Implementing a Low-Cost South African National Heritage Portal built from Metadata Aggregation

Honours Project Proposal

Alex Priscu
Department of Computer Science
University of Cape Town
Cape Town, South Africa
PRSALE003@myuct.ac.za

Toshka Coleman
Department of Computer Science
University of Cape Town
Cape Town, South Africa
CLMTOS001@myuct.ac.za

Ashil Ramjee
Department of Computer Science
University of Cape Town
Cape Town, South Africa
RMJASH002@myuct.ac.za

KEYWORDS

Metadata, Aggregation, Portal, Heritage, Archive, OAI-PMH.

1. PROJECT DESCRIPTION

South Africa has a remarkable heritage, characterised by multiculturalism, post-colonialism and apartheid [5]. Traditional methods of physical storage of cultural heritage objects are out of date due to artefact deterioration. However, technological developments such as XML databases, Internet and Web 2.0 can create a suitable environment for the preservation of digitalised object information that enables efficient and reliable archiving, search and evaluation of such objects [7]. Online digital repository systems, digital libraries and digital archives are common platforms for storing various types of digital content [8].

Academic documents and heritage collections are commonly archived in such repositories for the purpose of discovery and preservation [9]. These systems store and manage digital objects and provide access to them through the provision of search and browsing services via a Web-based interface [9]. The preservation of artefacts such as books, sketches and paintings is crucial for the benefit of future generations. Furthermore, social, environmental and political aspects are influenced by cultural heritage, which can directly affect the nature of government decisions [6]. In the current period of rising Web browser usage, these online systems benefit from general accessibility [8].

After ten years of attempts to create a national heritage portal, there is still no established central repository system storing collected metadata from multiple South African cultural heritage archives [4].
Early digital archives such as the Bleek and Lloyd Collection and early publications scanned by Digital Imaging South Africa (DISA) are all independent and not linked [12][13]. Cross-archive discovery is a problem that affects digital libraries, as there are many existing digital libraries hosted by different institutions [4].
The focus of our project is to develop a system that addresses this issue. We intend to develop the first central South African heritage portal that will provide users with access to historical documents from multiple domains. This will be implemented using Metadata Aggregation, where the central organisation operates services to improve the discovery of cultural heritage resources through the collection of their associated metadata.

2. PROJECT STATEMENT

This section identifies the central issue that our project tackles by outlining our problem statement, project aims, involved stakeholders, potential users and our project requirements.

2.1. Problem Statement

The two main limitations of existing software which provide access to heritage documents are as follows:

Cultural cross-archive discovery is not possible in South Africa, as existing cultural heritage archives are hosted by separate organisations, containing only focused content and independent of one another. Ideally, historians, researchers and students should be able to search across multiple archives.

Cross-archive discovery is currently only possible with a generic search engine such as Google. This is impractical due to its limitation in the specialisation of local heritage documents. Search engines, like Google, function by Web scraping, which limits the accessibility of documents. Only a Web Portal that explicitly searches through remote repositories can address this limitation.

2.2. Project Aims

This project aims to build a small-scale prototype of the South African National Heritage Portal, using best practises in low-resource heritage archive systems to collect metadata from multiple local heritage archives. It also aims 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 Africa heritage to access and search through a range of historical resources through a single Web Portal without experiencing the above-mentioned limitations linked to the usage of the current software available.

As a low-cost 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. Low cost relates to the complexity of the Web Portal development. If this system is low-cost, a minimal number of resources such as team members, servers, time and funding are required to build the system. Due to technology and research availability, a low-cost central repository was not developed for this domain. Previously, central repositories were designed with limitations determined by complexity, resources and funding. Currently, we may merge protocols and system designs used in separate projects to build a single low-cost system. If this system must be built in another geographic area such as a developing country with fewer resources, a low-cost system would be an effective solution.

Therefore, the overall objective would be to demonstrate the feasibility of this project as proof of concept by implementing the South African National Heritage Portal. However, system development will be discussed using the software engineering approach, as the project's primary components are each development projects.

2.3. Stakeholders & Potential Users of the Portal

The primary project stakeholders and potential users of the portal are listed below:

2.3.1. Project Supervisor

Hussein Suleman

2.3.2. Project Team

Alex Priscu
Toshka Coleman
Ashil Ramjee

2.3.3. Data Providers

Five Hundred Year Archive
Bleek and Lloyd Collection
Metsemegologolo

2.4. Project Requirements

The following requirements of the project will be implemented:

- The development of a central heritage portal that is developed as a metadata aggregator that aggregates from multiple South African heritage archives.

- The heritage portal is low resource in terms of financing for its development, server hosting and labour. Also, according to our prior research, it is developed using the best practises for low-cost development.

- It provides cross-archive discovery services to end-users, such as searching and browsing archives through the central Web Portal.

- The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), a low-complexity interoperability protocol that facilitates the incremental transfer of metadata between networked systems, is implemented for the aggregation of metadata.

- The metadata requires a database or file store.

- The harvester is implemented as a Web client application. It regularly collects metadata from multiple sites and creates a central store. Additionally, the harvester resolves the issue of various metadata formats.

- The three OAI-PMH-compliant Web interfaces for data providers are used to share metadata. Each site will need a unique data provider tool that can expose its metadata.

These are our initial project requirements that will be refined during requirements gathering.

3. PROCEDURES AND METHODS

This section discusses the design, development platform, implementation strategy and evaluation testing for our system.

3.1. Design Features

The metadata aggregator system will adopt a three-layer architecture for the development of the South African National Heritage Portal. Each layer is separable and communicates via a fixed low-cost protocol. As shown below in Figure 1, the system layers are: Data provider interfaces, Harvester, and Central Web Portal.

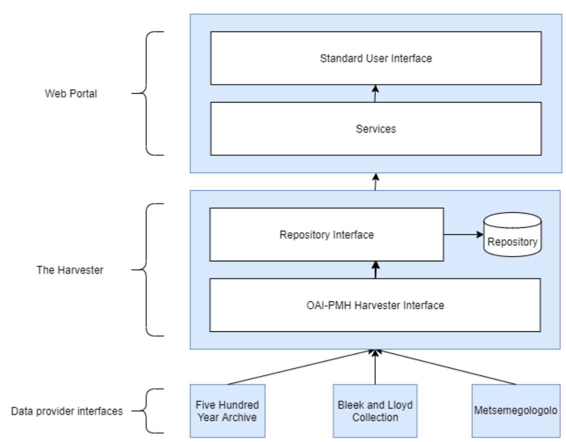

Figure 1: Overall System Architecture

Central Web Portal

The Web Portal layer includes the end-user component that will allow users to interact with the system through the Standard User Interface and provide them with access to the metadata database. The aggregation of metadata will be used to feed content to the Web Portal. As part of its Services component, the Web Portal will implement a national cultural heritage-adapted search engine tailored for the retrieval of metadata records from the data providers. Users will be able to use the search and browse services on the Web Portal by accessing indexed metadata records that link to the original documents in the source repositories.

The aggregated metadata records in the repository are accessed through a query via the Web Portal. Metadata records stored in the database are indexed. The system will use Apache SOLR to process search and browsing of indices. Finally, end-users of the metadata aggregator Web Portal can then search through a single archive or browse through a collection of archives using keywords and different filters, such as title, author and subject.

The Web Portal will also encapsulate and provide all standard Web services such as the OAI-PMH, Really Simple Syndication (RSS) and end-user services available to end-users in a single location. Additional features will include the integration of HTML pages such as “Home” and “About” pages.

The Harvester

The Harvester comprises of a Web client application that will periodically collect metadata from multiple remote data providers to a centralised repository. The OAI-PMH Harvesting Interface facilitates services that provide efficient interoperability between content repositories by presenting aggregated metadata through the end-user search portal.

The Harvester component will collect metadata from a set of remote cultural heritage archives. Initially, a full harvest of each remote repository will be made where all metadata records are retrieved and stored in the central repository. The metadata retrieved is stored in the shared Harvester and Repository data store with the use of flat files. Subsequently, the harvester shall carry out incremental harvesting where only metadata records are collected, which have been added since the last harvest date. Retrieved metadata records are converted and stored directly in the shared repository data store and the Repository uses the OAI-PMH to make them accessible. The OAI-PMH Harvester Interface performs initial validation checks to prevent malformed records from being sent to the Repository through the shared data store. Time intervals could be set for the harvesting cycle in order to maximise efficiency. The Harvester layer will implement an administrative end-user interface, control the harvesting process and manage the list of remote repositories.

Data Provider Interfaces

This layer contains Web interfaces for each site to share metadata supporting the OAI-PMH. At least three sites will be data providers to the Web Portal, each having its own metadata format. The three data providers are the FHYA, the Bleek & Lloyd Collection and Metsemegologolo. Each of these data providers will need to expose their metadata so that the OAI-PMH Harvester can run.

3.1.1. Development Platform

The Python Programming Language will be used to code the backend components of the system.

Web Portal

The Web Portal will be developed so that it can be accessed via a Web browser. The front-end interface of the portal will be implemented using XML to define user content structure, CSS to format content presentation, and Bootstrap, a front-end framework that provides CSS-based design templates for interface components.

The portal will use the Apache SOLR open-source search platform to search through records. The Web Portal will also include an RSS feed class for generating content updates. Lastly, our portal will be hosted on an Apache Web Server and will run using Amazon EC2.

The Harvester

The Harvester components will be implemented using Python and include relevant programs running at regular intervals to collect records from data providers.

These programs include: the OAI-Harvester class for harvesting the OAI metadata from different data providers; a class for presenting records in the correct format; a class for sending requests and receiving responses from the OAI sources and a database class for connecting to the database and storing records.

These records will be stored in the database as flat files in an XML format such as unqualified Dublin Core. Data provider repositories provide access to their records through the OAI-PMH interface. The Harvester will then use this protocol to harvest records from the remote server.

Data Provider Interfaces

Data providers will need to share their data in their archives. This will be done by developing an OAI-PMH interface for each of the sites. The interface will be coded using Python and each interface will be different. The data aggregated will belong to the data providers as the resources listed on the portal will be redirected to their source websites.

3.1.2. Implementation Strategy

This project will adopt the Agile software development methodology as it has proven to be more efficient than the Waterfall approach for projects with versatile user requirements and adjustments. Due to the complexity and separation of the project, certain elements of Agile, Extreme Programming, and Incremental frameworks will be used.

Therefore, our implementation strategy will follow a four-phased iterative approach. Each iteration shall comply with the requirements, analysis, design, implementation, integration and test-driven development. Deliverables will be defined before each iteration, and weekly meetings with all team members will be held to ensure that everyone is on track. In addition, after each iteration, refactoring from the Extreme Programming Framework will be done to refine the code. The advantages of the refactored code include increased efficiency, readability, comprehensibility and reusability, which is important to the research community.

The details of each iteration are shown below:

Iteration 1

- Initialise Communication with Data Providers
- Gather Requirements from Data Providers
- Acquire Data from Data Providers
- Set Up Web Server
- Develop Skeleton for Each Component
- Perform Software Testing on Web Portal Skeleton

Iteration 2

- Update Iteration Tasks According to Requirements
- Complete Skeleton for Each Component
- Integrate skeletons for each component
- Implement Search and Browse Service Within Web Portal
- Develop the Harvester within Harvester Component
- Develop the First Data Provider Interface
- Integrate Harvester and Data Provider Interface into Web Portal
- Perform Compliance Testing for Data Provider Testing
- Perform Correctness Testing for Harvester
- Perform Integration Testing

Iteration 3

- Update Iteration Tasks According to Requirements
- Develop Repository in the Harvester Component
- Perform Correctness Testing for Repository
- Develop the Second and Third Data Provider Interface
- Integrate additional Data Provider Interfaces into Web Portal
- Include Any Other Additional Features
- Perform Integration Testing
- Perform Acceptance Testing

Iteration 4

- Integrate Three Components into Final System
- Perform Performance and Correctness Testing for Web Portal
- Perform Integration Testing
- Perform Acceptance Testing
- Perform User Task Evaluation Testing for Web Portal
- Integrate Additional Changes Based on User Evaluations

By following this iterative approach, it will be easier to manage time and complexity constraints. We will be able to: respond to any drastic changes within the system; to assess and improve our architecture as well as identify and mitigate risks in the early project iterations.

3.1.3 Expected Challenges

The main challenge of the project is that we are working during a global Covid-19 pandemic with no expected outcome. It raises challenges such as working from home while team members are in different countries, and therefore advantageous in-person interactions between our supervisors, team members and data providers are not possible.

Data providers have yet to provide the cultural metadata that we will be working on, so it is currently a challenge to specify how we will convert the metadata in order to comply with the OAI-PMH repository interface. In addition, we need to ensure scalability for up to 30 000 data records per site.

Building on the previous challenge, we're aiming to build a low-cost Web Portal of cultural heritage resources that hasn't been done before while working with technologies and resources that we're not familiar with. It is also predicted that we will have to go through a learning cycle both before and during the implementation of the program.

3.2. Evaluation/Testing

This section outlines the evaluation procedures that we will perform for each component of the system. This includes user evaluations; use of software measurement metrics; compliancy tests; metadata schema checks; performance, unit and black box testing.

3.2.1. Central Portal

The Web Portal and its user services are the front-end component of our system. Using user task evaluations, the usability of this component will be tested. This assessment will consist of tasks that require users to interact with the interface and assess their ability to complete these tasks. This will include assessing their ability to: set up and load the portal and navigate the Web pages within the portal; locate the search engine; search using an identifier; browse the results and view the relevant heritage resources. Software measurement metrics including the response times, the amount of bandwidth used, and the time to perform the searches will also be tested for this component.

3.2.2. The Harvester

The Harvester component will be subject to Unit Testing in order to evaluate its individual units and to validate the performance of each unit as intended. White-box testing will also be used to test the internal structure of the component. Performance testing will be conducted as part of software measurement metrics to evaluate how the system performs in terms of its responsiveness and stability.

3.2.3. Data Provider Interfaces

Each data provider interface will need to be adapted to its metadata schema. In addition, it will have to comply with the OAI-PMH, which will allow the Harvester to perform its services. In order to do this, data providers need to specify their current metadata schema, which will allow us to cross-map and convert data while complying with the OAI-PMH. For each interface, the component will be tested using data with complete and incomplete metadata fields. In addition, different data sizes will be used to test the component.

3.2.4. Final System

The three components mentioned above will be involved in integration testing where the individual components will be combined and tested as a holistic system.

The final test will be Acceptance Testing, which will evaluate our system for its acceptability. Compliance of the system with project requirements and other standards will be assessed on whether it is acceptable for final delivery.

4. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

This section outlines the ethical, professional and legal issues regarding the project. Relevant areas include our user evaluations, data use and software.

4.1. User Evaluation Consent

In accordance with the Computer Science Ethics Committee, we are required to complete and submit an ethics application form that includes a proposed consent form for user evaluations. If the submission is accepted, this consent form will be directed to all participants partaking in our project's user evaluation. Their participation will be entirely voluntary, and they will be allowed to withdraw from the evaluation test at any time.

4.2. Data Providers Terms of Use

Our data providers will be made fully aware of the terms of use of their data provided.

4.3. Software

We will be using open-source software in accordance with the terms and conditions outlined by the software developers.

5. RELATED WORK

This section gives an overview of the related work with regards to Web Portals. Two metadata aggregator systems, namely, Europeana and the National Electronic Theses and Dissertations Portal (NETD) provide contrastingly useful insights towards the creation of a centralised Web Portal [1][10]. Europeana entails aspects of a high-cost scalable Web Portal that caters for rich complex metadata whereas NETD is a local ETD portal that uses a low-cost architecture that conforms to one metadata standard [2][10].

5.1. Data Providers

The National Electronic Theses and Dissertations (NETD) portal is a metadata aggregator in South Africa which provides access and cross-archive services to a collection of local Electronic Theses and Dissertations (ETD) by relying on the OAI-PMH interface to retrieve the remote repositories' metadata [10]. As shown in Figure 2 below, the NETD system consists of three separate layers to support future expansion and scalability: Harvester, Repository, and Web Portal.

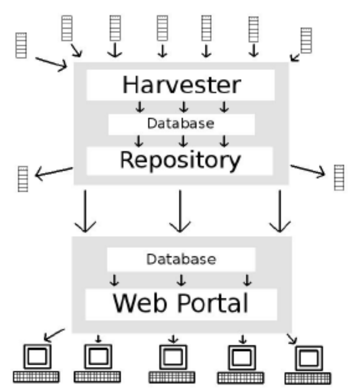

Figure 2: The NETD System Architecture [10]

Europeana is an example of a large-scale cultural heritage metadata aggregator consisting of a variety of European archives and other cultural resources that can be accessed via its Web Portal interface [1][2]. Essentially, by using RDF, Application Programming Interfaces (API) and the OAI Object Reuse and Exchange Framework (OAI-ORE), Europeana extracts, aggregates and exploits its metadata from heterogeneous digital objects [2][3]. OAI-ORE allows these resources to be recognised using aggregation resources described in the resource maps.

5.2. Web Portal

The metadata collected for the Europeana is stored and manipulated in a central Web Portal, which provides core user services such as search and browse [11]. Similarly, the NETD Web Portal encapsulates and enables standard end-user services, as well as Web services such as OAI-PMH and RSS, to be available to end-users at a single location [10]. This portal is a multi-tiered, simple repository architecture chosen to improve system interoperability by connecting to larger repository systems such as the Networked Digital Library of These and Dissertations (NDLTD) [11] and similar organisations.

5.3. Harvester

For Europeana, metadata is either collected by harvesting with the OAI-PMH or added to the central repository using an API [11]. All recovered and harvested data is stored in distributed libraries; therefore, the system architecture could be distributed. Europeana's digital objects are based on the Resource Description Framework (RDF), so services can take advantage of the rich relationships between digital objects and collections.

For NETD, The Harvester component uses the OAI-PMH to facilitate the transfer of metadata from digital repositories. The metadata retrieved is stored in the shared Harvester and Repository data store with MySQL used as the database layer. The Harvester performs initial validation checks to prevent malformed records from being sent to the Repository [10]. Metadata aggregation is a feasible solution for cross-archive search and browsing, as the local NETD system is an example of such an ETD domain Web Portal.

6. ANTICIPATED OUTCOMES

This section overviews our anticipated outcomes regarding our system. In addition, it identifies our major design challenges, expectations and key success factors.

6.1. System

At the end of the project, we aim to have developed a low-cost prototype of the South African National Heritage Portal that meets the functional requirements.

6.1.1. Software and Key Features

The overall system architecture will consist of a Web Portal, the Harvester that includes a repository and interfaces for data providers. The key features include automated metadata collection, data contribution interfaces and essential services, similar to any other aggregator.

6.1.2. Major Design Challenges

First, no low-cost Web Portal has been created, so the biggest design challenge is trying to build a low-cost portal. In addition, we need to determine how to design the "low-cost" aspect of the project that will cater for reusability and scalability in a low-cost environment.

Similarly, for other aggregators, data integrity will not be maintained for some collections during the conversion of metadata, as some fields will not be addressed. We therefore need to carefully analyse the data provided and choose the most appropriate standard.

Each component of the system will be assigned to work on an individual basis for each team member. However, each component may rely on other components to perform its services. We will therefore need to work closely and perform frequent integration testing to ensure that our overall system is integrated as intended.

6.2. Expectations

By the end of our project, we expect a fully functioning low-cost cultural heritage portal to be reusable and scalable for the research community. In addition, we expect our final design to be used as a basis for the development of low-cost aggregators.

We also anticipate that the heritage portal will provide users with the ability to search multiple South African heritage archives through a central system, and that the limitations of using generic existing software mentioned in Section 2.1 will be mitigated.

6.3. Key Success Factors

The success of our project will be indicated by its ability to fulfil its functional requirements. These include:

Table 1: Component Success Factors

Web Portal	Harvester	Data Provider Interfaces	Overall System
Able to search for records	Successfully retrieve data from multiple archives	OAI-PMH interfaces can share data	System developed as low-cost/complexity
Able to browse records	Resolve issues of different metadata formats		Evaluate success using software measurement metrics
Able to navigate pages of results			Feedback communicates the usefulness of the application in comparison to the existing software
User satisfaction with Interface			User satisfaction with response times
Search service spans multiple archives			Successfully integrate components

7. PROJECT PLAN

This section gives an overview of our project plan. Risk and mitigation strategies are provided in a risk matrix. Furthermore, the resources needed, deliverables expected and work allocation between team members are discussed. In addition, a Gantt chart and set milestones identify our time management for the project.

7.1. Risks and Risk Management Strategies

The project risks and their associated strategies for mitigation, monitoring and management are outlined in Appendix A.

7.2. Gantt Chart Timeline

Attached in Appendix B.

7.3. Resources Required

To successfully execute the project, a few resources need to be obtained. Firstly, all software, including Apache SOLR and the OAI-Framework, are open-source. The servers required to test, host and store data for the application will be provided by the University of Cape Town. Section 2.3 mentions the three data providers that will provide the data required and the people required for the successful implementation of the system.

7.4. Deliverables

- Literature review
- Project proposal
- Software feasibility demonstration
- Final Complete Draft of Final Paper
- Final paper
- Final code
- Final project demonstration
- Poster
- Web page
- Reflection

7.5. Milestones

- Literature review - 12/05/20
- Project proposal - 02/06/20
- Software feasibility demonstration - 03/08/20 - 11/08/20
- Final Complete Draft of Final Paper - 11/09/20
- Final paper - 21/09/20
- Final code - 25/09/20
- Final project demonstration - 05/10/20 - 09/10/20
- Poster - 12/10/20
- Web page - 19/10/20
- Reflection - 31/10/20

7.6. Work Allocation

Toshka Coleman – Central Web Portal

- Set Up Web Server
- Develop Web Portal
- Include Additional Features
- Perform Testing Related to Web Portal

Alex Priscu – Data Provider Interfaces

- Procure Data from Data Providers
- Develop Three Data Provider Interfaces
- Perform Testing Related to Data Provider Interfaces

Ashil Ramjee – Harvester

- Develop Repository Component
- Develop Harvester Component
- Perform Testing Related to Harvester

All group members will contribute to research, documentation, demonstrations and final project testing and deliveries.

REFERENCES

[1] Sally Chambers and Wouter Schallier, 2010. Bringing Research Libraries into Europeana: Establishing a Library-Domain Aggregator, *Liber Quarterly*.

[2] Martin Doerr, Stefan Gradmann, Steffen Hennicke, Antoine Issac, Carlo Meghini, Herbert van de Sompel. 2010. The Europeana data model (edm). In *World Library and Information Congress: 76th IFLA general conference and assembly.* 10-15.

[3] Nuno Freire, Antoine Isaac, Glen Robson, John Howard, and Hugo Manguinhas, 2017. A survey of Web technology for metadata aggregation in cultural heritage. Information Services & Use, Vol. 37, Issue 4, 425–436.

[4] Nuno Freire, Rene Voorburg, Roland Cornelissen, Sjors de Valk, Enno Meijers, E and Antoine Isaac, 2019. Aggregation of Linked Data in the Cultural Heritage Domain: A Case Study in the Europeana Network. Information, 10, 252.

[5] 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.

[6] Michele Pickover, Dale Peters. 2002. DISA: An African Perspective on Digital Technology. Innovation, 24.

[7] Hanumat G. Sastry, Lokanatha C. Reddy. User interface design principles for digital libraries. *International Journal of Web Applications*, *1*(2), 86-91. http://dirf.org/ijwa/v1n20109.pdf

[8] Hussein Suleman, Marc Bowes, Matthew Hirst and Suraj Subrun. Hybrid online-offline digital collections, 2010. Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '10, ACM Press, 421-425.

[9] Hussein Suleman, 2019. Reflections on Design Principles for a Digital Repository in a Low Resource Environment. Proceedings of HistoInformatics Workshop 2019, 13 September 2019, Oslo, Norway, CEUR.

[10] Hussein Suleman, Tatenda Chipeperekwa and Lawrence Webley, 2011. Creating a National Electronic Thesis and Dissertation Portal in South Africa. Proceedings of 14th International Symposium on Electronic Theses and Dissertations.

[11] Hussein Suleman, 2012. Design and architecture of digital libraries. Facet Publishing.

[12] 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.

[13] Hussein Suleman, 2008. An African Perspective on Digital Preservation, in Post-Proceedings of International Workshop on Digital Preservation of Heritage and Research Issues in Archiving and Retrieval, Kolkata, India, 29-31 October 2007. Available http://www.husseinsspace.com/research/publications/iwdph_2007_african.pdf

APPENDIX

A. Risk Matrix and Risk Management Strategies

Risk Condition	Consequence	Category	Probability	Impact	Mitigation	Monitoring	Management
Not able to deliver all requirements on time.	Crunchtime will be necessary for project completion. Working overtime will cause burnout of team members. The project schedule will be changed, as requirements will need to be completed at a later stage or removed entirely.	Development & Management	Medium	Critical	Realistically determine the project scope from the beginning. Revise the requirements and functionality frequently and thoroughly. Create a Gantt chart to keep track of milestones which prioritizes the most essential items first.	Record and monitor weekly meetings to check if the project is up to date with the project schedule. Ensure Gantt Chart is consistently checked and updated.	Communicate with the team as to which deliverables are still feasible to hand in and readjust the project scope.
Delay in the provision of data from data providers.	Project tasks requiring the data are delayed.	Communications	Medium	Critical	Confirm stakeholders' investment and interest in the project is communicated via email weekly. Ask for sample data before software development.	Establish consistent communication with the supervisor regarding the status of data provision. Set up a due date for the submission of data.	Begin software development with sample data. Hold a critical meeting with data providers.
Loss of project work.	The project will fall behind schedule. The amount of work will increase in order to restore lost work.	Maintenance	Medium	Critical	Ensure multiple backups of work are created regularly and stored externally from the project.	Monitor if project work is uploaded daily to a remote repository using a version control system.	Assess the amount of work lost, communicate the impact with the supervisor and re-prioritise the project so that the most important features are implemented before the deadline.
Unable to complete allocated components in parallel due to dependencies.	The project development will be shifted. Certain deliverables will not meet their allocated deadlines.	Development & Management	Medium	Critical	Create a Gantt Chart with correct dependencies of components clearly defined.	Monitor that the order of components in progress aligns with dependencies on Gantt Chart in weekly meetings.	Review dependencies and halt the progress of dependent components. Implement software test stubs to act as a temporary stimulation for the relative components.
Unable to integrate components.	System will not be functional and will be unable to perform the required user-end services.	Development & Management	Low	Catastrophic	Integrate components at an early stage. Apply extreme programming practices and test-driven development.	Review Gantt chart for component progress and dependencies.	Reduce the complexity of components so that focus shifts on components connecting rather than each component performing correctly.
Poor communication among the members of the team.	Team members will be unaware of project progress and challenges faced in certain tasks. Dependent tasks will be negatively affected by lack of cohesion.	Communications	Low	Critical	Set up a schedule for regular meetings. Use a platform that enables the project to be openly accessible by all team members.	Register that everyone is present during team meetings. Take minutes for every meeting.	Organise a meeting to discuss and to collectively deal with the consequences. Discuss and implement an improved medium for communicating.
Undecidability of methods for conducting user evaluations due to COVID-19 pandemic.	Incomplete or inadequate testing of the system.	Testing	High	Marginal	Plan various methods of testing according to all possible circumstances.	Monitor status of circumstances and prepare accordingly.	Choose the most suitable method planned or create a better-suited method based on circumstances,
Team members becoming ill/sick.	The sick team member will be unable to complete the assigned tasks. The team will fall behind the project schedule, since work will have to be reassigned to healthy team members.	Health & Safety	Medium	Critical	Delegate amounts of work that does not require overtime. Check-in with team members regularly and delegate work before a member becomes too ill to work.	Organise team meetings to receive feedback on workload and check up on the health of team members. Team members should report early if feeling sick.	Determine the impact of delayed work and fairly distribute tasks that other team members may complete. Reduce the amount of work by cutting back on scope.
Regional load shedding.	The team will resort to using battery power for coding the software. Web-based prototypes might not be tested fully.	Environmental	Medium	Marginal	Schedule project development to workaround load shedding hours.	Monitor what work can be done offline, and when power is on, to upload the offline work. Track load shedding timetable so the project schedule incorporates it.	Resort to using battery-powered devices for coding software.

B. Gantt Chart