UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2020

Title: Developing a Low-Cost National Heritage Web Portal built from Metadata Aggregation

Author: Toshka Coleman

Project Abbreviation: HERIPORT

Supervisor(s): Hussein Suleman

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	10
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	
System Development and Implementation	*0*	*20*	20
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Developing a Low-Cost National Heritage Web Portal built from Metadata Aggregation

Web Portal Component

Toshka Coleman
Department of Computer Science
University of Cape Town
Cape Town, South Africa
clmtos001@myuct.ac.za

ABSTRACT

Digital heritage portals have recently been introduced with the purpose of preserving and accessing historical information. Heritage portals that encompass archives from various external sources have been implemented internationally for projects such as the Europeana project, but this has not yet been developed for South African heritage archives. This project therefore aimed to create a low-cost local heritage Web portal encompassing multiple local heritage archives in a central system. Metadata harvesting via the OAI-PMH was used to harvest metadata from multiple archives that was displayed on the Web portal. The Web portal was required to facilitate search and browse services, and display metadata of archive items. Development was conducted using an Agile Methodology through iterative implementation, whereby four iterations were completed to enforce iterative testing and flexibility, and to minimize risks. Testing methods included Correctness, Integration, User Acceptance, Usability and User Satisfaction testing. The results of the tests indicated that the requirements of the Web portal were met successfully. Moreover, the feasibility of implementing a heritage Web portal built upon metadata aggregation was demonstrated and accomplished.

CCS Concepts

Heritage Portal, Metadata Harvesting, Digital Library Systems.

Keywords

Web portal, heritage portal, low-cost, search, browse

1. INTRODUCTION

1.1. Project Context

South Africa possesses a unique heritage with renowned cultural significance, characterized by multiculturalism, post-colonialism, and the Apartheid era [12]. This heritage is the reason for the diverse culture seen in South Africa presently.

One's knowledge of this heritage thus aids their understanding of our current society, its politics, historical monuments, and distinctive art [12]. For this reason, the preservation, as well as the accessibility of national heritage content is essential for learning about and conducting research on South Africa's unique heritage. Access to such content has increased with the growing popularity of digital libraries and digital archive systems that collectively store content online [8].

Using this technology, South African memory institutions have made various attempts to develop digital libraries for heritage content. The Bleek and Lloyd Collection [14] is an example of such an archive, but only contains a single domain of heritage. Europeana [3] is an example of a high-cost heritage portal in Europe, encompassing a vast assemblage of European archives in a central system. The South African National ETD Portal [17] is an example of a low-cost, local system developed for collating South African theses and dissertations resources.

The aforementioned and current related systems provide dedicated heritage content that are not linked to each other or commonly accessible. Therefore, there has not yet been a central system developed that provides cross-archive discoverability services across multiple domains of local heritage archives. The introduction of a central heritage system would allow historians, researchers, students and whoever else may be interested in South African heritage to have access to a wide variety of focused historical documents. This is currently only possible through tools such as Google Search, proving impracticable due to its lack of and limitation in the specialization of focused, local heritage documents.

1.2. Project Aims

This project aimed to develop a national heritage Web portal, HERIPORT, which applies practices in low-resource heritage archive systems to harvest metadata from multiple local heritage archives. It also aimed to provide cross-archive discovery services for end-users, such as searching and browsing archives through a central Web portal. Thereby, we anticipate that the introduction of a national heritage metadata aggregator system will allow users such as academics and anyone interested in South African heritage to access and search through a range of historical resources via a single Web portal without experiencing the above-mentioned limitations linked to the usage of the current software available.

We have procured Extensible Markup Language (XML) metadata

from The Five Hundred Year Archive (UCT History)[1], Bleek and Lloyd Collection (UCT Fine Art)[2], and Metsemegologolo (Wits/UP).

As a low-resource South African national heritage metadata aggregator has not yet been developed, and as we build on the philosophies of current low-resource software tools, this project would contribute significantly to the research community.

The overall objective was, therefore, to demonstrate the feasibility of this project as proof of concept by implementing a South African National Heritage Portal. System development is, however, discussed using the software engineering approach, as the project's primary components are each development projects. This paper will specifically discuss the implementation of the Web portal component including its implementation of the search, browse, and carousel features.

1.3. HERIPORT System Architecture

HERIPORT's system comprises three major components as seen in Figure 1 below. The layered architecture and the communication between the three components are shown in their order of interaction: from the Data Providers' Interfaces, to the Harvester and finally the Web portal.

The Web portal Component includes the Standard User Interface and the client services – search and browse. The search and browse services allow end-users to access the metadata harvested within the Harvester component. This metadata had previously been sent by and exposed through the Data Provider Interfaces component. Further detail on the Web portal component's architecture is presented in Section 3.2.

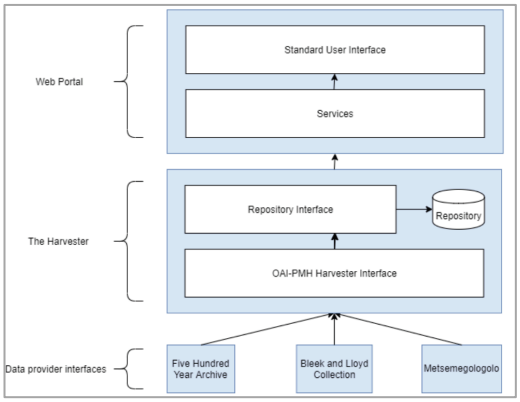

Figure 1: HERIPORT System Architecture

1.4. Report Structure

Firstly, background and related research on the topic is discussed. The design and implementation of the Web portal follows, including requirements gathering and the Agile development process and implementation. The Integration, Correctness, User Acceptance, Usability and User Satisfaction evaluation processes are then described. The results and discussion of these evaluations follow. Conclusions are then presented and, finally, future work as an extension of the project is described.

[1] http://emandulo.apc.uct.ac.za/

2. BACKGROUND AND RELATED WORK

2.1. Metadata Harvesting Overview

The Web portal is a user-centred interface that displays content gathered from the heritage portal's data providers which has been collated onto a central domain using metadata harvesting, implemented via the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) protocol [20].

Metadata harvesting facilitates the identification and collation of resources based on their associated metadata, gathering metadata from numerous source archives or repositories and aggregating them to a single database destination [20]. In recent years, the OAI-PMH [9] has become the standard protocol used for metadata harvesting. The OAI-PMH implements unqualified Dublin Core (DC) [5], a metadata schema consisting of 15 core fields, as the metadata standard in order to maintain interoperability. Consequently, Data Providers providing metadata have to ensure that their records are in the same format defined in XML [23]. The metadata on the Web portal is thus displayed in DC format.

The OAI-PMH contain requests that are sent and defined as HTTP (Hypertext Transfer Protocol) requests using GET and POST operations - the repositories are thus required to support these methods. The requests are mapped via a base URL which outlines the network host and port of the HTTP server that is the repository [23]. Thereafter, the responses to OAI-PMH requests are represented in XML and are encoded using UTF-8 representation of Unicode with character references instead of entity references. This is used in order for XML responses to be independent of external entity declarations [9]. These XML records are inserted into the repository that the Web portal extracts its content from.

2.2. Search and Browse Services in Digital Libraries

2.2.1. Background

A Digital Library System (DLS) is an online system containing a collection of electronic documents [21]. Searching within an DLS involves inputting items related to an identifier or field and retrieving relevant content from various remote databases containing digital objects. These databases may include metadata for relevant objects or entire objects such as an article, book or video [18].

Browsing is an information-seeking activity relatively less directed than searching as the resulting content is not necessarily predetermined [18] Browsing involves specifying a category to sort content by and is usually done through browsing by index such as browsing by date or author [21].

Search and browse are the core services in digital repositories and involve incorporating aspects of information retrieval [18]. The OAI-PMH implementation of search differs from Federated search [4]. In Federated search systems, search queries are performed concurrently within multiple data providers' collections. Results are subsequently returned from individual data provider's metadata collections and then aggregated together. Contrastingly, metadata is first aggregated centrally in OAI-PMH implementations, therefore search queries are performed on a central collection. In this way, search is facilitated by the search service provider as opposed to each metadata provider [4].

[2] http://pumbaa.cs.uct.ac.za/~balnew/

2.2.2. Implementation of Search Service in OAI-PMH Digital Libraries

The digital library, National Science Digital Library (NSDL) [1], implements its search service using the OAI-PMH to enforce discoverability of high-quality resources and tools related to science and mathematics [1].
In the implementation of the NSDL search service, a list of searchable items is acquired for the search operation by gathering the contents contained in the metadata repository using the OAI-PMH. Through the application of this protocol [8], the repository's contents can to be acquired initially, and the search component's indexes are updated frequently by harvesting new and modified items. The search engine interacts with the portals with the use of the Simple Digital Library Interoperability Protocol (SDLIP) [7] protocol, which specifies the methods by which queries are sent between clients and servers, as well as how results are returned. The search's results are then structured as a ranked list of items.

In the latest implementation of NSDL [1], a more high-performance technique for its search engine was implemented - a search service that uses Apache Lucene for indexing and query. Through the use of the OAI-PMH and a Metadata Repository, Lucene indexes the metadata as well as the full-text resources provided in the metadata records. In this way, the search service efficiently translates the metadata-centred data model to a resource-centred model [1].

The Illinois project, based at the University of Illinois, is another example of a Web portal that uses the OAI-PMH [4]. As with HERIPORT, it implements cross-archive discoverability services that allows users to search through aggregated cultural metadata in Dublin Core format. Implementation of the project included the creation of a SQL database, namely the University of Illinois Cultural Heritage Repository [3], and the development of a Web portal using XPAT indexing and search engine tools. The project found that the discovery of the heritage data from the Web Portal was impacted by how the Dublin Core field values were assigned, and implementation of search needed to account for this to maximise discoverability. For instance, a user may have searched for a creator's name that was assigned to the contributor field, and subsequently omitted results. This was resolved by implementing search across all elements within each record and by grouping similar elements together such as contributor and creator, and description and subject, in the search implementation. The final conclusion from the project was that the OAI-PMH Protocol is effective for implementing search and discovery services [4].

2.2.3 Implementation of Search and Browse for XML Data

A study conducted by Suleman [16] aimed to assess the performance of a search and browse system based solely within a browser and without a network connection and a software installation. Interestingly, it concluded that this system shows potential for efficiency with the use of JavaScript and pre-indexed data stored in XML files. Additionally, it displayed reasonable performance for basic operations varying from small to medium sized collection [16].

This system's implementation of the search service applies information retrieval principles and uses an extended Boolean model [22] while the browse service is supported by database formatted indices containing items that match a particular term. It includes two sets of indices that are stored as XML documents which allow them to be processed using built-in browser services. The search index for a term contains a list of identifiers of all items including the term. The browse index for a field name with a certain value includes a list of identifiers of all items in which the field has that specific value [16].

Essentially, the search system contains a Perl script that creates the indices needed for search or browse and a JavaScript file that performs search or browse operations and displays the relevant results on the web browser window. Subsequently, the relevant search and browse indices are loaded and the necessary processes are performed to produce results ordered by relevance and filtered by the particular browse fields [16].

SimplyCT [10] is a more lightweight system approach, developed with the purpose of creating a DLS using simple architecture. SimplyCT's framework is made up of a collection of archive data, and services within a hierarchical structured directory. The archive files are outlined in XML metadata files and its services adhere to a server-instance model [6].

Similarly to the previous project, indices and query operations are used for the search service, however, they are also implemented using the Xapian information retrieval library [24] as follows. When a user searches for items, a GET request is sent from the XML HTTP Request in the Python CGI scripts [6]. These CGI scripts are instances of the search service allocated to the particular archive containing the item. The search query is conducted using the shared code in the search directory which generates the paths for the archive and interfaces with Xapian [24]. Xapian manages the productions and searching operations of the indices. The XML-formatted search results are sent to the JavaScript on the client-side and is transformed into a XHTML sheet that is presented to the user [6].

3. DESIGN AND IMPLEMENTATION

3.1. Requirements Gathering

3.1.1. Data Provider Requirements Gathering Process

In order to meet the implementation expectations of our end-users and clients, requirements gathering was conducted before any development commenced. As the data providers are the clients who have invested interest in our project, we communicated this process with them. This was done by requesting the requirements from the members of the three data providers via email. In terms of the Web portal, this included a specification of what they would like to gain from the Web portal, the order of importance of the features, and which fields they would like to be able to browse by.

It was found that Search, Browse and viewing of metadata were the common top three features required from the data providers and that 'Title', 'Date', and 'Author' were the most common fields that were requested to be able to browse by. Additional features that were proposed included a language translation on the metadata records and orthographic features that could enhance the search of the database.

3.1.2. Web portal Functional Requirements

The initial functional requirements of the Web portal were:

- The development of a central heritage Web portal that is built on top of a metadata aggregator that aggregates metadata from multiple South African heritage archives.

- The Web portal provides cross-archive discovery services to end-users, such as searching and browsing archives.
- The metadata on the Web portal is displayed in Dublin Core format.

Following requirements gathering, additional specifications of the functional requirements were outlined. For the Web portal, this included:

- Prioritisation of search, browse and viewing of metadata over all features.
- The implementation of being able to by browse by "Title", "Date" and "Archive".

These functional requirements are presented in Figure 2 below. The system component represents the Apache Solr Server.

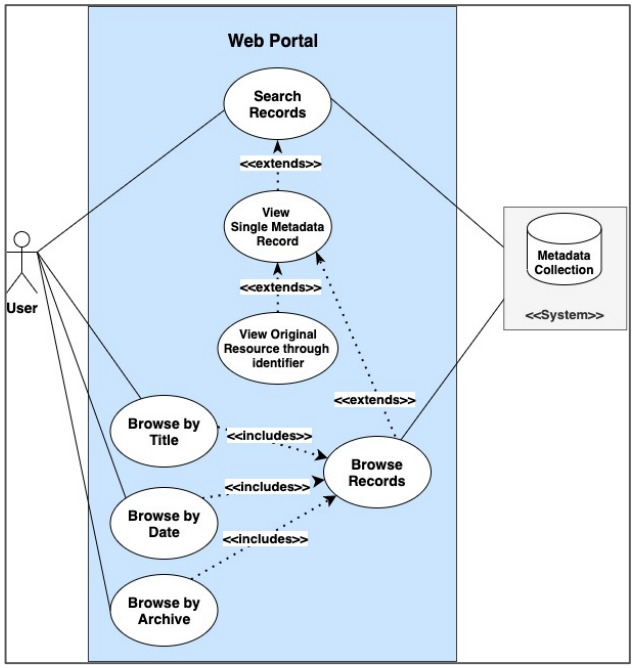

Figure 2: Use Case Diagram of Web Portal Functional Requirements

3.1.3. Web portal Non-functional Requirements

The following non-functional requirements were identified:

3.1.3.1. ***Maintainability*** – This requires that the Web portal is able to be restructured and supported over time.

3.1.3.2. ***Usability*** – This requires that the Web portal's user interface is intuitive and not difficult to use.

3.1.3.3. ***Portability*** – This requires that users are able to access the Web portal from any location.

3.1.3.4. ***Availability*** – This requires that the Web portal persistently runs and is available to users at all times.

3.1.3.5. ***Speed***– This requires that the Web portal is able to load and perform queries at an optimal speed.

3.1.3.6. ***Scalability*** – This requires that the Web portal is able to allow for and manage a large and expanding amount of data.

3.2. Web Portal Architecture

Figure 3 below presents a high-level architecture and the interactions of the Web portal, in which the Web server and a Web browser form the Web portal component.

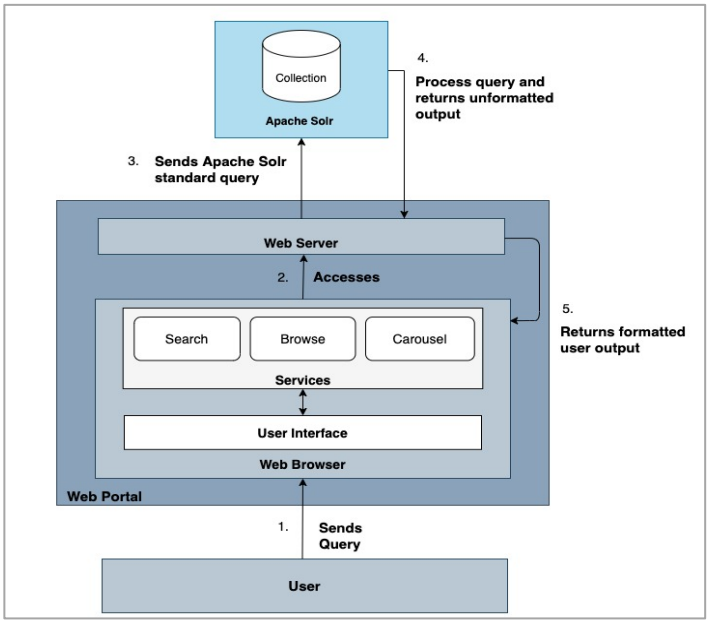

Figure 3: Web Portal UML Architecture Diagram

Following the diagram, a user sends a query on a Web browser. The query comprises interactions with the web browser including the services - search, browse and the carousel. The Web browser then accesses the Web server that the Web portal is hosted on. The Web server follows up by sending the user query in the appropriate Apache Solr standard format to the Apache Solr server hosting the metadata. Apache Solr processes the query and returns the unformatted output to the Web server. The Web server returns the user-formatted output to the user's Web Browser that is displayed on the user interface.

3.3. Agile System Development

The Agile Software Development Methodology was chosen as our methodology approach over the traditional Waterfall approach because it is more flexible in structure and thereby reduces risks [2]. As our project work has been divided, and due to the possibility of varying user requirements and adjustments, an Agile and iterative approach was the most appropriate choice of methodology to account for these factors.

Our implementation approach included four phased iterations, each of which adhered to the requirements, analysis, design, implementation, integration and test-driven development. The requirements and deliverables were defined before each iteration. Iterative communication with the data providers was conducted, and weekly meetings with our project supervisor, as well as daily meetings with all team members were held.

Iterative communication with the data providers helped ensure client satisfaction. It gave our clients the opportunity to be involved in the development process and allowed us to adapt their needs and preferences throughout the project. This communication also aided consistent refinements of our requirements, specifically for the Web portal that they would interact with. Weekly meetings with the project

supervisor ensured that the supervisor was up to date with the group's development and allowed for any issues to be raised and resolved quickly. Daily communication with the team helped the team members ensure that each of us were on track with implementation, that we were each aware of one another's progress and thereby aided integration of components.
In line with the Extreme Programming Framework, testing and refactoring of code was done at the end of each iteration [2].
For the Web portal, this involved correctness and integration testing. Iterative testing allowed for easier debugging in the smaller iterations and reduced the chance of potential risks later in the project. The advantages of the refactored code include increased comprehensibility and reusability.

3.3.1. Development Platform and Tools

3.3.1.1. Apache Solr 7.2 - Apache Solr 7.2 [19] was used to facilitate the indexing of metadata and implementation of the Search and Browse services. It enforced high scalability for the accumulated data and provided useful functionality and features for implementation such as filtered querying. The Apache Solr default schema file was edited and reconfigured to fit the metadata display requirements.
3.3.1.2. JavaScript 1.7 - JavaScript was used to implement the communication with Apache Solr using XML HTTP requests. It was also used for developing the methods and functionality behind the display of the search and browse services, navigation between pages, and for implementing the methods related to user input such as clicking on buttons and specifying keywords.
3.3.1.3. HTML - The 'Home', 'Browse', 'Search' and 'About' Web pages and their content displayed on the Web portal were created and structured using HTML.
3.3.1.4. CSS - CSS was used for the layout, styling and formatting of the HTML Web pages.
3.3.1.5. Bootstrap 4 - Bootstrap 4 [13] provided templates that enhanced the design of the Web pages, giving the website a more professional interface.
3.3.1.6. Git - Git was used for version control of code, allowing the code to be reverted to specific versions in the event of errors or mistakes. It allowed changes of source code to be tracked and facilitated collaboration of data and code with team members. It also implemented local development by facilitating pushing code developments to the server.

3.3.2 Iterative Development

In this section, the iterations are described with a high-level outline of the development process.

3.3.2.1. Iteration 1: Implement Web portal pages on server and initiate Search feature development

In the first iteration of the project development, a UCT server was used for hosting the Web portal through the use of a Virtual Private Network (VPN). A testing collection was created on the Apache Solr server and an XML file was added to the collection for testing. The items of the collection were indexed, and terminal commands were used for testing the indexing and retrieval of search items. Following the Apache Solr command testing, the HTML files for the Home, Browse, Search and About Web pages were created and the navigation between these Web pages was implemented. The interface formatting of the Web pages was developed using CSS and improved on by using Bootstrap for more professional styling.

The development of the search feature was initiated by setting up a connection to the Apache Solr collection using XML HTTP requests in JavaScript within the Search HTML file. This connection enabled JSON formatted responses to be retrieved based on the search item. Correctness testing was done by checking the output of the JSON response against expected metadata responses for multiple search items.
Initial requirements gathering was done in Iteration 1. This was described in Section 3.1.1.

3.3.2.2. Iteration 2: Refine Search feature and implement Browse Feature

The XML metadata files from the data providers were received and added to the Apache Solr server, thereby replacing the testing metadata. The search feature was subsequently refined to allow specific fields to be displayed in readable text format.

The browse feature was developed by firstly enabling all records in the collection to be displayed and thereafter by specifying which of the fields would be sorted on in either ascending or descending order. Correctness testing was done for the refined search by checking the output of the text format response against expected metadata responses for different field types. Error and bugs such as endless number of previous and next pages were fixed.

3.3.2.3. Iteration 3: Refine Search feature and Implement carousel and Integration

Following the initial testing of search, the search feature was refined to provide for searching of fields with multiple values e.g. contributor field values containing multiple contributor names. The Apache Solr Schema configuration file was edited to account for the DC formatted fields and such multi-valued fields.

Correctness testing was done by checking the output of the text response against expected metadata responses for multi-valued field types, and testing that the browse feature worked correctly.

The carousel was implemented whereby the home page displays the latest resources. This was done by sorting the items by date in descending order and limiting the number of results to 15. Correctness testing was done by checking the output of the text response against expected metadata responses for sorting on browsing fields.

3.3.2.4. Iteration 4: Testing and Refinements after testing

User testing was conducted in which 5 representatives from the Data Providers and 5 UCT computer science students completed a Task Evaluation as well as a standard User Satisfaction questionnaire - USE [11]. Their consent was then given via a consent form. Their feedback was gathered and used for additional refinements to the Web portal. These refinements are discussed in Section 5.1.2.

3.4. Final Web portal Features and Implementation

3.4.1. Home Page

3.4.1.1. Home Page Design

The first Web page that is displayed when HERIPORT Web portal loads is the Home page. It introduces the user to the website with a brief description of the website, and also displays the carousel, which loads the latest resources for the user. As seen in Figure 4 below, the Home page includes links to the other Web pages - Search, Browse and About, within a navigation header.

3.4.1.2 Carousel Implementation

The Carousel uses the same Browse service implementation discussed in 3.3.3.2. In this case, the field that the parameter uses to sort the data is 'dc_modDate' and a limit parameter of 15 is implemented and requested from the Apache Solr RequestHandler.

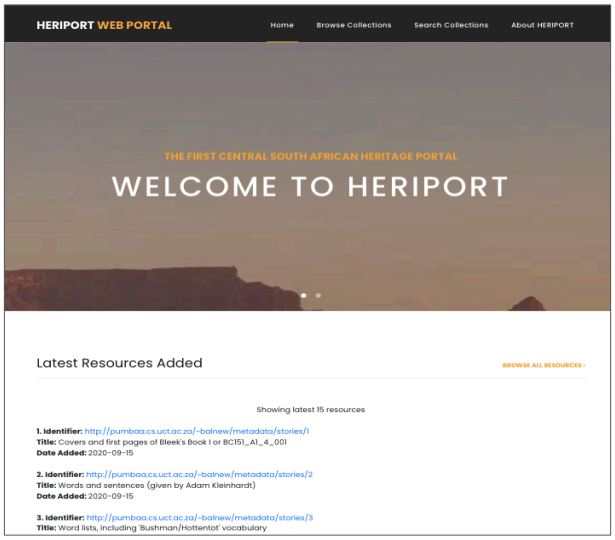

Figure 4: Home Page with Carousel

3.4.2. Search Page

3.4.2.1 Search Page Design

When "Search Collections" is clicked on in the navigation header, the Search page is loaded. As seen in Figure 5, the search feature allows the user to enter an item from any metadata field, producing results whereby this item appears in any of the metadata fields. The results display as seen in Figure 6, and the user is able to click on the title of any result to view the record's metadata as seen in Figure 7. 30 results are shown at a time and the user is able to use the "Previous" and "Next" buttons to navigate the results. This implementation fulfils Schneiderman's principle [15] in which a user is given an overview of data and can access further information by clicking on the record.

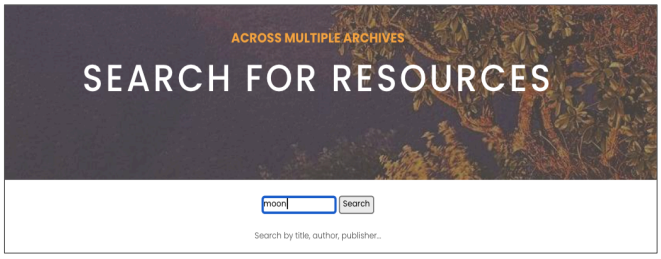

Figure 5: Search Page with search item "moon" entered

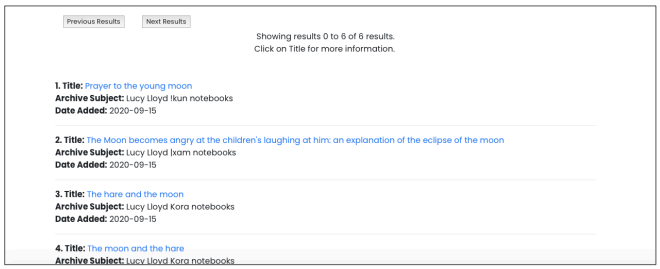

Figure 6: Search results displayed with "moon" as search item

Figure 7: Metadata of selected record displayed

3.4.2.2 Search Service Implementation

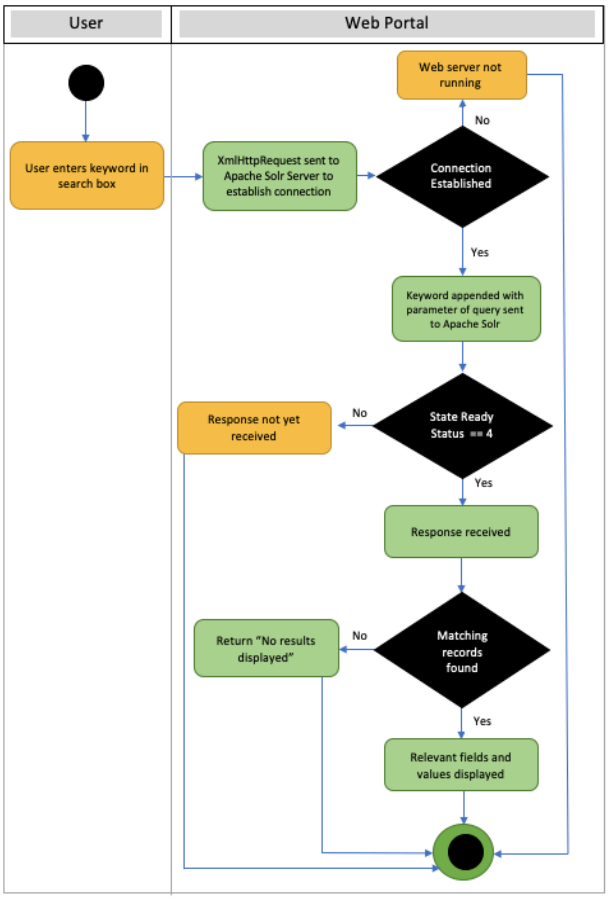

Figure 8: UML Activity Diagram of Search Flow Events

The functionality for the search service was developed in JavaScript. In the search() method, a connection to the Apache Solr collection was established by instantiating an XmlHttpRequest object. XmlHttpRequests are used to request data and make HTTP requests from Web servers: in this case, a connection to the sever port hosting the collection (the host URL) was requested. When the search keyword item is entered and the search button is clicked, the search() method is invoked. A query string is generated by retrieving the user input keyword(s) and appending it as part of a Solr query parameter that is sent as a query to Apache Solr. This query is placed as the

search value of either of the fields in the collection, depending on which field it occurred in, by using Boolean logic operators (if it does appear in any field in the collection). Apache Solr then processes the query through its Request Handler.

The event handler – onreadyStateChange - defines a function that is executed when the readyState status changes. When the readyState's status is 4, this means that the request has been processed and a response is ready. In this event, the response from the request to the host URL is sent as a parameter to the displayResponse() method. This response is in JSON format.
The response includes various fields and respective values related to the response content. 'numFound' is first retrieved to return and display to the user the number of records that include the query item within them. Subsequently, the Dublin Core fields 'title', 'modDate', 'subject, etc. are also returned and displayed as introductory metadata. A for loop is used to iterate through all relevant results and retrieves these field values from each record. The 'title' value is a hyperlink that invokes the function viewMetadata() when clicked on, which returns and displays the rest of the DC fields as well for the user. A high-level flow of events of the above process is represented visually in Figure 8 above. The sequence of events and methods invoked are also demonstrated in the Sequence Diagram in Supplementary Information.

3.4.3. Browse Page

3.4.3.1. Browse Page Design

When "Browse Collections" is clicked on in the navigation header, the Browse page is loaded as seen in Figure 9, the Browse page provides three options for the user to browse by: Title, Date and Archive subject. When either of these options are clicked on, all the records in the central repository are loaded and sorted via the chosen field. As with search, 30 results are shown at a time and the user is able to use the "Previous" and "Next" buttons to navigate results.

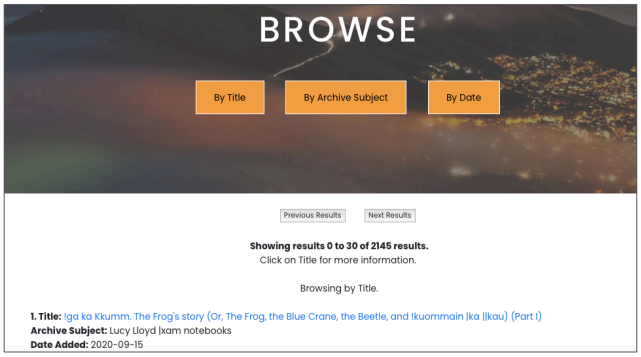

Figure 9: Browse by Title displayed

3.4.3.2. Browse Service Implementation

The Browse service script establishes a connection with the Apache Solr Collection in the same way as the Search service, using an XmlHttpRequest. When the user clicks on a field that they want to browse by, the browseRequest() method is called with the chosen field as the parameter. The query in this case includes requesting all items in the collection then invoking onreadyStateChange as with Search. The query also includes a sorting parameter based on the chosen field to browse by. As with Search, a for loop is used, but iterates through all the records in the collections and retrieves the introductory field values from each record. The 'Title' value is once again a hyperlink that invokes the function viewMetadata() when clicked on, which returns and displays the rest of the DC fields as well for the user.

3.4.4. About Page

Finally, when "About" is clicked on in the navigation header, the About page is loaded. This is an additional Web page on the Web portal that provides the user with more information about the project in terms of why it was developed and what it does. The group members are also shown here.

4. EVALUATION AND TESTING

User Acceptance, Usability, Integration, and Correctness testing was conducted for the Web portal.

4.1. User Evaluations

Before User Testing was conducted, we requested and received ethical clearance from the Faculty of Science Research Ethics Committee. This ensured that the evaluations did not violate the rights of or harm the livelihoods of potential participants. Participants completed consent forms prior to partaking in the evaluations in order to ensure their informed consent for participation.

4.1.1. User Acceptance Requirements Testing

User Acceptance Requirements Testing took place at the beginning of Iteration 4. 5 Data Providers and 5 UCT students completed the evaluations. The 5 Data Providers included 2 representatives from the Bleek and Lloyd collection, 2 representatives from FHYA and 1 representative from Metsemegologolo.

Due to the restrictions caused by the COVID-19 pandemic, the task evaluation was conducted online, and the tasks were sent via e-mail with instructions included. The evaluation consisted of tasks that tested the requirements of the Web portal. This included loading the website, navigating the website, searching using keywords, viewing full metadata records, redirecting to the original source through the identifier tag and browsing by title, date and archive subject. A section asking for additional feedback and comments from participants then followed.

4.1.2. Usability and User Satisfaction Testing

Usability and User Satisfaction Testing proceeded the User Acceptance Testing. A standard questionnaire - USE [11] - for testing the categories of usefulness, ease of use, ease of learning, and user satisfaction was conducted. The same participants who completed the User Acceptance Testing completed this questionnaire. The questionnaire was structured with 30 questions pertaining to the above 4 categories. This allowed the perceived usefulness and usability of the Web portal to be assessed by the target groups.

4.2. Integration Testing

Integration testing was done during each iteration between the Web portal and the metadata harvested from the Harvester stored on Apache Solr. This included testing that the data was able to be retrieved and was in the appropriate format. Each field and value were tested to assess if the content would be displayed correctly on the website.

4.3. Correctness Testing

Each component and function of the Web portal was tested multiple times during each iteration. This included testing various search items for each field and archive and testing the browse service for relevant fields. This has been outlined in Section 3.3.2.

5. RESULTS AND DISCUSSION

5.1. User Evaluation Results and Discussion

5.1.1. User Acceptance Testing Results

Task	Percentage Passed
Home Page Loads	100
Resource Identifiers Load	90
Navigation Works Correctly	100
All Pages Displays Correct Content	100
View Metadata Record Works	100
Previous And Next Buttons Works	100
Search Works Correctly	100
Browse Works Correctly	100

Table 1: User Acceptance Testing Results

As seen in Table 1, the large majority of participants were able to successfully complete all the tasks related to the functional requirements of the Web portal. The only participant from the Metsemegologolo archive was unable to be redirected to the original resource through the identifier as Metsemegologolo does not currently have a website hosting their data.

5.1.2. Participant Feedback Section

5.1.2.1. Bleek and Lloyd Data Provider Feedback

The representatives found that compared to the old Digital Bleek and Lloyd site's search function, the Web portal's search function is effective and described as a "wonderful resource". Suggestions were included.
It was unanimously proposed that the Home Page include more information related to the purpose of the Web portal and information about the collections, without repeating from the "About" page. It was suggested that it be made clear on the Home Page that the website does not actually contain the resources, but displays their metadata and provides a link to the original source.
In terms of design, a different header image from the Table Bay Mountain was suggested that does not signify tourism as much, and the removal of an orange preloader, which flashes an original overlay when changing Web pages, was suggested as it appeared too flashy.

It was found that the limitation of the amount of data on the website made it difficult for them to conclusively determine whether the browse feature was working correctly as a lot of the data structure appeared the same.

5.1.2.2. Five Hundred Year Archive Data Provider Feedback

When instructed to click on a resource identifier that links back to the original resource's website, a participant misunderstood that the original resource was not on the Web portal and tried to navigate from the original resource's website back to the Web portal through the original resource's website.

As with the Bleek and Lloyd representative, it was found difficult to conclusively determine whether the browse feature was working correctly as a lot of the data structure was the same.

5.1.2.3. Metsemegologolo Data Provider Feedback

The representative was satisfied and stated that the Web portal could become really useful. They suggested that the browse navigation include intermediate pages with additional means of browsing. It was suggested that additional visual cues be added to guide the user on the different features' functionality.

5.1.2.4. UCT Students

The students were all able to complete the user tasks successfully and it was stated that everything ran smoothly and worked. There was also a bit of confusion about the identifier original data not being part of the Web portal with attempts to navigate back to the website via the original source website. When queried, this was quickly cleared up and they were able to navigate successfully.

5.1.2.5. User Acceptance Testing Discussion

The effect of the similar data on testing the browsing of the Web portal seemed to be a recurrent issue. Practically, this would be alleviated when more data is added as the data structure between records will vary more.

Another recurrent issue was misunderstanding that metadata identifier linked to a Web page on the Web portal. As it was suggested, this had been alleviated by clearly stating in the Home and About page that the identifier links back to the original website.
Overall, there seemed to be a general agreement that the functionality works as required and that the website could be useful. Suggestions were taken into account and implemented when refining the features. This included the above-mentioned solution of clearly defining the identifier's purpose, change of the header image from Table Mountain to a less tourism-related image, and the removal of the preloader object.

5.1.3. Usability and User Satisfaction Testing Results

Category	Data Providers Average	UCT Students Average	Overall Average
Usefulness	3.70	4.35	4.02
Ease of Use	3.91	4.62	4.26
Ease Of Learning	4.35	4.80	4.58
Satisfaction	3.51	4.29	3.90
Average	**3.87**	**4.51**	**4.20**

Table 2: Usability Test Results

Table 2 displays the Usability Test results of the 4 testing categories for the 2 groups, as well as the overall results. All results were rated with a score out of 5, with a score of 5 denoting "Strongly Agree" and a score of 1 denoting "Strongly Disagree". Figure 10 provides a graphic representation of these results whereby the average is also grasped visually.

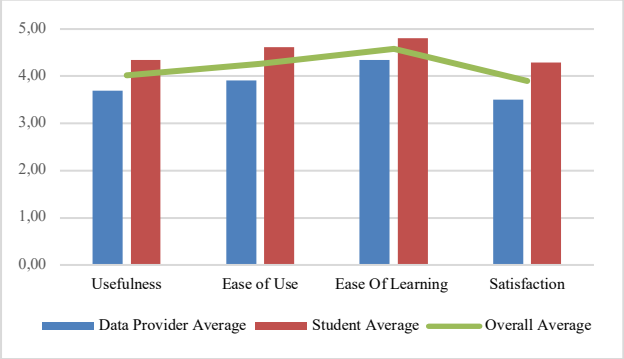

Figure 10: Usability Test Results

5.1.4. Usability and User Satisfaction Testing Discussion

5.1.4.1. Usefulness

The usefulness of the Web portal was rated at an average of 4.02 overall. This is a high rating, which indicates that the participants believe that Web portal could be used for practical purposes in various scenarios. The Data Providers rated the usefulness at an average of 3.70 while the students rated it at an average of 4.35. The difference in rating may be because students find the Web portal useful for research and have not experienced a platform such as it before, while the data providers have more experience with the type of platform. Also, the data providers tested from the FHYA and Bleek and Lloyd archives (with only one representative available to be tested from Metsemegologolo) already have Web sites that contain their relevant data and services that they are interested in and may not see the usefulness of accessing other metadata.

5.1.4.2. Ease of Use

The ease of use of the Web portal was rated at an average of 4.26. This is a high rating, which indicates that the participants were able to use the Web portal and apply its functionality quite easily. The Data Providers rated the ease of use at an average of 3.91 and the students rated it at an average of 4.62. This lower score given by the data providers may be due to the functionality issue raised of the browsing not always being clear due to the similarly of the data structure. Additionally, it may be due to the misunderstanding related to the identifier linking back to the original website discussed before. It may further be due to this being the first national heritage Metadata Aggregator, while the data providers are more accustomed to focused heritage digital libraries.

5.1.4.3. Ease of Learning

The ease of learning of the Web portal was rated at an average of 4.58. This is the highest rating received of the four categories, which indicates that the participants were able to learn the Web portal and its functionality with ease. The Data Providers rated the ease of learning at an average of 4.35 and the students rated it at an average of 4.80. These are both high scores. This result may be due to the consistent and familiar design used for the Web portal interface. For example, buttons and search bars are components commonly used on the Web.

5.1.4.4. Satisfaction

The satisfaction of the Web portal was rated at an average of 3.90. This is the lowest rating but also indicates that the participants were more in agreement of their satisfaction of the Web portal than not. The Data Providers rated their satisfaction at an average of 3.51 and the students rated it at an average of 4.62. This lower score given by the data providers may be due to the aforementioned issues raised in their feedback related to browsing and the identifier link redirection.

5.2. Integration Testing Results and Discussion

Metadata Field	Iteration 1	Iteration 2	Iteration 3	Iteration 4
dc_title	Passed	Passed	Passed	Passed
dc_identifier	Passed	Passed	Passed	Passed
dc_contributer	Failed	Failed	Passed	Passed
dc_creator	Failed	Failed	Passed	Passed
dc_modDate	Passed	Passed	Passed	Passed
dc_publisher	Failed	Failed	Passed	Passed
dc_subject	Passed	Passed	Passed	Passed
dc_rights	Passed	Passed	Passed	Passed
dc_source	Passed	Passed	Passed	Passed
dc_description	Passed	Passed	Passed	Passed
dc_format	Passed	Passed	Passed	Passed
dc_coverage	Passed	Passed	Passed	Passed

Table 3: Integration Testing Results

Table 3 displays all relevant Dublin Core fields harvested and their result of being able to display correctly on the Web portal per iteration. The results indicate that the display of the fields dc_creator, dc_publisher and dc_contributer was not always functioning correctly. This was due to issues related to these fields containing multiple values. This issue was resolved by editing the Apache Solr schema configuration file appropriately to accommodate these fields' format.

5.3. Correctness Testing Results and Discussion

Iteration 1			Iteration 2			Iteration 3		
Testing search items' JSON responses for testing data			**Testing search items' text responses for data providers' data**			**Testing search items' text responses for multi-valued fields and browse items.**		
Test case	**Input**	**P/F**	**Test case**	**Input**	**P/F**	**Test case**	**Input**	**P/F**
Genre field	genre:fantasy	P	Title field	moon	P	Creator field	Hoesar	P
Author field	author:Matthew	P	Subject field	stories	P	Contributor field	Henry Callaway	P
Title field	title:computer	P	Creator field	David Hoesar	F	Publisher	Wilhelm Bleek	P
			Contributor field	Henry Callaway	F	Title field	Browse Title	P
						Date field	Browse Date	P
						Subject field	Browse Subject	P

Table 4: Correctness Testing Results

Table 4 above presents the correctness testing results for different test cases in the iterations 1, 2 and 3. 'P' or 'F' denote "Pass" or "Fail" in terms of displaying correctly on the Web browser. In Iteration 1, search items' JSON responses from the testing data was tested. All test cases passed. In Iteration 2, search items' text responses for the data providers' data were tested. The fields only containing single values passed but the fields containing multiple values such as 'creator' or 'contributor' failed to display correctly. This was resolved during iteration 3, whereby search items' text responses for multi-valued fields and browse items were tested. All test cases passed.

6. CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

6.1.1. Functional Requirements Conclusions

The aims of the project were fulfilled, whereby a heritage Web portal, built upon metadata aggregation, was successfully developed. Particularly, the resulting Web portal met its functional requirements of providing search and browse services and displaying metadata information to users through a user interface. The expansion of these requirements communicated by the data providers during requirements gathering included the prioritisation of search, browse and viewing of metadata over all features, and the implementation of browsing by "Title", "Date" and "Author". As mentioned, the former was achieved, and users are able to browse by these fields, with author replaced by subject as author is not a Dublin Core standard field.

The success of the functional requirements was tested and deduced through User Acceptance, Integration and Correctness Testing whereby the results indicated that these requirements were achieved.

6.1.2. Non-functional Requirement Conclusions

6.1.2.1. Maintainability – The overall system implements a layered architecture and the code has been developed with high coupling and low cohesion. The Web portal component is thus maintainable as its implementation is separable from the other components and its inner functions are loosely coupled.
6.1.2.2. Usability – This was enforced through the use of consistent and familiar features and by applying design principles. The results from the Usability testing indicate that the users were mostly able to use the Web portal with ease.
6.1.2.3. Portability – The Web portal is portable as it is run on a server and can be accessed through its URL.
6.1.2.4. Availability – As the Web portal runs on a Web server, it cannot be run offline and requires that the user has an Internet connection. It may also not run if the server is faulty or stops responding.
6.1.2.5. Speed – The Web portal runs on a server that has a specification of 32 GB RAM with 4 CPU cores. This suffices for the requirements of the Web portal. Apache Solr is used for handling search and browse requests. Solr does not have the constraints that conventional Relational Database Management Services have (RDBMS) and does not use B-trees for search and is thus efficient.
6.1.2.6. Scalability – The metadata collection is stored through Apache Solr, which is able to support the amount of data harvested.

6.1.3. Further Conclusions

The successful implementation of the project sees this as the first national heritage metadata aggregator. As it has been built upon the philosophies of current low-resource software tools, this project could add meaningful contribution to the research community. The Web portal has also been successfully implemented as a stand-alone component that could be transferable to other metadata archives.

6.2. Challenges and Limitations

This project was implemented during the global COVID-19 pandemic. Tasks such as requirements gathering, meetings and evaluations were thus limited as a result of not being able to have in-person interaction. This was mitigated by using VOIP technology for communicating with team members and our supervisor. Also, the data providers were communicated with via e-mail and the evaluations took place online. Although these mediums of communication allowed us to maintain communication, network problems often became an issue and managing group-work activities would have been more effective in person. Also, evaluations were limited in terms of participants not being able to immediately communicate issues or questions as they would in person. This may have resulted in them overlooking a query they might have had (although they were encouraged to query in the introduction to the evaluation).

The final Web portal was limited by only having final user evaluations and not iterative user evaluations. Iterative user evaluations would have maximized the level of satisfaction from the target users as their feedback would have been used for consistent refinements. Another challenge that was anticipated in our risk analysis was the delay in reception of data from all the data providers. Testing data was used for a lot of the feature development process and implementation changes had to be made to account for the different schema when all the data was received from the data providers. Finally, there was a delay in confirmation of progressing with evaluations with the data providers. There was thus less time available to implement their feedback.

6.3. Future Work

The Web portal could be extended to include language translations of the data and fields for national and international users, as the Web pages and content are currently only displayed in English.

Another extension would be to improve the interface of the Web portal when used on mobile devices such a smartphones and tablets. This will encourage users to access it from anywhere, facilitating mobility. A further extension of this would be to create a mobile application as mobile applications have become more popular, and is a more convenient and readily available medium of access to the Web portal.
Finally, the Web portal could include a feature of being able to add admin users. This would enforce quicker development modification from all team members and would facilitate the expansion of data providers more efficiently and conveniently.

7. ACKNOWLEDGEMENTS

I would firstly like to thank my supervisor, Hussein Suleman, for his commitment and guidance throughout the project course. I would subsequently like to thank my team members, Alex Priscu and Ashil Ramjee, for being consistently dependable and helpful with the project work. An expression of thanks is extended to the data providers at the Bleek and Lloyd Collection, FHYA and Metsemegologolo, for providing us with data as well as partaking in the user evaluations. A thank you to the students who took part in the evaluations as well. Finally, I would like to express my gratitude to the UCT Computer Science Department for providing a Web server and Craig Balfour for assisting in its configurations.

REFERENCES

[1] William Y. Arms, Naomi Dushay. Dave Fulker, and Carl Lagoze, 2002. A Case Study in Metadata Harvesting: the NSDL. *Library Hi Tech*, 21 (2).

[2] VenuGopal Balijepally, Nils Brede Moe, Torgeir Dingsoyr, Sridhar Nerur, 2012. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software 85, 6 (2012),* 1213-1221.

[3] Sally Chambers and Wouter Schallier, 2010. Bringing Research Libraries into Europeana: Establishing a Library-Domain Aggregator. *Liber Quarterly 20, no. 1 (September 2010),* 105.

[4] Timothy Cole, Joanne Kaczmarek and Sarah Shreeves, 2003. Harvesting cultural heritage metadata using the OAI Protocol. *Library Hi Tech.* 159-169.

[5] DCMI: DCMI Schemas. 2020. *Dublincore.org*. https://dublincore.org/schemas/.

[6] Azhar Desai, 2010. SimplyCT Online Search*, CS10-04-00*, Department of Computer Science, University of Cape Town.

[7] Carl Lagoze, Walter Hoehn, David Millman, William Arms, Stoney Gan, Dianne Hillmann, Christopher Ingram, Dean Krafft, Richard Marisa, Jon Phipps, John Saylor, Carol Terrizzi, James Allan, Sergio Guzman-Lara, and Tom Kalt, 2002. Core Services in the Architecture of the National Science Digital Library (NSDL)", *In Proceedings of Second ACM/IEEE-CS Joint Conference on Digital Libraries*, 201-209.

[8] Carl Lagoze , Dean Krafft, Tim Cornwell, Naomi Dushay, Dean Eckstrom, and John Saylor, 2006. Metadata aggregation and 'automated digital libraries': a retrospective on the NSDL experience, in Nelson, M. L., and Marshall, C. C. (eds): *Proceedings of the 6nd ACM/IEEE-CS Joint Conference on Digital Libraries, ACM,* 230-239. doi:10.1145/1141753.1141804

[9] Carl Lagoze, Herbert Van de Sompel, Simeon Warner and Michael Nelson, 2001. The Open Archive Initiative Protocol for Metadata Harvesting, Open Archives Initiative. Available www.openarchives.org/OAI/openarchivesprotocol.htm

[10] Phiri, Lighton, and Hussein Suleman, 2011. simplyCT: A Lightweight Digital Library Framework, Department of Computer Science, University of Cape Town.

[11] Arnold Lund, 2001. Measuring usability with the use questionnaire. *Usability Interface: The usability SIG newsletter of the Society for Technical Communications,* 8(2), 3-6.

[12] Joann McGregor. and Lyn Schumaker, L, 2006. Heritage in Southern Africa: Imagining and Marketing Public Culture and History. *Journal of Southern African Studies* 32, 4, 649-665.

[13] Mark Otto, a. 2020. Introduction. *Getbootstrap.com*. https://getbootstrap.com/docs/4.4/getting-started/introduction/.

[14] Lighton Phiri and Hussein Suleman, 2012. In Search of Simplicity: Redesigning the Digital Bleek and Lloyd, *DESIDOC Journal of Library & Information Technology*, 32, 306-312, DESIDOC, Ministry of Defence, India.

[15] Ben Schneiderman, 1987. Designing the user interface strategies for effective human-computer interaction. *ACM SIGBIO Newsletter* 9, 1, 6.

[16] Hussein Suleman, 2019. Investigating the effectiveness of client-side search/browse without a network connection, *Proceedings of 21st International Conference on Asia-Pacific Digital Libraries (ICADL),* Kuala Lumpur, Malaysia, Springer.

[17] Hussein Suleman, Tatenda Chipeperekwa and Lawrence Webley, 2011. Creating a National Electronic Thesis and Dissertation Portal in South Africa, in Olivier, E., and Suleman, H. (eds): *Proceedings of 14th International Symposium on Electronic Theses and Dissertations (ETD 2011), Cape Town, 13-15 September.* Available http://dl.cs.uct.ac.za/conferences/etd2011/papers/etd2011_webley.pdf

[18] Hussein Suleman, 2012. Design and architecture of digital libraries. in Chowdhury, G.G. and Foo, S. (Eds), *Digital Libraries and Information Access: Research Perspectives,* Facet, London, pp. 13-28.

[19] Apache Solr, 2020. *Lucene.apache.org*. https://lucene.apache.org/solr/

[20] Herbet Van de Sompel, Michael Nelson, Carl Lagoze and Simeon Warner, 2004.Resource Harvesting within the OAI-PMH Framework. *D-Lib Magazine* 10, 12.

[21] Gary Wan and Zao Liu, 2008. Content-Based Information Retrieval and Digital Libraries. *Information Technology and Libraries 27*, 1, 41.

[22] Ian H. Witten, Alistair Moffat, and Timothy C. Bell, 1999. Managing gigabytes: compressing and indexing documents and images. Morgan Kaufmann.

[23] Mary Woodley, 2008. Crosswalks, metadata harvesting, federated searching, metasearching: Using metadata to connect users and information. In M. Baca (Ed.), Introduction to metadata 38–62, Los Angeles, CA: Getty Research Institute

[24] The Xapian Project. 2020. *Xapian.org*. https://xapian.org/.

SUPPLEMENTARY INFORMATION

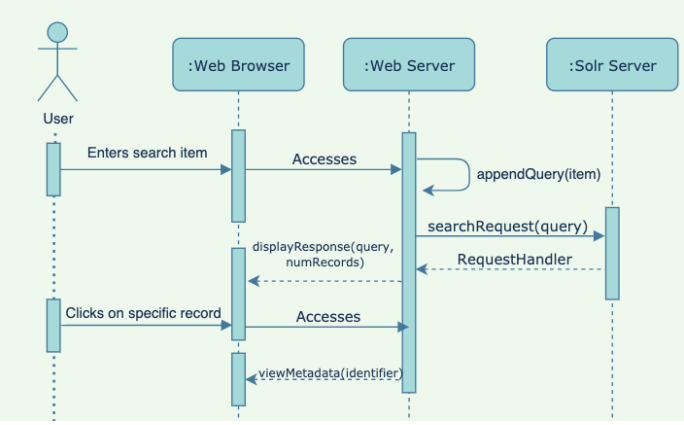

Figure 11a: UML Sequence Diagram for Search

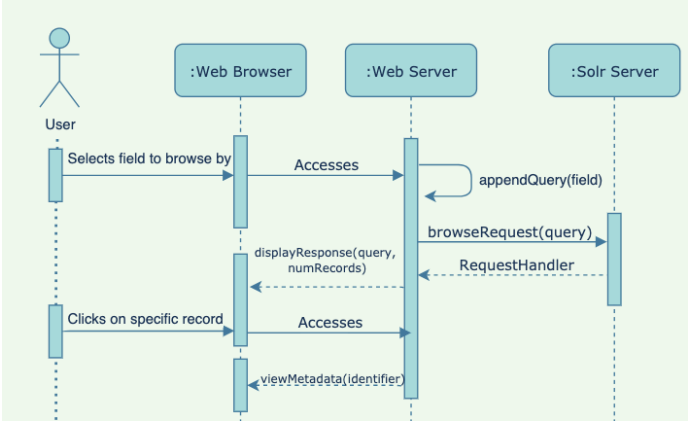

Figure 11b: UML Sequence Diagram for Browse

Evaluation Results

	Participant Number										
	Data Providers					Students					
	Metsemegologolo	Bleek and Lloyd		FHYA							
Usefulness	Justine Winjes	Pippa Skotnes	Niek De Greef	Henry Fagan	Rifqah Khan	Sam Rosslind	Victoria McDonalad	Rifqah Francis	Suhail Kapdi	Bradley Malgas	Average
1. It helps me be more effective.	5	5	3	4	3	5	5	4	3	4	4,1
2. It helps me be more productive.	5	3	3	4	3	5	5	4	3	3	3,8
3. It is useful.	5	5	4	5	3	5	5	5	3	5	4,5
4. It gives me more control over the activities in my life.	5	4	3	3	3	4	5	4	4	4	3,9
5. It makes the things I want to accomplish easier to get done.	5	5	3	4	3	5	5	5	4	4	4,3
6. It saves me time when I use it.	4	3	3	4	3	5	5	4	4	3	3,8
7. It meets my needs.	4	3	3	4	2	5	4	5	4	5	3,9
8. It does everything I would expect it to do.	4	3	3	4	3	5	5	4	3	5	3,9
Ease of use											**4,025 Category Ave**
9. It is easy to use.	4	5	4	4	4	5	5	5	5	5	4,6
10. It is simple to use.	4	5	4	4	4	5	5	5	5	5	4,6
11. It is user friendly.	4	5	3	4	4	5	5	4	5	5	4,4
12. It requires the fewest steps possible to accomplish what I want to do with it.	4	5	4	4	4	5	5	5	4	4	4,4
13. It is flexible.	5	3	3	4	3	5	4	4	4	3	3,8
14. Using it is effortless.	5	5	4	4	2	5	5	5	4	5	4,4
15. I can use it without written instructions.	5	5	4	4	4	5	5	4	4	5	4,5
16. I don't notice any inconsistencies as I use it.	5	3	4	2	1	5	5	5	3	4	3,7
17. Both occasional and regular users would like it.	5	4	3	4	3	5	5	4	4	3	4
18. I can recover from mistakes quickly and easily.	5	3	4	4	2	5	5	5	2	5	4
19. I can use it successfully every time.	5	5	4	4	2	5	5	5	5	5	4,5
Ease of learning											**4,263636 Category Ave**
20. I learned to use it quickly.	5	5	4	4	4	5	5	5	5	5	4,7
21. I easily remember how to use it.	5	5	4	4	4	5	5	5	5	5	4,7
22. It is easy to learn to use it.	5	5	4	4	4	5	5	5	5	5	4,7
23. I quickly became skillful with it.	5	5	4	3	4	5	5	4	4	3	4,2
Satisfaction											**4,575 Category Ave**
24. I am satisfied with it.	4	4	3	3	2	5	5	5	4	3	3,8
25. I would recommend it to a friend.	5	5	3	4	3	5	4	5	5	4	4,3
26. It is fun to use.	4	3	3	3	3	5	5	4	4	4	3,8
27. It works the way I want it to work.	4	4	3	4	3	5	5	5	4	3	4
28. It is wonderful.	4	5	3	4	2	5	5	4	4	3	3,9
29. I feel I need to have it.	5	3	3	4	2	4	4	4	3	3	3,5
30. It is pleasant to use.	4	4	3	4	3	5	5	4	4	4	4
											3,9 Category Ave