



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Advanced Scholarly Search Engine

By Jake Mason (256602)

Supervisors: John Williams & Andrew Green

Project submitted as part of the requirements for the
award of BSc (Hons) Applied Computing
18 March 2022

Abstract

The first aim of this project was to investigate currently existing scholarly search engines, to obtain an estimation for how long it would take a student to get valid information from the sources provided, this was done via primary research, getting participants from a variety of studies to find sources relevant to their course and to record the time. The second aim was to create a system which minimizes research time, success was ensured with a follow-up primary research study using the same participants to research using the new system. The final aim was testing of the new system, ensuring it worked properly and making improvements based on feedback. A wide range of research methods were used. Methods used included positivism and deduction, questionnaires were used for primary research and a Gantt chart to create a time horizon. The major discoveries made during this project were the time it takes students to conduct secondary research, the process of creating a search engine and getting it online, as well as how to conduct future projects in a similar fashion – by utilising similar methodologies and forms of user testing, as they proved effective here.

Contents Page

- Abstract - Page 1
- Contents - Page 2
- Introduction - Page 3
 - Aims and Objectives - Page 3
- Research & Review and Literature - Page 3 – 6
 - Google Scholar - Page 4 - 5
 - Microsoft Academic Search - Page 4
 - University Webpages - Page 4
 - RefSeek - Page 4 - 5
 - Issues - Page 5 – 6
 - Access to Information - Page 5
 - Finding relevant information - Page 5 – 6
 - Coding - Page 6
 - Accessing repositories - Page 6
 - Mitigating inaccessible results - Page 6
 - Summary - Page 6
- Methodologies - Page 6 – 12
 - Research Methodologies - Page 7 – 10
 - Philosophy - Page 7
 - Approach - Page 7
 - Strategy - Page 7
 - Time Horizon - Page 7, 8
 - Gantt Chart - Page 8
 - Data Collection - Page 8 – 10
 - Secondary Research Methodology - Page 8 - 9
 - Primary Research Methodology - Page 9
 - Evaluation criteria - Page 9 – 10
 - Methods of testing - Page 10
 - Development Methodology - Page 10 – 12
 - Time Boxing - Page 10
 - Software - Page 11
 - Hardware - Page 11
 - Materials and Tools - Page 11 – 12
- Design & Implementation - Page 12 – 32
 - Design - Page 12 – 19
 - Project Requirements - Page 12
 - System interfaces - Page 12 - 14
 - File Structure - Page 15 - 17
 - Data Structures - Page 17 - 19
 - Getting system online - Page 19
 - Implementation - Page 19 – 32
 - Problems and successes - Page 19 - 20
 - Unique Code Snippets - Page 20 – 22
 - Combining unique results - Page 20 - 22
 - Changing result sort system - Page 22
 - Entity Relationship Diagrams - Page 22 - 23
 - Use Case - Page 23
 - Testing - Page 23 – 32
 - Unit Testing - Page 23 - 28
 - Usability Testing - Page 28 - 30
 - Acceptance Testing - Page 30 – 32
 - Time Boxes - Page 32
- Evaluation - Page 32 - 33
- Conclusions & Recommendations - Page 33 - 34
- Reflection - Page 34
- References - Page 34 - 38
- Appendix - Page 38 - 39

Introduction

When conducting secondary research for formal essays, scholarly sources were obtained in some form. These sources can include journals, university papers, court documents and so on – obtaining said sources nowadays is mostly done through the internet using scholarly search engines, however these search engines sometimes have papers unique to themselves, this means searching on one rather than another means limiting what you will be able to access. The solution to this is to search on all these search engines, however this can take a lot of time, on top of that finding the information which is useful to your search query still requires a fine read through the sources presented to you, which can in turn also lengthen the secondary research process. Time is of the essence when drafting essays or research papers, with most of that time be spent towards reading through secondary sources can be incredibly time consuming, eating into time which may be better spent writing and furnishing the essay or research paper itself.

This project has investigated a limited number of themes regarding the research question – such as the current systems in place for use in scholarly document searching, the issues which arise from their use such as limited access to information, and the time sink it requires to find specific relevant information – as well as a solution to these issues.

A possible solution which was conceptualised is to build a web-based application which could access the results of multiple scholarly search engines to generate results, as well as using the keywords of the search to find relevant information within the search results and present these relevant findings. The latter desire for the application may not be perfect and may require the user to still do some further reading into a specific source to find information.

Aims and Objectives

1. Investigate existing scholarly source search engines to obtain an estimation of time researching satisfactory sources takes
 - Find different scholarly search engines to test.
 - Choose a specific subject and set of information to research using said engines.
 - Record the time it takes to find sources, and within said sources find the information that is relevant.
2. Design and develop a system which can minimize the time needed to find needed information from scholarly sources by filtering through search results and presenting the relevant information in the search results directly, to avoid the process of reading the entire scholarly source.
 - Find an appropriate language and IDE to create the system.
 - Get the system online using appropriate software
 - Make sure it is easy to use and looks presentable.
3. Test the project to ensure it satisfies its goal without significant error.
 - Ensure it can search for relevant sources and present the information needed directly.
 - Allow users to test it, and present feedback.
 - Use feedback to make improvements.

Research & Review and Literature

This literature review will be divided into sections, including what currently exists regarding the field the project aims to have an edge in, looking into their issues, what they have to offer, their unique elements and flaws. Citations are noted, when necessary, with their attributions being discussed briefly. The main issues found will be spoken about in a more general way in the following section, as implied – citations will be noted when necessary, and will be elaborated upon in instances when it is important, how each of the systems investigated in the previous section mitigate these issues will be spoken about, when possible, too, in this section.

Google Scholar

Google Scholar is the most popular student essay search engine currently existing. It is free and indexes the full text and metadata of sources deemed as scholarly literature, this is in many forms such as PDF files, online Google books and HTML. Court papers, theses, reports, and abstracts are included in its archive alongside academic books and journals [1].

Google scholar has come under criticism for not being updated [2], it has also been claimed that Google Scholar updates every day (or every other day) [43] [44], at the cost of reliability, since it will now include “anything that looks... remotely like an article”[43] , with no quality control [45] [43]– the inclusion of journals has lowered its database’s overall validity [3]. Google scholar’s search results are ranked by how much a paper has been cited, meaning older sources are given a higher priority [4], this is considered to be inefficient as it means information that is not up to date are given priority. How often should a scholarly search engine be updated is an important question to ask, ideally a balance should be struck between quality control and up to date information, however if one must be chosen over the other then having the most available sources should come first as at least the sources can have their reliability analysed, however if an archive is not updated then crucial sources may be missing.

Microsoft Academic Search

Opened in 2006 and shut in 2012, before being relaunched in 2016 – Microsoft Academic Search was created to compete with Google Scholar [5]. It contains 27.2 million records for books, conference papers and journals [6]. Development into Microsoft Academic Search has been regarded as “slow” relative to its competition, with its archive of resources not updated as often, according to one research paper attributed to Spanish researchers who specialise in web development – Microsoft Academic Search is largely ignored and not taken seriously by academics and bibliometricians [7] [47] [46], this consensus seemed to be true among those working at MIT too [46] – whom state that its archive is generally not up to date, and not due to quality control reasons too with some of the repositories for information containing articles that did not go through “the peer review process” [46].

University Webpages

Universities often provide their own library searches which allows students to browse their archives of sources on subject matters. Often these sources are incredibly high in validity, but the capabilities of their searches are low, as it typically must pertain to a specific subject taught in the university – although this is different per school’s own system. Usually, access requires the user to be a student or staff member in said school too, which is understandable for their own sake, however for a student at another university trying to browse as many sources as possible, this could be disadvantageous – although it should be stated that if a source is being used by one university, it is most likely in circulation among the scholarly search systems of the others too.

RefSeek

RefSeek has a similar UI design to Google Scholar but differs from it and its competition by allowing users to view regular websites and another option for scholarly documents. The section for all websites does filter out what you find on a normal search engine such as journalist articles and commercial sites [36]. The flaws of RefSeek however come from its low archive, having no exclusive contents, and a smaller library of documents than its competition, while it does provide easier access to PDF documents than Google Scholar, it does have any filters for searching, quick ways to cite or any way to save documents [36]. The primary source on RefSeek comes from a journal entry in information in the modern age from 2016 written for the Chitkara College of education for women, with its purpose being for educating, with no ulterior motives, originating from an academic background.

Issues

Here issues concluded to exist in the present field of scholarly search engines is detailed, based on the research found above, further research is cited here, when more information was needed than what was initially found above.

Access to information.

The suggested articles from Google Scholar are not always accessible, an older source from 2005 from Yale university found Google Scholar to often suggest sources which are locked behind paywalls, such as ones from the 'Journal of the American Society for American Information Science and Technology' – the intent of said paper compared Scholar's quality to paid scholarly search engines, claiming it to be less updated [17]; while the latter criticism may not be true anymore, the former is. A search for phenomenology brings up over a million results, the first of which being a book locked behind a paywall, which may inspire users to go out of their way to find free PDF files, hosted on unofficial, unlicensed servers for the books they intend to browse.



Here is a screenshot of the first result when searching for phenomenology.

When searching on BASE, it seems this issue is not as prevalent, with inaccessible texts not appearing on the top of the search when the same text is entered [18]. My citation is from the library journal site InfoDocket, in a paper from the sites founder Gary Price, a librarian, writer and frequent speaker at Washington DC.

Finding relevant information

The main part of the issue being investigated is the speed in which it takes to find relevant information to the keywords being searched. Searching for ‘internet of things privacy requirements’ will bring relevant sources on the top of the search, as expected – although the first source is from 2014, when newer sources from 2018 and 2017 are below it – This 2014 source does not go into the privacy requirements for IoT devices or applications in its entirety, with all info regarding that area being briefly mentioned in its introduction. Security requirements are quite detailed however, but that was not what the search was for [19].

When searching for what intentions may be for intruding on privacy, for IoT – valuable selections of text could be found in an unrelated, political paper, regarding how American political parties utilise social media [20], when searching related keywords, this source would have taken a while to appear on the scholarly search engine, due to its irrelevance, but it holds more relevance than the previously discussed source on IoT Privacy requirements did to its relevant search.

Coding

Accessing repositories

Some scholarly search engines have search results which others do not. The Cyber-metrics Lab at the Spanish National Research Council has a page dedicated to showing scholarly sources that cannot be found on Google Scholar [56]. A tutorial to adding new repositories is given by the Bielefeld University [42], using this tutorial and the list of repositories not found on Google Scholar, as well as the repositories given by the Bielefeld University, a large database can be created. Adding certain repositories to the archive requires simple XML code [57], ensuring only accessible documents are available by the engine will be detailed in the next section.

Mitigating inaccessible results

Google Scholar uses “automated processes for indexing” [49] in regard to its repositories, this process however leads to certain inaccessible items being indexed [48]. In 2007 Google Scholar used Eprints, which indexes primarily based on if an archive has an available home page [50], which again leads to not available documents being found in Google Scholar’s archive at times. A solution to this could be to use a different method of indexing repositories, such as DSpace, which had been used by Google Scholar in the past – DSpace can index just PDF documents [51][52], meaning it will index just the source itself, and not the home page advertising the PDF – while it is behind maybe a paywall – which is sometimes the case with Google Scholar, as well as Microsoft Academic Search [54][55] and BASE [53]. DSpace will search the repositories given and only index the documents which are accessible publicly [51][52].

Summary

The most important pieces to take away from the research conducted is that access to information and the speed in which relevant information can be found are the biggest issues with current academic search engines as a whole, however individual search engines have their own issues, namely small archives of data (in the case of Microsoft Academic Search), and a search filter system which does not put the most relevant sources at the top, going for most citations, as a way to present the most reliable instead of valid source first – which while having advantages, serves only to make finding relevant information take more time, the proposed system here will mitigate this issue by using a more standard search algorithm system which will prioritise relevance, access to information will be attempted to be solved by searching for the actual source itself, not where it may be hosted on – most scholarly search engines will search for webpages based on the keywords of the search term, webpages which host the desired documents for the user, and the host page will contain a list of terms and keywords which make it appear in searches – while the document may be

hidden behind a wall of some kind – by searching however for just terms that appear in the desired document, then the results will only bring up the desired documents, no walls of inaccessibility, no host page to struggle through to find where the download/view button is for the document they are hosting.

Methodologies

Research Methodology

This paper has cited literature from scholarly articles that investigate relevant fields or contain information relevant to this subject area – regardless of its original theme – the purpose, validity and reliability has been measured for each secondary source. Less scholarly sources were utilised, but in a limited quantity, and not to draw any conclusions.

Philosophy

Positivism was the research philosophy chosen for the user testing surveys as it finds its strengths dealing with quantitative data that is easier to analyse [14]. Some parts of the feedback needed to use open questions, thus large, detailed qualitative responses, for this I utilised a phenomenology philosophy, as it is more equipped to handle more subjective contexts – such as open answers given in primary research. [15].

Approach

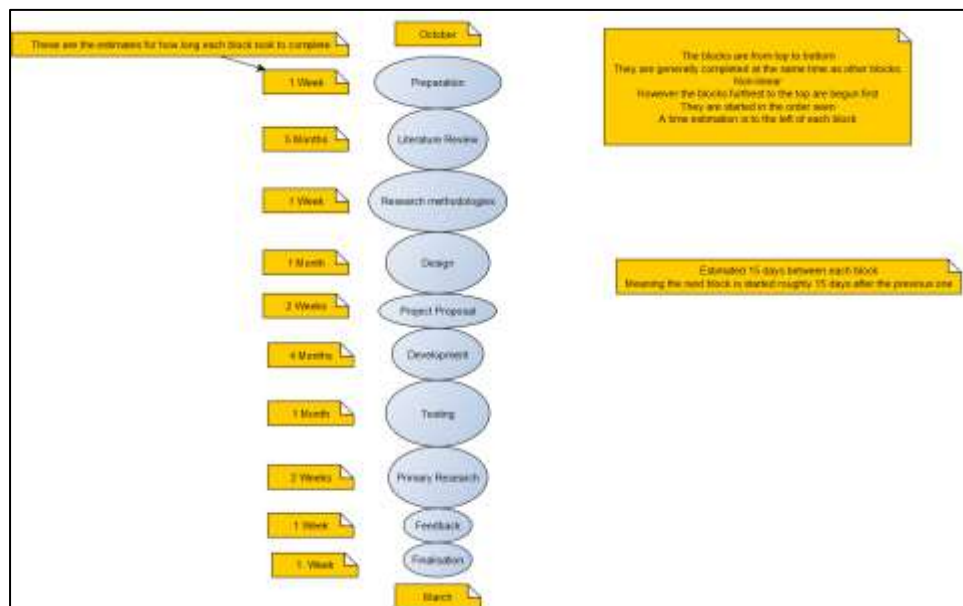
Deductive reasoning was used as the objectives for this solution have been clearly thought out. This approach is known for saving time, and for having use in situations (such as this one) where the information necessary has already been concluded and is existing [12] [13].

Strategy

Questionnaires were created for user testers of the solution, to draw conclusions regarding the effectiveness of the device and the quality of its UI, and ease of use. Questionnaires are useful for quick distribution and results, as well as obtaining criticisms, but limited in guidance - according to J Rowley's book on 'Management Research'[11], which details designing effective questionnaires and their usage – this comes to use here as the questionnaires are used to gain feedback on what does and does not work, thus what needs to be fixed and changed, as opposed to being used for guidance on what to create for the solution, this final point leads me onto the approach which was utilised. Archival Research was conducted for the secondary research, as the secondary sources are sources that already exist. Extracting data from relevant, reliable sources which was discovered for use here in either finding errors in currently existing search engines, or in help in the construction of the application.

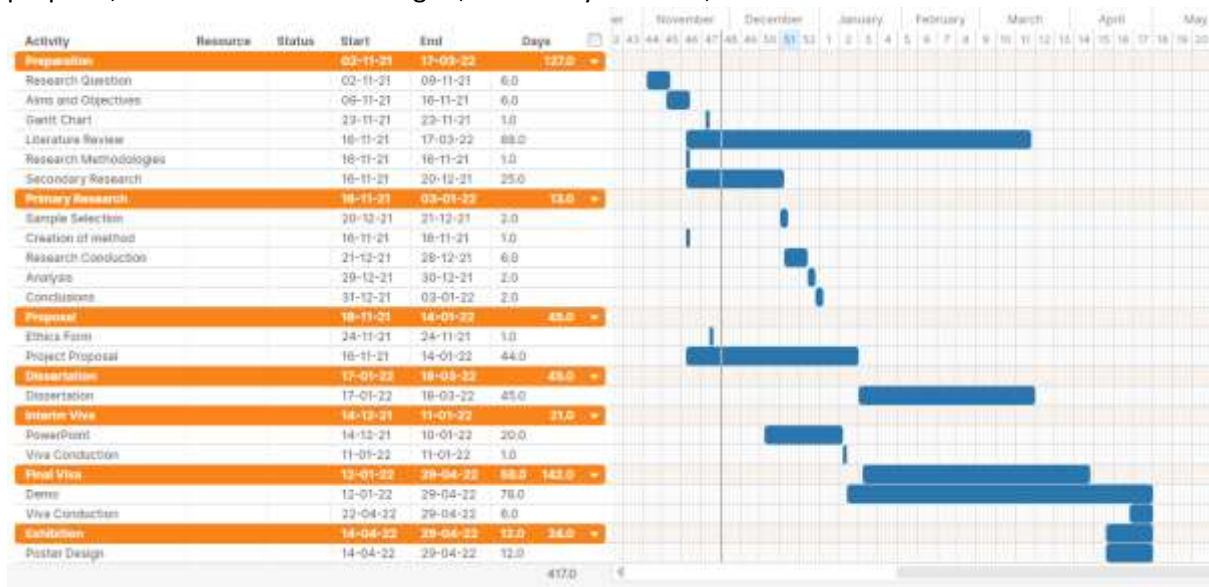
Time Horizon

Time horizons for the different points in the creation of this project were conceptualised at the project's genesis. Development and areas needing to be done before development had a 6-month span to be completed, and thus the time horizon was created to revolve around this 6-month schedule.



Gantt Chart

A Gantt chart has been designed to manage each factor of the project, including the research, the proposal, the research methodologies, secondary research, and the creation of the solution itself.



The dates given on each section are not concrete and are generally flexible, the Gantt chart works more for organisation's sake and to provide a basic frame for when things need to be done by, as to keep progress on the solution on the right track, on a consistent pace. Providing each needed section its own category and time estimation, gives it a clear indication of where it comes in the process of the app's creation, which in turn can also provide a guide in the contents of said section, as well as what can and cannot be worked on at the time as a specific task.

Data Collection

Secondary Research Methodology [8][9][10]

Secondary research can be analysed with ease [8][9][10]. Secondary research is economical both in resources and time, as the studies conducted to draw the results told in the data was already done

by someone else, the acquisition of said data does not take too long neither because of the internet [10]. Secondary research originating from third parties adds another perspective that may not have been considered before [10]. Secondary sources provide qualitative and some quantitative data. The qualitative data is primarily used however, this is because most of the secondary research needed is non-statistical, looking into statements by experts on specific relevant subject areas [9][10].

Quantitative is used for areas where objective, more mathematical data was necessary [8] – such as the size of the source archives for search engines – however most of the secondary research investigated criticisms and compliments made towards current scholarly search engines.

Quantitative data however was used for sources which can assist into the construction of the application itself, such as information on developing web-based applications, as well as what relevant APIs may exist and find use.

Qualitative and quantitative data both have their pros and cons. Qualitative data can provide more insight than quantitative, and because of its descriptive, more elaborative nature, is also higher in overall validity [9][10]– but this can come with the downside of it taking longer to analyse, and being lower in reliability, however [25], to counter this possible lower reliability only scholarly sources were cited, on top of CRAAP tests being conducted. Quantitative data is higher in reliability, compared to qualitative, and takes less time to analyse, but this comes at the cost of its usefulness to a specific subject area, with its data being vaguer [25].

Primary Research Methodology

The primary research used a mixed methodology. Asking the sample to provide answers, some of which being quantitative and others qualitative. The primary data was utilised to source feedback on the application's quality for its intended audience, as well as what improvements could be made.

Quantitative data here consisted of quality ratings for the application and its relevance, qualitative for asking participants to justify their quantitative answers and for obtaining suggestions for improvements on the application – this ensured the primary research yielded reliable quantitative and valid qualitative data, providing easy to analyse results in regards to their feelings on the project, as well as more in-depth, valid, but harder to analyse thoughts that can be more useful for feedback, and making improvements[25][26][27].

Evaluation criteria

The criteria in which the project will be evaluated against is mostly based on assessing how well it meets its purpose and satisfies its target audience. Furthermore, its ability to progress already existent concepts, as well as its likelihood of engineering a fresh, successful technology will be considered. The last criteria in which this application will be evaluated upon is its benefits over the competition, whether any do exist [21].

The proceeding adjectives will be assigned accordingly, based on how well the project fits one of the descriptions.

Excellent	Addresses every area of criteria in a convincing manner, displaying performance and showing no weakness, and all requirements fulfilled.
Good	Requirements fulfilled; minor weaknesses exist.
Satisfactory	Requirements met, but with some significant weaknesses displayed, or many insignificant weaknesses shown.
Unsatisfactory	Not all requirements met, aim fulfilled. Cons outweigh pros.

Unacceptable	Failure to meet all requirements, unlikely to ever meet success. Little to no positives found, nothing salvageable.
--------------	---

[21]

Methods of testing

Black box testing will be explored, the testers will not have any knowledge of the workings of the application, remaining oblivious to its internal intricacies – interacting merely with the interface, providing inputs to obtain outputs. Black box testing has the advantage of code access not being required, and allows a larger sample to test, as expertise is not required. Black box testing distinguishes the perspective of the user and the developer clearly too, useful for this project as the target audience will not necessarily have the knowledge of the project, or any of the elements which make up the project – university students being the audience, but not all university students have any interest, knowledge or need to know of the functions used [22].

Feedback on errors is asked for from testers, which may cause some issues when using black-box testing, as users may not be able to highlight exactly how it went wrong or provide solutions on how to fix. This disadvantage will not be as evident here as it may be in other tests where black box is utilised, this is because much of the code being used is based on pre-existing code which already does function as intended and has been thoroughly tested and proven to work [22][23].

Large code segments are being utilised in this project, alternatives like white-box testing may prove tedious here compared to black box because of this, as white box will require specific details and analysis of the large code segments – making testing take longer and reducing the sample size [22] [23].

The test sample was university students, believed to be representative of the desired user base to the application – as most university courses need to conduct research using the most reliable sources, there was no filter on which study each student was focusing on, because of this a broad range of students was selected – studying a large range of different courses.

Development Methodology

Time Boxing

Time boxing was incorporated for personal time management, specifying units of time for certain activities done for the duration of development – each ‘timebox’ had its own deliverables. Time boxing fit into the development of this project as the aims and their objectives distinguish deliverables for quite different sections of development. A variety of highly different activities were needed to be completed, so separating them into timeboxes to be completed one by one made management a lot easier. Timeboxing is like the Gantt chart created for this project too, with different tasks separated in their own sections with time limits [28]. Timeboxing manages risk well, by reducing the scope through lowering the impact of requirements, when dividing them up – extra time constraints ensure schedule is stuck too [29].

The separation of deliverables presented by timeboxing however does mean the bigger picture becomes more out of sight, leading to a lower quality in the result of specific deliverables, or the quality of them on their own remains intact – but when the deliverables come together as a finished project, said project may not represent the sum of its parts [28][30].

Software

- Operating System: Windows 10 Pro
- Microsoft Office (Word, PowerPoint)
- Brackets
- Google Forms
- XAMPP
- DSpace

The operating system being used is Windows 10 Pro, this is because of its access to Microsoft Office programs which are pre-installed with the OS such as Word and PowerPoint. Microsoft word being the tool which was utilised for the literature review, and PowerPoint being used for the creation of the necessary presentations for displaying the interim and final vivas of the creation.

Brackets was the IDE used for creating the web application, because the software is free, optimised, provides useful addons and is easy to use – having an experience with it in the past.

Google Forms was used for creating the questionnaires, this is because it is free and provides an easy way of distribution via sharable URLs.

XAMPP was needed for testing if the application can go live. XAMPP makes transitioning to a live web server from a localised server for testing possible. Providing MySQL and Apache HTTP server scripts. XAMPP is also free, and requires only one zip, tar, 7z or .exe file to be ran; with minimal configuration of the several components needed in the creation of a web server.

DSpace is an open-source repository package used for scholarly repositories, while it is not the only software like this (ESpace for example is what Google Scholar uses), DSpace focuses on access, with its registry list containing over three thousand accessible repositories [57], all of which are accessible for users [51] [52].

Hardware

- Graphics Card:
- Processor: Intel® Core™i7-10700K CPU @ 3.80GHz
- 16 GB RAM
- Storage (1.8TB HDD, 477GB SSD, 932GB External HDD)

These are the specifications of the desktop that was used for the creation of the web-based application as well as the conduction of the research – both primary and secondary. The more high-end components ensure there was little issue running necessary hardware, with there being no lag during testing – despite this, the application is being designed with low end specifications in mind, with it being usable by university students on even the worst computers on the market.

Materials and Tools

HTML is the coding language intended on being used, as is CSS. This is because the application being created is going to be web based, and websites use HTML for their creation – as all web browsers support it, CSS will be used to improve the appearance of the app.

Windows 10 Pro specifically was chosen over Home because of increased security (having BitLocker device encryption and Windows Information Protection) and because it is better for deployment, because of its business management capabilities – having mobile device management , and group policy[24].

JavaScript adds interactivity to webpages, which will be essential for this web application in particular. Hamburger menus are planned to be used for compressing essential features when reducing the default resolution of the web application down [37][38]. 'Node.js' will be used because it is useful for bringing event driven programming to a webpage, useful here for improving user interactivity and input [39][40], allowing the webpage to react and change based on a user's action [41].

Design & Implementation

Design

Project Requirements

The project being constructed is not an original concept, but a refinement of an already existing system, thus several requirements will be the same as those which apply to the currently existing scholarly search engines.

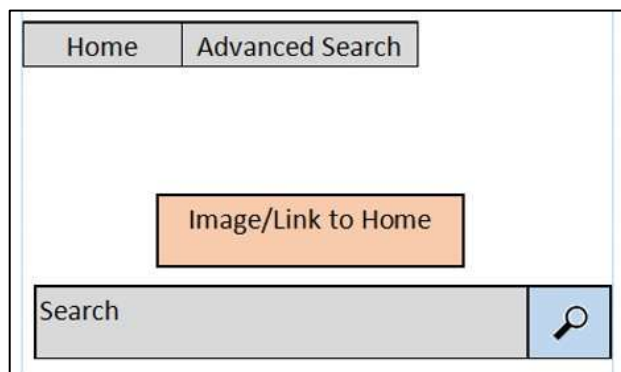
The application being created must be able to take in user inputs and complete a transaction with said information, creating an output. In this context the input would be a user search (with or without use of the advanced search option), and the outputted information would be the search results.

The application must be able to provide an advanced search system, allowing the user to alter specificities of the search results of their input, for example searching by just the 'author' or 'title' of a document, or be desiring only results showing a specific format such as a journal, image, video, or patent. This is a requirement fulfilled by what is already existing in the field of scholarly search engines, however in this application the advanced search will be more detailed than that provided by pre-existing engines offering users the ability to select 'access' type, a larger range of document types (and subcategories of said document types. Specified in full in the appendix), nation of origin, the publication year and publication date range.

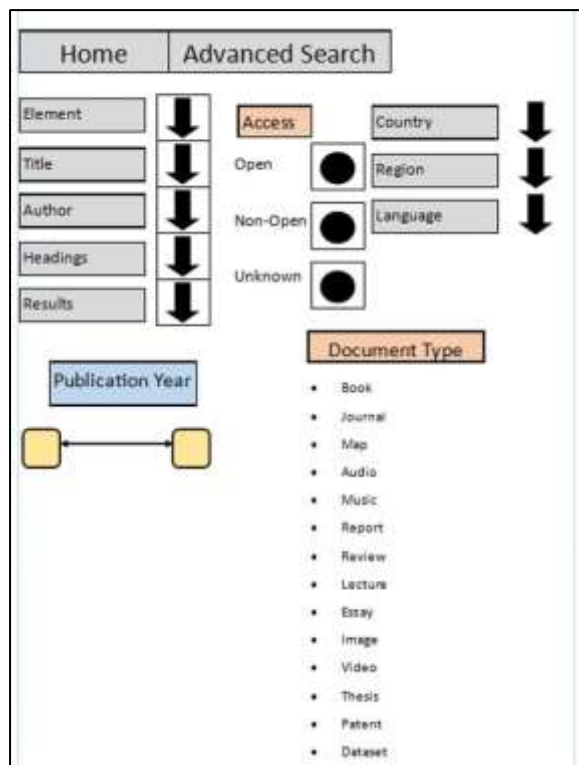
The search results should be ordered using an algorithm which places the most relevant search result on the top, as opposed to the most common method which the most popular scholarly search devices utilise – which is to organise search results by placing the result with the most citations at the top, as a way to better ensure the most reliable result is at the top of the page [4], this system is flawed as it does ensure more reliable search results are at the top, but this does mean the results most likely to appear first are not necessarily relevant to the search term and could be useless to the user, it also means longer pieces of text are favoured as they can fit in more citations – this further increases the time for the user to obtain the information they need. A system which favours relevance prioritises validity over reliability would decrease the time necessary for users to find information they need, and because the sources will be smaller – tests for reliability will take less time.

The last requirement of this application is to output search results which would otherwise be exclusive to specific scholarly search engines. For example, a specific document which is related to 'Data Warehousing' may appear on google scholar but will not appear on Microsoft Academic Search. It is a requirement of this application to be able to provide a universal scholarly search (so to speak) – providing results from varying engines, as to further lower the time it takes for a student to research – requiring them to only stop at one search site.

System interfaces

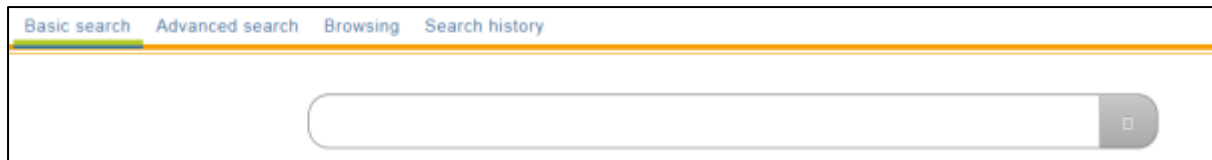


This is a design idea for the home page. Displaying the home page and the advanced search section, which contains the search filters. A placeholder for the logo exists, this logo would link users back to the home page.



The pre-code prototype for the advanced search contains options for geography, language, aspect of paper, type of paper, publication period, attribution, and the ability to limit the number of results that appear. The down arrows indicate a drop-down box, while the circles in the squares represent multiple choice. The double ended arrow with two yellow boxes is a range selector, the lower year being placed on the left and the later year on the right. Document type is also meant to be a multiple-choice feature.

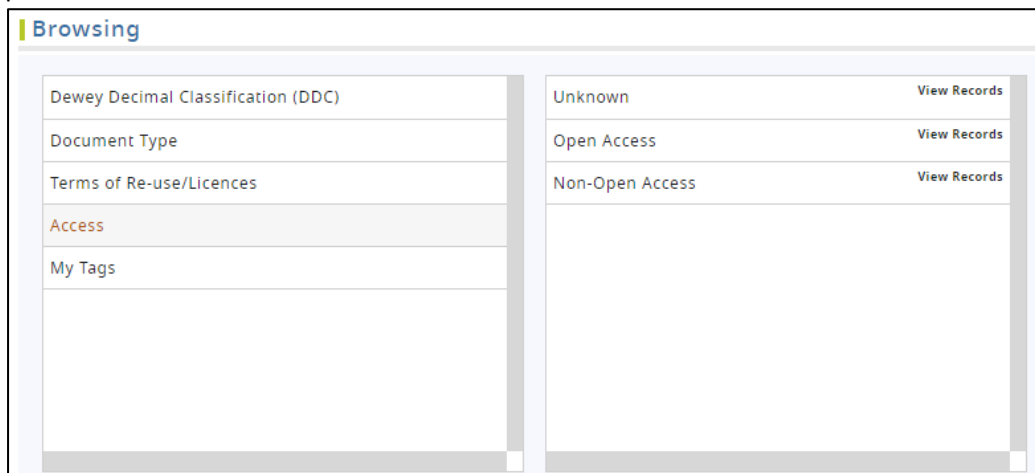
A html interface was created based on this design, the basic search acting as the homepage. The search bar was put into the top/middle of the page to be in line with modern search engines, a design choice done for user preference [31][32], as it draws the user immediately to the primary function of the app, saving a brief time which would be spent scanning the page, it sounds miniscule, but this immediate convenience does impact user experience [33].



The logo was dropped entirely, as the home page can be re-accessed by selecting basic search in the header, which is available on every page. A noticeable change here compared to the prototype is the inclusion of the 'browsing' and 'search history' options. The browsing option being a holdover from the API used, initially not planned for use in this application, but it provided users with another option for searching, browsing by selecting the document types first, it was a simple option to get functioning and did provide users with another option, so it was left in once it was working properly.

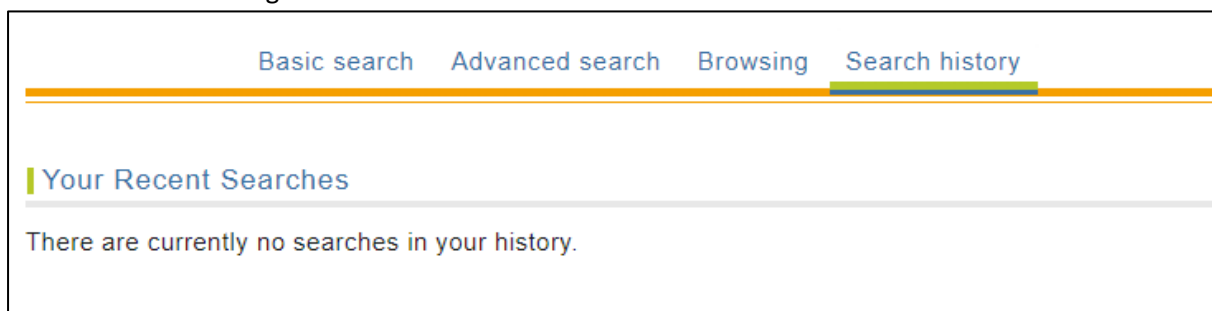


It works from left to right, with options in a box appearing based on what was selected from the previous box.



With the options available being based on the advanced search.

The search history tab will keep track of your recent searches for a limited time, taking inputted information and saving it.



To make this search history option save information permanently a login system was conceptualised, allowing certain sources to be saved to the search history page, for archival sake – being useful for the user, allowing them to re-access sources for future use, an option not supported by Google Scholar presently [34][35].

The advanced search in concept was not changed much from the original design, other than being changed to 'advanced search' from 'filters,' this change was made for more accuracy – filter implies funnelling results already searched to a specific sect, while advanced search refers to the actual search process being changed – which is closer to what was is done on this page. Advanced search is also what it is called on Google Scholar, which provides an extremely limited advanced search option.

File Structure

This application is HTML based, so the file system is integral to ensure consistency, with files required for certain interfaces needing to be in specific folders. Files relating to the app were put in the same directory.

Project HTML

Within this the HTML documents were contained, as well as the files containing the JavaScript and CSS for certain pages which need it. Non-essential documents were included here too such as the pseudo-code.

```
<Title></Title>

<link rel="stylesheet" href=min.css
<link rel="stylesheet" href=bootstrap-theme.min.css
<link rel="stylesheet" href=base.css

[copy + paste lightbox content from index]

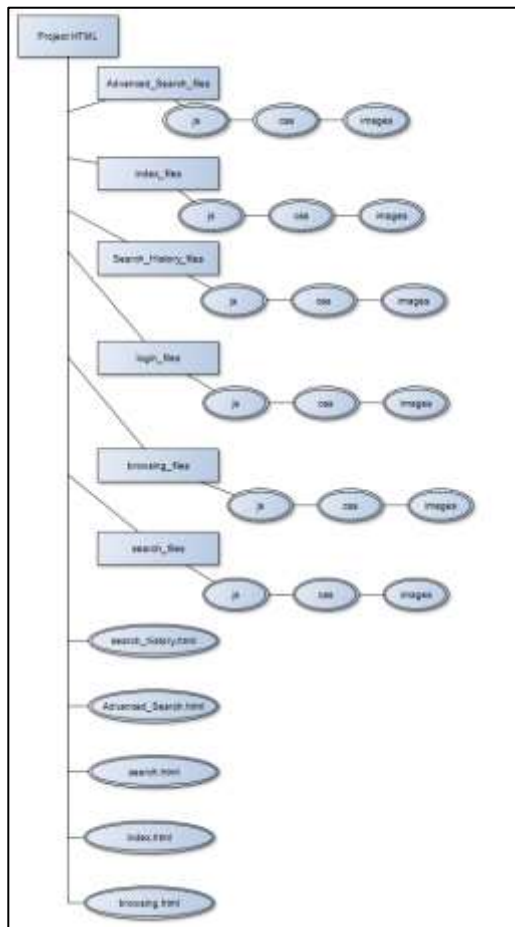
[header content from index]

<!-- Main -->
search/history.tpl
<h1 class="sr-only">Search History</h1>
<div class="row">

<html>
<head>
<script>
function goBack() {
    window.history.back()
}
</script>
</head>
<body>

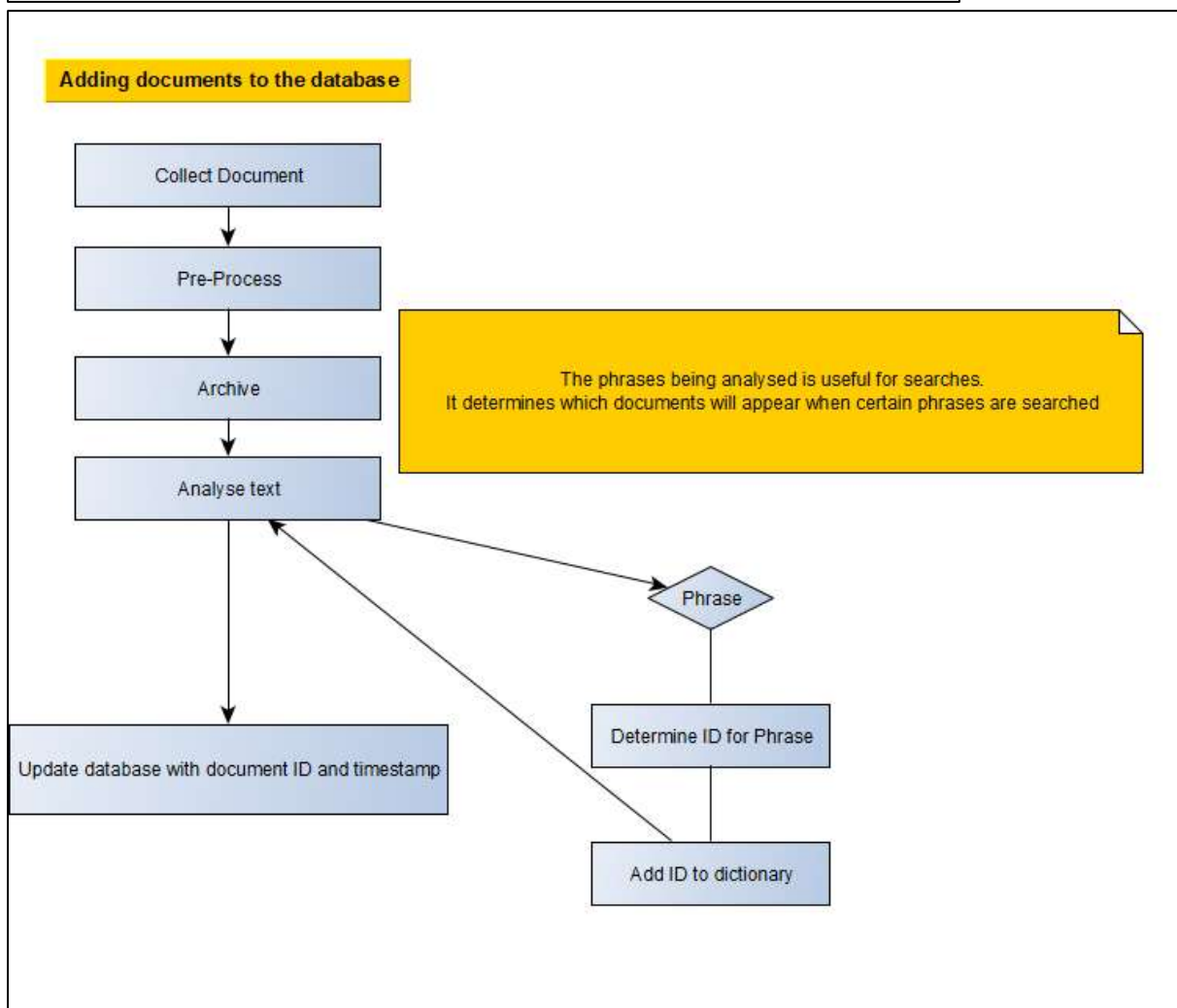
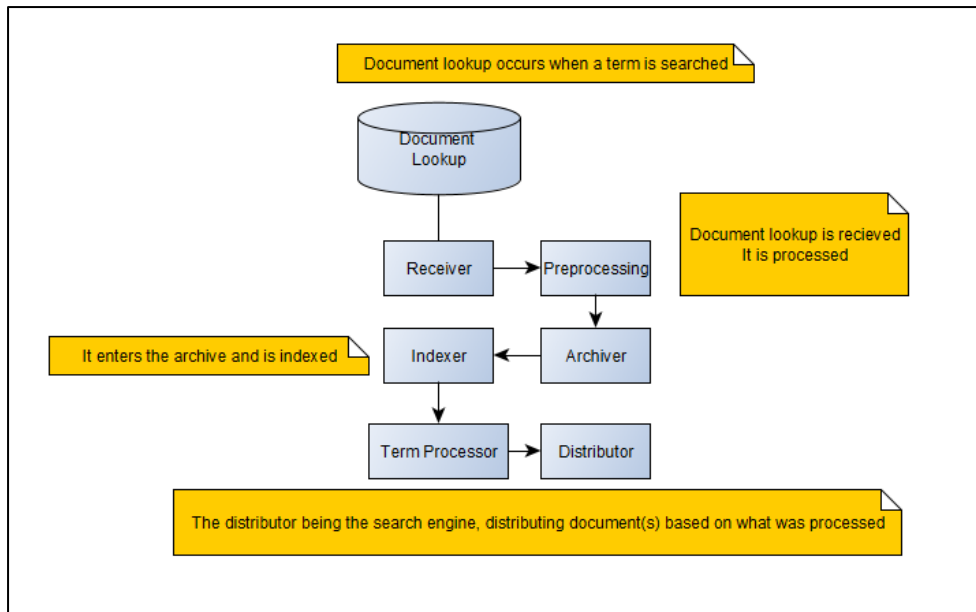
<input type="button" value="Back" onclick="goBack()">

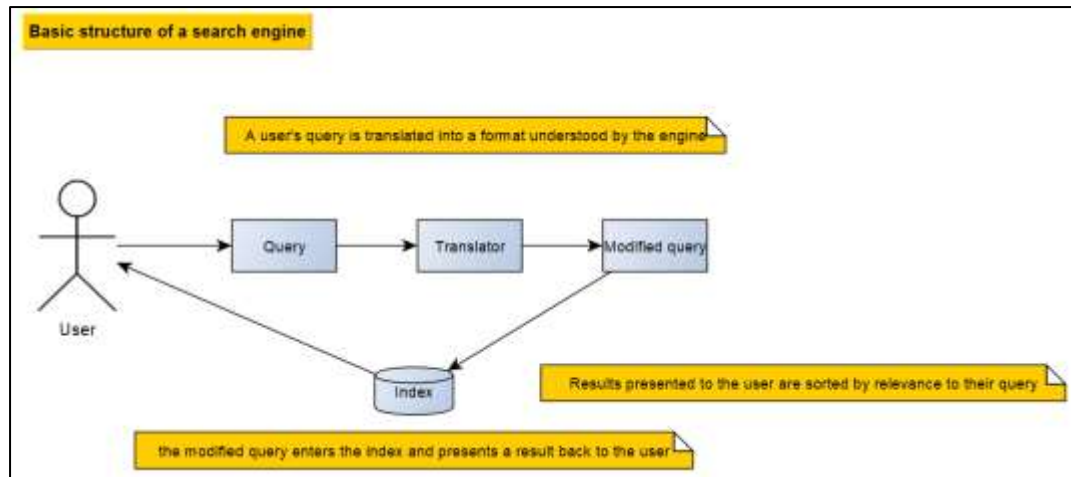
</body>
</html>
```



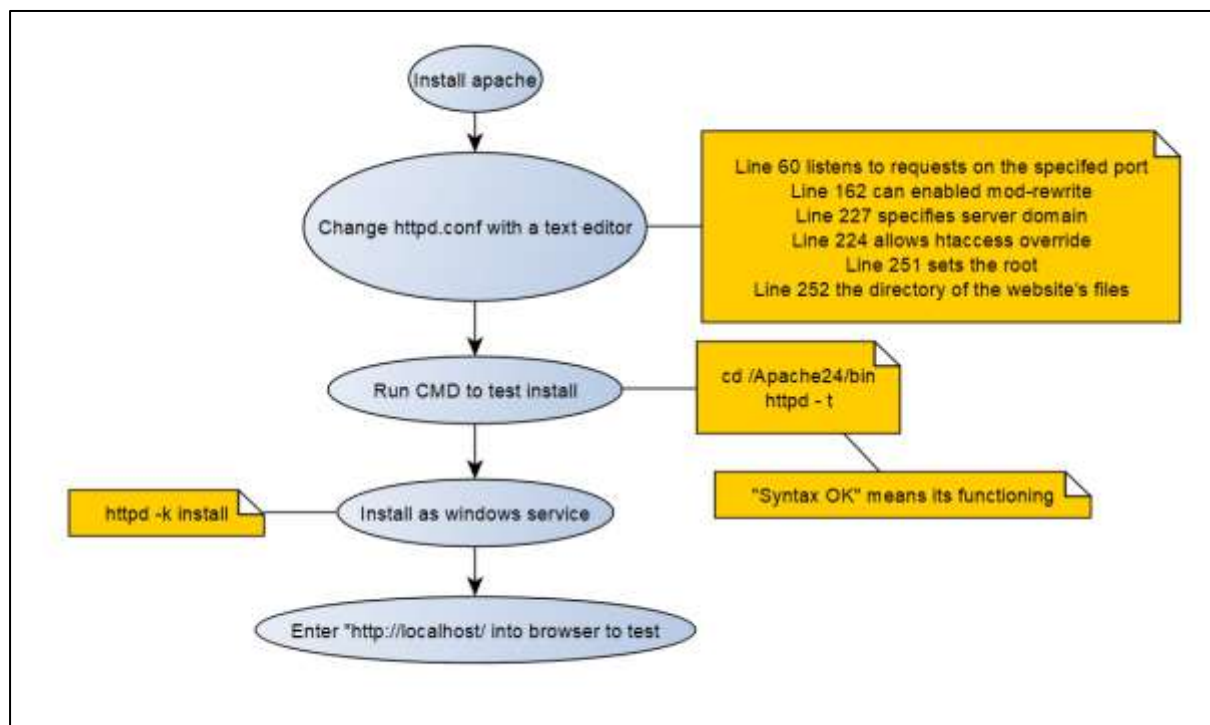
This is a diagram showing the file hierarchy, the rectangles symbolising folders, the ovals being documents.

Data Structures





Getting system online



The plans for getting the system online are simple and go as follows in the diagram. Each section has notes which detail specific details/instructions to go along with the planned step. The second bubble in the diagram's note specifies important lines to alter in the "conf" file, while the third and fourth contain notes detailing the code needing to be entered on CMD, and how to tell if it is working.

Implementation

Problems and successes

The success of some of the methods of testing could have been improved on, for example the primary research may have been more successful and yielded more valid results if the participants testing were being observed, that way more in-depth indications of success could have been drawn. For example, for testing the navigational success of the application was solely based on user's direct feedback, relying on their word, not that it should be assumed that they would lie, but rather interpret their ability to navigate the engine as being better or worse than it actually was – while

having an observer who knows the intended length of time navigation is supposed to take on the engine perform an observation could draw different conclusions, or even make more astute observations as to how users may use the engine, what their first actions are, what pages they remain on the most and so on.

A difficulty encountered during implementation was utilisation of the Base API requiring the IP range of the user to be authorized, which was possible by submitting a form, which took a while to be responded to, meaning time which could have been spent developing and troubleshooting was wasted [16]. This issue was partially mitigated by working out how to implement algorithms which would not have been included in the API before obtaining access to it, while evaluating them was difficult, their creation beforehand was simple enough; as was putting extra focus into areas which would have begun work later in the duration of this project, such as primary research.

In the project proposal, the idea of getting the application online was discussed, however it was realised during development that this would have taken up development time which could not have been lost in other areas, so the idea was scrapped as it was deemed ill-essential for finding a solution to the problem or meeting the requirements.

Successes was found in the development of making the search engine universal, including results from all the present alternatives, as the BaseAPI [16] has a system for including repositories from other sources, so including repositories from the alternatives was an easy feat to accomplish, despite it initially presenting itself as a daunting task. Similarly, the new sorting algorithm is the default way items are searched, so nothing needed to be changed here in the code, however hoping to still satisfy users, a “sort by” system was implemented, allowing further criteria for changing searches by users.

Unique Code snippets

Most of the code written was not entirely original, already existing in some form in other similar applications, however there are unique snippets here which highlight key elements of the application.

Combining unique results

A method for doing this is provided in the BASE API, although is not used in the BASE search engine itself [16]. Which is to change to add repositories for what would’ve been unique responses to the

XML.

```
<?xml version='1.0' encoding='UTF-8'>
<collection>
  <collection_name>cau</collection_name>
  <list_repositories>
    <repository>
      <name>Zürich Open Platform</name>
      <name_en>Zürich Open Platform</name_en>
      <activation_date>2022-03-10</activation_date>
      <internal_name>ftunivzuerichzob</internal_name>
    </repository>
    <repository>
      <name>Körszolgálati Tudásportál Nemzeti Körszolgálati Egyetem</name>
      <name_en>Knowledge Portal University of Public Service</name_en>
      <activation_date>2022-03-10</activation_date>
      <internal_name>ftunivnkedc</internal_name>
    </repository>
    <repository>
      <name>TAUJA: Repositorio de Trabajos Académicos de la Universidad de Jaén</name>
      <name_en>TAUJA: Repositorio de Trabajos Académicos de la Universidad de Jaén</name_en>
      <activation_date>2022-03-10</activation_date>
      <internal_name>ftunivjaentaufa</internal_name>
    </repository>
    <repository>
      <name>Scientia. Digital Information Deposit of the Department of Health</name_en>
      <activation_date>2022-03-10</activation_date>
      <name>Scientia. Dipòsit d'Informació Digital del Departament de Salut</name>
      <internal_name>ftscientia</internal_name>
    </repository>
    <repository>
      <name>Linguistic Data and NLP Tools CLARIN</name>
      <activation_date>2022-03-10</activation_date>
      <name_en>Linguistic Data and NLP Tools CLARIN</name_en>
      <internal_name>ftclarinit</internal_name>
    </repository>
    <repository>
      <name>docomomo journal</name>
      <name_en>docomomo journal</name_en>
      <activation_date>2022-03-10</activation_date>
      <internal_name>ftjdocomomo</internal_name>
    </repository>
    <repository>
      <name>FreiDok plus Zeppelin Universität</name>
      <name_en>FreiDok plus Zeppelin Universität</name_en>
      <activation_date>2022-03-09</activation_date>
      <internal_name>ftufriedrichshfd</internal_name>
    </repository>
    <repository>
      <activation_date>2022-03-09</activation_date>
      <name_en>Akademia Morska w Szczecinie</name_en>
      <name>Akademia Morska w Szczecinie</name>
      <internal_name>ftamszczecin</internal_name>
    </repository>
    <repository>
      <activation_date>2022-03-09</activation_date>
      <name_en>Marmia-Mazury Digital Library</name_en>
      <name>Marmińsko-Mazurska Biblioteka Cyfrowa</name>
      <internal_name>ftwbcolsztyn</internal_name>
    </repository>
  </list_repositories>
</collection>
```

Repositories will have their available documents contained within.

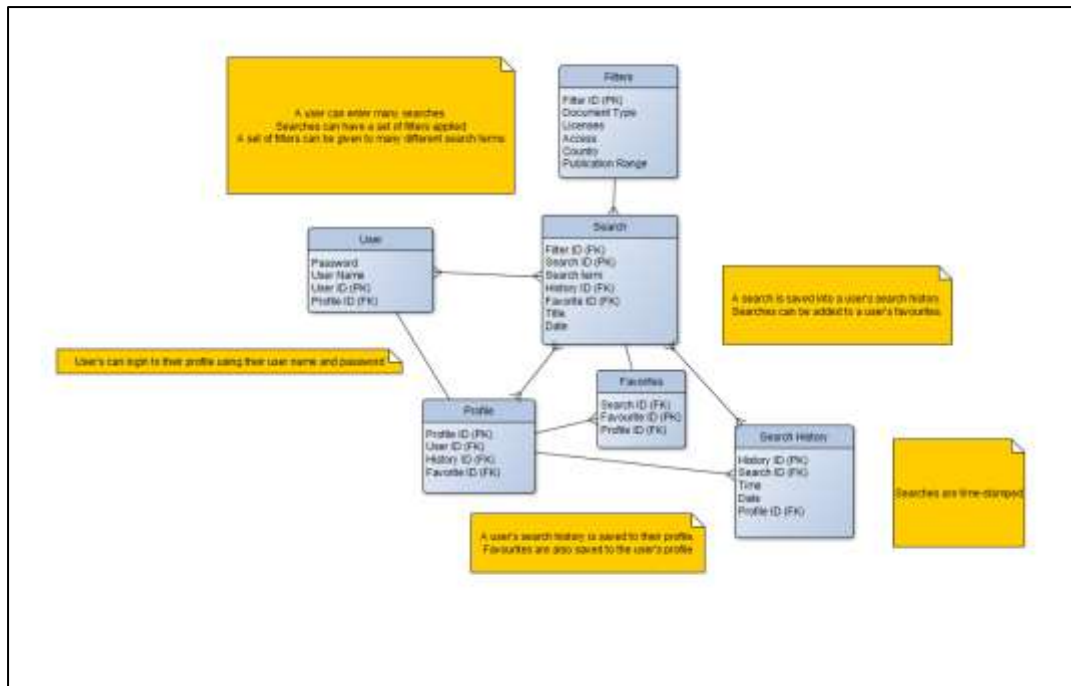
```
<?xml version='1.0' encoding='UTF-8'><?xsi:schemaLocation='http://www.loc.gov/standards/mods/mods.xsd http://www.loc.gov/standards/mods/mods.xsd'/?><mods xmlns='http://www.loc.gov/standards/mods/mods.xsd'><!-- Title --><title></title><!-- Author --><author></author><!-- Publisher --><publisher></publisher><!-- Date --><date></date><!-- Identifier --><identifier></identifier><!-- Description --><description></description><!-- Subject --><subject></subject><!-- Type --><type></type><!-- Language --><language></language><!-- Rights --><rights></rights><!-- Availability --><availability></availability><!-- Usage --><usage></usage><!-- Other information --><otherInformation></otherInformation></mods></?xml>
```

Changing result sort system

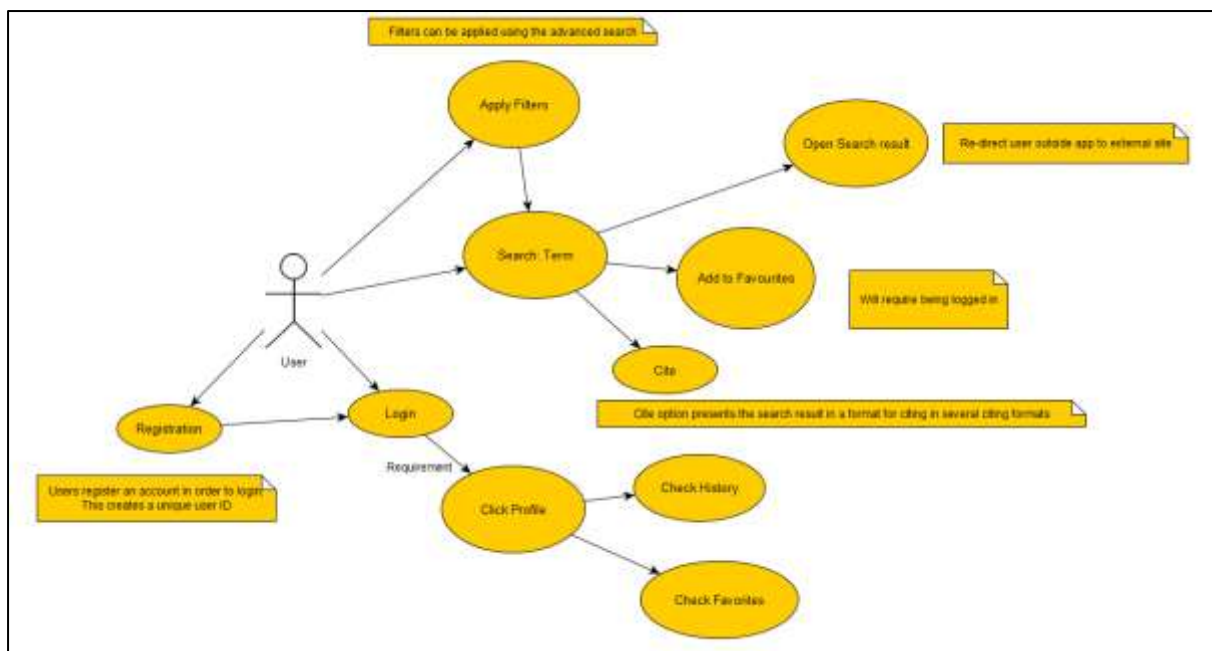
The common system scholarly search engines use for sorting results – as previously stated – is by total citations of each source. The change this system to go by relevancy was not difficult. Relevancy is the default system search tools use regardless, so implementing this required no extra code, to allow users the option of how to search for documents a “sortby” parameter was implemented.

```
<noscript>
  <select name="sort" onchange="this.form.submit();" id="sortBy" class="form-control input-sm" aria-label="Choose a Sort Order">
    <option value="score_desc,dc_title_sort asc" selected="selected">Relevance</option>
    <option value="dcreator_sort asc,id asc">Author, A-Z</option>
    <option value="dcreator_sort desc,id asc">Author, Z-A</option>
    <option value="dc_title_sort asc,id asc">Title, A-Z</option>
    <option value="dc_title_sort desc,id asc">Title, Z-A</option>
    <option value="dyear_sort desc,id asc">Date of publication, descending</option>
    <option value="dyear_sort asc,id asc">Date of publication, ascending</option>
  </select>
</noscript>
<button type="submit" class="btn btn-default input-sm margin-top-5 pull-right">Go</button></noscript>
```

Entity Relationship Diagrams



Use Case



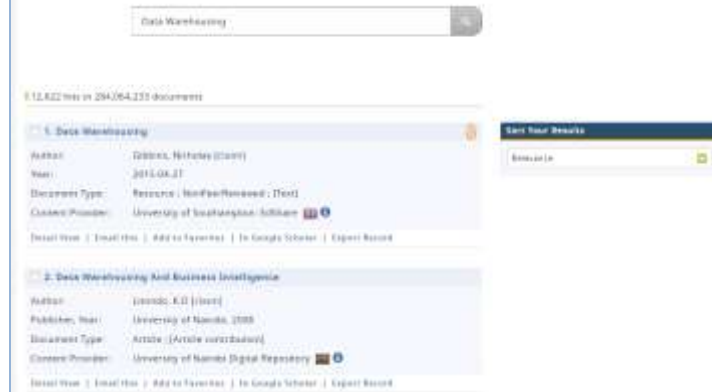
Testing

Unit testing

Category	Test Case	Description
Code	1.1	Sorting
	1.2	Favourites
	1.3	Applying filters
	1.4	Search History

Test Case 1.1 – Sorting

Test Description	The sorting system should organise search results correctly corresponding to their sort criteria
Testing Method	Enter search terms and sort the results under each of the sorting categories.
Expected Results	All results sorted as intended
Actual Results	Relevance:



Author:



Title:



Date:

	
Action Needed	None
Retest Results	

Test Case 1.2 – Favourites

Test Description	Search results should have the option to be added to an archive stored on the user's profile
Testing Method	Add articles to favourites and check if they appear on the profile.
Expected Results	Favourited articles should be saved in the correct section of the profile
Actual Results	
Action Needed	None
Retest Results	

Test Case 1.3 – Applying filters

Test Description	The user should have the option to apply search filters, which will alter the content of the results, to “filter” out those which do not fit the specifications
Testing Method	Enter searches without any filters, then use advanced search to each filter to see if results are different.
Expected Results	Search results should appear differently based on the filters used.
Actual Results	The term searched was “Digital Forensics.”

No filters:

30,539 hits in 284,064,233 documents

Book:

2,130 hits

Journal:

3,494

Audio:

28 hits

Map:

No documents found.

Image:

292 hits

Report:

300 hits

Review:

5 hits

Course Material:

6 hits

Lecture:

14 hits

Manuscript:

2 hits in 284,064,233

Patent:

5 hits

Thesis:

954 hits

Software:

13 hits

Dataset:

124 hits

Unknown:

1,868 hits

Open Access:

11,872 hits

Non-Open Access:

	654 hits
Provider, United Kingdom:	1,936 hits i
Provider, United States:	16,445 hits i
Provider, Europe:	8,238
Provider, Asia:	1,340 hits
Publication Year, 0 – 1992:	101 hits
Publication Year, 1992 – 2022:	8,089 hits
Action Needed	None
Retest Results	

Test Case 1.4 – Search History

Test Description	Timestamps of search results should be stored temporarily.
Testing Method	Enter search results, note the time, date and IP used when searching. See if it is stored correctly.
Expected Results	Accurate search history results

Actual Results

14.03.2022, 14:33	digital forensics comment:neu year (1992 TO 2022)	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:32	digital forensics comment:neu year (0 TO 1991)	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:32	digital forensics comment:neu year (* TO 1992)	Verbform search: On Boosting: Open access documents
14.03.2022, 14:32	digital forensics comment:neu	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:31	digital forensics comment:cas	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:30	digital forensics country:us	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:30	digital forensics country:uk	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:30	digital forensics doctype:f	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:29	digital forensics access:0	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:29	digital forensics access:1	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:28	digital forensics doctype:7	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:28	digital forensics doctype:6	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:28	digital forensics doctype:18*	Verbform search: On Boosting: Open access documents
14.03.2022, 14:27	digital forensics doctype:18	Additional word forms: On Boosting: Open access documents
14.03.2022, 14:27	digital forensics doctype:19	Additional word forms: On Boosting: Open access documents

14.03.2022, 14:19	digital forensics
14.03.2022, 14:18	Hard Drive forensics
14.03.2022, 14:12	Britain and free trade
14.03.2022, 14:09	Kanye West
14.03.2022, 14:09	The Beach Boys
14.03.2022, 14:09	Data Warehousing

Action Needed None

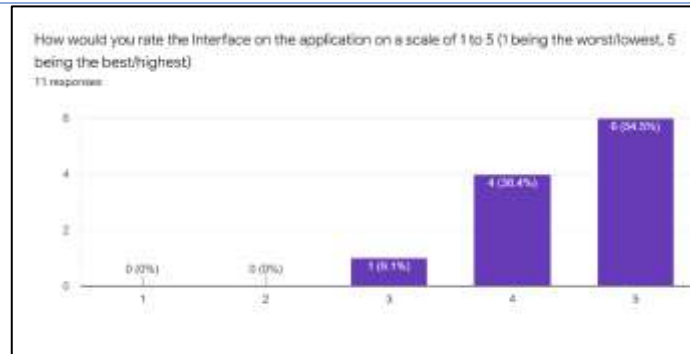
Retest Results None

Usability testing

Category	Test Case	Description
Usability to users	1.1	Interface
	1.2	Navigation
	1.3	Enjoyment

Test Case	1.1 – Interface
Test Description	Interface design should satisfy users.
Testing Method	Create and distribute survey for all, asking for a rating on the interface and for feedback
Expected Results	Interface should satisfy users.

Actual Results



Can you elaborate on the reason you rated it the way you did?

4 responses

it works, simple, clean lookin'

Simple and easy to use, understood immediately. Love the filters.

Very generic

easy to use, not very pretty

Action Needed None

Retest Results None

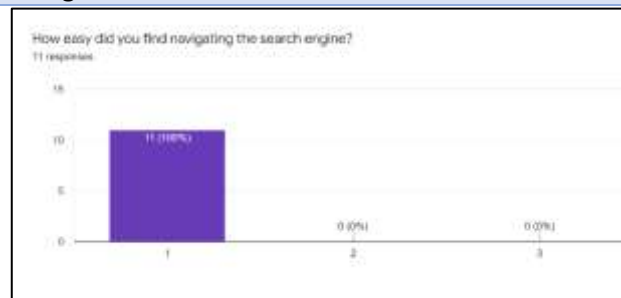
Test Case 1.2 – Navigation

Test Description Users should be able to navigate the pages with little to no assistance.

Testing Method Questionnaire to gauge individual opinions on how easy navigation was.

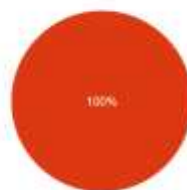
Expected Results Navigation should be of ease to all users.

Actual Results

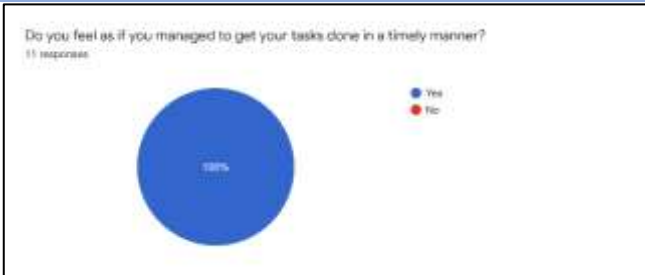


Did you need assistance in navigation at any point?

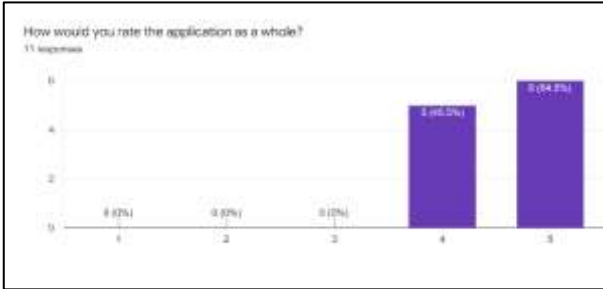
T1 responses



Yes
No

	 Do you feel as if you managed to get your tasks done in a timely manner? 11 responses 100% Yes
Action Needed	None
Retest Results	None

Test Case 1.3 – Enjoyment

Test Description	Ensure users enjoyed using the application.
Testing Method	Questionnaire, asking users to rate the application, and what they think about what they tested.
Expected Results	Satisfies users, most results being around the middle, with some more leaning positive than negative.
Actual Results	 How would you rate the application as a whole? 11 responses Can you provide any suggestions for feedback, or additional features? 2 responses idk what all the filters mean lol Works as expected, faster than scholar and my university. google scholar can fit more results on a page at once tho
Action Needed	None
Retest Results	None

Acceptance testing

Category	Test Case	Description
Research time reduction	1.1	Overall time reduction

Test Case 1.1 – Overall time reduction

Test Description	Make sure the product satisfies its goal of reducing the time it takes students to do secondary research
Testing Method	Do a study, using university students as a sample – the study should take the time of research on present applications, then have them continue research, but on this created app. Times will be recorded for both, and compared.

Expected Results The time to find valid research should be shorter on the created application here.

Actual Results First Study:

How long did it take for you to find necessary information? (If your answer is an estimation, please state so).

20 responses

- Over an hour
- 20 minutes
- 38 minutes i think, this is a previous search tho and im just guessing how long it took
- Dont know, around 40 minutes maybe. It was a good read.
- 3-10 minutes i think, estimation /shrug
- 22 minutes
- an hour
- 3-8 mins
- 5m

- 30 minutes
- 11 minutes
- 10-20 minutes
- 20-30 minute
- 7 minutes
- about a minute
- estimated half an hour
- 34m
- 6 minutes

Less than a minute.

10 minutes

Second Study:

How long did it take for you to find necessary information? (If your answer is an estimation, please state so).

11 responses

- 2 minutes
- 9 minutes
- 1< minute
- 6 minutes
- 4 minutes
- less than a minute
- around 10 to 20 minutes?
- 14 minutes
- 11 minutes

	7 minutes
Action Needed	Not as successful as hoped, but still a success in time reduction for research. Not action needed.
Retest Results	None

Time Boxes

Time boxes were created to assist with time management, based on the Gantt chart devised previously, but the time boxes were briefer and looked more into how many hours specific tasks would take to complete and how often routines in development would be completed.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Weeks	Quarter
Secondary Research	2h	1h	1h	1h	0h	0h	2h	8	Q1, Q2, Q3
Design	2h	0h	0h	0h	0h	0.5h	3h	4	Q1, Q2
Implementation	3h	1h	0h	2h	2h	0h	2h	12	Q2, Q3, Q4
Testing	1h	0h	0h	0h	2h	2h	1h	3	Q3, Q4
Feedback	1h	0h	0h	0h	2h	2h	1h	12	Q2, Q3, Q4
Coding	0.5h	2h	0h	2h	2h	0.5h	0.5h	12	Q2, Q3, Q4

The quarters are rough being general estimates of when during development these activities occurred/will occur. 'Weeks' representing how many weeks this routine will occur for the specified activity. As with the Gantt chart, this was not fixed, serving more as a fluid measurement to ensure not too much time is put into specified areas, and that a decent amount of time was still dedicated towards the project, serving to not run over schedule.

Feedback days were chosen around the most convenient times in which the researcher and participants could conduct the primary research, testing days were selected around this same justification.

Coding took place throughout in lesser amounts, as it is a tedious process, but a lengthy one, so it was chipped away slowly so to speak.

Design and implementation would eventually rely upon feedback and testing to generate new features and ideas, as planned ones would all be designed and coded – finished with – at some point.

Secondary research took place throughout although inconsistently, including what weeks it would be picked up on, secondary research served incredibly important early on for getting the project off the ground where a large bulk of the secondary sources come from, but eventually served more for picking up pieces of information regarding coding and licensing.

Evaluation

Data was drawn over the duration of this project from two places, the primary and secondary research. What can be concluded upon based on the former is that the system has satisfied its aim of reducing the time it takes for students to conduct research, as shown by the questionnaires given

to students. The secondary research gives much which can be concluded on regarding the quality of what is presently existing the way of scholarly search engines, which served the basis for what functions the system created here would complete, this is detailed more in the literature review.

The literature mostly critiques search engines such as Google Scholar using a tone which implies there are methods to fixing the issues found, if this implication was used as a definitive conclusion from the primary research, then it could be stated logically that this system did backup the literature review, as it displayed that a lot of the pitfalls discovered can mended.

The aims and objectives were mostly met, the system was created successfully and through user testing it was found to not have significant error and was also found to have satisfied its goals – as previously stated. Investigations into existing scholarly search engines was conducted with success, and their quality in providing haste in research for students was investigated via the participants of the primary research. Little constructive feedback was given by participants which could be worked with however for the system, so this objective went unsatisfied. The second aim's objectives were not entirely satisfied, an IDE was chosen (Brackets), and the system was presentable in the end as seen by the testing, but the system was not given online functionality, as it was deemed too time consuming.

The results found from the research, development and deployment of the system were all expected, the system worked fine, satisfying the user base and fulfilling its aim. There are some gripes however, the unconstructive feedback was unexpected, which had the consequence of perhaps the system not implementing original ideas or fixing existing ones and being as good as it could be. It was not expected that nearly half of the participants of the first study would not partake in the follow-up, this reduced the reliability of the conclusions which could be drawn from the second study.

If development were to be conducted again then less effort would be put into areas which did not come to see it into deployment, on top of this more effort may be put into the appearance of the application, as it was remarked that the interface did look "plain." Primary research would be conducted using larger samples, as to ensure any follow-up studies do not have small sample sizes in case of a drop in returning participants – which is always likely.

Conclusion & Recommendations

A literature review was done, analysing the secondary sources that were found in the research of established, well-used search engines based on conducting research. Planning was done based on Google Scholar's layout and features, hoping to find improvement more on this engine over others because of its popularity and familiarity and lastly coding was done using the BaseAPI as it provided high functionality for most of the features desired for this system. Primary research saw two studies being conducted, one looking into user satisfaction with existing systems, and the second into the created system. This latter study was used heavily as evidence for the new system's success, participants of these studies being testers of the system's functionality.

Other approaches which could have been conducted in hindsight include perhaps a smaller system, which more heavily incorporates existing APIs such as the one used, and implements solely new repositories – features such as the new search algorithm and sort by system further satisfy the project's aim, but at the cost of the reliability of research, which is a fair trade off for many (if not most), but in reality a lot of time went into said venture, with it providing little benefit.

Expansion of the work done here can be done, suggestions for improvements had been listed – including an interface overhaul – but more could be investigated, with a larger development time, such as being able to fit more results onto a page, removing the option to go to a source’s original repository (which most of the time was Google Scholar) would benefit this system and give it more of a unique identity too. The interface design which was done could be described as a more primitive version of Google Scholar and that provided by the BaseAPI, while the makers of said API had stated they are fine with use of their design, it may have been beneficial to create something more appealing, rather than stripping away an already not perfect design.

Time constraints led to the abandonment of online functionality, which was one of the objectives for this project, research was done into online functionality though nonetheless, this fruitless effort put into researching took away attention from other areas. Similarly, too much research went into implementing a sorting system which turned out to be simple to create using quite simple code, but the idea of creating an original algorithm was investigated instead, wasting time. Furthermore, development time was temporarily halted when the IP range used to access the API was changed and time needed to be taken to wait for a new IP to be authorized for the API to be used again – showing the flaws in relying heavily on third parties.

Reflection

There are many lessons to take away from the experiences of this project. Firstly is that when aims and objectives are being drawn up, ensure each one is integral for fulfilling the goal of the solution being created, in this case online functionality was written up as an objective, but was never delivered on, but it was not essential to satisfying the goal of the system in the first place; nonetheless heavy amounts of planning and research was conducted into it, just for it to ultimately end up fruitless, and to take away attention from other areas. Secondly is to try and gain a larger sample for primary research, for reasons previously said. A skill that was developed upon over the course of this work include working with HTML, and to think of developing long HTML pages as the sum of smaller pieces of code which function, rather than as developing one long piece which comes together in full in the end – this change in development mindset makes the task of HTML development far less tedious and easier to plan, research and troubleshoot.

A lot of the theory used for this project gave off a different impression of how development would be, portraying the impression that it would be a lot more complex and server-based than it would end up. In practice a lot of what was done was a matter of advancing already existing HTML code and CSS for interfaces, and some small snippets for changing the search system. Adding new repositories was simple enough, albeit tedious, with each repository needing to be added individually. Part of this ease came from using the BASE API already existing to provide the framework for the webpage(s), then just advancing it from there using what was learnt from research.

References

1. Google. Google Scholar Help [Internet]. *Scholar.google.com*. 2021 [cited 30 November 2021]. Available from: <https://scholar.google.com/intl/en/scholar/inclusion.html#crawl>
2. E. Falagas M, I. Pitsoun E, A. Malietzis G, Pappas G. *Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses* [Internet]. faseb.onlinelibrary.wiley. 2021 [cited 30 November 2021]. Available from: <https://faseb.onlinelibrary.wiley.com/doi/10.1096/fj.07-9492LSF>
3. Beall J. *Google Scholar is Filled with Junk Science* [Internet]. scholarlyoa.com. 2021 [cited 30 November 2021]. Available from:

<https://web.archive.org/web/20141107175135/http://scholarlyoa.com/2014/11/04/google-scholar-is-filled-with-junk-science/>

4. Beel J, Gipp B. *Google Scholar's Ranking Algorithm: An Introductory Overview* [Internet]. sciplore. 2021 [cited 30 November 2021]. Available from: https://web.archive.org/web/20181215175040/http://www.sciplore.org/publications/2009-Google_Scholar's_Ranking_Algorithm_-_An_Introductory_Overview_-_preprint.pdf
5. Carlson S. *Challenging Google, Microsoft Unveils a Search Tool for Scholarly Articles* [Internet]. Chronicle of Higher Education. 2021 [cited 6 December 2021]. Available from: <https://eric.ed.gov/?id=EJ773667>
6. Harzing AW. Microsoft Academic (Search): a Phoenix arisen from the ashes?. *Scientometrics*. 2016 Sep;108(3):1637-47. [cited 6 December 2021]. Available from: <https://link.springer.com/article/10.1007/s11192-016-2026-y>
7. Orduña-Malea E, Martín-Martín A, M. Ayllon J, Delgado López-Cózar E. *The silent fading of an academic search engine: the case of Microsoft Academic Search* | Emerald Insight [Internet]. Emerald.com. 2014 [cited 6 December 2021]. Available from: <https://www.emerald.com/insight/content/doi/10.1108/OIR-07-2014-0169/full/html>
8. B. Atkinson A, Brandolini A. *Promise and Pitfalls in the Use of "Secondary" Data-Sets: Income Inequality in OECD Countries As a Case Study* [Internet]. American Economic Association. 2001 [cited 8 December 2021]. Available from: <https://www.aeaweb.org/articles?id=10.1257/jel.39.3.771>
9. W Stewart D. *Secondary Research: Information Sources and Methods* [Internet]. Google Books. 2021 [cited 8 December 2021]. Available from: https://books.google.co.uk/books?hl=en&lr=&id=Oe3MrNsOjkkC&oi=fnd&pg=PP11&dq=secondary+research&ots=D1W5oJ1oz9&sig=sGCaLJ6OwB-ZmauXespYTKvmbAY&redir_esc=y#v=onepage&q=secondary%20research&f=false
10. Hox J, Boeije H. Data Collection, *Primary vs. Secondary* [Internet]. 2005 [cited 8 December 2021]. Available from: <http://dspace.library.uu.nl>
11. Rowley J. *Designing and Using Research Questionnaires* [Internet]. E-space.mmu.ac.uk. 2021 [cited 8 December 2021]. Available from: <https://e-space.mmu.ac.uk/579515/1/Designing%20and%20using%20Research%20QuestionnairesREV18042013.pdf>
12. P. N. Johnson-Laird. *DEDUCTIVE REASONING* [Internet]. Media.gradebuddy.com. 199 [cited 8 December 2021]. Available from: <https://media.gradebuddy.com/documents/281093/43550409-e1b3-477e-bfaf-c9292cd7272f.pdf>
13. GALLAIRE H, MINKER J, NICOLAS J. *Logic and Databases: A Deductive Approach* [Internet]. Citeseerx.ist.psu.edu. 1984 [cited 8 December 2021]. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.5645&rep=rep1&type=pdf>
14. J.H Bridges. *A General View of Positivism* [Internet]. Google Books. 2015 [cited 9 December 2021]. Available from: https://books.google.co.uk/books?hl=en&lr=&id=GDAKAQAAMAAJ&oi=fnd&pg=PR1&dq=positivism&ots=4TH9SGEupZ&sig=a14bidfElRg6nOkOAITlaQKH95Y&redir_esc=y#v=onepage&q=positivism&f=false
15. Flood A. *Understanding phenomenology* [Internet]. journals.rcni. 2010 [cited 9 December 2021]. Available from: <https://journals.rcni.com/nurse-researcher/understanding-phenomenology-nr2010.01.17.2.7.c7457>
16. Pieper D. *[InetBib] BASE Migration* [Internet]. 2011 [cited 14 December 2021]. Available from: <https://inetbib.de/listenarchiv/msg44882.html>
17. Bauer K, Bakkalbasi N. *An Examination of Citation Counts in a New Scholarly Communication Environment* [Internet]. Dlib.org. 2005 [cited 21 December 2021]. Available from: <http://dlib.org/dlib/september05/bauer/09bauer.html>
18. Price G. Content from *Bielefeld University's BASE Database Now Searchable in EBSCO Discovery Service* [Internet]. LJ infoDOCKET. 2015 [cited 21 December 2021]. Available from: <https://www.infodocket.com/2015/12/07/content-from-bielefeld-universitys-base-database-now-searchable-in-ebSCO-discovery-service/>
19. Alqassem I, Svetinovic D. *A Taxonomy of Security and Privacy Requirements for the Internet of Things (IoT)* [Internet]. Tarjomefa.com. 2021 [cited 21 December 2021]. Available from: <http://tarjomefa.com/wp-content/uploads/2018/10/9347-English-TarjomeFa.pdf>

20. M Pennacchiotti, Popescu A. *Democrats, republicans and starbucks aficionados* | Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining [Internet]. Dl.acm.org. 2021 [cited 21 December 2021]. Available from: <https://dl.acm.org/doi/10.1145/2020408.2020477>
21. *SAMPLE EVALUATION CRITERIA AND RATING SCALE* [Internet]. Energy.gov. 2022 [cited 26 February 2022]. Available from: <https://www.energy.gov/sites/prod/files/2017/04/f34/ATTACHMENT%206%20Sample%20Eval%20Criteria%20and%20Rating%20Plan%20Apr2017.pdf>
22. Nidhra S, Dondeti J. *Black box and white box testing techniques-a literature review. International Journal of Embedded Systems and Applications (IJESA)*. 2012 Jun 30;2(2):29-50. [cited 26 February 2022]. Available from: https://www.researchgate.net/profile/S-Nidhra/publication/276198111_Black_Box_and_White_Box_Testing_Techniques_-_A_Literature_Review/links/570e313f08ae2b772e46aa40/Black-Box-and-White-Box-Testing-Techniques-A-Literature-Review.pdf
23. Khan ME, Khan F. *A comparative study of white box, black box and grey box testing techniques. Int. J. Adv. Comput. Sci. Appl.* 2012 Jun;3(6). [cited 26 February 2022]. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.1887&rep=rep1&type=pdf#page=22>
24. Fritzsche G. *Understanding windows 10*. In: Proceedings of the 2015 ACM SIGUCCS Annual Conference 2015 Nov 9 (pp. 75-78). [cited 27 February 2022]. Available from: <https://dl.acm.org/doi/abs/10.1145/2815546.2815577>
25. Almalki S. *Integrating Quantitative and Qualitative Data in Mixed Methods Research--Challenges and Benefits*. Journal of education and learning. 2016;5(3):288-96. [cited 1 March 2022]. Available from: <https://eric.ed.gov/?id=EJ1110464>
26. Kaplan B, Duchon D. *Combining qualitative and quantitative methods in information systems research: a case study*. MIS quarterly. 1988 Dec 1:571-86. [cited 1 March 2022]. Available from: <https://www.jstor.org/stable/249133>
27. Adamson J, Gooberman-Hill R, Woolhead G, Donovan J. *'Questerviews': using questionnaires in qualitative interviews as a method of integrating qualitative and quantitative health services research*. Journal of Health Services Research & Policy. 2004 Jul 1;9(3):139-45. [cited 1 March 2022]. Available from: <https://journals.sagepub.com/doi/abs/10.1258/1355819041403268>
28. Jalote P, Palit A, Kurien P, Peethamber VT. *Timeboxing: a process model for iterative software development*. Journal of Systems and Software. 2004 Feb 1;70(1-2):117-27. [cited 1 March 2022]. Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0164121203000104>
29. Coyle S, Conboy K. *A study of risk management in DSDM*. In: International Conference on Agile Processes and Extreme Programming in Software Engineering 2009 May 25 (pp. 142-148). Springer, Berlin, Heidelberg. [cited 1 March 2022]. Available from: https://link.springer.com/chapter/10.1007/978-3-642-01853-4_18
30. Onofri S, Spagnuolo C. *Project Time Management: gestione per fasi e timeboxing*. Project Time Management: gestione per fasi e timeboxing. 2014:43-6. [cited 1 March 2022]. Available from: <https://www.torrossa.com/en/resources/an/2934760>
31. Chau M, Wong CH. *Designing the user interface and functions of a search engine development tool*. Decision Support Systems. 2010 Jan 1;48(2):369-82. [cited 9 March 2022]. Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0167923609002103>
32. Jin BS, Ji YG. *A Study about Search Engine Interface Design including User's Search Goal*. The Journal of Society for e-Business Studies. 2008;13(4):111-24. [cited 9 March 2022]. Available from: <https://www.koreascience.or.kr/article/JAKO200801440596547.page>
33. Palanisamy R, Liu Y. *User Search Satisfaction In Search Engine Optimization: An Empirical Analysis*. Journal of Services Research. 2018 Oct 1;18(2):83-120. [Cited 9 March 2022] Available from: Palanisamy R, Liu Y. User Search Satisfaction In Search Engine Optimization: An Empirical Analysis. Journal of Services Research. 2018 Oct 1;18(2):83-120.
34. Jacso P. *As we may search—comparison of major features of the Web of Science, Scopus, and Google Scholar citation-based and citation-enhanced databases*. Current science. 2005 Nov 10;89(9):1537-47. [Cited 9 March 2022] Available from: <https://www.jstor.org/stable/24110924>
35. Jacsó P. *Google Scholar Metrics for Publications: The software and content features of a new open access bibliometric service*. Online information review. 2012 Aug 3. [Cited 9 March 2022]

- Available from:
<https://www.emerald.com/insight/content/doi/10.1108/14684521211254121/full/html>
36. Rani M. Information seeking tools for specific information seekers in digital age. *International Journal in Management & Social Science*. 2016;4(1):143-9. [Cited 9 March 2022]. Available from: https://www.researchgate.net/profile/Mamta-Arora-4/publication/293823093_Information_seeking_tools_for_specific_information_seekers_in_digital_age/links/5746bd5108aea45ee8562028/Information-seeking-tools-for-specific-information-seekers-in-digital-age.pdf
 37. Crockford D. *JavaScript: The Good Parts: The Good Parts*. " O'Reilly Media, Inc."; 2008 May 8. Available from: https://books.google.co.uk/books?hl=en&lr=&id=PXa2bby0oQ0C&oi=fnd&pg=PR7&dq=javascript&ots=HLpop5q5hH&sig=djNsGGeO8ymSyh4ECiRv9wLRZG0&redir_esc=y#v=onepage&q=javascript&f=false
 38. Richards G, Lebresne S, Burg B, Vitek J. *An analysis of the dynamic behavior of JavaScript programs*. In *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation 2010 Jun 5* (pp. 1-12). [Cited 15 March 2022]. Available from:
 39. Cantelon M, Harter M, Holowaychuk TJ, Rajlich N. *Node.js in Action*. Greenwich: Manning; 2014. Available from: <https://dl.acm.org/doi/abs/10.1145/1806596.1806598>
 40. Tilkov S, Vinoski S. Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*. 2010 Nov 1;14(6):80-3. [Cited 15 March 2022]. Available from: <https://ieeexplore.ieee.org/abstract/document/5617064>
 41. Milicevic A, Jackson D, Gligoric M, Marinov D. *Model-based, event-driven programming paradigm for interactive web applications*. In *Proceedings of the 2013 ACM international symposium on New ideas, new paradigms, and reflections on programming & software 2013 Oct 29* (pp. 17-36). [Cited 15 March 2022]. Available from: <https://dl.acm.org/doi/abs/10.1145/2509578.2509588>
 42. F. Summann, R. Mitrenga. *BASE HTTP Interface* [Internet]. Base-search.net. [cited 15 March 2022]. Available from: <https://api.base-search.net/cgi-bin/BaseHttpSearchInterface.fcgi?func=ListRepositories&coll=ceu>
 43. López-Cózar ED, Orduña-Malea E, Martín-Martín A, Ayllón JM. *Google Scholar: the big data bibliographic tool*. In *Research Analytics 2017 Oct 25* (pp. 59-80). Auerbach Publications. [cited 15 March 2022]. Available from: <https://www.taylorfrancis.com/chapters/edit/10.1201/9781315155890-4/google-scholar-big-data-bibliographic-tool-emilio-delgado-lópez-cózar-enrique-orduna-malea-alberto-martín-martín-juan-ayllón>
 44. Bastopcu M, Ulukus S. *Who Should Google Scholar Update More Often?* [Internet]. Arxiv.org. 2022 [cited 15 March 2022]. Available from: <https://arxiv.org/pdf/2001.11500.pdf>
 45. Haddaway NR, Collins AM, Coughlin D, Kirk S. *The role of Google Scholar in evidence reviews and its applicability to grey literature searching*. *PloS one*. 2015 Sep 17;10(9):e0138237. [cited 15 March 2022]. Available from: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0138237><https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0138237>
 46. Wang K, Shen Z, Huang C, Wu CH, Dong Y, Kanakia A. *Microsoft academic graph: When experts are not enough*. *Quantitative Science Studies*. 2020 Feb 1;1(1):396-413. [cited 15 March 2022]. Available from: <https://direct.mit.edu/qss/article/1/1/396/15572/Microsoft-Academic-Graph-When-experts-are-not>
 47. Jacsó P. *The pros and cons of Microsoft Academic Search from a bibliometric perspective*. *Online Information Review*. 2011 Nov 29. [cited 15 March 2022]. Available from:
 48. Orduña-Malea E, Delgado López-Cózar E. *The dark side of Open Access in Google and Google Scholar: the case of Latin-American repositories*. *Scientometrics*. 2015 Jan;102(1):829-46. [cited 15 March 2022]. Available from: <https://www.emerald.com/insight/content/doi/10.1108/14684521111210788/full/html?fullSc=1&fullSc=1&mbSc=1>
 49. López-Cózar ED, Robinson-García N. *Repositories in Google Scholar Metrics or what is this document type doing in a place as such?*. [cited 15 March 2022]. Available from: <https://digital.csic.es/handle/10261/174568>

50. Beazley MR. *EPrints institutional repository software: A review*. Partnership: the Canadian Journal of Library and Information Practice and Research. 2010;5(2). [cited 15 March 2022]. Available from: <https://journal.lib.uoguelph.ca/index.php/perj/article/view/1234>
51. Smith M, Barton M, Bass M, Branschovsky M, McClellan G, Stuve D, Tansley R, Walker JH. *DSpace: An open source dynamic digital repository*. [cited 15 March 2022]. Available from:
52. Yang L. *Making search engines notice: An exploratory study on discoverability of DSpace metadata and PDF files*. Journal of Web Librarianship. 2016 Jul 2;10(3):147-60. [cited 15 March 2022]. Available from: <https://www.tandfonline.com/doi/abs/10.1080/19322909.2016.1172539>
53. Pieper D, Summann F. *Bielefeld Academic Search Engine (BASE): an end-user oriented institutional repository search service*. Library Hi Tech. 2006 Oct 1. [cited 15 March 2022]. Available from: <https://www.emerald.com/insight/content/doi/10.1108/07378830610715473/full/html>
54. Shao H, Sun D, Wu J, Zhang Z, Zhang A, Yao S, Liu S, Wang T, Zhang C, Abdelzaher T. *paper2repo: GitHub repository recommendation for academic papers*. In Proceedings of The Web Conference 2020 2020 Apr 20 (pp. 629-639). [cited 15 March 2022]. Available from: <https://dl.acm.org/doi/abs/10.1145/3366423.3380145>
55. Haunschild R, Hug SE, Brändle MP, Bornmann L. *The number of linked references of publications in Microsoft Academic in comparison with the Web of Science*. Scientometrics. 2018 Jan;114(1):367-70. [cited 15 March 2022]. Available from: <https://link.springer.com/article/10.1007/s11192-017-2567-8>
56. AGUILLO I, BLEDA A, PÉREZ ÁLVAREZ S, ÁLVAREZ F, LÓPEZ BENITO C. *Ranking Repositories of the Web* [Internet]. <https://www.csic.es/es>. [cited 15 March 2022]. Available from: <https://repositories.webometrics.info/en/node/37>
57. Kurtz M. *Dublin Core, DSpace, and a brief analysis of three university repositories*. Information technology and libraries. 2010 Mar 1;29(1):40-6. [cited 16 March 2022]. Available from: <https://ejournals.bc.edu/index.php/ital/article/view/3157>

Bibliography

- Hagedorn G, Mietchen D, Morris RA, Agosti D, Penev L, Berendsohn WG, Hobern D. *Creative Commons licenses and the non-commercial condition: Implications for the re-use of biodiversity information*. ZooKeys. 2011(150):127. [Accessed 9 March 2022] Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3234435/>
- Cabrio E, Palmero Aprosio A, Villata S. *These are your rights*. In European Semantic Web Conference 2014 May 25 (pp. 255-269). Springer, Cham. [Accessed 9 March 2022] Available from: https://link.springer.com/chapter/10.1007/978-3-319-07443-6_18

Appendix

1. System Testing.

Category	Test Case	Description
Functionality	1.1	Validation

Test Case	1.1 – Validation
Test Description	Test HTML on page to check for errors and warnings.
Testing Method	Use datayze's site-validator to run over the HTML to check for errors. • Repeat for all pages.
Expected Results	No major warnings or errors which will inhibit the site's functionality

Actual Results

No "li" element in scope but a "li" end tag seen.

From line 1181, column 29 to line 1181, column 33

Code Extract:

```
</li>↵
```

This error appeared on all pages, however upon closer inspection, it appears there it is attached to an opening list tag.

```
<li role="heading"><a href="search.html">Basic search</a></li>
```

Action Needed

Validate code again, try rewriting line from scratch to ensure no human error.

Retest Results

Every time validation was checked for each page, the same error appeared. Upon testing the actual pages, themselves, this specific line functioned as intended.

No "li" element in scope but a "li" end tag seen.

From line 1181, column 29 to line 1181, column 33

Code Extract:

```
</li>↵
```

Primary Research Surveys:

- https://docs.google.com/forms/d/e/1FAIpQLSeXyQv0w5OIdkuRxqcU3dnsQJnep9mFbi-XlopqP2G63QSjKA/viewform?usp=sf_link
- https://docs.google.com/forms/d/e/1FAIpQLSepzqwK5EpuNX0KiJldCt7-ge7QmhEROzoIMYIGJCz64IM-yg/viewform?usp=sf_link

7989 words