ARCHMAN: Simple Archive Management Tool

PROJECT PROPOSAL

Laylaa Varachia
Department of Computer Science
University of Cape Town
Cape Town, South Africa
vrclay001@myuct.ac.za

Rushil Vallabh
Department of Computer Science
University of Cape Town
Cape Town, South Africa
vllrus002@myuct.ac.za

Kamilah Louw
Department of Computer Science
University of Cape Town
Cape Town, South Africa
lwxkam003@myuct.ac.za

1 PROJECT DESCRIPTION

Digital archives contain collections of digital objects such as numerical data, images, maps, videos, and audio files that are available via the internet [1]. In most cases, digital archives are easily accessible and free, making them ideal to preserve history and culture.

In South Africa, projects such as the Bleek and Lloyd Collection [3] and Emandulo [4] give the general public the opportunity to browse through artefacts that document the language and culture of indigenous groups in Southern Africa.

Emandulo is a project belonging to the Archives and Public Culture (APC) centre. It is made up of subcollections, each comprising of its own hierarchical organisation. Emandulo was initially built using AtoM [5], which was later successfully replaced by Simple DL [2].

Simple DL is a system that essentially has no software being executed at runtime, making it a 'software-less' solution to implement digital libraries with ease [2]. The Simple DL toolkit provides a means for archivists to create digital libraries that do not require much computational power or maintenance, enabling them to be implemented successfully in low-bandwidth environments.

As it stands, Simple DL has limited functionality for its administrative system [2]. Administrators are unable to change configurations or edit files online. In order to edit a file that is part of an archive, it must be downloaded, edited using external software and finally, reuploaded. Additionally, the current user interface for the administrative system is unintuitive and difficult to navigate. There is a need to extend and improve the current functionality of Simple DL's administrative system. The solution is a redesigned version of the administrative system that is easy to use and can accomplish the following tasks: change configurations, manage files and edit files online.

2 PROBLEM STATEMENT AND AIMS

The current administrative system of Simple DL lacks certain functionality. Firstly, administrators are unable to edit files online without downloading them first, using external software to edit the files and then reuploading the files. Secondly, the configurations of the site cannot be changed directly from the administrative interface, but rather has to be changed from the command line. Configurations can be complex to set up, and many configurations currently use predefined rules that limit the extent of what the user can do with currently available configurations. Currently, these rules cannot be edited with the current system. Lastly, the file management system is unintuitive and lacks a user-friendly interface.

2.1 Aim

The goal of this project is to develop a usable administrative system for Simple DL. This will be achieved by allowing administrators to change configurations, manage files and edit files online.

- *Configuration Management:*
    - Create an easy-to-use interface for the configuration of the system
    - Ensure all existing functionality of the current configuration system are included and work as intended
    - Rebuild configurations if needed based on needs of system.
- *File Management:*
    - Create an intuitive, usable interface for the management of files and folders on the system
    - Ensure the current functionality, such as uploading, deleting and creating of files and folders remains.
- *In-Place Editing:*
    - Add the ability to open and view files that are already uploaded
    - Add the feature to edit existing CSV and XML files online, through the browser

2.2 Potential Users

Potential users of the administrative system could be researchers, historians and students. Anyone interested in curating their own digital archive may also be potential users.

2.3 Requirements

The requirements listed below are initial requirements, which will be refined when requirements gathering takes place.

- The new administrative system must allow administrators to change the configuration of the system from the interface itself without having to use the command line.
- The configurations will need to include tasks such as changing the appearance of the website, managing directives, managing accepted formats of online resources, etc.
- A more intuitive and visually appealing file management system should be implemented.
- Users should be able to create and delete files or folders as well as easily move files into folders or delete files from folders.
- The editing of files should be done in-place [6] and online through the browser, making it easier and faster to make changes to files.
- The Simple DL toolkit was designed to be as simple as possible. Therefore, above all, the additional functionality of the administrative system should be low maintenance and easy to use.

2.4 Project Scope

The scope of the project only includes the replacement of Simple DL's administrative system. Additional functionality will be added on top of the toolkit's existing code.

2.5 Stakeholders

- *Professor H Suleman (supervisor) and Dr. Abayomi Agbeyangi (co-supervisor):* Providing guidance throughout the project. Responsible for ensuring that the work produced by the team is of a high standard.

- *Project Team:* Responsible for carrying out the project according to the requirements set out by the client and adhering to the project schedule.

- *Clients:*
  - *Emandulo:* proposed the project and specified high-level requirements.
  - *Professor H Suleman:* specified the exact features for the system.

- *Users*:
  - Researchers, historians and students.

3 PROCEDURES AND METHODS

3.1 High-Level Design

The administrative system of Simple DL consists of a frontend and a backend. The system will be split vertically to ensure that each team member can work independently of one another. This modular approach considers the risk of a team member not completing their part of the project. Refer to Figure 1 for a diagram of the high-level system architecture.

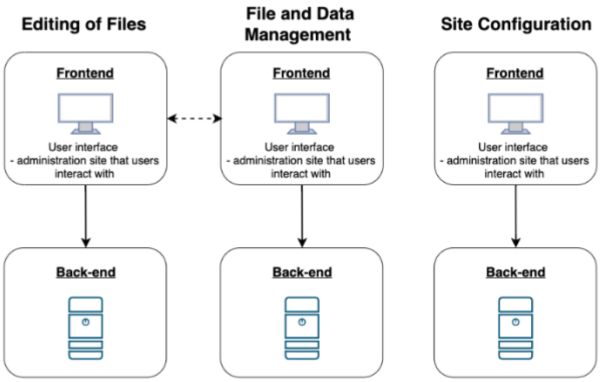

Editing of Files
File and Data Management
Site Configuration
Frontend
User interface - administration site that users interact with
Frontend
User interface - administration site that users interact with
Frontend
User interface - administration site that users interact with
Back-end
Back-end
Back-end

Figure 1: High-level System Architecture

3.2 Development Platform

The ARCHMAN will be designed as a website and will be compatible with majority of commonly used browsers, such as Google Chrome, Microsoft Edge, and Internet Explorer. Typescript will be used for the frontend and python will be used for the backend. A simple python Web-hosting framework such as Flask [8], Web2py [9] or Python CGI [10] will be used to manage the website.

3.3 Implementation Strategy

We will be using an iterative approach to the implementation strategy, where each component of the project will be carried out simultaneously by the respective team member responsible. An iterative approach recognizes that software development can be volatile and allows the team to respond to changes that emerge throughout the project.

The agile methodology [11] is ideal for small groups and thus will be used for project management. The Kanban [12] method will be used to implement the agile methodology. A Kanban board will be used to visualize the workflow and limit the work in progress at each stage. The board will show the assigned work of each team member along with associated priorities, allowing developers to focus on a few tasks at a given time. Kanban makes use of a continuous workflow in which work items flow from one column, representing a workflow, to the next. The workflow stages we plan to include are *Backlog*, *Prioritized*, *In Progress*, *In Testing*, *Complete* and *Passed UAT*.

Implementation of the project will be done in 4 phases. Refer to Appendix A for a Gantt chart showing a timeline of the phases.

- *Phase 1: Planning*
  Consists of installing all required software such as Simple DL and development environments. The ethics application will also take place in this stage. As soon as ethics clearance is granted, requirements gathering through interviews will take place.
- *Phase 2: Design*
  Based on the requirements gathered in the planning phase, a low fidelity prototype [13] will be created. The use of a low-fidelity prototype is advantageous because they are simple, inexpensive, and quick to create, allowing for the exploration of alternative designs and ideas. We will make use of Nielsen's 10 Usability Heuristics [20] to conduct heuristic evaluations [14], first of the low fidelity paper protype and later of the evolutionary prototype. The heuristic evaluations will be conducted with the project supervisors who have the technical expertise to provide adequate feedback.
- *Phase 3: Development*
  Initial development will begin using the resulting prototype from the design phase. Core functionality regarding the editing of files, changing configurations and managing files will be implemented.
- *Phase 4: Finalization*
  The software will be refined and any additional requirements that arose after the planning phase may be satisfied based on time constraints. The user interface will be finalized towards the end of development and system evaluation will take place.

Integration of the project with the existing Simple DL toolkit is trivial since the administrative system will have its own URL and a simple redirect from an archive will allow administrators to access the administrative system.

3.4 Expected Challenges

Implementation of each component of ARCHMAN will be a challenge as the team is not very well versed with the implementation of digital archives and unfamiliar with Simple DL. Further research and experimentation with the Simple DL toolkit will be required from the team to gain a better understanding of how it works before building onto the existing administrative system.

The functionality needed to allow administrators to change configurations and manage files requires the team members responsible for those aspects to understand the way in which the current configuration settings and file management works before they can be improved. This may delay the beginning of the implementation.

Since the participants required to test the system need to have had some experience with Simple DL, finding such persons who are both willing to participate and have some expertise, all within a short timeframe, may prove to be a challenge.

Lastly, overcoming the learning curve faced by one of the teammates who has not had experience with TypeScript in the past might pose difficulties during the initial stages of the project's implementation. Overcoming this learning curve might be a challenge.

3.5 System Evaluation

For testing purposes, we will be using a digital archive from Emandulo to test whether the functionality we add to the Simple DL administrative system works as expected with an archive.

- *Performance Testing:*
  To test the system, each team member will conduct unit tests for the features that they implement. Once features are implemented, we will conduct system performance tests [15] such as load-testing to examine the speed and reliability of the system. Load testing will be used for the file editing tool and the file management tool to determine the impact that many or very large files have on the performance of the system. Additionally, functional testing will be used to ensure that all the functions have been implemented correctly.

- *Usability Testing:*
  Users will include people in the digital library lab and the FHYA (Five-hundred-year archive).

  To test the usability of the system, custom Likert scales [19] containing questions specific to the administrative system of Simple DL will be used.

  We will be using a qualitative approach to evaluate the system. Some metrics to be assessed are:

  - How easy is the system to use?
  - How intuitive is it to find what a user is looking for?
  - How learnable is the system?
  - How comfortable and confident does the user feel while using the system?

  Additionally, we plan to use Nielsen's Heuristics once again to conduct a final evaluation with the users. This easy and inexpensive approach will ensure that any remaining errors are identified, and all possible improvements are made.

  The heuristics we will be using are [29]:

  - Visibility of system status
  - Match between system and real world
  - User controls and freedom

- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recover from errors
- Help and documentation

Interviews may also be conducted to get suggestions from the participants on how to improve the system.

- *Acceptance testing:*
  Acceptance tests [16] will be created prior to the implementation of any requirements. These tests will be drawn up in conjunction with the client and serves as clarification of what needs to be done to ensure a high-quality product is produced. A workflow item will only move to the final stage, *Passed UAT*, on the Kanban board (refer to section 3.3) once acceptance tests are complete and the client is satisfied.

4 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

Ethical clearance will be obtained through the UCT Research Ethics Committee before any interaction with people. Once ethical clearance has been obtained, the team can start the process of requirements gathering.

Human participants will be needed to test the system. Due to the nature of this project being a software system, participants will be at minimal risk. There will be no rewards or compensation offered to participants and thus, no coercion. Participants will include existing users of Simple DL who will benefit from an improved administrative system. The project supervisor will assist with getting in touch with current users of Simple DL. Participant selection will not be biased since the team will not have any influence on the selection of participants. The project will be explained beforehand, and participants will be informed that they can withdraw at any time. Personal information of the participants will remain private unless the participant wishes otherwise. If any interviews are going to be recorded, the participant will be informed, and consent will be obtained.

Results will be represented fairly and accurately. Appropriate parties will be credited for their work where applicable throughout the project.

The final project will remain as property of the UCT Department of Computer Science and will be available for download through the project website.

5 RELATED WORK

5.1 In-Place Editing

In-place editing is sometimes referred to as live site editing, in context editing or in-situ editing [36]. It is commonly used for the editing of Web pages. The in-place editing model allows users to make changes to the content of Web pages. Sparrow [27] was the first in-place editing system for the Web. Other examples of in-place editing systems for Web pages are DirectEdit and ISAWiki [37]. All these systems share philosophies of a Wiki but what differentiates them is that Wikis, at the time of development, required users to have HTML knowledge where these systems did not. The functionality offered by Sparrow differs from that required by Simple DL in several ways. Firstly, Sparrow focusses on community-shared documents meaning that there is a single author who is the creator of a document and then users that form part of the community can all edit the content. Simple DL [2] will only allow administrators to make changes to any documents that are part of a specific archive. Secondly, what is meant by ‘documents’ differs in the context of Simple DL and Sparrow. Sparrow refers to a web page as a community shared document while ‘documents’ in the context of Simple DL would be files in CSV or XML format that form part of an archive’s configuration.

5.2 Configuration Management

In the past, system administrators were necessary to set up digital library tools since configuration interfaces were not given attention [21]. Package management tools such as MSI and emerge can be used to mitigate end-user installation of these packages. Currently, it is up to the repository to present the configuration settings in an appealing interface. The web interface for DSpace uses JavaServer Pages (JSP) for its configuration or the Apache Cocoon framework (XMLUI) [22]. In terms of a configuration task such as access control, Fedora's digital repository uses XACML (eXtensible Access Control Markup Language) for supporting users and accessing specific digital objects. However, users still found some difficulty with configuration tasks such as changing the overall layout, working with HTML code from many JSP files [21]. A solution proposed suggested the use of a high-level system tool to help users by removing the lower-level details. This relates to the configuration management part of this project as it is stated that there is not much work done on administrative usability and more work is required to develop an easy tool for all users. The work being done currently is not sufficient. Improving on it or creating a new methodology will help develop a simpler configuration interface.

A Model-View-Controller Wizard with a persistence layer was used to create an interface for configuration management based on the requirements of a digital repository [30]. When a user opens the wizard, they are welcomed with a message and the purpose of the wizard. The various digital library’s configuration is detailed, and users have the option to click on them to view more information such as the general settings of configuration or changing the home directory of the repository. Once all configuration has been made, they are prompted with a summary page of the configurations and the option to go back to change if necessary. The final page is a confirmation that the changes have been made. This can be used as a foundation for creating the user interface of this project as it does cover principles conveyed in Nielsen heuristics for building a good user interface and can be applied to other parts of the project as well.

5.3 File Systems

File management software includes some popular ones such as Google Drive [31], Dropbox [32] and OneDrive [38]. Some similar toolkits such as DSpace and EPrints also have file management functionalities [26]. DSpace organizes data in such a way that it reflects the structure of the organization using it [25]. EPrints has a less structural division of sections and collections with the central concept being that all files are equivalent and do not form a hierarchical system. Just as with DSpace, users can upload documents with the necessary metadata for the records by filling out a web form with all the information.

Google Drive allows users to create folders, move items to a folder, make copies of files or delete a file or a folder [33]. The names of the folders are defined on Google Drive when creating the folder. To move a folder or a file, users can manually do this by right-clicking on the item they want to move, or they can make use of the drag and drop functionality, which allows users to drag items to new locations and drop them by releasing the file or folder. Copying a file/folder works the same as one would normally copy and paste by right-clicking and deleting a file/folder is also achieved by right clicking and removing the folder. Items can also be restored if they are accidentally deleted.

Dropbox offers similar functionality, however; functionality capabilities differ depending on the users' account [34]. For example, once a file is deleted a user may recover the file up to 30 days after deletion for the Dropbox Basic, Plus and Family accounts and up to 180 days for the Dropbox Professional and team accounts.

This kind of implementation will not be used in this project since the aim is not to make money or force users to pay for specific accounts to have access to certain benefits.

As with Google Drive, OneDrive also allows users to upload, create, delete and share files and folders [35]. OneDrive makes use of both a manual method and a drag and drop method for moving and copying and pasting files and folders. New files and folders can also be created in OneDrive using the "New" button. This is also where the name of the file or folder can be entered. Deleted files and folders can be restored from the recycling bin however if the user is signed into OneDrive with a Microsoft account, items in the bin will be automatically deleted after 30 days but if the user is signed in with a work or school account, items are only removed from the bin after 93 days unless the administrator changes this.

The main functionalities such as deletion, creation, copying and pasting, and uploading of files and documents will be implemented in the project. However, users will not be required to pay for any additional features. The restoring of items will be implemented as users could accidentally delete items and may need to recover them. The drag-and-drop functionality is a nice option to have as it is very intuitive. All these functionalities will be accessible through the administration site.

6 ANTICIPATED OUTCOMES

6.1 System Outcomes

A fully functional administrative system for Simple DL that can run through a browser. The features that will be developed are as follows:

- *A file management tool:*

  This will allow users to create and delete files and folders as well as move files in and out of folders. The improved file management system should be more intuitive and visually appealing than the existing one.

- *A configuration customization tool:*

  This tool will allow users with no CSS or HTML knowledge to customize their digital archives without having to use a command line. Customizations include changing the appearance of the website, managing directives, and managing accepted formats of online resources.

- *A file editing tool:*

  This tool will allow users to view and edit CSV and XML files online, through the browser. This should eliminate the need for users to download the file, use external software to edit it and then reupload the file.

6.2 Expected Impact

By improving the administrative system of Simple DL, making the toolkit easier to use, more customizable and more professional, we hope to give everyone, especially those lacking the expertise, the ability to create their own digital archives.

6.3 Key Success Factors

The success of the project is based on the ability of administrators to change configurations, manage files, and edit files that are part of archives built using the Simple DL toolkit. In general, the system should be usable, satisfy the project requirements and function as expected without any bugs. There are several software success factors that can be used to determine whether we have achieved these goals.

- The ability to open and view files that are already uploaded.

- The ability to edit existing CSV and XML files online, through the browser.
- The ability to enter configuration settings from the administrative interface.
- The ability to change administrative settings.
- The ability to change accepted file types.
- The ability to create, delete and modify folders.
- The ability to upload files and delete files.
- The ability to move files in and out of folders.

7 PROJECT PLAN

7.1 Risks

Refer to Appendix B for a risk matrix.

7.2 Resources Required

This project requires various software for the development of the system as well as users to conduct testing. There will be no specialized equipment beyond personal computers needed for development.

Software:

- Visual studio code with development environments set up for typescript and python.
- Simple DL downloaded on personal computers.
- Python based Web hosting framework.
- Emandulo suggested the project, so the team will use their archives for the data

Hardware:

- Server space

People:

- 2-3 people from Emandulo for requirements gathering.
- 6-8 users of Simple DL for testing.
- Supervisors for general guidance.

7.3 Deliverables

Key deliverables include the following:

- Project proposal
- Low fidelity prototype
- Evolutionary prototype
- Feasibility demonstration
- Project paper
- Project code
- Project poster
- Project website
- School of IT demonstration

7.4 Project Milestones

Below is a list of the project milestones. These milestones include the official deadlines set out for the honour's project as well as the deadlines we have set for ourselves as a team.

- Project proposal draft 21/04/2023
- Proposal presentation dry run 24/04/2023
- Proposal presentation 25/04/2023
- Final proposal 29/04/2023
- Project progress demonstration 17/07/2023 – 21/07/2023
- Ethics application 12/05/2023
- Requirements gathering 17/05/2023
- Low fidelity prototype 01/06/2023
- Evolutionary Iteration 1 08/07/2023
- Evolutionary Iteration 2 15/08/2023
- Draft of project paper 28/08/2023
- Final project paper 11/09/2023
- Project code 15/09/2023
- Project poster 09/10/2023
- Project website 16/10/2023
- School of IT demonstration 24/10/2023

Refer to Appendix A for a Gantt chart detailing the project timeline containing all project milestones.

7.5 Work Allocation

- *Laylaa Varachia:* The in-place editing of files.
- *Rushil Vallabh:* Site configurations.
- *Kamilah Louw:* File management.

REFERENCES

[1] Cheryl Mason Bolick. 2006. Digital Archives: Democratizing the Doing of History. *International Journal of Social Education* 21, 1 (2006), 122-134.

[2] Hussein Suleman. 2021. Simple DL: A Toolkit to Create Simple Digital

[3] Libraries. *Towards Open and Trustworthy Digital Societies* (November 2021), 325-333. DOI:https://doi.org/10.1007/978-3-030-91669-5_25

[4] Hussein Suleman. 2007. Digital Libraries Without Databases: The Bleek and Lloyd Collection. Research and Advanced Technology for Digital Libraries (2007), 392-403. DOI:https://doi.org/10.1007/978-3-540-74851-9_33

[5] EMANDULO. Retrieved April 12, 2023 from http://emandulo.apc.uct.ac.za/metadata/Presentations/7549/index.html

[6] Jessica Bushey. 2011. International Council on Archives (ICA) "Access to Memory" (AtoM): Open-source software for archival description. (2011).

[7] Brook Jesse Novak. 2010. Seamlessly Editing the Web (Thesis, Master of Science (MSc). University of Wakaito, New Zealand (2010).

[8] Dinnie Muslihat. 2018. Agile Methodology: An Overview. *Zenkit*. Retrieved April 19, 2023 from https://zenkit.com/en/blog/agile-methodology-an-overview/

[9] Fankar Armash Aslam, Hawa Nabeel Mohammed, and P.S. Lokhande. 2015. Efficient way of web development using python and flask. *International Journal of Advanced Research in Computer Science* 6, 2 (March 2015), 54-57.

[10] Massimo Di Pierro. 2011. web2py for Scientific Applications. *Computing in Science & Engineering* 13, 2 (March 2011), 64-69. DOI:https://doi.org/10.1109/mcse.2010.97

[11] Aaron Watters, James C. Ahlstrom, and Guido Van Rossum. 1996. Internet programming with Python. *Henry Holt and Co., Inc.* (1996).

[12] Gillian M. Nicholls, Neal A. Lewis, and Ted Eschenbach. 2015. Determining When Simplified Agile Project Management Is Right for Small Teams. *Engineering Management Journal* 27, 1 (2015), 3-10. DOI:https://doi.org/10.1080/10429247.2015.11432031

[13] Muhammad Ovais Ahmad, Jouni Markkula, and Markku Oivo. Kanban in software development: A systematic literature review. *2013 39th Euromicro Conference on Software Engineering and Advanced Applications* (September 2013), 9-16. DOI:https://doi.org/10.1109/seaa.2013.28

[14] Reinhard Sefelin, Manfred Tscheligi, and Verena Giller. 2003. Paper prototyping - what is it good for? *CHI'03 extended abstracts on Human factors in computing systems* (2003), 778-779. DOI:https://doi.org/10.1145/765891.765986

[15] Jacob Nielsen. 1995. How to conduct a heuristic evaluation. 1, 1 (1995).

[16] B.M. Subraya and S.V. Subrahmanya. 2000. Object driven performance testing of Web applications. *Proceedings First Asia-Pacific Conference on Quality Software* (October 2000), 17-26. DOI:https://doi.org/10.1109/apaq.2000.883774

[17] Ken Pugh. 2010. Lean-Agile Acceptance Test-Driven-Development. Pearson Education.

[18] James R. Lewis. 2018. The System Usability Scale: Past, Present, and Future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577-590. DOI:https://doi.org/10.1080/10447318.2018.1455307

[19] Lewis R. James. 1995. Computer system usability questionnaire. *International Journal of Human-Computer Interaction* (1995).

[20] Susan Jamieson. 2004. Likert scales: how to (ab)use them. *Medical Education* 38, 12 (2004), 1217-1218. DOI:https://doi.org/10.1111/j.1365-2929.2004.02012.x

[21] Jacob Nielsen. 1993. Usability Heuristics. *Usability Engineering* (1993), 115-163. DOI:https://doi.org/10.1016/b978-0-08-052029-2.50008-5

[22] George Pyrounakis, Mara Nikolaidou, and Michael Hatzopoulos. 2014. Building Digital Collections Using Open Source Digital Repository Software: *International Journal of Digital Library Systems* 4, 1 (2014), 10-24. DOI:https://doi.org/10.4018/ijdls.2014010102

[23] Nils Körber and Hussein Suleman. 2008. Usability of Digital Repository Software: A Study of DSpace Installation and Configuration. *Digital Libraries: Universal and Ubiquitous Access to Information* (December 2008), 31-40. DOI:https://doi.org/10.1007/978-3-540-89533-6_4

[24] Hesamedin Hakimjavadi and Mohamad Noorman Masrek. 2012. Evaluation of interoperability protocols in repositories of electronic theses and dissertations. *Journal of Basic and Applied Scientific Research* 2, 47 (2012), 10843-10852. DOI:https://doi.org/10.1108/00330331211296303

[25] R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan, and M. Smith. 2003. The DSpace institutional digital repository system: current functionality. *2003 Joint Conference on Digital Libraries, 2003. Proceedings.* (2003). DOI:https://doi.org/10.1109/jcdl.2003.1204846

[26] Lakshmi Verma and Nishant Kumar. 2018. Comparative Analysis of Open Source Digital Library Softwares: A Case Study. *DESIDOC Journal of Library & Information Technology* 38, 5 (2018), 361-368. DOI:https://doi.org/10.14429/djlit.38.5.12425

[27] Mike R. Beazley. 2010. Eprints Institutional Repository Software: A Review. *Partnership: The Canadian Journal of Library and Information Practice and Research* 5, 2 (2010). DOI:https://doi.org/10.21083/partnership.v5i2.1234

[28] Anders Orn. 2017. The Pros and Cons of the System Usability Scale (SUS) - Research . *Research Collective*. Retrieved April 19, 2023 from https://research-collective.com/sus/

[29] Jakob Nielsen. 1994. 10 Usability Heuristics for User Interface Design. *Nielsen Norman Group*. Retrieved April 19, 2023 from https://www.nngroup.com/articles/ten-usability-heuristics/

[30] Rodrygo L. T. Santos, Pablo A. Roberto, Marcos André Gonçalves, and Alberto H. F. Laender. 2006. Design, Implementation, and Evaluation of a Wizard Tool for Setting Up Component-Based Digital Libraries. *Research and Advanced Technology for Digital Libraries* (2006), 135-146. DOI:https://doi.org/10.1007/11863878_12

[31] Google, "Easy and secure access to your content," [Online]. Available: https://www.google.com/drive/. [Accessed 19 April 2023].

[32] Dropbox, "Everything teams need, all in one place," [Online]. Available: https://www.dropbox.com/official-teams-page?&_tk=paid_sem_goog_biz_b&_camp=1033325450&_kw=dropbox%7ce&_ad=650036281368%7c%7cc&gclid=CjwKCAjwov6hBhBsEiwAvrvN6Dv0nGhb9ADEsauVS6TrNY53L7anME70yLjDWydfq6-dTEd-XgRT3RoC6tYQAvD_BwE&gclsrc=aw.ds. [Accessed 19 April 2023].

[33] Organize your files in Google Drive - Computer - Google Drive Help. Retrieved April 19, 2023 from https://support.google.com/drive/answer/2375091?hl=en&co=GENIE.Platform%3DDesktop

[34] How to Recover Deleted Files in Dropbox - Dropbox Help. Retrieved April 19, 2023 from https://help.dropbox.com/delete-restore/recover-deleted-files-folders

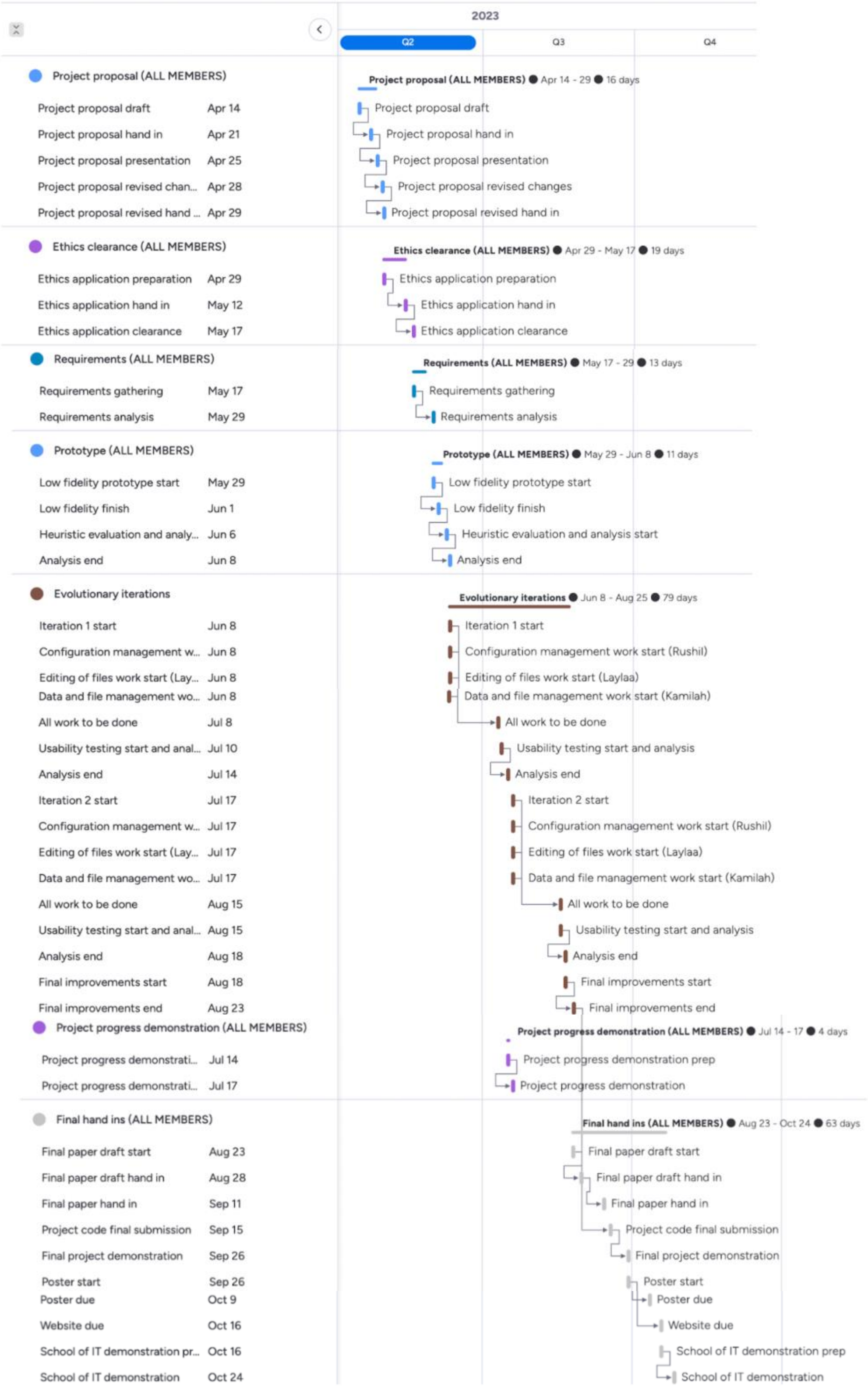
[35] OneDrive help & learning. Retrieved April 20, 2023 from https://support.microsoft.com/en-us/onedrive

[36] Peter Deutch and Butler Lampson. An Online Editor. *Communications of the ACM* 10, 12 , 793-799. DOI:https://doi.org/doi.org/10.1145/363848.363863

[37] A. Di Iorio and F. Vitali. 2005. Web Authoring: A Closed Case? *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (2005), 95. DOI:https://doi.org/10.1109/hicss.2005.682

[38] Microsoft, "Securely save and share what's important with OneDrive," [Online]. Available: https://www.microsoft.com/en-us/microsoft-365/onedrive/online-cloud-storage-b?ocid=cmmpt8dho3c&activetab=tabs%3afaqsregion3. [Accessed 19 April 2023].

APPENDIX A: Gantt Chart



● Project proposal (ALL MEMBERS)	
Project proposal draft	Apr 14
Project proposal hand in	Apr 21
Project proposal presentation	Apr 25
Project proposal revised chan...	Apr 28
Project proposal revised hand ...	Apr 29
● Ethics clearance (ALL MEMBERS)	
Ethics application preparation	Apr 29
Ethics application hand in	May 12
Ethics application clearance	May 17
● Requirements (ALL MEMBERS)	
Requirements gathering	May 17
Requirements analysis	May 29
● Prototype (ALL MEMBERS)	
Low fidelity prototype start	May 29
Low fidelity finish	Jun 1
Heuristic evaluation and analy...	Jun 6
Analysis end	Jun 8
● Evolutionary iterations	
Iteration 1 start	Jun 8
Configuration management w...	Jun 8
Editing of files work start (Lay...	Jun 8
Data and file management wo...	Jun 8
All work to be done	Jul 8
Usability testing start and anal...	Jul 10
Analysis end	Jul 14
Iteration 2 start	Jul 17
Configuration management w...	Jul 17
Editing of files work start (Lay...	Jul 17
Data and file management wo...	Jul 17
All work to be done	Aug 15
Usability testing start and anal...	Aug 15
Analysis end	Aug 18
Final improvements start	Aug 18
Final improvements end	Aug 23
● Project progress demonstration (ALL MEMBERS)	
Project progress demonstrati...	Jul 14
Project progress demonstrati...	Jul 17
● Final hand ins (ALL MEMBERS)	
Final paper draft start	Aug 23
Final paper draft hand in	Aug 28
Final paper hand in	Sep 11
Project code final submission	Sep 15
Final project demonstration	Sep 26
Poster start	Sep 26
Poster due	Oct 9
Website due	Oct 16
School of IT demonstration pr...	Oct 16
School of IT demonstration	Oct 24

2023
Q2
Q3
Q4
Project proposal (ALL MEMBERS) ● Apr 14 - 29 ● 16 days
Project proposal draft
Project proposal hand in
Project proposal presentation
Project proposal revised changes
Project proposal revised hand in
Ethics clearance (ALL MEMBERS) ● Apr 29 - May 17 ● 19 days
Ethics application preparation
Ethics application hand in
Ethics application clearance
Requirements (ALL MEMBERS) ● May 17 - 29 ● 13 days
Requirements gathering
Requirements analysis
Prototype (ALL MEMBERS) ● May 29 - Jun 8 ● 11 days
Low fidelity prototype start
Low fidelity finish
Heuristic evaluation and analysis start
Analysis end
Evolutionary iterations ● Jun 8 - Aug 25 ● 79 days
Iteration 1 start
Configuration management work start (Rushil)
Editing of files work start (Laylaa)
Data and file management work start (Kamilah)
All work to be done
Usability testing start and analysis
Analysis end
Iteration 2 start
Configuration management work start (Rushil)
Editing of files work start (Laylaa)
Data and file management work start (Kamilah)
All work to be done
Usability testing start and analysis
Analysis end
Final improvements start
Final improvements end
Project progress demonstration (ALL MEMBERS) ● Jul 14 - 17 ● 4 days
Project progress demonstration prep
Project progress demonstration
Final hand ins (ALL MEMBERS) ● Aug 23 - Oct 24 ● 63 days
Final paper draft start
Final paper draft hand in
Final paper hand in
Project code final submission
Final project demonstration
Poster start
Poster due
Website due
School of IT demonstration prep
School of IT demonstration

APPENDIX B: Risk Matrix

Risk	Consequence	Impact(out of 10)	Probability(out of 10)	Mitigation	Monitoring	Management
Delay in obtaining databases to test	Limited time to test thus lack of evaluation and improving on design	3	1	Use sample database	Check in with supervisor for updates for the timeline for gaining the database	Check in supervisor weekly
Functionality of configurations, files management or file editing not working	System not functional thus goal/aims cannot be reached	10	4	Allocate enough time to ensure functionality is performed correctly.	Check in with supervisor weekly to see if project progress is where it is at	Check unit tests for each component to see if passed or failed
Unable to meet deadlines	Not able to complete project before hand in and unable to do adequate testing and evaluation on the system	10	1	Communicate effectively with project stakeholders well in advance to help where needed	Assess the timeline of each group member to see if anyone is behind	Schedule weekly meetings with supervisor
Loadshedding affecting working on project	Wasting unnecessary time	3	2	Plan ahead to make preparations for load shedding	Checking loadshedding schedule	Ensure backup plan
misunderstanding on tasks to accomplish	Wasted time going back and changing what has been done	6	3	Record all meetings for reference	Review tasks that were discussed in previous meeting at current meeting	Schedule weekly meetings with supervisor
Team member leaving honours	Project work not complete will lead to more work for each group member or delays where overlap is needed.	10	2	Have each group members work have minimum overlay with each other	Have check-ups with group members often	Split tasks independent to each other and where overlay is needed ensure ample time is given for consideration of delays.