BRIEF

TABLE OF CONTENTS

INTRODUCTION: PURPOSE AND OBJECTIVES

This project aims to improve the previous version of Search Aberdeen Registers SAR, which was created in 2017. We will recreate the search tool using PHP, aligning with current practice and aiding future support and maintenance. The front end of the webpage will be developed using JavaScript and CSS. GitHub will be the source code repository and version control, and PHPUnit will be the testing framework. The target audience is academic researchers and students with a specialist interest in the content. A secondary audience includes those who are curious about the records and transcripts.

TARGET AUDIENCE AND INTENDED USAGE

The key demographic of the webpage and search tool will consist of users aged 18 and above with a range of technological skills and experiences. The primary audience includes academic researchers and students with specific interests in the subject material. The secondary audience includes non-specialists who have a general interest in records and transcripts. To ensure that the webpage and search tool are accessible to all, steps will be taken to create a usable and accessible user interface with clear directions and instructions.

The webpage and search tool will be used to search and navigate the Aberdeen registers, a valuable historical resource. The webpage will access the search tool and information about the tool and registers. It will also include external links to the University of Aberdeen registers page and the Aberdeen registers blog page.

FUNCTIONAL REQUIREMENTS

Most importantly, our version of the search tool must retain the existing functionality of Search Aberdeen Registers. What this involves, is creating both basic and advanced search functions, and giving the user the ability to browse through each volume, page and entry.

The application should process the XML files as per TEI standards, and the user should be able to search through these files using XQuery.

In the advanced search, the user should be able to search for a word, expression or phrase and filter the results in various ways (i.e., page, date, content, language). They should also be able to search at the beginning, middle and end of words for their desired string of characters, and check for spelling variants. The search results should display a list of entries that include the desired expression, with the matches highlighted in the entry. Additionally, the search results should have the option of sorting by volume & page, frequency and chronology, as well as the option of choosing the number of results per page. The document ID, volume, page and date should be shown for every entry.

While browsing or searching, if the user selects a page, the application should display a picture of the page, as well as transcriptions of all the entries on that page, noting the ID, date and language. Hovering over the picture should zoom in on where the cursor is. There should be the option to switch to an XML view of the transcription. Lastly, selecting entries should allow for download of a pdf of those specific transcription(s).

Lastly, the current edition of the tool does not allow for highlighting and copying of the text of a search result, so this will be added to the new version, as well as any additional fuctions as the client wishes (i.e., new searches, quality of life improvements)

NONFUNCTIONAL REQUIREMENTS

Capacity & scalability

The database contains 5MB of transcribed XML files. It is considered a medium sized database and not expected to increase. However, if the search tool has an increase in the number of users, the website should still be able to handle the influx of activity, especially by following good programming practices that optimize efficiency in every aspect.

The method to access and manage the pictures of the original documents is currently unknown, so we still have to discuss this matter with the client.

Compatibility

Website will be designed for desktop and mobile versions, and should be able to run on any operating system that has an internet browser. Internet Explorer, Chrome, and Firefox may work the best in terms of highlighting search terms within search results.

Security, Reliability & Availability

All pages of the website should be available all the time. 24 hours a day, 7 days a week. In the event of critical failure, the user should be redirected to an appropriate error page. There should also always be a convenient way of contacting the university development team in case part(s) of the website are not working correctly.

The web application will comply with the latest OWASP Top 10 security awareness document. This will keep packages up to date and prevent injections to the database from unauthorized users. These security measures will also include but are not limited to using stored procedures, parameterisation, validating user inputs, and HTTPS.

To check that security measures are consistent and reliable in practice and not just in theory, we will constantly run numerous tests throughout the development process using a suitable testing framework.

Performance, Maintainability & Manageability

The University of Aberdeen currently develops their internal systems with the Laravel PHP framework, which is a server side scripting language. As such, this web application will also be developed mainly in the Laravel PHP framework. This will enable the University of Aberdeen software team to support and maintain the product for the future. Additionally, some functionality will be written in JavaScript using Vue framework, a client side scripting language. This will allow some functionality to be run directly on the user's local machine without needing to wait for the university's server to receive, process, and respond to the user request. In effect, this will reduce latency and ensure that web interactions feel instantaneous to the user.

Source code will be uploaded to a github repository and managed by the university of aberdeen at the end of this project.

Usability & accessibility

The website should be developed to fit WCAG 2.2 standard. This is a set of criteria which will help content to be more accessible to people with a range of disabilities such as blindness, hearing impairments, speech disabilities, photosensitivity, and cognitive limitations. Tools such as WAVE, Axe, and chrome dev tools can be used to validate compliance in different browsers. The user experience should be efficient and satisfying. The purpose of each interaction should be clear and easily understood to the user, and there should be a logical followup response to any user action. Our team will design the website centered around the user's needs because navigating the website should be simple, intuitive, and overall as user friendly as possible. Furthermore, we aim to achieve a simple but aesthetic layout to maximize user enjoyment and satisfaction. The UI should be designed in a way that is professional, learnable, and cohesive for people that have a range of disabilities.

Programming and scripting languages

- Front end
  - The website will be constructed with HTML
  - CSS styling will be done using SAAS framework
  - Functionality between the server and the client (user) will be written in PHP using Laravel Framework
  - Local functions will be written in Javascript using Vue Framework
- Backend
  - Python will be used for querying data in the database
- Testing
  - To perform all of our tests, we will use PHPUnit testing framework

---

PROJECT TIMING

The development methodology chosen to carry out this project is agile development. Following this, it was concluded that the one week sprints are most appropriate length given both the time-frame and other responsibilities of our coursework. This layout will give us the best environment to optimize our performance. Our sprint planning sessions will be held every Monday at 16:00 P.M. and further into the week, the classic sprint workflow will be kept to. Each iteration will allow us to incrementally progress towards a deliverable product for the client, for which we will have a further three or four sprints for.

DEFINITIONS AND ACRONYMS

- Aberdeen Registers Online: 1398-1511 (ARO) is a digital transcription of the first eight volumes of the Aberdeen Council Registers. ARO Definition
- Law in the Aberdeen Council Registers, 1398-1511: Concepts, Practices, Geographies (LACR) was a project to create a digital textual resource from the registers. This was achieved by creating a transcription and TEI-compliant mark-up of the entire corpus of text from the registers covering 1398-1511. LACR Project Definition
- Search Aberdeen Registers is a prototype web application to facilitate search within the Aberdeen Registers Online (ARO) corpus. SAR Definition

ARO - Aberdeen Registers Online
LACR - Law in the Aberdeen Council Registers
SAR - Search Aberdeen Registers
TEI - Text Encoding Initiative (An organisation which has developed guidlines for presenting text online for research, teaching and preservation)
XML - Extensible Markup Language (Used for data representation and storage)
PHP - Hypertext Preprocessor (oginally stands for Personal Home Page) - powerful programming langage that is commonly used when building websites
UI - User Interface
UX - User Experience
HTML - Hypertext Markup Language (Used for the front end of a website, basic layout etc)
CSS - Cascading Style Sheets (Used for the front end, to simplify the html, and to make the UX uniform)
