn't quite managed to complete the selection options for the borrower occupation section. This is complete now – you can select and deselect occupations at any level and corresponding occupations at higher or lower levels of the hierarchy will also select / deselect as required. I have also added in autocompletes for borrower settlement and street. If you start typing into one of these boxes (e.g. 'black' in settlement) a list of matching options will appear from which you can select one. Note that the entered text can appear anywhere in the settlement name (e.g. 'parish of Blackford' is brought back) and we might want to change this to just match the beginning of settlements.

Street works in the same way (e.g. type in 'king'). A couple of things to point out, though. Firstly: the selection of settlement and of street are currently in no way connected. E.g. if you select 'Blackford' as a settlement and then attempt to type in a street the system doesn't limit this to just streets within Blackford. I could update things to connect the two search boxes in such a way, though. Secondly: I think we'll have to give people freedom to ignore the autocomplete if they want. For example, if you enter 'king' in 'street' you'll see lots of very specific addresses (e.g. '15 great king street'). If you select one of those you're obviously limiting your search quite considerably. Whereas if we allow people to enter 'great king street' to bring back all borrowings at all addresses on this street the search might be more useful. I've also added in borrower gender which (as specified in the requirements document) allows one single gender to be selected. Thinking about it, we might want to make this a multi-select like other things instead.

The book author section is exactly the same as the 'simple' search and the book work section is pretty straightforward (and still awaits the addition of genre). In the book edition section the ESTC field is an autocomplete. This works slightly differently in that it matches the beginning of the ESTC only (e.g. 'T1001' matches IDs beginning with that text) and three characters rather than two need to be entered. Even three gives a very long list and I may make it four characters before the list appears. The last autocomplete field is place of publication. This matches text anywhere in the place. For example, type in 'lon' and you'll see all of the places involving London, but also places like Bouillon. I did wonder about making this a multi-select instead, but there are possibly too many to list all at once.

There are also two further multi-select areas for language and format. Format wasn't listed in the requirements document as being a multi-select but I think it makes sense for it to be one.

Each of these areas lists the number of book editions that have the language / format and (as previously discussed) the data needs tidying up as it's a bit messy. So, that's the 'advanced' form complete, although the layout is not finalised so it will eventually look a lot nicer (I hope). But there's no getting around the fact that there are an intimidatingly large number of search options listed and we might need to think some more about this.

Also this week I inserted a missing page into the records for a register for Leighton library and sent the data for the 62 borrowers that have been erroneously assigned the mid-tier 'Minister/Priest' occupation to Katie to be assigned a final tier occupation instead.

20th February 2023

I dealt with some data correction issues for Haddington library that one of the researchers had spotted, including swapping page images around and moving borrowing records to different pages. I also corrected the occupation errors that we'd spotted with some borrowers from Innerpeffray library using a spreadsheet that Katie sent me. She had noticed that there were also several duplicate borrowers in the data and had noted which records needed to be amalgamated so I dealt with these as well.

My main task for the week was to update the Solr data we use for the quick search to incorporate all of the data that we will also need to use for the advanced search. On Monday and Tuesday I spent some time reworking the Solr instance running on my laptop so as to get it ready to handle the advanced search. This involved adding new fields for all of the types of data the advanced search needs to query.

I also figured out how to get around the stemming issue for fields like occupation. For text fields Solr creates stemmed versions of all recognisable words in the fields, so for example 'searching', 'searched', 'searches' etc all have the stemmed form 'search'. This then allows free-text searches to find data that's of relevance, which can be really useful. Unfortunately when displaying search filters it's the stemmed forms that get returned and displayed beside the checkboxes and these can be a bit confusing. I figured out that you can create copy fields for these text fields in Solr where the text is stored as strings rather than text, and strings do not get stemmed. The search can then use the text field and the search filters can then use the string field. Pressing on a search filter then searches the string field, which is case sensitive, but this isn't an issue as what's being searched is the full text of the checkbox label (e.g. 'Religion and Clergy') which will always match the string form Solr stores. This means that the search filters now say something like 'Education' rather than 'educat' and full author names now get displayed, which is great.

I also added in borrower title and ESTC as search filters as I thought these might be useful. Plus I've fixed the issue of fields that hold multiple values not being sortable. For example, a borrowing record may have multiple occupations associated with it as there may be multiple borrowers and each borrower may have several occupations. Because of this it was not possible to sort the search results by borrower occupation. The fix for this was to generate a

further field for each that stores all of the multiple values as a single string. For borrower occupation for sorting purposes the occupation at the bottom of the hierarchy appears first, so if a borrowing record features a borrower with occupation 'Law -> Advocate' the record will be sorted under 'Advocate' then 'Law'. For borrower names and author names the ordering is surname then forename.

With all of these changes in place I took a copy of the live database (also taking the opportunity to deactivate all of the test libraries in the system), regenerated the JSON files that Solr indexes and then ingested them into my updated Solr instance on my laptop. After that I ran some tests to check all was working fine. After that I sent the data to the IT people at Stirling (I need to get them to import the data into the Solr instance on the server) and on Wednesday morning they imported it all and thankfully everything went smoothly.

With the new data in place I updated the API and the search results page to add in the new filters (Borrower title and ESTC) and to switch the filters over to using the string versions for display so we now have full occupation and author names displayed. I also updated the 'Order by' facility to allow all sorting options to work. Unfortunately whilst doing so I spotted that I'd forgotten to add in the code to populate the book edition title single field so I'm afraid sorting by this field doesn't work yet, but other options such as borrower occupation and author and borrower name are now working. I updated my Solr data generation script to add in the book edition title now so next time I regenerate the data this will work.

I then started to work on implementing the advanced search. I decided to change the way the API is referenced for the search. Previously there was going to be one endpoint for the quick search, which would accept one search parameter, and another for the advanced search, which would accept multiple parameters. I decided instead to amalgamate the two into one single search endpoint as in reality both search facilities will need to do the same things: format the search options for Solr, work out the pagination, deal with ordering options and work out which filters need to be applied.

In order to amalgamate the endpoints I needed to rework the quick search facility that I had already created, and this meant breaking the quick search for a while. Thankfully I managed to put it all back together again with the quick search working once more, but with slightly different URLs and a differently structured API call. With this in place I began to add the advanced search data types to the API so as to construct the query that will be passed to Solr to return the advanced search results. This basically allows specific fields in the Solr data (e.g. author names, library names, dates) to be queried rather than querying all fields, which the quick search does. As I left things off on Friday I was in the middle of adding in the option of searching author birth and death years, but I'd run into a little difficulty when processing negative years (i.e. BC years) that I'm going to have to investigate further next week.

27th February 2023

I continued to develop the front-end for the Books and Borrowing project for most of this week, completing work on an initial version of the advanced search facility. Last week I decided to change the way the API is referenced for the search. Previously there was going to be one endpoint for the quick search, which would accept one search parameter, and another for the advanced search, which would accept multiple parameters. I decided instead to amalgamate the two into one single search endpoint as in reality both search facilities will need to do the same things: format the search options for Solr, work out the pagination, deal with ordering options and work out which filters need to be applied.

In order to amalgamate the endpoints I needed to rework the quick search facility that I had already created, and this meant breaking the quick search for a while. Thankfully I managed to put it all back together again with the quick search working once more, but with slightly different URLs and a differently structured API call. With this in place I began to add the advanced search data types to the API so as to construct the query that will be passed to Solr to return the advanced search results. This basically allows specific fields in the Solr data (e.g. author names, library names, dates) to be queried rather than querying all fields, which the quick search does.

As I worked on this I ran into a spot of bother with author years of birth and death that were negative (i.e. BC). They just weren't working as they should have done and a bit of investigation revealed that this was because I was storing the years as strings rather than integers. I regenerated the data on my laptop, saving the years as integers, and after that negative dates worked. However, I soon realised why I hadn't been saving the years as integers: some author dates are not integers but are things like '1650?' or '16__'. When I tried ingesting these into Solr the records gave errors and failed to get added. I therefore had to add a further check to avoid any non-integer dates getting added to Solr. This means the associated records now get added but don't have the offending dates. This isn't a huge issue as the dates would never have been searchable anyway. For now this update is not present on the live site as I will wait until the next data export to add this, so in the meantime negative author dates will not work but the issue has been sorted.

I also ran into another issue with how I was structuring the URLs for the advanced search. Short URLs as I've previously used work fine, but the advanced search is going to potentially result in some very long URLs, with advanced search fields and values stored in a specific section of the URL between slashes, for example:

/search/advanced||register|270||register|287||register|273||transcription|transcription||bdate|1789_1791||btitles|Miss||bfnames|forename||bsnames|surname||bothernames|othername||boccs|Church%20of%20Scotland||boccs|Presbyterian%20Seceder||bettlements|Selkirk/

However, such URLs were resulting in a 403 forbidden error on the server. I contacted Stirling IT Services to enquire about this and discovered that the issue wasn't the length of the URL but the length of the text between slashes in the URL. The file system only allows filenames to be 255 characters in length and even though the above URL isn't actually referencing filenames but is split up into variables by my script, the server first has to treat the URL as if it contained filenames (well, folder names) and it's the server at a very fundamental level that is preventing things from working.

Unfortunately this meant I had to go back to the drawing board regarding how the search URLs would work. Previously (as shown above) the search variables appeared first, with variable names and values separated by a bar and each pairing separated by a double bar. After that things like filter queries, pagination and ordering options are included in the URL. I needed to split the variable pairings up with a slash instead to avoid the lengthy text between slashes, but this would mean I could no longer be certain where in the URL things like pagination would appear. Instead I needed to switch around the order of things in the URL, ensuring pagination, filter queries and sorting options appear first and then all of the search criteria follow, as many as are required. This took quite some time to implement and does unfortunately mean that none of the existing links I've sent the team will work any more, but we are now in a better place and the search's lengthy URLs will now work.

This week I fully updated the API to enable any and all combinations of the advanced search fields to be queried. I have updated the JavaScript that picks out the search options that have been selected in the search forms and processes the search URLs and I have connected the front-end to the API to enable the advanced search to function and return data. For example a 'simple' search can be limited to Edinburgh and Glasgow University libraries for borrowing records containing authors with forename 'William' and surname 'Shakespeare'. Above the search results there is a grey box that lists what search options have been selected and gives users the option of refining their search or starting a new one. I also worked on the formatting of the search options that get displayed (e.g. libraries show their full name rather than the 'slug' used in the search and registers show their name rather than their ID). The 'refine' option works, and displays either the 'simple' or 'advanced' search form as applicable with all search terms 'remembered' in the form. Users also have the option to clear the search forms and start again if required too.

So for example an advanced search is for borrowing records in two Edinburgh registers (Da.2.10 and Da.2.11) where the borrower surname is 'smith', the book edition language is 'english' and the format is '8v0' gives results and if you press 'refine search' the 'advanced' tab is displayed with these options already filled in, allowing you to update them as required.

I also investigated and fixed an issue with selecting / deselecting the third level religion occupations and I began to make some of the information in the search results searchable. I've currently added in 'click to search' options for the borrowed date and the borrower name. These now appear with a dotted line under them and if you press on one you will immediately

see all of the associated borrowings The click through for borrower name still needs a bit of work as it is actually a new search option not present in the search form – a search for borrower ID. At the moment it's only the ID that appears in the 'you searched for' section but I will fix this.

There are still some things that are not yet working properly. Search filters that feature slashes or bars or ampersands currently cause things to break. Also Solr is sometimes being too clever for its own good in bringing back records that are of relevance but don't actually match the search criteria. For example a search for the transcription field containing 'betsy thoughtless' currently brings back records that don't include this text in the transcription field but in other fields, meaning Solr returns the records because it thinks they might be of interest. This can be avoided by using quotes but I need to investigate whether there is a better way to deal with this.

On Friday I dealt with the normalisation of data that Matt and the team had been working on. This rationalises the data in fields such as borrower title and book edition language so that the same form is always used for the same thing. For example there is just one form for 'captain' rather than there being 'capt', 'capt.', 'captain' etc. I wrote scripts to process all of these updates and further scripts to pick out forms that need additional checking. I also regenerated the distinct forms for things like borrower title so Matt could check that no further unwanted forms had been added since I last exported the forms.

6th March 2023

This week I continued to focus primarily on the development of the Books and Borrowing front-end, making various updates to the search and browse facilities. I investigated an issue with authors not appearing in the Solr data and located the cache generation scripts I'd previously written but hadn't re-run recently. These include generating cached data for authors and also things like number of borrowings. I have now updated my 'Solr data generation how to' document to note that these cache generation scripts need to be re-run so hopefully this shouldn't be an issue in future. I've also re-run all of the scripts now but this won't affect anything until I send new data to the Stirling IT people.

I also updated my Solr generation scripts to split up edition languages and place of publication on the bar character (and also separating out any places in square brackets). This means each language and place will be independently searchable in future. I have also updated the advanced search form so that languages and places are split by the bar character. For language this means that where a language had a bar (e.g. 'Latin | Arabic') the associated count then gets added to each of the individual occurrences of the language. The list of languages on the advanced search form is now much more reasonable and less cluttered.

I was considering replacing the auto-complete for place of publication with a multi-select, but I've decided against this as even with the bar splitting there are still about 350 different places, which is too much. Instead I've updated the autocomplete to bring back individual places. For example previously 'Stirling' appeared in a long list of places all separated by bars and this is what was returned when you typed 'Stirling' in. Now only 'Stirling' appears.

I then created a new API endpoint to retrieve the data for a single borrower and I now use this call when a user presses on a borrower name in the search results, enabling me to display the borrower name in the 'You searched for' section rather than just the borrower ID. I have also updated things so that if you do press on a borrower name to search for all records involving the borrower, plus you can now press 'refine your search' and the borrower's details (title, forename, surname, othernames) will populate the advanced search form, which is quite nice. Note however that if you do this then press the 'Search' button this then searches on these fields and not the specific borrower ID, so the results may be very different (e.g. multiple John Smiths).

I also realised that I will need to rethink how the pubdate search works as I'd forgotten that we have both 'pubdate' and 'pubdateend' fields. Currently I have just been using 'pubdate' but this is not going to give accurate results where we have a range of years. Instead I'm generating and saving each year in the range. This allows the search to work without updating the code, and after testing it out all would appear to work very well.

I then worked on adding in more 'click to search' options to the search results page. I added in a search option for 'Holding title' that when clicked on searches for the holding title's ID (and populates the search form with the holding title if you choose to refine your search). Unfortunately I realised that I had somehow not included the book holding ID as a field in the Solr data. I therefore updated the schema on my laptop and regenerated and ingested the data into Solr on my laptop to check that the process worked. Thankfully all went smoothly, but it does mean that until I next update the live Solr data the 'click to search' for holding title doesn't actually provide any results.

I also added in 'click to search' for book edition and book work title. I had included the book edition and work IDs in the Solr data so thankfully I was able to get these searches working fully. As with book holding, when you click on a title to perform the search and then choose to refine your search the edition / work title appears in the advanced search form.

I also added in a similar option for authors. You can now click on an author's name at any of the levels of association and a search for author ID will be performed, with author forename and surname appearing in the advanced search form if you 'refine'.

I then moved on to adding in 'click to search' for edition language and publication place. These also had their own challenges as I needed to split up languages / publication places into individual clickable areas based on the bar character and also the square brackets for publication place. I managed to get it all working, but of course this search won't work properly until the Solr data is updated anyway.

13th March 2023

I continued to add 'click to search' options to all of the data. In the search results the format, editors and translators all now can be clicked on. I've also separated out the ESTC. The button linking to the BL site is still present on the right, but ESTC is also now listed like other data types on the left, and as with other data types you can click on it to perform a search in our site for the ESTC.

I also updated everywhere else that the data is displayed to incorporate the 'click to search' options too. This includes the register pages and the 'books' and 'borrowers' tabs when you're looking at a library. I've also set the 'page' view to default to the 'image and text' view, which was requested a while back.

I then began working on Section 5 of the requirements document, which is the 'Browse Books' feature through which book editions will be listed. I needed to make some changes to the database to implement this but I encountered a problem with the online database – when I look at the structure of a table it should list the columns, but for each table this list is blank. I was a little worried about adding new columns so I contacted Mike at Stirling but unfortunately he was out of office. Therefore I decided to work on things on my laptop instead of the live database. I made a good start with the API endpoints that will be required for this feature, and have written new scripts to generate cached data. I've not completed the feature yet but hopefully it won't take too much longer to implement.

20th March 2023

I completed an initial version of the 'Browse book' page in the Books and Borrowing front-end. The page works pretty much exactly as it was specified in the requirements document, apart from a limit by genre not being in place yet. The page presents all of the book editions in the system and works in a similar way to the 'browse book holdings' tab when viewing a library. By default the book editions are displayed by title and there are tabs for each initial letter and a count of the number of book editions with titles beginning with the letter. The data for each book edition (and associated book work where available) is listed, with searchable items clickable (e.g. click on an edition title to search for all borrowing records involving this edition). The number of borrowings is also listed as are all of the individual holding records associated with the edition.

You can change the way the book edition data is displayed to select ordering by author or the 'top 100' view in addition to title. There is also a date slider that allows you to limit the view to books published within a specific period. There is currently one book in the system with a publication date of 1968 which is why the range currently goes that far – once this has been fixed the range will automatically fix itself too. The range slider is double-ended – drag each end to match the period you're interested in then press 'Update' and the page will reload and will only display book editions published in your chosen range. So for example you can view the 'top 100' for books published in different periods and compare them.

Next week I will begin and hopefully finish the 'Browse borrowers' page. That then leaves the 'facts and figures' (both library and system-wide) plus the integration of genre to tackle. Plus working on the user interface, migrating to Bootstrap and fixing a number of little issues I've spotted.

27th March 2023

I pretty much completed the library borrowers page. As with the list of borrowers within a library, you can view the list by borrower surname or view the 'top 100' most prolific borrowers. In addition to this, there are a couple of limiting options. As with book editions, you can limit the list of borrowers by a date range. Here the date range is 'active period'. I've created new fields to store the year of the first and last borrowing for each borrower and this is the borrower's 'active period'. Note that this is something different to limiting the list to borrowers who actually borrowed a book within the selected period. If your selected range is '1780-1790' and a borrower's first borrowing was in 1777 and their last borrowing was in 1799 then the borrower will be returned as they are considered 'active' in your chosen period. It may well be that they didn't actually borrow anything in the period 1780-1790 but they'll still be returned. If instead of 'active period' we wanted to query exact years of borrowing this would be a much more complex query as each individual borrowing record for every borrower would need to be queried. I could create another Solr index for borrowers that would allow this, but I'm hoping that 'active period' is sufficient for the borrowers page. In addition to the borrower's active period you can also limit by gender, and you can combine the two limits. For example, listing all of the female borrowers by surname that were active between 1750 and 1800, or viewing the 'top 100' female borrowers that were active between 1800 and 1850.

I also added in a 'limit by occupation' feature to the borrowers page. Occupations are listed in the same way as they are on the advanced search form – you can select and deselect checkboxes. Note that I've also removed the unassigned occupations from both this list and the advanced search list. Unlike the advanced search list, the occupations that are displayed vary depending on the other limit options you have selected. For example, if you limit the list of borrowers to 'Female' then only those occupations that have been assigned to female borrowers are displayed, together with the number of assigned borrowers. When you select one or more occupations the returned borrowers and the tabs update to reflect your choice, as they do with gender and borrowing period. I haven't had time to add in a limit by library yet, but I will do so eventually.

10th April 2023

I regenerated the Solr data for the Books and Borrowing project ahead of a demonstration of the front-end in Liverpool at the end of the week. I wasn't at this event but apparently the demonstration went very well. I also tracked down and processed missing images for three library registers from Glasgow that had somehow not been incorporated into the system and I spent some time going through the site and making tweaks and fixes. These included adding a link to borrowers page from dev site homepage and adding a 'download full image' option when viewing a register page, which opens the full image in a new browser tab and from there you can save it. I also fixed an issue with book item part number ordering – previously the volumes were ordered messily and now they are in numeric order wherever volumes are listed. I also removed author years of birth and death from the 'simple' advanced search, as requested, and added in

borrower forename and surnames to the 'simple' advanced search, as requested. I also fixed the search results sort by date not working. This was caused by a bug that took a while to track down. All of the search results ordering options should now be working. I then fixed an issue with empty tabs appearing in borrower list alphabetical tabs. This was being caused by new borrowers being added since generating cached data. I've fixed the issue now so the empty tab should no longer appear. I also managed to fix the bar chart legend being cut off the bottom of the visualisations on the 'libraries' and 'library intro' pages and updated the colours used in the bar chart to differentiate them a bit better. I also fixed the date filter in the search results so it no longer displays the bar chart or 'view decade' button if a single year is searched for. It was doing so previously and then pressing on the options broke things. Finally, I fixed an issue with the apostrophe in the occupation "Advocate's Clerk" breaking the search. I also spent some time preparing a demonstration of the site that I'll be giving at the project's conference in Stirling next week.

17th April 2023

On Monday this week I attended the Books and Borrowing conference in Stirling. I gave a demonstration of the front-end I've been developing which I think went pretty well. Everyone seems very pleased with the site and its features.

1st May 2023

I ran a query and generating data about all of the book holding records that currently have no associated book edition record in the system (there are about 10,000 such records). We had also received the images for the final two registers in the Advocates Library from the NLS digitisation unit and I spent some time downloading these, processing the images to remove blank pages and update the filenames, uploading the images to our server and then running a script to generate register and page records for each page in both registers. These should be the last registers that need to get added to the system so it's something of a milestone.

8th May 2023

I sorted out the page numbers for one of the new registers I added to the system last week (the other one didn't need sorting). Page numbering was a real mess in this one and it's taken quite some time to get our page numbers aligned with what is written in the images.

I also spent some time investigating a couple of issues that Matt had reported. The first one was empty, unlabelled book items sometimes appearing in book holding records. I think there is a bug in the CMS that can result in a blank book item getting created in certain circumstances. As we're sort of nearing the end of the project and debugging the CMS would likely be a long and painful process I pointed out that I would rather run a script to strip out all unlabelled book items that have no associated borrowings as a quick fix instead. We could then run this script multiple times as required. I ran a query that identified all unlabelled book items that have no associated borrowings, and there are about 2,400 of them. I could then run a script to delete all of these items, which would remove the various 'Volume (0 borrowings)' from the book holding records. What it won't do is fix the unlabelled volumes that do have borrowings (e.g. 'Volume (1 borrowing)' items). However, I could run another script to output a list of all book items that have no label but have at least one borrowing if that might help to fix these.

The second issue Matt pointed out is that the number of borrowing records listed for book holdings in the overall list of book editions does not match up with the number of search results found when you perform a search for the holding. This is because the counts on the list of books page are the total number of times each book item for each book holding have been borrowed (so in Matt's example adding up all of the times each book item was borrowed gives 159). The search results display a count of borrowing records, and a single borrowing record can include multiple book items. For example one borrowing record may involve three book items but only counts as one borrowing record. In Matt's example the 108 search results are borrowing records, not book items.

This is a bit confusing and I wonder what we could do to clarify this. I did wonder whether the search results could also tally up the number of book items in the results and display this alongside the 'your search matched x borrowing records', but unfortunately we only have this data for a subset of results (up to 100 on each page of the results). Perhaps we need to add some explanatory text somewhere, or perhaps the number of borrowings on the 'Books' page should also count the number of associated borrowing records rather than the total number of times each item was borrowed. Or I could update the Solr index to cache the total number of book items for each borrowing record.

I also updated the overall list of borrowers in the front-end, adding in a 'limit by library' option. I had hoped that this would only take a couple of hours to implement but it ended up being rather complicated to get working as I needed to update several API endpoints to incorporate the limit by library, update the way the URLs work on the borrowers page and update the JavaScript that processes the changing of the view and the selection of filter items, in addition to incorporating the list of libraries into the page. However, I manage to get the feature completed by the end of the week.

One thing to note is that there is often a discrepancy between the number of borrowers listed in the occupations section and the number in the tabs. For example, when limiting the list to female borrowers with an 'Arts and Letters' occupation the 'Arts and Letters' occupation shows 7 borrowers but if you count the borrowers in the tabs there are only 5. The reason is a borrower can have multiple occupations. For example, Anne Grant is both poet and author so counts as one borrower in the 'G' tab but one each for 'Poet' and 'Author' in the occupation counts. I also wonder whether I should add in counts of borrowers at each library to the 'limit by library' page. Something to consider, anyway.

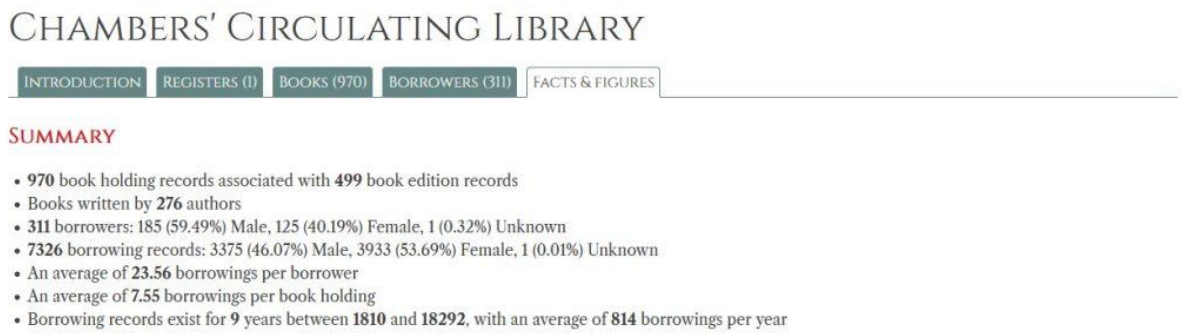
I also encountered some weirdness with the site today – specifically any searches for things with spaces in them (e.g. occupation 'Arts and Letters') just displayed a 403 Forbidden page. This was before I'd made any updates to the code so it wasn't that I'd inadvertently broken something. It turned out that a recent update to the Apache server software was breaking any URL that had a space in it. If you really want to find out more you can read here:

https://stackoverflow.com/questions/75684314/ah10411-error-managing-spaces-and-20-in-apache-mod-rewrite

I had to spend some time figuring out how this change has affected the site and making updates to avoid the 403 error pages. I think I've sorted most things out when using the front-end, although accessing the API directly is still causing problems. I'll need to sort this out next week. I'll also deal with the Hunterian images, which it turns out had not yet been added to the system, and will hopefully move on to the 'Facts and Figures' pages and their visualisations next week too.

15th May 2023

I spent the majority of this week continuing to work for the Books and Borrowing project. I added in images of both the Hunterian and Inverness Kirk Sessions library register pages, as these had not yet been integrated. I then began working on the library 'Facts and Figures' page and added in the API endpoints and processing of library statistics, which now appear in a 'summary' section at the top of the page, as you can see for the Chambers library in the following screenshot:



We may need to include some further explanation of the above. The number of borrowing records does not match the numbers split by gender, which may be caused by some borrowing records not having an associated borrower or others having more than one associated borrower. Also as you can see a borrowing record in this library has an erroneous year, making it look like the borrowing records stretch into the distant future of 18292.

Beneath the summary are a series of 'top ten' lists, featuring the top ten borrowed books, authors and most prolific borrowers, both overall and further broken down by borrower gender, as you can see below:

Most borrowed books by Male borrowers

1. Pelham; or the Adventures of a Gentleman (**125**)
2. Blackwood's Edinburgh Magazine (**110**)
3. Disowned (**87**)
4. Edinburgh Literary Journal (**84**)
5. Anne of Geierstein (**74**)
6. Devereux (**74**)
7. Adventures of a Kuzzilbash (**73**)
8. Zillah: a Tale of the Holy City (**70**)
9. Borderers (**69**)
10. Sailors and Saints (**66**)

Most borrowed books by Female borrowers

1. Disowned (**126**)
2. Blackwood's Edinburgh Magazine (**109**)
3. Borderers (**99**)
4. Pelham; or the Adventures of a Gentleman (**90**)
5. New Forest (**83**)
6. At Home (**77**)
7. Salathiel (**73**)
8. Rank and Talent (**67**)
9. Zillah: a Tale of the Holy City (**66**)
10. Hungarian Tales (**66**)

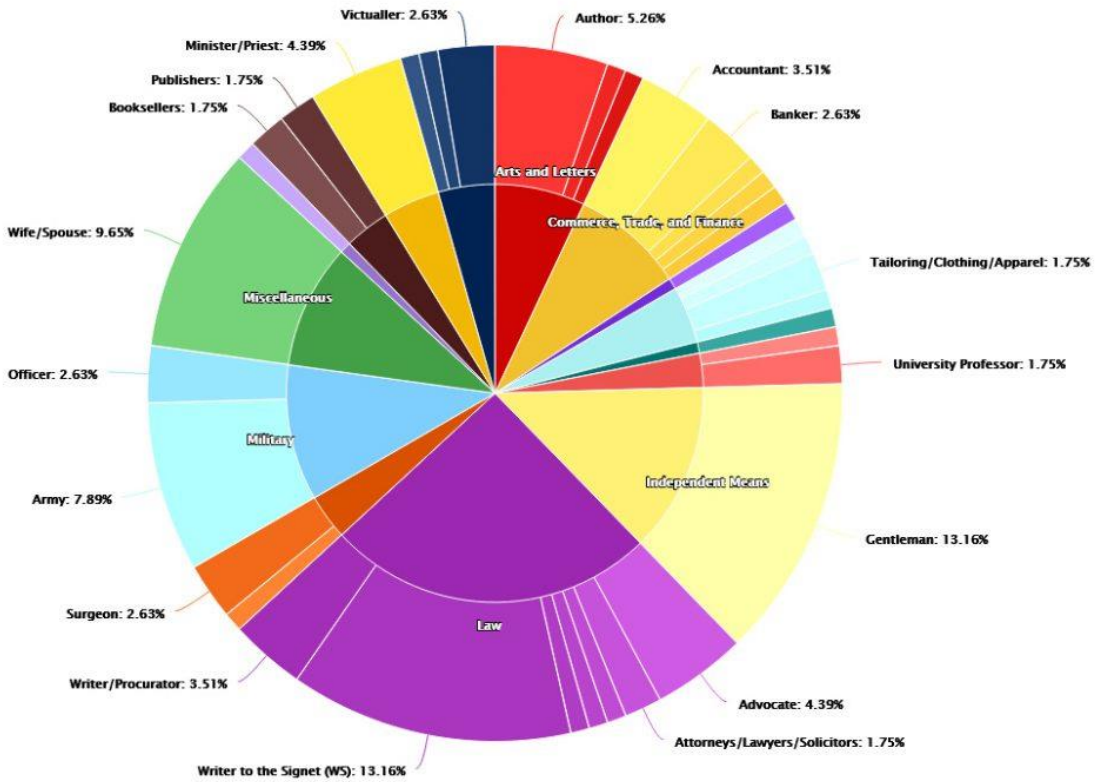
These all link through to search results for the item in question. Note that we once again face the issue of numbers here reflecting volumes borrowed while the search results count borrowing records, which can include any number of volumes. I'm going to have a serious think about what we can do about this discrepancy as it is bound to cause a lot of confusion. There is also something very wrong with the author figures that I need to investigate. I think the issue is authors can be associated with any level of book record and I suspect authors in this list are getting counted multiple times.

There are also issues caused by the data still being worked on while the Solr cache and other cached data become outdated. I spent quite a while on Tuesday trying to figure out why a book holding wasn't appearing where it should or with the expected number of borrowings until I realised the data in the CMS had been updated (including the initial letter of the book title) while the cache hadn't.

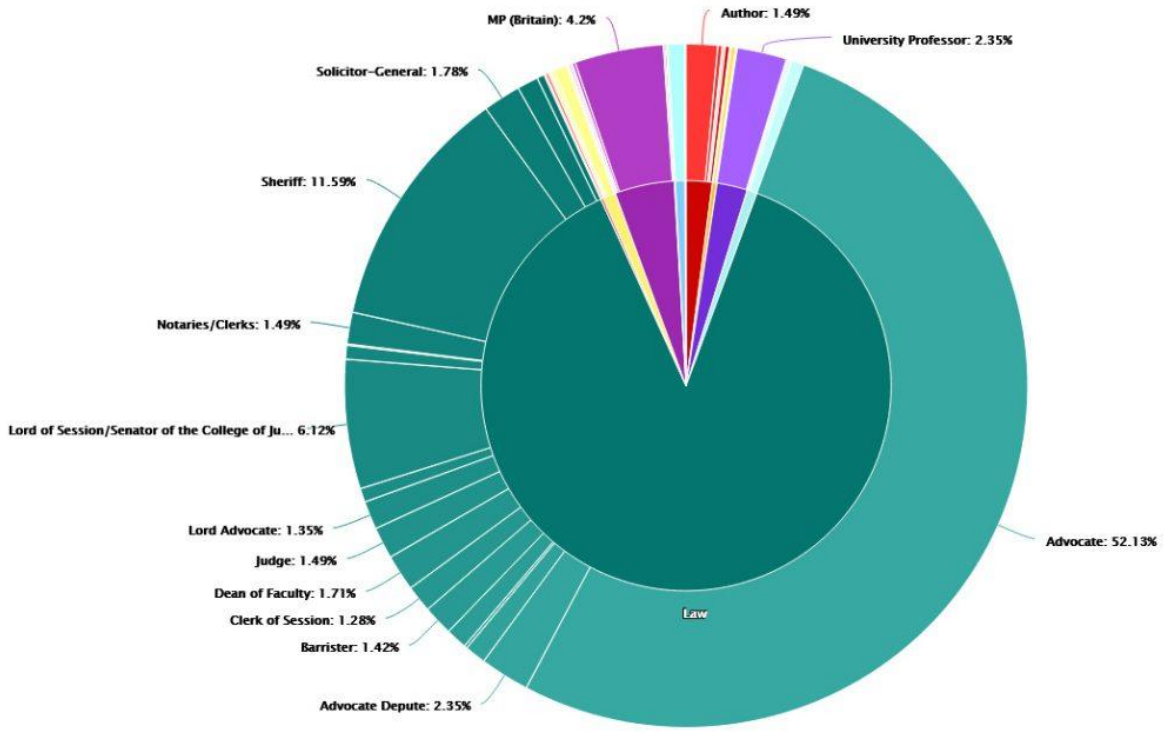
I then moved onto working on some visualisations. I created a borrower occupations donut chart as described in the requirements document. This chart shows the distribution of borrower occupations across library borrowers in a two-level pie chart with top-level occupations in the middle and these then subdivided into secondary level occupations in the outer ring. Note that I haven't further split 'Religion and Clergy' > 'Minister/Priest' into its third

level occupations as it is the only category that has three levels and it would have made the visualisation too complicated (both to develop and to comprehend in the available space). Instead the individual ‘minister / priest’ categories are amalgamated into ‘minister / priest’.

The charts are a nice way to get an overall picture of the demographics of the selected library and also to compare the demographies of different libraries. For example, Chambers has a very broad spread of occupations:



Whereas Advocates (as you’d expect) is more focussed:



You can hover over each segment to view the title of the occupation and the percentage of borrowers that have the occupation. You can also click on a segment to perform a search for the occupation at the library in question. You should bear in mind that a borrower can have multiple (or no) occupations so the number of borrowers and the number of occupations will be different. The next thing I'll do is create a further donut chart showing the number of borrowings per occupation, but I'll leave that to next week.

I also thought some more about the thorny issue of number of borrowings versus number of borrowing records. What I think I'll try and do is ensure that the only number that is shown is the number of associated borrowing records. I'll need to experiment with how (or if) this might work, which I will leave until next week.

22nd May 2023

The PI Katie had suggested that we add author gender as a new field and I engaged in an email discussion about how best to implement this. It will take some time as although it's simple enough to add the extra field to the database, the CMS, the API, the Solr data export, the front-end searches, browses and display of data will all need to be updated to incorporate the new option. I generated a list of the current authors in the system as a spreadsheet and sent this to Katie, who is going to fill in a new 'gender' field and send it back to me to work with, probably next week.

One of the major issues I tackled this week was what to do about the numbers of times a book is borrowed (which appears in several places throughout the front-end) not matching the number of associated borrowing records (as found in the search results). The issue is caused by a book holding possibly consisting of multiple volumes. One borrowing record may involve any number of volumes, so although in the search results we have one borrowing record, in the counts of borrowed books the total number for a book holding may be very different.

I decided to updated the system so that the number of borrowing records in addition to the number of times borrowed is stored in the cache and displayed on the site. This meant updating the database and writing a new cache generation script, then updating the API to bring back the new data and the front-end to display it. So for example the the Co-I Matt noticed a couple of weeks ago that "the Buffon [book] is listed like this: 'Number of borrowings: 159'. However, if you click on the holding title, the resulting search has 108 results". After the update the number of borrowings for the holding record now states: "Volumes of this book were borrowed 159 times in 108 borrowing records".

I've also updated book edition counts, for example: "Volumes associated with this edition were borrowed 316 times in 233 borrowing records". The 'top 100' list also now uses the number of borrowing records to order things, although this generally doesn't change things too drastically. It should be noted that discrepancies between number of times borrowed and number of associated borrowing records only happen when a book has multiple volumes. Therefore if a book doesn't have multiple volumes then the text just states the number of borrowings.

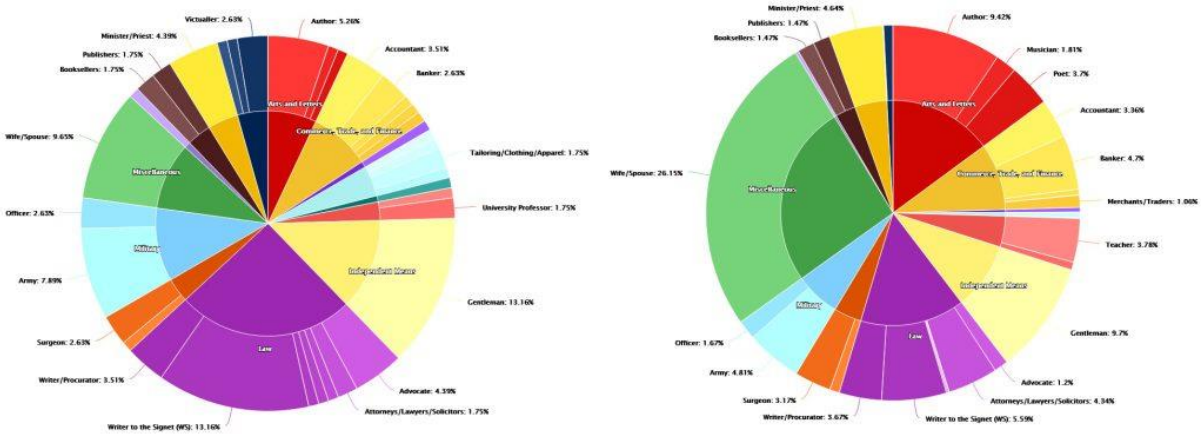
I have updated the book edition page to display the new borrowing record information too, and I've also updated the 'top ten lists' in the 'facts' page to use the borrowing record figures. In addition I have fixed the issue with the numbers in the 'top ten authors' lists being wrong. Figures relating to borrowers rather than books should have been correct anyway so I haven't needed to update these. Hopefully this will considerably cut down on the potential for confusion.

Also this week I wrote a little script to add in the normalised borrower occupation for borrowers from the Royal High School and returned to the visualisations on the library 'facts and figures' page. I've implemented a further donut chart for borrower occupations, this time showing the number of borrowings per occupation. I've also updated the text above the chart to show more information about how the figures are derived. For example for Chambers it now says:

"Of the 311 library borrowers 101 have one or more identified occupations. The total number of occupations associated with borrowers at this library is 114 and these are represented in the following chart".

Hopefully this makes it clearer that the numbers represent occupations and these are not necessarily present for all borrowers and some borrowers may have more than one occupation.

The chart of borrowings per occupation makes for an interesting comparison with the chart for borrower occupations, for example in Chambers the occupation 'Wife/Spouse' (show in green) represents 9.65% of the total occupations in the library but 26.15% of the borrowing records, as you can see from the following screenshot:



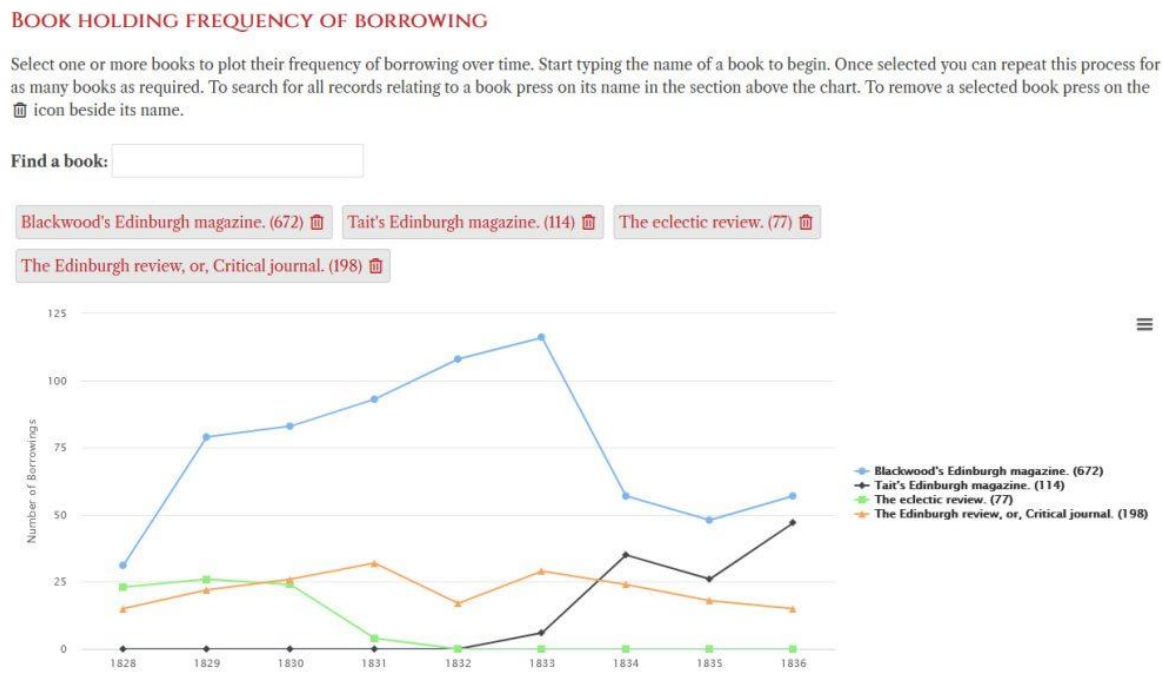
29th May 2023

Unfortunately my access to the Stirling University VPN stopped working on Wednesday and access wasn't restored until late on Thursday afternoon. As I am unable to access the project's server and database without VPN access this limited what I could do, although I did manage to work on some code 'blind' (i.e. without uploading it to the server and testing it out) for much of Wednesday.

For the project this week I generated a list of Haddington book holdings that don't have any borrowings so the team could test some things. I also added in author gender as a field and wrote a script to import author gender from a spreadsheet, together with tweaks made to all of the author name fields. I still need to fully integrate author gender into the CMS, API and front-end, which I will focus on next week.

I spent the rest of my time on the project this week developing a new visualisation for the library facts and figures page. It is a line chart for plotting the number of borrowing records for book holdings over time and it features an autocomplete box where you can enter the name of a book holding. As you type, any matching titles appear, with the total number of borrowings in brackets after the title. If the title is longer than 50 characters it is cropped and '...' is added. Once you select a book title a line chart is generated with the book's borrowings plotted. You can repeat this process to add as many books as you want, and you can also remove a book by pressing on the 'delete' icon in the list of selected books above the chart. You can also press on the book's title here to perform a search to view all of the associated borrowing records.

Hopefully this chart will be of some use, but its usefulness will really depend on the library and the range and number of borrowings. So for example the image below for Wigtown shows a comparison of the borrowings for four popular journals, which I think could be pretty useful. A similar comparison at Chambers is less useful, though, due to the limited date range of borrowing records.



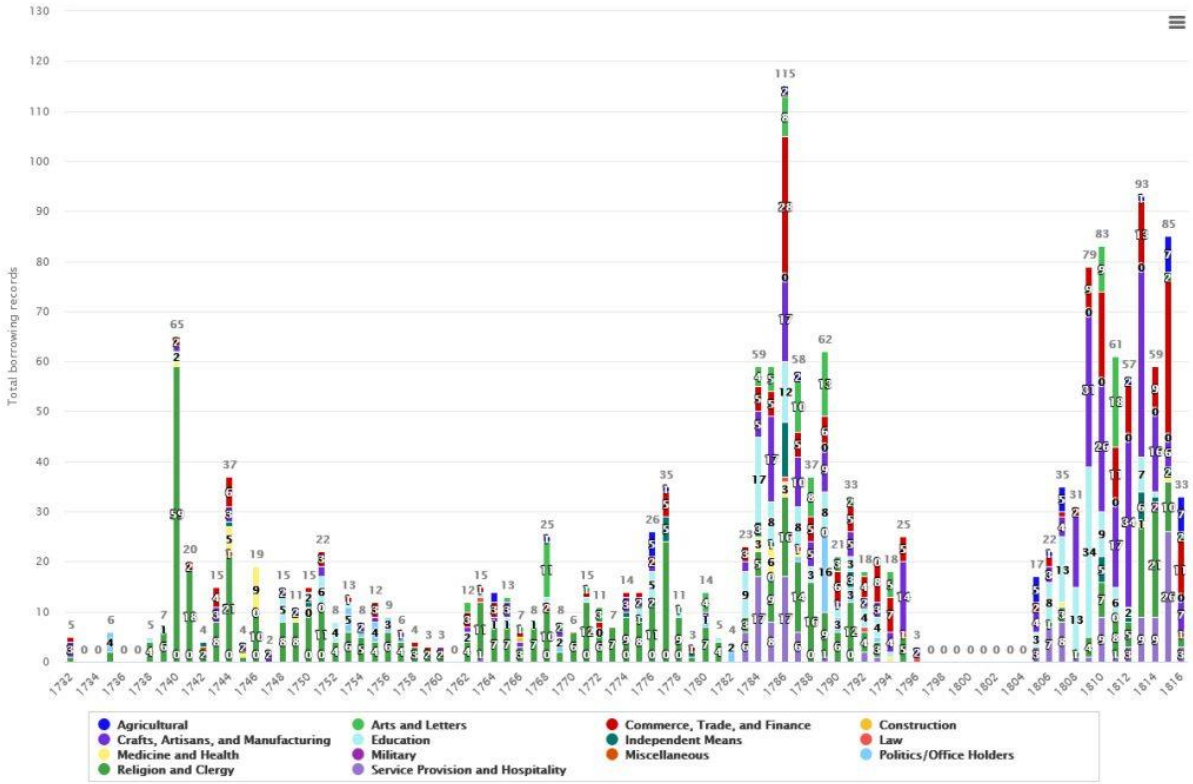
5th June 2023

I continued to work on the Books and Borrowing project for most of this week. One of my main tasks was to fully integrate author gender into the project's various systems. I had create the required database field and had imported the author gender data last week, but I still needed to spend some time adding author gender to the CMS, API, Solr and the front-end. It is now possible for the team to add / edit author genders through the CMS wherever the author add / edit options are available and all endpoints involving authors in the API now bring back author gender. I have also updated the front-end to display gender, adding it to the section in brackets before the dates of birth and death.

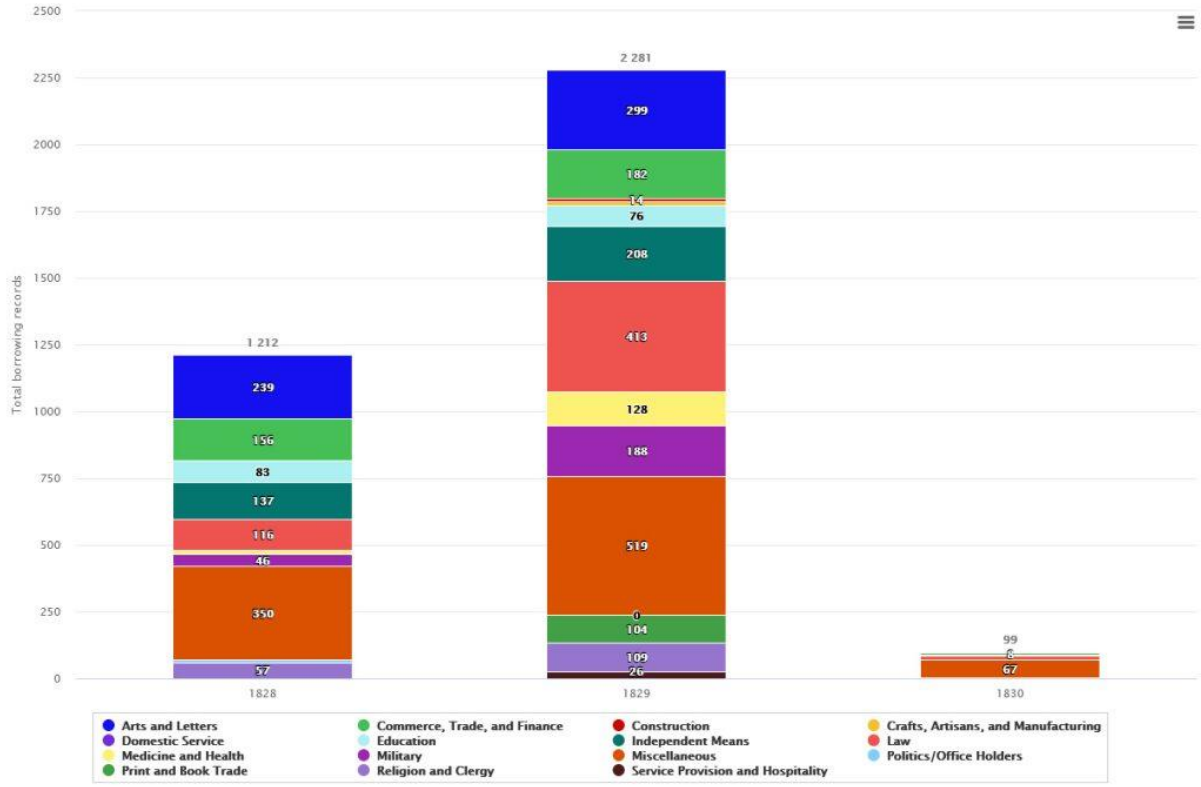
I have also added author gender to the advanced search forms but a search for author gender will not work until the Solr instance is updated. I prepared the new data and have emailed our IT contact in Stirling to ask him to update it, but unfortunately he's on holiday for the next two weeks so we'll have to wait until he gets back. I have tested the author gender search on a local instance on my laptop, however, and all seems to be working. Currently author gender is not one of the search results filter options (the things that appear down the left of the search results). I did consider adding it in (as with borrower gender) but there are so few female authors compared to male that I'm not sure a filter option would be all that useful most of the time – if you're interested in female authors you'd be better off performing a search involving this field instead rather than filtering an existing search.

I then moved on to developing a further (and final) visualisation for the library facts and figures page. This is a stacked column chart showing borrowings over time divided into sections for borrower occupation (a further version with divisions by book genre will be added once this data is available). I've only included the top level occupation categories (e.g. 'Education') as otherwise there would be too many categories; as it is the chart for some libraries gets very cluttered.

It took quite some time to process the data for the visualisation but I completed an initial version after about two days of work. This initial version wasn't completely finished – I still needed to add in the option to press on a year, which will then load a further chart with the data split over the months in the chosen year. Below is a screenshot of the visualisation for Haddington library:

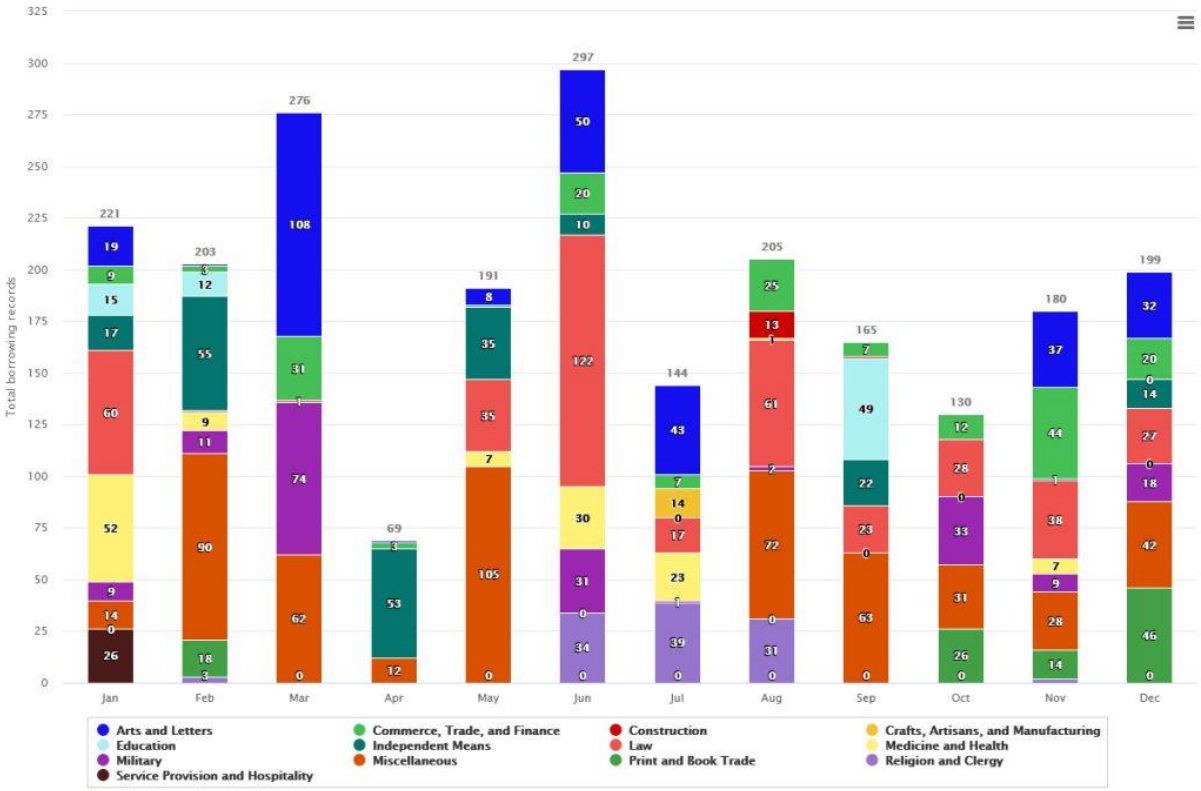


With the above example you'll see what I mean about the chart getting rather cluttered. However, you can switch off occupations by pressing on them in the legend, allowing you to focus on the occupations you're interested in. For some libraries the visualisation is a little less useful, for example Chambers only has data for three years, as you can see below:



I also had to place a hard limit on the start and end years for the data because Chambers still has some dodgy data with a borrowing year thousands of years in the future which caused the script to crash my browser as it tried to generate the chart. Note that the colours for occupations match those in the donut charts displayed higher up the page to help with cross-referencing.

Later in the week I completed work on the 'drilldown' into an individual year for the visualisation. With this feature you can press on one of the year columns and the chart will reload with the data for that specific year, split across the twelve months. This drilldown is pretty useful for Chambers as there are so many borrowings in each year. It's useful for other libraries too, of course. You can press on the 'Return to full period' button above the chart to get back to the main view and choose a different year. Below is an example of a 'drilldown' view of the Chambers library data for 1829:



As I developed the drilldown view I realised that the colours for the sections of the bar charts were different to the top-level view when the drilldown is loaded, as the colours are assigned based on the number of occupation categories that are present, and in the drilldown there may only be a subset of occupations. This can mean 'Religion' may be purple in the main view but green in the drilldown view, for example. Thankfully I managed to fix this, meaning the use of colour is consistent across all visualisations.

The next item on my agenda is to create the site-wide 'Facts and Figures' page, which will work in the same way as the library specific page, but will present the data for all libraries and also allow you to select / deselect libraries. I spent some further time updating the API endpoints I'd developed for the visualisations so that multiple library IDs can be passed. Next week I'll create the visualisations and library selection options for the site-wide 'Facts' page. After that I will probably move onto the redesign of the front-end to use Bootstrap, during which I will also redesign many of the front-end pages to make them a bit easier to use and navigate. For example, the facts and figures page will probably be split into separate tabs as it's a bit long at the moment and the search form needs to be reworked to make it more usable.

12th June 2023

I completed an initial version of the site-wide facts and figures page. The page functions in a very similar way to the library-specific facts and figures page as discussed in previous weeks, but here you can view the data for any or all libraries rather than just a single one. Libraries are selectable from a series of checkboxes as in the advanced search, allowing you to group together and compare library data, for example the data for the three University libraries or data from libraries in southern Scotland. However, the page can take about 30 seconds to load as it's processing an awful lot of data for the default view (which is data for every library). I'm going

to create a cached version of the data for this particular view and possible others, but I haven't got round to it yet.

The 'Summary' section provides an overview of the amalgamated data for your chosen libraries and the 'top ten lists' are also amalgamated. The book holding and prolific borrowers lists include the relevant library in square brackets after the number of associated borrowing records as these lists contain items that are related to specific libraries (borrowers and book holdings 'belong' to a specific library). I've also added in the top ten borrowed book editions, as these are site-wide, as are authors. The links through to the search results for these lists include your selected libraries to ensure the search results match up. For example, Sir Walter Scott is the most borrowed author when viewing data for Dumfries, Selkirk and Wigtown and when you press on the author's name you search is limited to these libraries. The occupations and borrowings through time visualisations contain data for all selected libraries and rather than a 'book holding frequency of borrowing' chart at the bottom of the page there is a 'book edition' version as editions are site-wide. As with other data, the editions and borrowings over time returned here are limited to your chosen libraries and links through to the search results also incorporate these.

I still need to address a couple of things, though. Firstly, book edition titles have not currently been cut off after 50 characters as I realised doing so for titles with non-Latin characters (e.g. Greek) broke the page. This is because characters such as Greek take up multiple bytes while Latin characters take up one byte each. The default substring method for chopping up strings is not 'multibyte safe', meaning non-Latin characters can get split in the middle of the data, which results in an error. There is a handy multibyte version of the substring method but unfortunately multibyte functions have not been installed on the server so I can't use it. Once I've managed to get this installed I'll add in the limit. Secondly, I've noticed that the visualisations don't work very well on touchscreens so I'm going to have to rework them to try and improve matters. For example, the donut charts allow you to click on a section to perform a search for the section's occupation, but as touchscreens have no hover state this means on a touchscreen it's not possible to view the tooltip that includes the name of the occupation and the exact percentage. Also, the click through to a year on the 'borrowings through time' chart don't seem to work on my iPad and the book edition autocomplete doesn't seem to be firing either.

Also for the project this week I assigned some borrowing records for a borrower in Wigtown to another borrower, as it turned out these were the same person. This also involved updating the borrower's total number of borrowings and active borrowing period. I then began to look into migrating the website's theme to Bootstrap. I'm going to be working on this on a local version of the site running on my laptop and I began getting things set up.

19th June 2023

I continued to work for the Books and Borrowing project this week, switching the search facilities over to use a new Solr index that includes author gender. It is now possible to incorporate author gender into searches, for example bringing back all borrowing records

involving books written by women. This will be a hugely useful feature. I also fixed an issue with a couple of page images of a register at Leighton library that weren't displaying.

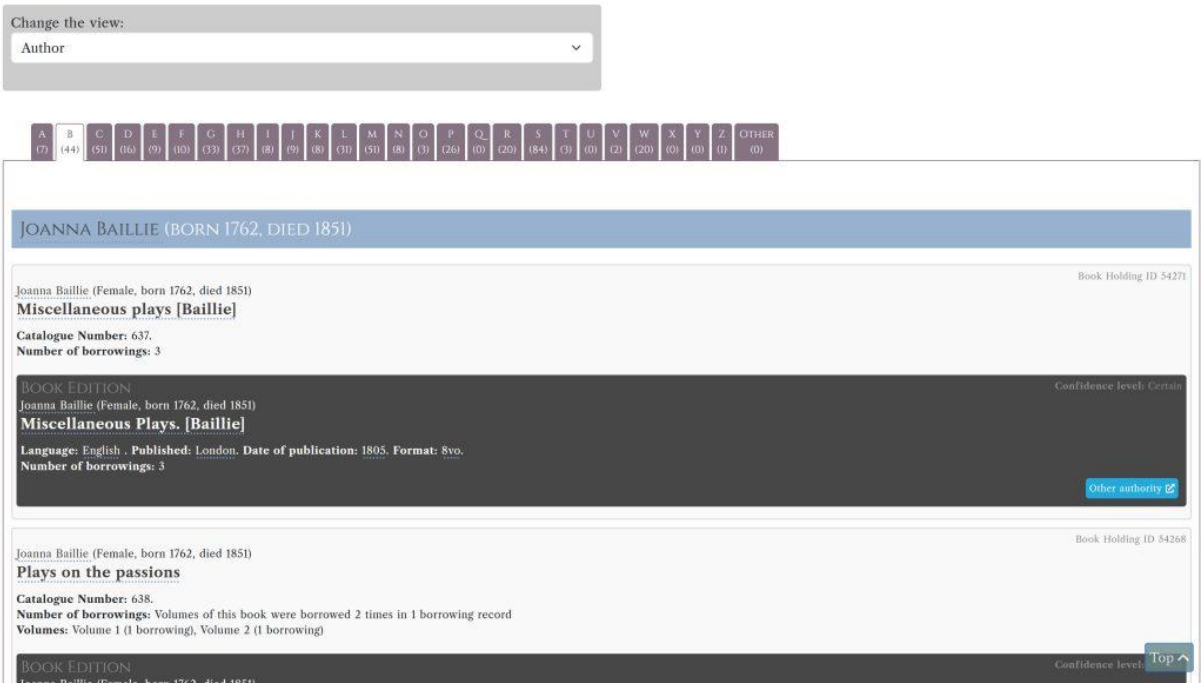
The rest of my time this week was spent developing a new Bootstrap powered interface for the project's website, which is now live (https://borrowing.stir.ac.uk/). You'd struggle to notice any difference between this new version and the old one as the point of creating this new theme was not to change the look of the website but to make Bootstrap (https://getbootstrap.com/) layout options available to the dev site. This will allow me to make improvements to the layout of things like the advanced search forms. I haven't made any such updates yet, but that is what I'll focus on next.

It has taken quite a bit of time to get the new theme working properly – blog posts with 'featured images' that replace the site's header image proved to be particularly troublesome to get working – but I think all is functioning as it should be now. There are a few minor differences between the new theme and the old one. The new theme has a 'Top' button that appears in the bottom right when you scroll down a long page, which is something I find useful. The drop-down menus in the navbar look a bit different, as does the compact navbar shown on narrow screens. All pages now feature the sidebar whereas previously some (e.g. https://borrowing.stir.ac.uk/libraries/) weren't showing it. Slightly more text is shown in the snippets on the https://borrowing.stir.ac.uk/project-news/ and other blog index pages. Our title font is now used for more titles throughout the site. I've also added in a 'favicon' for the site, which appears in the browser tab. It's the head of the woman second from the right in the site banner, although it is a bit indistinct. My first attempt was the book held by the woman in the middle of the banner but this just ended up as a beige blob.

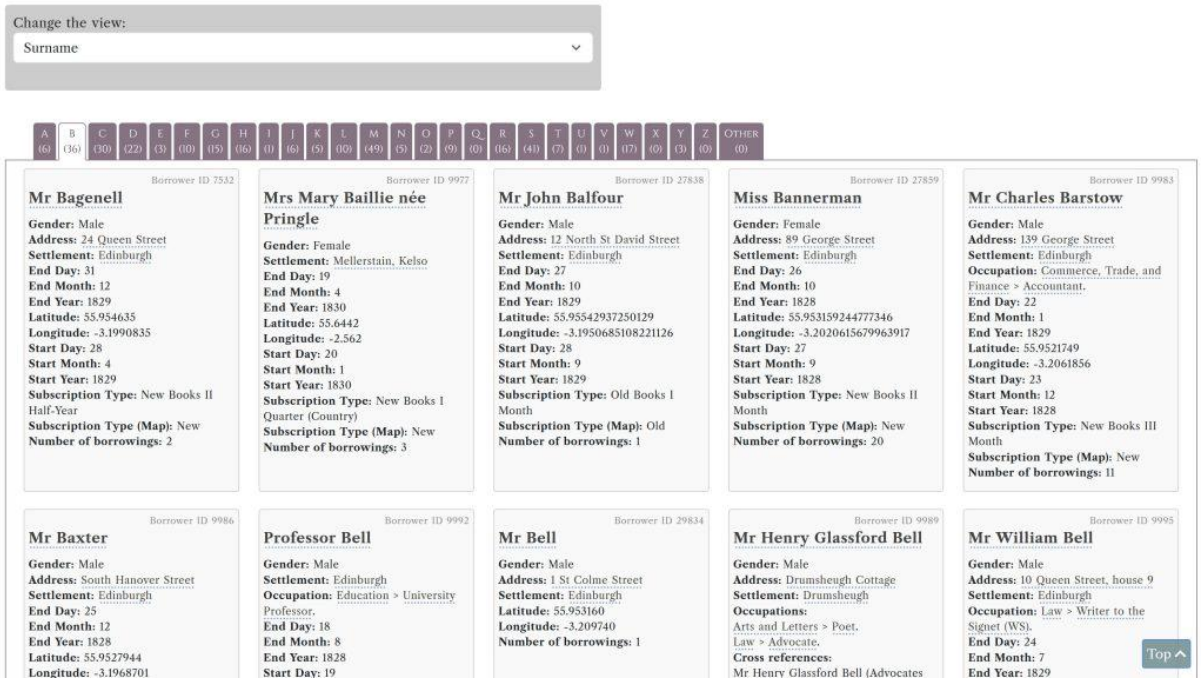
26th June 2023

I continued to work on the front-end for the Books and Borrowing project this week, and have now completed the migration of the dev site to Bootstrap. There are still some aspects that I would like to tweak further, but on the whole the layout is now much improved on all screen sizes. On the page that displays a page from a library register I have updated the navbar to make it position better on all screen sizes and I updated the image section so that it now now has a dark grey background. Text and image panels now work better on all screen sizes and 'Image view' now has a larger image viewer.

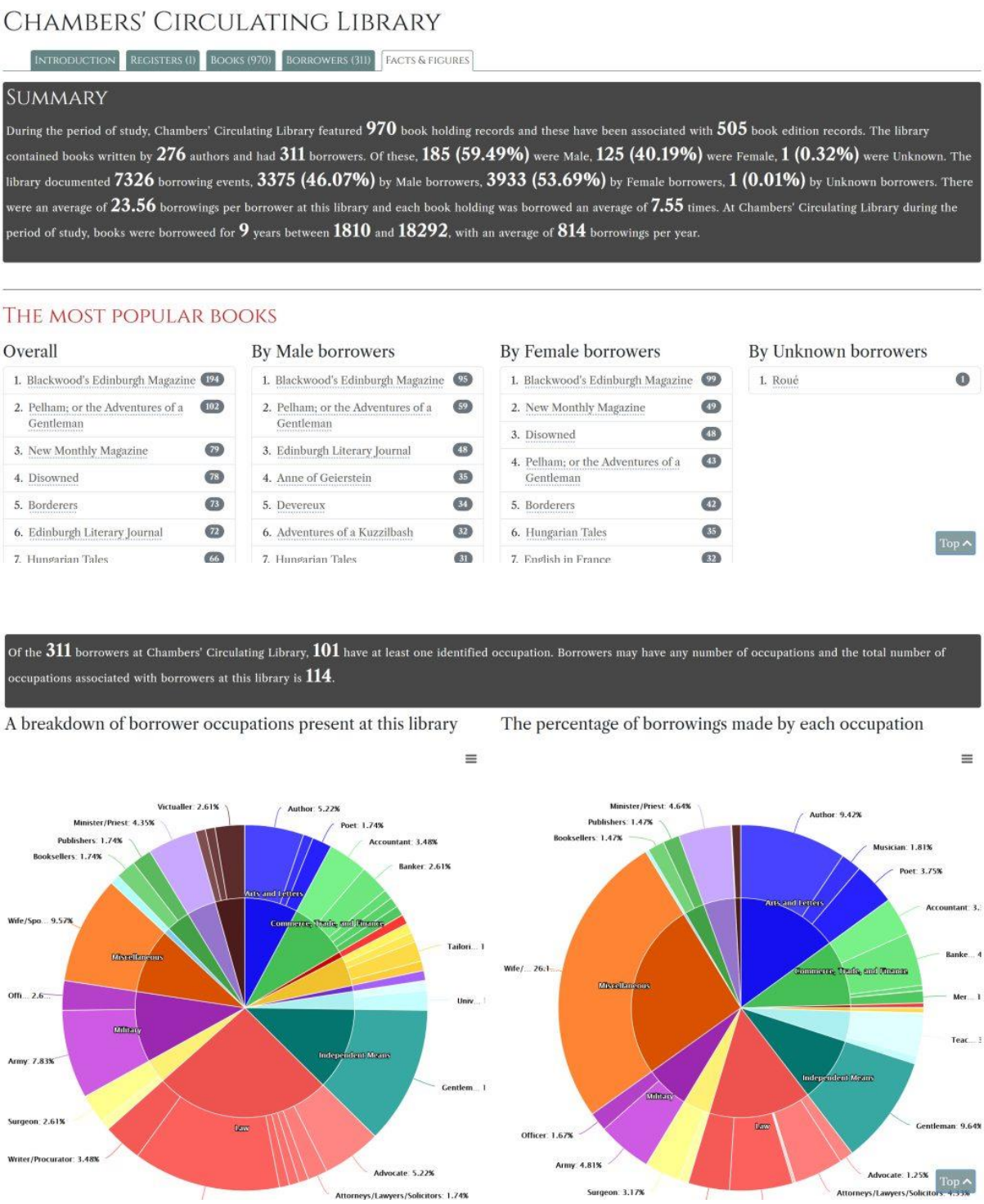
On the library books page I've improved that layout of the 'change the view' feature and when viewing books grouped by authors the author now has a blue background to help spot where the divisions between authors are. Layout of 'top 100' icon and the ESTC listing have also been improved:



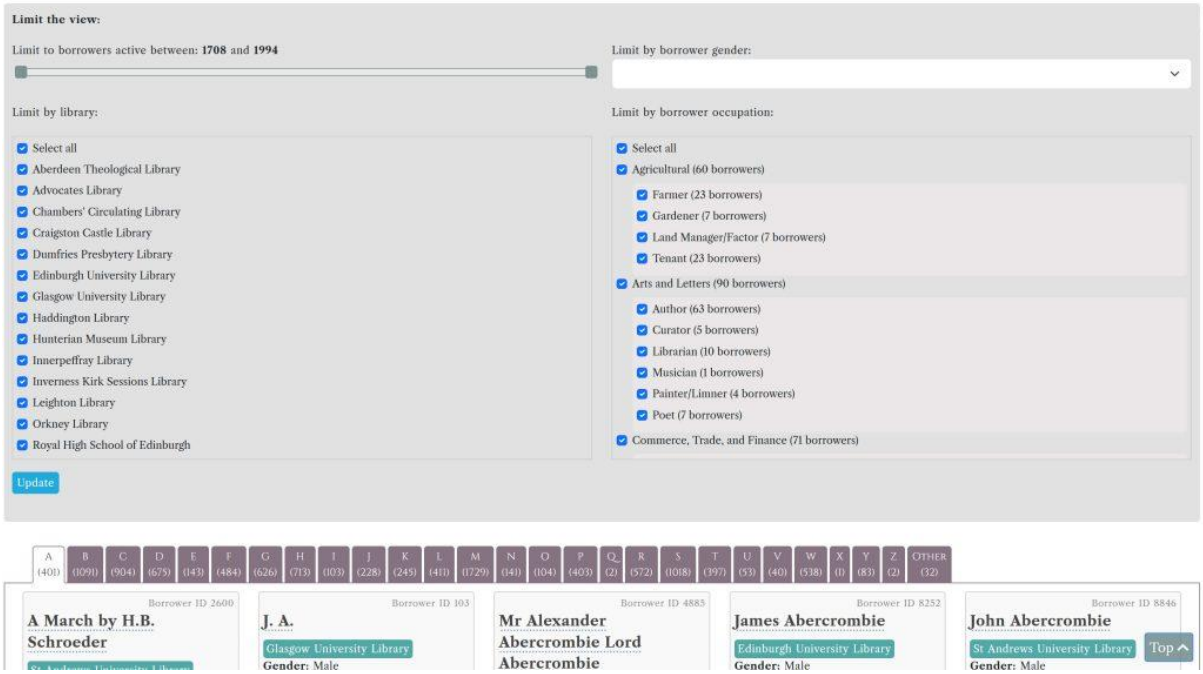
I also updated the borrowers so these are now listed in a grid with four per row to make better use of the available space:



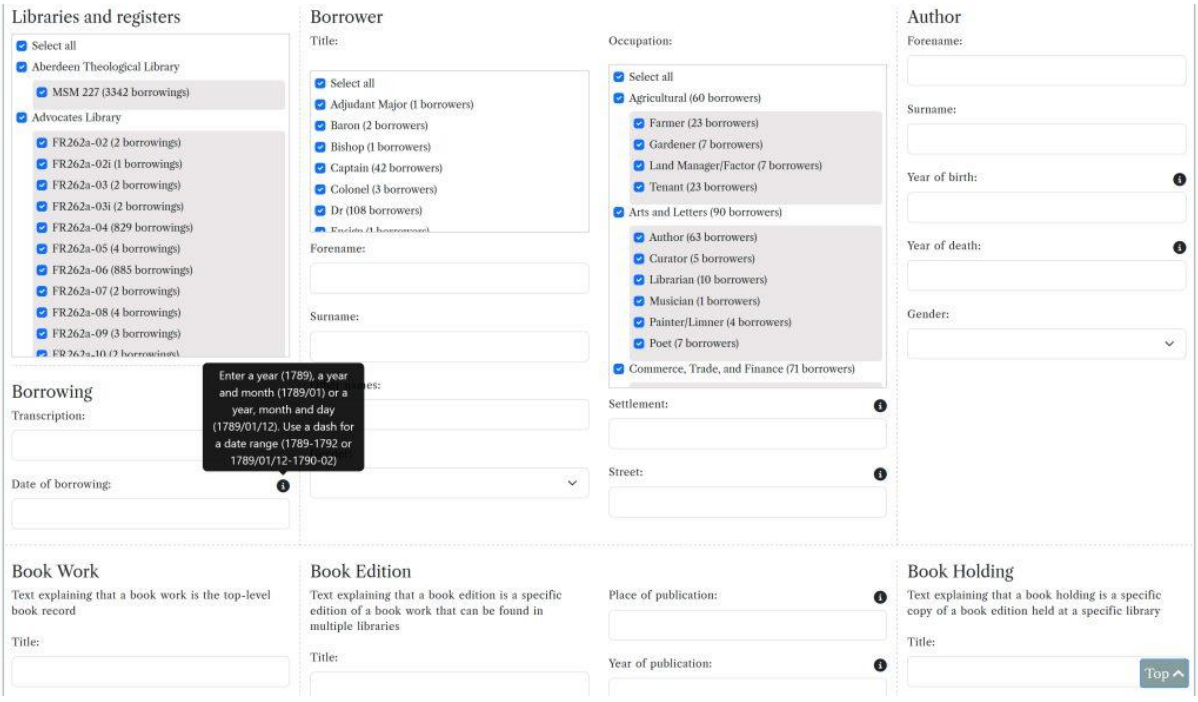
In the library ‘Facts’ page the ‘summary’ section now has a narrative flow rather than a bullet point list of figures. The ‘top 10’ lists now appear in columns with up to four per row. Layout of each list is improved. The borrower occupations summary is also more of a narrative flow and the two donut charts appear side by side (if there’s room). Similar improvements to the summary text in the other sections:



In the site-wide 'books' page the 'change view' and 'limit' sections now appear side by side and author view has the same blue backgrounds behind the author names. On the site-wide 'borrwers' page the 'Limit the view' section layout has been overhauled, giving it a two-column layout, ensuring it takes up much less space. It is still rather large, though. We could potentially hide it until the user chooses to open it. The list of borrowers is grid-based with up to five borrowers per row:



I've also massively overhauled the search forms in both the 'simple' and 'advanced' tabs. They now consist of multi-column displays grouped by data type. Tooltips are used where help information is included. Dotted lines are used to divide different types of data. Library registers with zero records are now excluded from the list:



There is still a fair amount to be done, including implement book genre, sorting out the highlighting in search results, investigating some situations when the year bar chart in the search results doesn't display properly, adding in 'cite this…' options, adding in an 'On this day' feature and a 'download data as CSV' feature, update the API index page to add in information about licensing and to ensure all endpoints are listed with good examples, updating the API to

ensure CSV output works properly for data that are arrays, integrating the dev site with the live site and ensuring all still works (e.g. the chambers map) and to add in a quick search option and navigation items as required, add in a cache for the facts and figures so it loads quicker and update the Solr index once work on the data is complete. But we're getting there!

24th July 2023

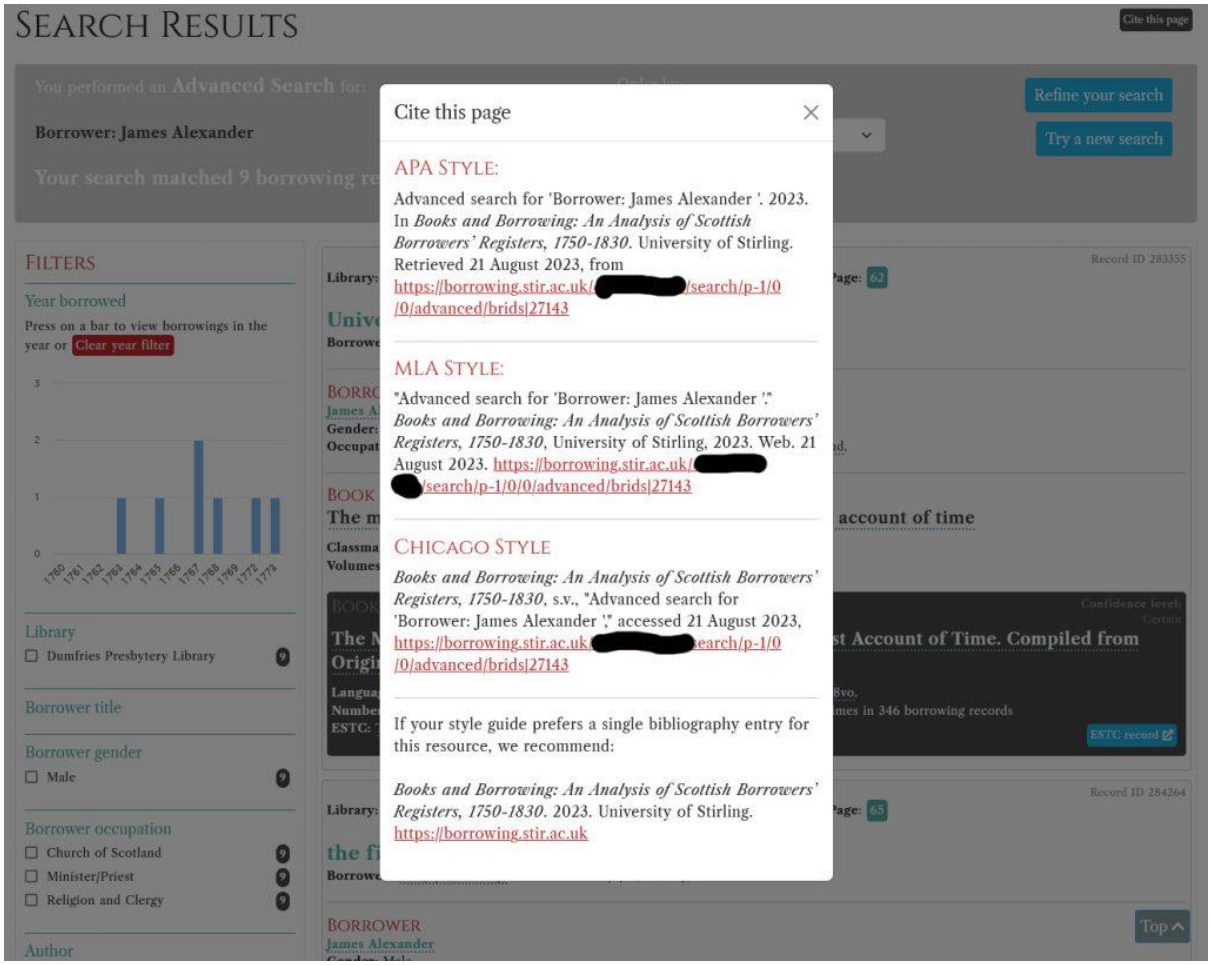
I implemented a first version of the 'On this day' feature, which I've currently added to the homepage of the dev site. What the feature does is to pick out a random borrowing for the current day and display information about it, for example:

"On this day in 1829, Mr Robert Allan, a borrower at Advocates Library borrowed 2 volumes of Histoire de la Vie et de la Mort des deux illustres fréres Corneille et Jean de Witt. by Cornelis de Witt."

The feature picks out and displays the borrower, the library, the number of volumes borrowed (if this is over 1), plus the title and the author of the book borrowed. The borrower, title and author are links to perform a search while the library is a link to the library page. There is a 'reload' button in the bottom left and when you press on this the area scroll ups and then scrolls down again with a new randomly selected borrowing from the day. There is also a link in the bottom right to view all of the borrowings for the current day. These are presented on a new page that also features options to select a different day and month, in case people want to see what was borrowed on their birthday, for example. On this day items on this page are listed in date order and then by library. It's maybe not the most serious and academic of features, but I think it's a nice addition and makes the data feel more alive.

14th August 2023

I implemented a 'cite this page' feature for the Books and Borrowing project, and the feature now appears on every page that features data. A 'Cite this page' button appears in the right-hand corner of the page title. Pressing the button brings up a pop-up containing citation options in a variety of styles. I've taken this from other projects I've been involved with (e.g. the Historical Thesaurus) and we might want to tweak it, but at the moment something along the lines of the following is displayed (full URL crudely 'redacted' as the site isn't live yet):



Developing this feature has taken a bit of time due to the huge variation in the text that describes the page. This can also make the citation rather long, for example:

Advanced search for ‘Borrower occupation: Arts and Letters, Borrower occupation: Author, Borrower occupation: Curator, Borrower occupation: Librarian, Borrower occupation: Musician, Borrower occupation: Painter/Limner, Borrower occupation: Poet, Borrower gender: Female, Author gender: Female’. 2023. In Books and Borrowing: An Analysis of Scottish Borrowers’ Registers, 1750-1830. University of Stirling. Retrieved 18 August 2023, from [very long URL goes here]

I haven’t included a description of selected filters and ‘order by’ options, but these are present in the URL. I may add filters and orders to the description, or we can just leave it as it is and let people tweak their citation text if they want.

The ‘cite this page’ button appears on all pages that feature data, not just the search results. For example register pages and the list of book editions. Hopefully the feature will be useful once the site goes live.

21st August 2023

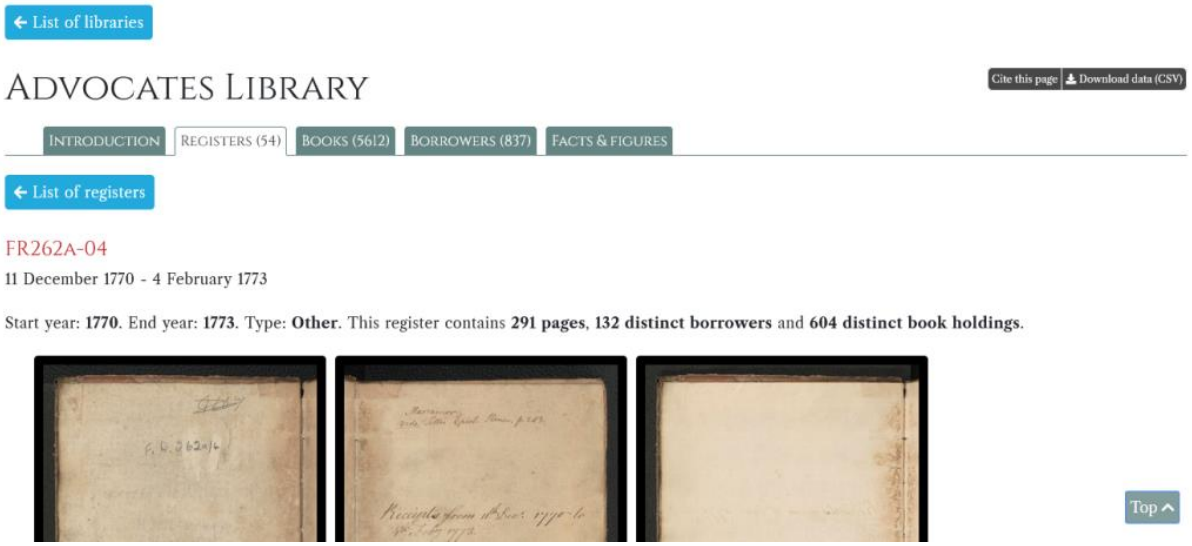
I fixed some issues with images for one of the library registers for the Royal High School for the Books and Borrowing project. These had been assigned the wrong ID in the spreadsheet I'd initially used to generate the data and I needed to write a little script to rectify this.

18th September 2023

One thing on my 'to do' list is to sort out the API. There are a few endpoints that I haven't documented yet, plus the existing documentation and structuring of the API could be improved. I spent some time adding in a license statement and a 'table of contents' that lists all endpoints. I'm currently in the middle of adding in the missing endpoint descriptions. After that I'll need to ensure the examples given all work and make sense and then I need to ensure the CSV output works properly for all data types. I'm fairly certain that some data held in arrays will not output properly as CSV at the moment and this definitely needs sorted.

25th September 2023

Last week I had begun reworking the API to make it more usable and this week I completed this task, adding in a few endpoints that I'd created but hadn't added to the documentation. I then moved onto the task of adding 'Download data' links to the front-end. These links now appear as buttons beside the 'Cite' button on any page that displays data, as you can see in the following screenshot:



Pressing on the button loads the API endpoint used to return the data found on the page with 'CSV' rather than 'JSON' selected as the file type. This then prompts the file to be downloaded by the browser rather than loading the data into the browser tab. It took a bit of time to add these links to every required page on the site, but I think I've got them all. However, the CSV downloads still needed quite a lot of work doing to them. When formatted as JSON any data

held in nested arrays are properly transformed and usable, but a CSV is a flat file consisting of columns and rows and the data has a more complicated structure than this. For example, if we have one row in the CSV file for each borrowing record on a register page the record may have multiple associated borrowers, each with any number of occupations consisting of multiple fields. The record's book holding may have any number of book items and may be associated with multiple book editions and there may be multiple authors associated with any level of book record (item, holding, edition and work). Representing this structure in a simple two-dimensional spreadsheet is very tricky and requires the data to be 'flattened'. In order to do so a script needs to work out the maximum number of each variable items a record in the returned data has in order to create the required columns (with heading labels) and to pad out any other records that don't have the maximum number of items with empty columns so that the columns of all records line up.

So, for example, when looking at borrowers: If borrowing row number 16 out of 20 has a borrower with five occupations then column headings need to be added for five sets of occupation columns and the data for the remaining 19 rows needs to be padded out with empty data to ensure any columns that appear after occupations continue to line up. As a borrowing may involve multiple borrowers this then becomes even more complicated.

I managed to update the API to ensure nested arrays were flattened for several of the most complicated endpoints, such as a page of records and the search results. The resulting CSV files can become quite monstrously large, with over 200 columns of data a regular occurrence. However, with the data properly structured and labelled it should hopefully make it easier for users who are interested in the data to download the CSV and then delete the columns they are not interested in, resulting in a more manageable file. I still need to complete the 'flattening' of CSV data for a few other endpoints, which I hope to tackle next week.

2nd October 2023

I completed my work on the CSV endpoints for the Books and Borrowing project, ensuring all nested arrays are 'flattened' when producing the two-dimensional CSV file. This has been a lengthy and tedious task, but it's good that it's done, and it should mean that future researchers will be able to extract and reuse the data in a relatively straightforward manner.

9th October 2023

Also this week I did some further work on the Books and Borrowing project, which included generating several different spreadsheets of book holdings that have no associated borrowing records and discussing the options of creating downloadable bundles of all data associated with each specific library.

6th November 2023

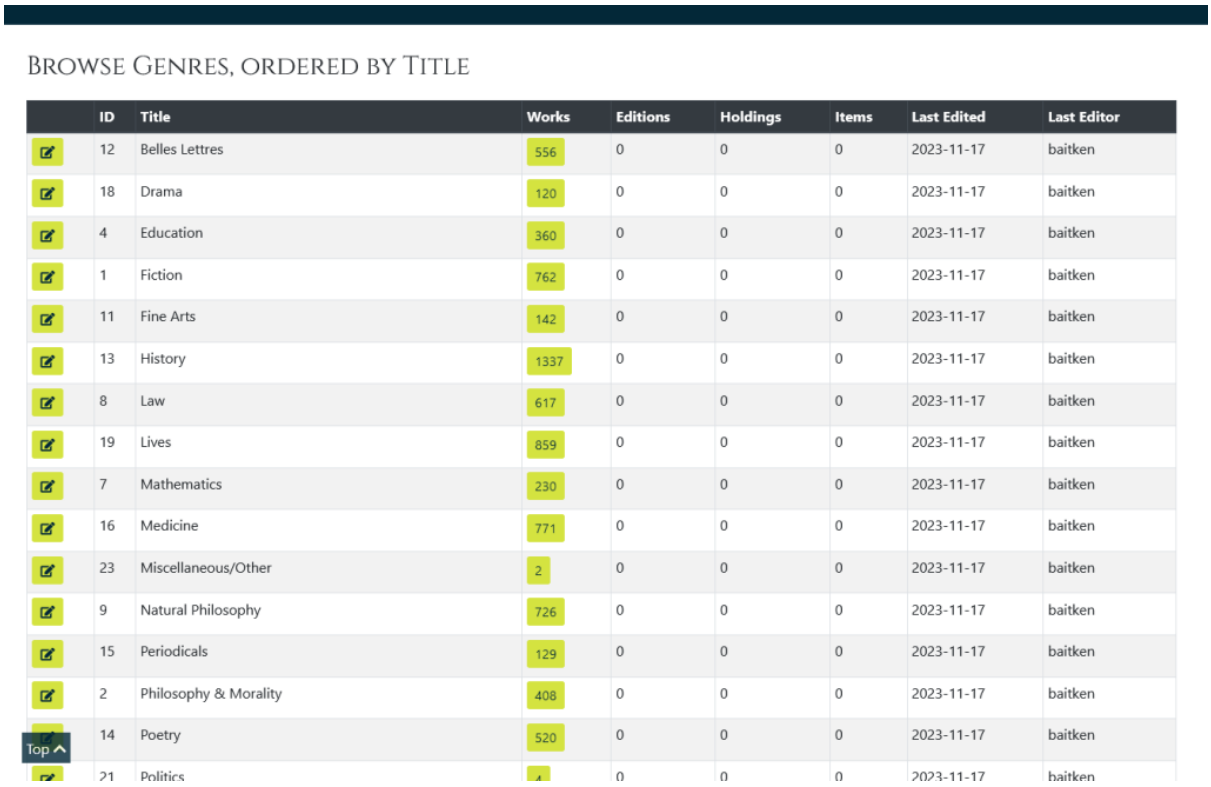
I received the data about Genre from Matt Sangster for the Books and Borrowing project and have arranged to focus on implementing this over the coming weeks. This will involve setting up the data structures to store genre data, writing and testing the scripts to import the data then running the scripts, updating the CMS to enable data to be managed, updating the Solr index and the scripts for generating Solr data for the front-end search to incorporate genre and

updating the front-end to incorporate genre including the display, search and browse of genre. This is going to be a pretty major job.

13th November 2023

I began the major task of integrating book genre with the Books and Borrowing dataset. The team had been working on a spreadsheet that enabled them to assign top-level Book Work records to more than 13,000 Book Edition records and also assign up to three genres to each Work. I had to write a script to parse this data, which involved extracting and storing the distinct genres, creating Book Work records, assigning Book Work authors, adding in associations to Book Edition records, deleting any author associations at Edition level and creating associations between Works and genres. It took the best part of two days to create and test the script, running it on a local version of the data stored on my laptop. After final testing the number of active Book Works increased from 75 to 9808 and the number of active Book Editions that have a Work association grew from 72 to 13,099. The number of genre connections for Works stood at 11,536 and the number of active Book Works that have at least one author association stood at 9,808, up from 70, while the number of active Book Editions with at least one direct author association decreased to 2,191 from 14,384, due to the author association being shifted up to Work (and it will cascade from there).

With the data import sorted I then moved onto updating the project's content management system to incorporate all facilities to add, edit, browse and delete genres. This included creating facilities for associating genres with book records at any level (from Work down to Item) wherever books can be edited in the CMS. The 'Browse Genres' page works in a similar way to 'Browse Authors', giving you a list of genres and a count of the number of each book at each level that has an association, as the following screenshot shows:

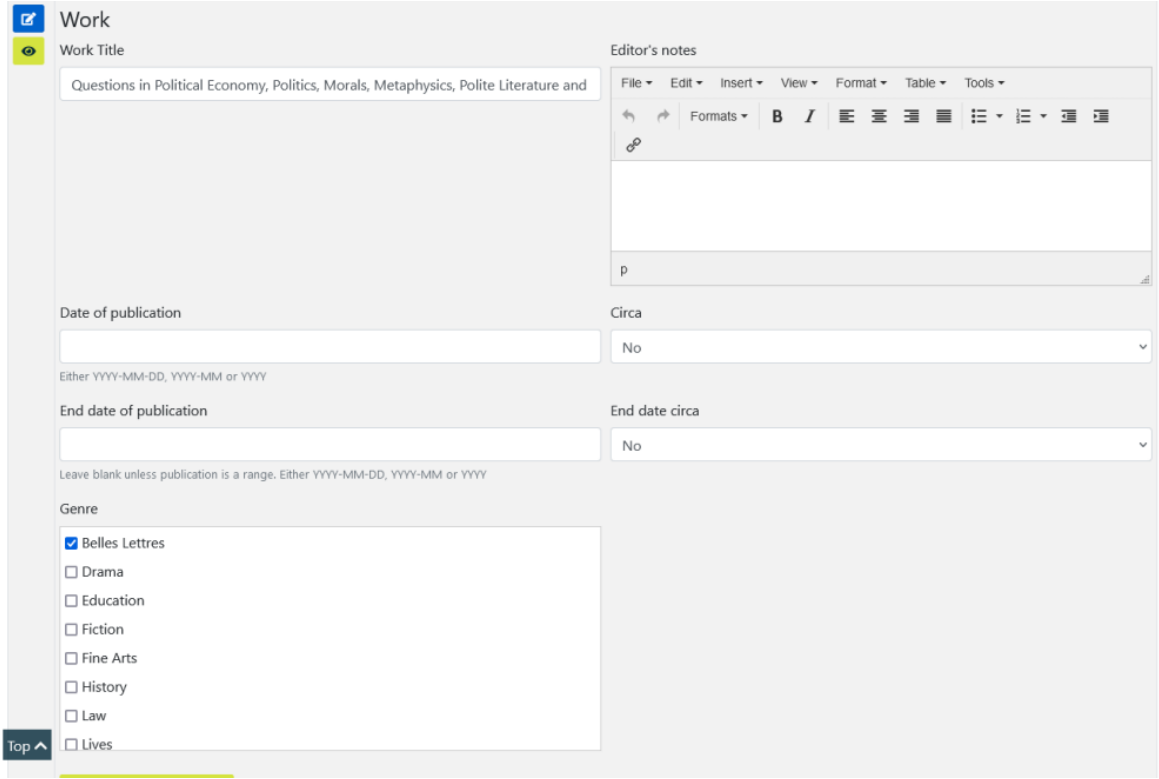


BROWSE GENRES, ORDERED BY TITLE

	ID	Title	Works	Editions	Holdings	Items	Last Edited	Last Editor
	12	Belles Lettres	556	0	0	0	2023-11-17	baitken
	18	Drama	120	0	0	0	2023-11-17	baitken
	4	Education	360	0	0	0	2023-11-17	baitken
	1	Fiction	762	0	0	0	2023-11-17	baitken
	11	Fine Arts	142	0	0	0	2023-11-17	baitken
	13	History	1337	0	0	0	2023-11-17	baitken
	8	Law	617	0	0	0	2023-11-17	baitken
	19	Lives	859	0	0	0	2023-11-17	baitken
	7	Mathematics	230	0	0	0	2023-11-17	baitken
	16	Medicine	771	0	0	0	2023-11-17	baitken
	23	Miscellaneous/Other	2	0	0	0	2023-11-17	baitken
	9	Natural Philosophy	726	0	0	0	2023-11-17	baitken
	15	Periodicals	129	0	0	0	2023-11-17	baitken
	2	Philosophy & Morality	408	0	0	0	2023-11-17	baitken
	14	Poetry	520	0	0	0	2023-11-17	baitken
	21	Politics	4	0	0	0	2023-11-17	baitken

Top

Pressing on a number opens a pop-up containing a list of the associated books and you can connect through to each book record from this. As with authors, genre will cascade down from whichever level of book it is associated with to all lower levels. You only need to make an association at a lower level if it differs from the genre at a higher level. The counts in the 'browse' page show only the direct associations, so for now there are no editions or lower with any numbers listed. Wherever a book at any level can be edited in the CMS a new 'Genre' section has been added to the edit form. This consists of a list of genres with checkboxes beside them, as the following screenshot demonstrates:



You can tick as many checkboxes as are required and when updating the record the changes will be made. I tested out the new genre features in the CMS and all seem to be working well. I also imported all of the genre data so hopefully everything is now in place. Next week I will move onto the front-end, where there is much to do – not only making genre visible wherever books are viewed but updating the search facilities and adding in a number of new visualisations for genre as well. I also fixed a few issues with images of Registers from the Royal High School – a few that were missing I added in and the order of others needed to be updated.

20th November 2023

I spent most of this week working towards adding genre to the Books and Borrowing front-end, working on a version running on my laptop. My initial task was to update the Solr index to add in additional fields for genre. With the new fields added I then had to update my script that generates the data for Solr to incorporate the fields. The Solr index is of borrowing records so as with authors, I needed to extract all genre associations at all book levels (work, edition, holding, item) for each book that was associated with a borrowing record, ensuring lower level associations replaced any higher level associations and removing any duplicates. This is all academic for now as all genre associations are at Work level, but this may not always be the case. It took a few attempts to get the data just right (e.g. after one export I realised it would be

good to have genre IDs in the index as well as their names) and each run-through took about an hour or so to process, but all is looking good now. I'll need to ask Stirling IT to create a new Solr core and ingest the new data on the server at Stirling as this is not something I have the access to do myself, and I'll do this next week. The screenshot below shows one of the records in Solr with the new genre fields present.


```
{
  "bnid":"161595",
  "lid":25,
  "slug":"edinburgh-university",
  "lname":"Edinburgh University Library",
  "rid":58,
  "rname":"Da.2.11",
  "instats":"Y",
  "syear":1775,
  "eyear":1777,
  "rtype":"Student",
  "pid":11003,
  "fnum":"314",
  "transcription":"History of Greece",
  "bday":13,
  "bmonth":1,
  "byear":1777,
  "borrowed":"1777-01-13T00:00:00Z",
  "bdayofweek":"Monday",
  "bhid":21200,
  "standardisedtitle":"History of ancient Greece : from the earliest times, till it became a Roman province | Alletz | Robertson",
  "brids":[13086],
  "bfnames":["John"],
  "bsnames":["Anderson"],
  "bfullnames":["John Anderson"],
  "boccs":["University Student",
    "Education"],
  "boccs-copy":["University Student",
    "Education"],
  "bgenders":["Male"],
  "bfullname-single":"Anderson John, ",
  "bocc-single":"University Student Education ",
  "aids":[3225],
  "afnames":["Pons Augustin"],
  "asnames":["Alletz"],
  "afullnames":["Pons Augustin Alletz"],
  "afullnames-copy":["Pons Augustin Alletz"],
  "abyear":[1705],
  "adyear":[1785],
  "agenders":["Male"],
  "afullname-single":"Alletz Pons Augustin, ",
  "genre":["History"],
  "genre-copy":["History"],
  "gids":[13],
  "genre-single":"History ",
  "beids":[11003],
  "bwids":[5614],
  "bwtitles":["History of Ancient Greece; from the Earliest Times, till it Became a Roman Province [Pons Augustin Alletz]"],
  "edtitles":["The history of ancient Greece; from the earliest times, till it became a Roman province."],
  "edtitle-single":"The history of ancient Greece; from the earliest times, till it became a Roman province., ",
  "estcs":["T114424"],
  "langs":["English"],
  "pubplaces":["Edinburgh"],
  "pubyears":[1768],
  "formats":["12mo"],
  "translators":["Robertson, William, 1740-1803"],
  "_version_":1783365982728749056},
```


With Solr updated I then began updating the front-end, in a version of the site running on my laptop. This required making significant updates to the API that generates all of the data for the front-end by connecting to both Solr and the database as well as updating the actual output to

ensure genre is displayed. I updated the Advanced Search forms (simple and advanced) to add in a list of genres from which you can select any you're interested in (see the following two screenshots) and updated the search facilities to enable the selected genres to be searched, either on their own or in combination with the other search options.

Books and Borrowing 1750-1830

An Analysis of Scottish Borrowers' Registers

Advanced Search

Select a tab below to swap between a simpler and a more complete advanced search. More detailed instructions will go here eventually.

Clear search options

Simple Options | Advanced Options

Top

Libraries and registers

- Select all
- Aberdeen Theological Library
  - MSM 227 (3342 borrowings)
- Advocates Library
  - FR262a-02 (2 borrowings)
  - FR262a-02i (1 borrowings)
  - FR262a-03 (2 borrowings)
  - FR262a-03i (2 borrowings)
  - FR262a-04 (829 borrowings)
  - FR262a-05 (4 borrowings)
  - FR262a-06 (885 borrowings)
  - FR262a-07 (2 borrowings)
  - FR262a-08 (4 borrowings)
  - FR262a-09 (3 borrowings)

Borrowing

Transcription:

Date of borrowing:

Borrower

Title:

- Select all
- Adjudant Major (1 borrowers)
- Baron (2 borrowers)
- Bishop (1 borrowers)
- Captain (42 borrowers)
- Colonel (5 borrowers)
- Dr (115 borrowers)

Forename:

Surname:

Other names:

Gender:

Occupation:

- Select all
- Agricultural (60 borrowers)
  - Farmer (23 borrowers)
  - Gardener (7 borrowers)
  - Land Manager/Factor (7 borrowers)
  - Tenant (23 borrowers)
- Arts and Letters (93 borrowers)
  - Author (65 borrowers)
  - Curator (5 borrowers)
  - Librarian (10 borrowers)
  - Musician (1 borrowers)
  - Painter/Limner (4 borrowers)
  - Poet (8 borrowers)
- Commerce, Trade, and Finance (71 borrowers)

Settlement:

Street:

Author

Forename:

Surname:

Year of birth:

Year of death:

Gender:

Book Work

Text explaining that a book work is the top-level book record

Title:

Book genre:

- Select all
- Belles Lettres
- Drama
- Education
- Fiction
- Fine Arts
- History

Book Edition

Text explaining that a book edition is a specific edition of a book work that can be found in multiple libraries

Title:

ESTC:

Language:

- Select all
- Anglo-Saxon (3 editions)
- Arabic (16 editions)
- Aramaic (3 editions)
- Chaldean (2 editions)
- Chinese (1 editions)
- Dutch (2 editions)

Editors:

Place of publication:

Year of publication:

Format:

- Select all
- 12mo (2389 editions)
- 12mo (duodecimo) (997 editions)
- 16mo (33 editions)
- 1818 (1 editions)
- 18mo (26 editions)
- 24mo (11 editions)

Translators:

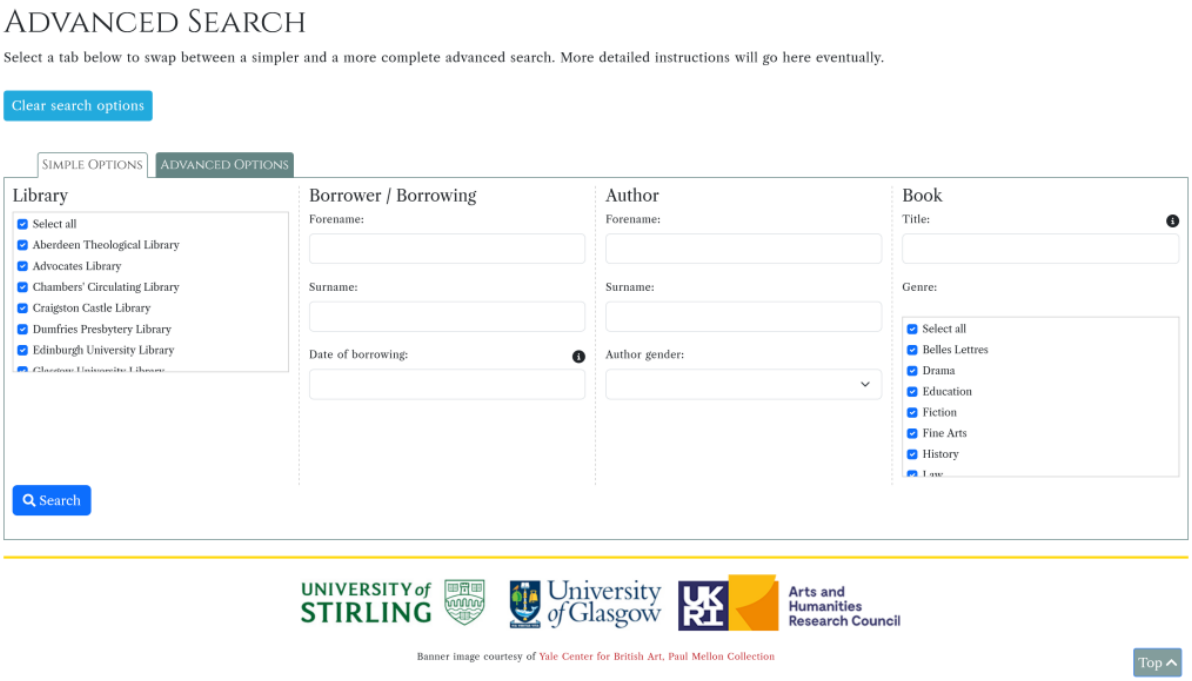
Book Holding

Text explaining that a book holding is a specific copy of a book edition held at a specific library

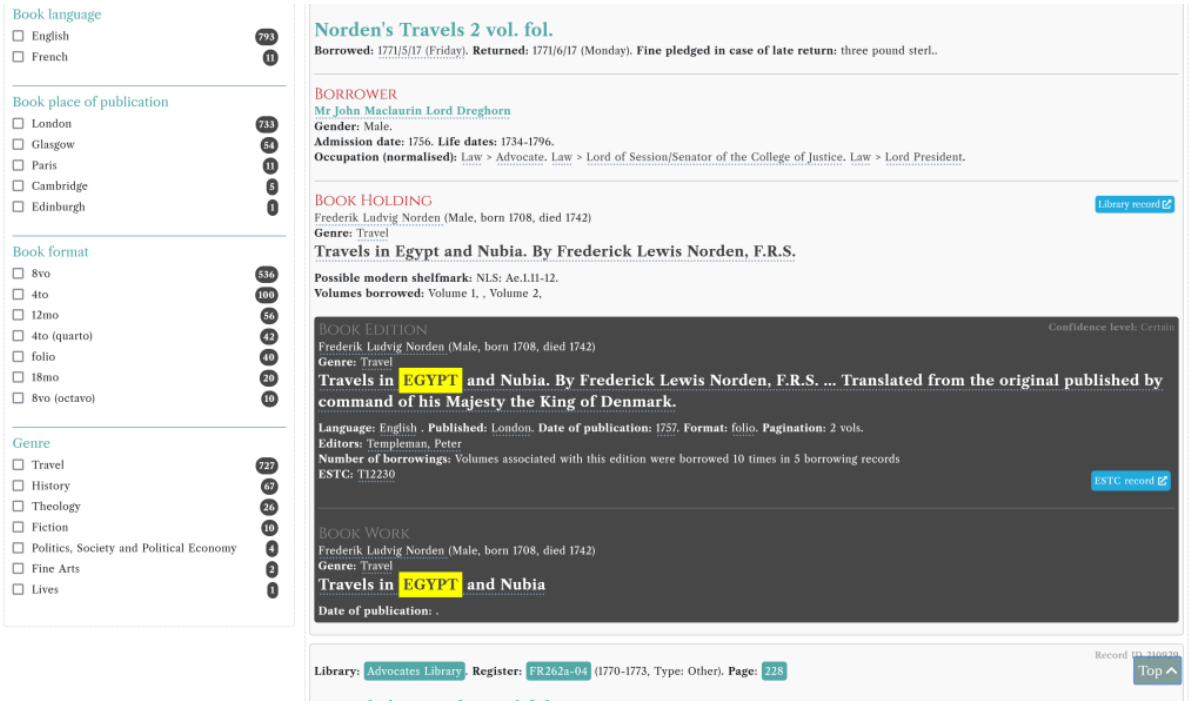
Title:

Search

University of Stirling | University of Glasgow | Arts and Humanities Research Council



On the search results page any genres associated with a matching record are displayed, with associations at higher book levels cascading down to lower book levels (unless the lower book level has its own genre records). Genres appear in the records as clickable items, allowing you to perform a search for a genre you're interested in by clicking on it. I've also added in genre as a filter option down the left of the results page. Any genres present in the results are listed, together with a count of the number of associated records, and you can filter the results by pressing on a genre, as you can see in the following screenshot, which shows the results of a quick search for 'Egypt', displaying the genre filter options and showing the appearance of genre in the records.



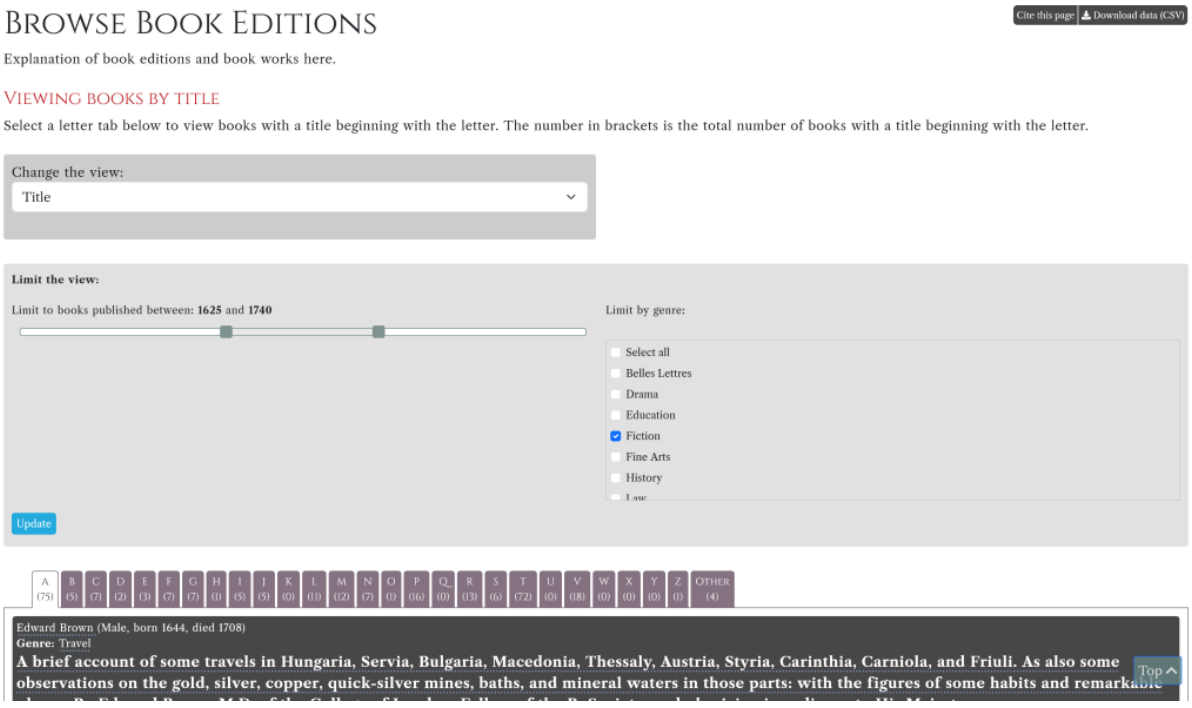
Genre is displayed in a similar way wherever book records appear elsewhere in the site, for example the lists of books for a library, the top-level 'book editions' page and when viewing a specific page in a library register.

There is still more to be done with genre, which I'll continue with next week. This includes adding in new visualisations for genre, adding in new 'facts and figures' relating to genre and adding in facilities to limit the 'browse books' pages to specific genres.

27th November 2023

I completed work on the integration of genre into the Books and Borrowing systems this week. It took a considerable portion of the week to finalise the updates but it's really great to have it done, as it's the last major update to the project.

My first task was to add genre selection to the top-level 'Browse Editions' page, which I'm sure will be very useful. As you can see in the following screenshot, genres now appear as checkboxes as with the search form, allowing users to select one or more they're interested in. This can be done in combination with publication date too. The screenshot shows the book editions that are either 'Fiction' or 'Travel' that were published between 1625 and 1740. The selection is remembered when the user changes to a different view (i.e. authors or 'top 100') and when they select a different letter from the tabs.



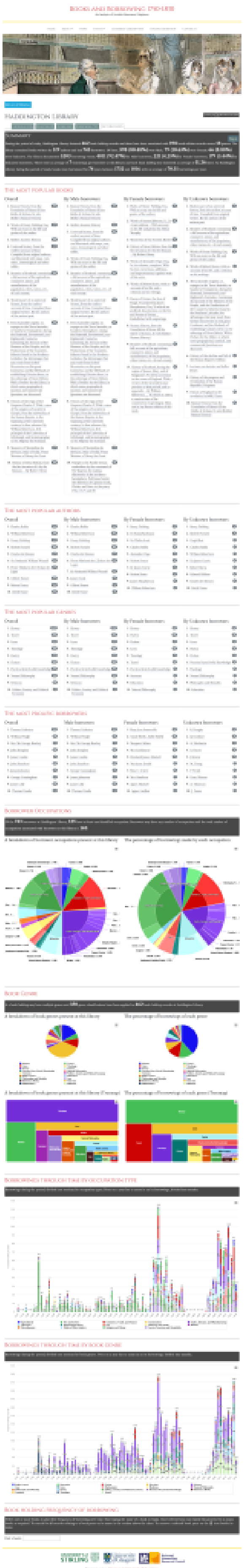
It proved to be pretty tricky and time-consuming to implement. I realised that not only did the data that is displayed need to be updated to reflect the genre selection, but the counts in the letter tabs needed to be updated too. This may not seem like a big thing, but the queries behind it took a great deal of thought. I also realised whilst working on the book counts that the counts in the author tabs were wrong – they were only counting direct author associations at edition level rather than taking higher level associations from works into consideration. Thankfully this

was not affecting the actual data that was displayed, just the counts in the tabs. I've sorted this too now, which also took some time.

With this in place I then added a similar option to the in-library 'Book' page. This works in the same way as the top-level 'Editions' page, allowing you to select one or more genres to limit the list of books that are displayed, for example only books in the genres of 'Belles Lettres' and 'Fiction' at Chambers, ordered by title or the most popular 'Travel' books at Chambers. This did unfortunately take some time to implement as Book Holdings are not exactly the same as Editions in terms of their structure and connections so even though I could use much of the same code that I'd written for Editions many changes needed to be made.

The new Solr core was also created and populated at Stirling this week, after which I was able to migrate my development code from my laptop to the project server, which meant I could share my work with others, which was good.

I then moved onto adding genre to the in-library 'facts' page and the top-level 'facts' page. Below is a very long screenshot of the entire 'facts' page for Haddington library and I'll discuss the new additions below:


Books and Borrowing 1750-1830
Haddington Library
Summary
The most popular books
Overall
By Male borrowers
By Female borrowers
By Unknown borrowers
The most popular authors
The most popular genres
The most prolific borrowers
Borrower occupations
Book Genre
Borrowings through time by occupation type
Borrowings through time by book genre

The number of genres found at the library is now mentioned in the 'Summary' section and there is now a 'Most popular genres' section, which is split by gender as with the other lists. I also added in pie charts showing book genres represented at the library and the percentage of borrowings of each genre. Unfortunately these can get a bit cluttered due to there being up to 20-odd genres present, so I've added in a legend showing which colour is which genre. You can hover over a slice to view the genre title and name and you can click on a slice to perform a search for borrowing records featuring a book of the genre in the library. Despite being a bit cluttered I think the pies can be useful, especially when comparing the two charts – for example at Haddington 'Theology' books make up more than 36% of the library but only 8% of the borrowings.

Due to the somewhat cluttered nature of the pie charts I also experimented with a treemap view of Genre. I had stated we would include such a view in the requirements document, but at that time I had thought genre would be hierarchical, and a treemap would display the top-level genres and the division of lower level genres within these. Whilst developing the genre features I realised that without this hierarchy the treemap would merely replicate the pie chart and wouldn't be worth including.

However, when the pie charts turned out to be so cluttered I decided to experiment with treemaps as an alternative. The results currently appear after the pie charts in the page. I initially liked how they looked – the big blocks look vaguely 'bookish' and having the labels in the blocks makes it easier to see what's what. However, there are downsides. Firstly, it can be rather difficult to tell which genre is the biggest, due to the blocks having different dimensions – does a tall, thin block have a larger area than a shorter, fatter block, for example. It's also much more difficult to compare two treemaps as the position of the genres changes depending on their relative size. Thankfully the colour stays the same, but it takes longer than it should to ascertain where a genre has moved to in the other treemap and how its size compares. I met with the team on Friday to discuss the new additions and we agreed that we could keep the treemaps, but that I'd add them to a separate tab, with only the pie charts visible by default.

I then added in the 'borrowings over time by genre' visualisation to the in-library and top level 'facts' pages. As you can see from the above screenshot, these divide the borrowings in a stacked bar chart per year (other month if a year is clicked on) into genre, much in the same way as the preceding 'occupations' chart. Note however that the total numbers for each year are not the same as for the occupations through time visualisation as books may have multiple genres and borrowers may have multiple occupations and the counts reflect the number of times a genre / occupation is associated with a borrowing record each year (or month if you drill down into a year). We might need to explain this somewhere.

We met on Friday to discuss the outstanding tasks. We'll probably go live with the resource in January, but I will try to get as many of my outstanding tasks completed before Christmas as possible.

4th December 2023

I dealt with a few data issues for the Books and Borrowing project, including generating spreadsheets of data for checking (e.g. list of all of the distinct borrower titles) and then making updates to the online database after these spreadsheets had been checked. I also fixed a bug with the genre search, which was joining multiple genre selections with Boolean AND when it should have been joining them with Boolean OR.

11th December 2023

I spent most of the remainder of the week working on the Books and Borrowing project. I updated the languages assigned to a list of book editions that had been given to me in a spreadsheet and added a few extra pages and page images to one of the registers. I then returned to my 'to do' list for the project and worked through some of the outstanding items. I moved the treemaps on the library and site-wide 'facts' pages to separate tabs and I went through the code to ensure that the data for all visualisations only uses borrowing records set to 'in stats'. This wasn't done before so many of the visualisations and data summaries will have changed slightly. I also removed the non-male/female 'top ten' lists in the library facts page, as requested.

I then moved on to creating a cache for the facts page data, which took about a day to implement. I firstly generated static data for each library and stored this as JSON in the database. This is then used for the library facts page rather than processing the data each time. However, the site-wide facts page lets the user select any combination of libraries (or select all libraries) and the 'top ten' lists therefore have to dynamically reflect the chosen libraries. This meant updating the API to pull in the 'facts' JSON files for each selected library and then analyse them in order to generate new 'top tens' for the chosen libraries. For example, working out the top ten genres for all selected libraries meant going through the individual top ten genre lists for each library, working out the total number of borrowings for each genre and then reordering things after this merging of data was complete.

Despite still requiring this processing the new method of using the cached data is considerably faster than querying and generating the data afresh each time the user requests it. Previously displaying the site-wide 'facts' page for all libraries was taking up to a minute to complete whereas now it takes just a few seconds. I also made a start on updating the site text that Katie had sent me earlier in the week. A large number of tweaks and changes are required and this is likely to take quite a long time, but I hope to have it finished next week.

18th December 2023

The Books and Borrowing project officially comes to an end on the 31st of December, although we're not going to launch the front-end until sometime in January. I still had rather a lot of things to do for the project and therefore spent the entirely of my four days this week working for this project. Of course, the other team members were also frantically trying to get things finished off, which often led to them spotting something they needed me to sort out, so I found myself even busier than I was expecting. However, by Thursday I had managed to complete all of the tasks I'd hoped to finish, plus many more that were sent my way as the week progressed.

At the end of last week I'd begun updating the site text that the project PI Katie had supplied me with, and I completed this task, finally banishing all of the placeholder text. This also involved much discussion about the genre visualisations and what they actually represent, which we thankfully reached agreement on. I also added in some further images of a register at Leighton library and wrote a script to batch update changes to the publication places, dates of publication and formats of many book edition records. One of the researchers also spotted that the 'next' and 'previous' links for the two Selkirk registers were not working, due to an earlier amalgamation of page records into 'double spread' records. I therefore wrote another script to sort these out.

I then added new 'Top ten book work' lists to the site-wide 'Facts' page (overall and by the gender of borrowers). This required me to update the script that generated the cache that I developed last week, to rerun the script to generate fresh data, to update the API to ensure that works were incorporated into the output and to update the front-end to add in the data. Hopefully the information will be of interest to people.

I then overhauled the highlighting of search terms in the search results. This was previously only working with the quick search, and only when no wildcards were used in the search. Instead I used a nice JavaScript library called mark.js (https://markjs.io/) that I'd previously used for the DSL website to add in highlighting on the client-side. Now any the values in any search fields that are searched for will be highlighted in the record, including when wildcards are used. I also updated the highlight style to make it a bit less harsh.

It should be noted that highlighting is still a bit of a blunt tool – any search terms will be highlighted throughout the entire record where the term is found. So if you search for the occupation 'farmer' then wherever 'farmer' is found in the record it will be highlighted, not just in the normalised occupation list. Similarly, if you search for 'born' then the 'born' text in the author information will be highlighted. It's not feasible to make the highlighting more nuanced in the time we have left, but despite this I think that on the whole the highlighting is useful.

I reckoned that the highlighting could end up being a bit distracting so I added in an option to turn results highlighting on or off. I added a button to process this to the search results page, as part of the buttons that include the 'Cite' and 'Download' options. The user's choice is remembered by the site, so if you turn highlighting off and then navigate through the pages of results or perform a filter the highlights stay off. They will stay off until you turn them on again, even if you return to the site after closing your browser.

One of the researchers noticed that an unnecessary near-duplicate genre had somehow been introduced into the system ('Fine Art' instead of 'Fine Arts') so I removed the and reassigned any records that were assigned to the erroneous version. The PI Katie also spotted some odd behaviour with the search form boxes. When using the browser's 'back' button search data

was being added to the wrong search boxes. This took quite some time to investigate and I couldn't replicate the issue in Firefox (the browser I use by default), but when using a Chrome-based browser (MS Edge) I experience the issue. It turns out it's nothing to do with my code but a bug in Chrome (see https://github.com/vuejs/vue/issues/11165). The fix mentioned on this page was to add 'autocomplete="off"' to the form and this seems to have sorted the problem. It's crazy that this issue with Chrome hasn't been fixed as the posts on the page identifying the issue started in 2020.

Katie also spotted another issue when using Chrome. Applying multiple filters to the search results wasn't working in Chrome, even though it worked fine in Firefox. This time it was caused by Chrome encoding the bar character to %7C while Firefox keeps it as '|'. My filter script was splitting up filters on the actual bar character and as this wasn't present in Chrome multiple filters were not working (even though they were working fine in Firefox). Thankfully once identified this was relatively easy to fix.

I also managed to implement a 'compact' view of borrowing records this week, something that had been on my 'to do' list for a while. Borrowing records can be extremely verbose and rather overwhelming so we decided to give the option to view compact versions of the records that contain a narrower set of fields. I added a compact / full record view switcher to the bar of options in the top right of the search results and library register page, beside the 'Cite' option. As with the highlighting feature I previously discussed, the choice is remembered in your browser, even if you return to the site in a later session (so long as you're using the same device and browser, of course).

For the compact view I decided to retain the links to the library, register and page as I figured it would be useful to be able to see these. Also included are the borrowed and returned dates, the borrowers (names only), the title of the Book Work (or Works) if the record has such an association and the title of the Holding if not, any associated authors and genres, plus a list of the volumes borrowed (if applicable). The following screenshot shows what the compact looks like:

My final tasks of the week were to add in a cookie banner for the site and install Google Analytics. In the New Year I'll need to regenerate the Solr index and then integrate the development site with the live site. This will include making updates to paths throughout the code, ensuring the existing Chambers Maps continues to function, adding links to the pages of the development site to the site menu and adding the quick search option to the site header. It will be great once the site is fully accessible.

8th January 2024

I spent a bit of time this week on the Books and Borrowing project, including participating in a project team Zoom call on Monday. I had thought that we'd be ready for a final cache generation and the launch of the full website this week, but the team are still making final tweaks to the data and this had therefore been pushed back to Wednesday next week. But this week I updated the 'genre through time' visualisation as it turned out that the query that returned the number of borrowing records per genre per year wasn't quite right and this was giving somewhat inflated figures, which I managed to resolve. I also created records for the first volume of the Leighton Library Minute Books. There will be three such volumes in total, all of which will feature digitised images only (no transcriptions). I processed the images and generated page records for the first volume and will tackle the other two once the images are ready.

15th January 2024

Also this week I did some further work for the Books and Borrowing project. I fixed an image that wasn't working in one of the Royal High School registers and reassigned images to one of the Orkney registers due to a duplicate image throwing everything from that point onwards off by one. I also updated the website interface to ensure that tables work better on narrow screens. I applied a minimum width of 800px to tables (e.g. the tabular list of libraries and the lists or registers), meaning on very narrow screens the tables no longer get squashed into

spaces that make them entirely unusable but instead extend beyond the width of the screen and you need to scroll to view their full contents. It seems to work ok on my rather ancient phone.

On Wednesday we were finally ready to generate the final pre-publication cache files, so I completed this task – updating all of the caches and generating a new Solr index file. Once this had been set up for us by Stirling's IT people I then updated the connection details in the API. Also in this update I fixed an issue with the place of publication in the advanced search. Previously if you typed in 'london' (lower case) and didn't select 'London' form the drop-down list and then pressed the 'search' button you found no results as publication place was case sensitive. The correct results are now returned. I wasn't able to 'go live' with the new site this week, however, due to other commitments. I'll start on this first thing next week.

22nd January 2024

I'd hoped that we'd be able to go live with the full Books and Borrowing site this week, but unfortunately there are still some final tweaks to be made so we're not live yet. I did manage to get everything ready to go, but then some final checking by the team uncovered some additional issues that needed sorting.

Migrating all of the pages in the development site to their final URLs proved to be quite challenging as it was not simply a matter of copying the pages. Instead I needed to amalgamate three different sets of scripts and stylesheets (the WordPress site, the dev site and the Chambers map), which involved some rewriting and quite a lot of checking. However, by the end of Monday I had completed the process and I sent the URLs to the team for a final round of checking.

The team uncovered a few issues that I managed to address, including some problems with the encoding of ampersands and books and borrowers without associated dates not getting returned in the site-wide browse options. I needed to regenerate the cache files after sorting some of these, which also took a bit of time.

I also realised that the default view of the 'Browse books' page was taking far too long to load, with load times of more than 20 seconds. I therefore decided to create a cache for the standard view of books, with separate cache files for books by first letter of their title and their author. These caches would then be used whenever such lists were requested rather than querying everything each time. When filters are applied (e.g. date ranges, genres) the cache would be ignored, but such filtered views generally bring back less books and are returned quicker anyway. It took a while to write a script to generate the cache, to update the API to use the cache and then add the cache generation script to the documentation, but once all was in place the page was much quicker to load. It does still take a few seconds to load in as there are almost 3000 books beginning with the letter 'A' in our system and even processing a static file containing this much data takes time. But it's a marked improvement.

On Friday afternoon the project PI Katie Halsey got back to me with a list of further updates that I need to make before the launch, one of which will require the Solr index structure to be

updated and all of the Solr data to be regenerated, which I'll tackle next week. Hopefully after that we'll finally be able to launch the site.

29th January 2024

I spent a lot of time this week working on the Books and Borrowing project, making final preparations for the launch of the full website. By the end of the week the website was mostly publicly available (see https://borrowing.stir.ac.uk) but it wasn't as smooth a process as I was hoping for and there are still things I need to finish next week.

My first task of the week was to write a script that identified borrowers who have at least one active borrowing record but of these zero are records on pages that are active in registers that are active. This should then bring back a list of borrowers who are only associated with inactive registers. It took quite some time to get my head around this after several earlier attempts didn't do exactly what was requested and my final script identified 128 borrowers that were then deactivated, all from St Andrews.

I then moved on to an issue that had been noticed with a search for author names. A search for 'Keats' was bringing back matches for 'Keating', which was clearly not very helpful. The cause of this was Solr trying to be too helpful. The author name fields were stored at 'text_en' and this field type has stemming applied to it, whereby stems of words are identified (e.g. 'Keat') and a search for a stem plus a known suffix (e.g. 's', 'ing') will bring back other forms of the same stem. For names this is hopeless as 'Keating' is in no way connected to 'Keats'.

It turned out that this issue was affecting many other fields as well, such as book titles. A search for 'excellencies' was finding book titles containing the forms 'excellency' and also 'excellence', which again was pretty unhelpful. I did some investigation into stemming and whether a Solr query could be set to ignore it, but this did not seem to be possible. For a while I thought I've had to change all of the fields to strings, which would have been awful as strings in Solr are case sensitive and do not get split into tokens, meaning wildcards would need to be used and the search scripts I'd created would need to be rewritten.

Thankfully I discovered that if I stored the text in the field type 'text_general' then stemming would be ignored but the text would still be split into tokens. I created a test Solr index on my laptop with all of the 'text_en' fields set to 'text_general' and searching this index for author surname 'Keats' only brought back 'Keats' and book title for 'excellencies' only brought back 'excellencies'. This is exactly what we wanted and with the change in place I was able to fully regenerate the cached data and the JSON files for import into Solr (a process that takes a couple of hours to complete) and ask for the online Solr index to be updated.

I also updated the help text on the website, adding in some new tooltips about editors and translators to the advanced search form. With no other issues reported by the team I then began the process of making the site live, including adding a 'quick search' bar to the header of

all pages, adding in menu items to access the data and adding the 'on this day' feature to the homepage.

However, I noticed one fairly sizeable issue, unfortunately. The stats on the 'facts' page do not correspond to the results you get when you click through to the search results. I therefore removed the top-level 'Facts & figures' menu item until I can figure out what's going on here. I first noticed an issue with the top ten genre lists. The number of borrowings for the most popular genre (history) were crazy. Overall borrowings of 'history' is listed as '104849', which almost as many as the total number of borrowing records than we have in the system. Also the link to view borrowings in the search was limiting the search to Haddington, which is clearly wrong. Other numbers in the page just weren't matching up to the search results either. For example for Chambers the most prolific borrower is Mrs Thomas Hutchings with 274 listed borrowings, but following the link to the search results gives 282.

It took several hours of going through the code to ascertain what was going on with the top ten genre issue and it was all down to a missing equals sign in an 'if' statement. Where there should have been two (==) there was only one, and this was stopping the following genre code from processing successfully.

There is unfortunately still the issue of the figures on the facts pages not exactly matching up with the number of results returned by the search. I had noticed this before but I'd hoped it was caused by the database still being updated by the teasm and being slightly out of sync with the Solr index. Alas, it's looking like this is not the case. There must instead be discrepancies in the queries used to generate the Solr index and those used the generate the facts data. I fear that it might be the case that in the chain of related data some checks for 'isactive' have been omitted. Each register, page, borrowing record, borrower, author, and book at each level has its own 'isactive' flag. If (for example) a book item is set to inactive but a query fails to check the book item 'isactive' flag then any associated borrowing records will still be returned (unless they have each been set to inactive). I'm going to have to check every query to ensure the flags are always present. And it's even more complicated than that because queries don't necessarily always include the same data types. E.g. a borrower is related to a book via a borrowing record and if you're only interested in borrowers and books the query doesn't need to include the associated page or register. But of course if the page or register where the borrowing record is located is inactive then this does become important. I might actually overhaul the 'facts' so they are generated directly from the Solr index. This would mean things should remain consistent, even when updates are made to the data in the CMS (these would not be reflected until the Solr index is regenerated). Something I'll need to work on next week.

5th February 2024

I continued to make updates to the Books and Borrowing website this week after the soft launch last week. I had noticed last week that the figures appearing on the 'Facts' page didn't correlate with the number of results returned through the search facilities. I reckoned this was because the 'Facts' page, which queries the database directly was not necessarily including the full chain of interrelated tables and their individual 'isactive' flags when returning figures whereas

the search facilities use the Solr index, and the data stored within this were generated using the full chain. So for example, counts of borrowing records for books may not incorporate the register page record, but if the register page is set to 'inactive' then it is important to factor in the register page.

Another issue was the 'in stats' flag. We have this flag for borrowing records to decide whether the record should appear in the stats or not, so for example a duplicate record could be omitted from the stats, but would still be findable when viewing the register page. The search results were finding records with 'instats' set to 'no' but these were not included in the 'Facts' page, meaning the numbers would never match up.

To try and make the figures more consistent I decided to update the 'Facts' page to use the Solr index for calculating borrowing records. This slightly changed the figures that appear in the summary section on this page, but as a borrowing record can involve multiple borrowers the total borrowings broken down by gender may not equal the total borrowings. If a borrowing has two borrowers, one male and one female then it will count as one borrowing in the total borrowings and one each in the totals per gender.

Further investigation into the discrepancies between figures and on the 'Facts' page and the number of records returned in a related search did seem to suggest that these are caused by the 'instats' flag. For example, for Chambers library, Fiction is the most popular genre with 4417 borrowings listed. This is the figure for borrowings with 'instats' set to 'Y'. But following the link through to the search results listing borrowing records involving the genre 'Fiction' at Chambers displays 4420 borrowings. This is because the search does not take into consideration the 'instats' field. I tried running the queries (and others) through Solr and this does appear to be the reason for the discrepancies. After discussion with the team we decided therefore to update the search page to avoid returning any 'instats' results. These records will still be findable when viewing register pages, but will not now appear in any search results. With this update in place the figures on the 'Facts' page began to match up with the search results.

However, after further testing I noticed that the stats for authors and genres in the site-wide facts page were still not aligned with the number of search results returned and I spent most of Wednesday investigating this. It was a very long and complex process but I finally managed to sort it. I updated the 'top ten' cache files for each library to use data from the Solr index to calculate the number of borrowings, but this alone wasn't giving the correct figures. The reason was that each library had its own 'top ten' for authors and genres, both overall and for each gender. Then when the facts page was being generated for multiple libraries these lists were then brought together for each library to formulate overall top tens. If an author or genre was not in the top ten for a library the data for this item was not included in the calculations as only the top ten were being stored. For example, if 'Fiction' was the eleventh most borrowed genre at a library then its borrowings were not found and were therefore not getting added to the total. What I've had to do instead is store all genres and authors for each library in the cache rather

than just the top tens, thus ensuring that all items are joined and their number of borrowings are compared. Unfortunately this does mean the cache files are now a lot bigger and more processing needs to be done. But at least the figures do now match up. Also this week I updated the Chambers Library Map to add a link to view the relevant borrowing records to the borrower popups (see https://borrowing.stir.ac.uk/chambers-library-map).

12th February 2024

I created an entry for the Books and Borrowing project on this site (see https://digital-humanities.glasgow.ac.uk/project/?id=160). On Friday afternoon I also investigated a couple of issues with the search that Matt Sangster had spotted. He noticed that an author surname search for 'Byron' wasn't finding Lord Byron, and entering 'Lord Byron' into the surname search was bringing back lots of results that didn't have this text in the author surname.

It turned out that Byron hadn't been entered into the system correctly and was in as forename 'George', surname 'Gordon' with 'Lord Byron' as 'othername'. I'll need to regenerate the data once this error has been fixed. But the second issue, whereby an author surname search for 'Lord Byron' was returning lots of records is a strange one. This would appear to be an issue with searches for multiple words and unfortunately it's something that will need a major reworking. I hadn't noticed previously, but if you search for multiple words without surrounding them by quotes Solr searches the first word against the field and the remaining words against all fields. So "surname 'Lord' OR any field 'Byron'". Whereas what the query should be doing is "surname 'Lord' AND surname 'Byron'". This is something that will probably affect all free-text fields. I'm going to have to update the search to ensure multi-word searches without quotes are processed correctly, which will take some time and I'll try to tackle next week. I also need to create a 'copy' field for place of publication as this is being tokenised in the search facet options. So much for thinking my work on this project was at an end!

19th February 2024

I addressed a couple of issues with the Books and Borrowing search that had been identified last Friday. Multi-word searches were not working as intended and were returning far too many results. The reason being (as mentioned last week) a search for 'Lord Byron' (without quotes) was searching the specified field for 'Lord' and then all fields for 'Byron'. It was rather tricky to think through this issue as multi-word searches surrounded by quotes need to be treated differently, as do multi-word searches that contain a Boolean. We don't actually mention Booleans in the search help, but AND, OR and NOT (which must be upper-case) can be used in the search fields.

I wrote a new function that hopefully sorts out the search strings as required, but note that search strings containing multiple sets of quotes are not supported as this would be much more complicated to sort out and it seemed like a bit of an edge case. This new function has been applied to all free-text search fields other than the quick search, which is set to search all fields anyway. After running several tests I made the update live, and now searching author surnames for 'Lord Byron' finds no results, which is as it should be.

Here are some examples that do return content.

1. If you search book titles for ‘rome’ you currently find 2046 records:

https://borrowing.stir.ac.uk/search/0/0/0/simple/bookname|rome

2. If you search book titles for ‘rome popes’ you currently find 100 records (as this is the equivalent of searching book titles for ‘rome’ AND ‘popes’:

https://borrowing.stir.ac.uk/search/0/0/0/simple/bookname|rome%20popes

3. Using Boolean ‘AND’ gives the same results:

https://borrowing.stir.ac.uk/search/0/0/0/simple/bookname|rome%20AND%20popes

4. A search for ‘rome OR popes’ currently returns 2046 records, presumably because all book titles containing ‘popes’ also contain ‘rome’ (at least I hope that’s the case):

https://borrowing.stir.ac.uk/search/0/0/0/simple/bookname|rome%20OR%20popes

5. A search for ‘rome NOT popes’ currently brings back 1946 records:

https://borrowing.stir.ac.uk/search/0/0/0/simple/bookname|rome%20NOT%20popes

6. And searches for a full string also work as intended, for example a search for “see of rome”:

https://borrowing.stir.ac.uk/search/0/0/0/simple/bookname|%22see%20of%20rome%22

With this update in place I then slightly changed the structure of the Solr index to add a new ‘copy’ field that stores publication place as a string, rather than text. This is then used in the facts, ensuring the full text of the place is displayed rather than being split into tokens. I then regenerated the cache and asked the helpful IT people in Stirling to update this on the project’s server. Once the update had been made everything then worked as it should.

11th March 2024

I added some statements to a couple of pages of the Books and Borrowing website referencing the project’s API and giving some information about it, and spoke to B&B project PI Katie Halsey about creating a preservation dataset for the project and depositing it with a research repository.

8th April 2024

I had discussions with the Books and Borrowing people about the official launch of the resource that’s taking place in a couple of weeks. I’m going to be speaking at the launch so I needed to figure out what I should be talking about. I also returned to the ‘Browse book editions’ page on the website (https://borrowing.stir.ac.uk/books/) which was at this point taking a long time to load. This is because the page defaults to displaying all book editions in the system that have a title beginning with ‘A’- almost 3000 books. I did consider adding pagination to the facility, but I personally find it easier to scroll through a long page rather than flicking between many smaller pages, plus it means a user can use ‘Find’ in their browser to search the listing. Another option I considered was to limit the default display to a particular genre of book rather than all genres, but I decided that this might confuse people if they don’t notice the limit has been applied. Instead I set the page to not load a specific letter tab by default. The tabs load, but to view the content of one of them the user actually has to select one. This means the page now loads

instantaneously and people get to choose what options they want to view without having a long wait.