Good!

See comments -h.

UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project
Final Paper 2022

Title: Simple Archive Management

Author: Kamilah Louw

Project Abbreviation: ARCHMAN

Supervisor(s): Professor Hussein Suleman

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	0 ?
System Development and Implementation	*0*	*20*	
Results, Findings and Conclusions	*10*	*20*	
Aim Formulation and Background Work	*10*	*15*	
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**	

✓

ABSTRACT SEEMS A BIT LONG

Type text here

The Simple Archive Management Project

Kamilah Louw
Department of Computer Science
University of Cape Town
Cape Town
LWXKAM003@myuct.ac.za

✓ ONE WORD

ABSTRACT

Simple DL is a digital archiving toolkit. It was built to enable users to build and maintain their own digital archives without prior technological expertise. This project is aimed at improving the current Simple DL system by introducing new features and creating a more usable interface. This paper focuses on the administrative system's data and file management aspects, encompassing tasks like uploading, deleting, creating and moving folders and files. The project aims to address key usability challenges while expanding the capabilities of the archival system. During requirements gathering, it was found that all the users preferred to work in one directory at a time rather than a tree-like structure. Many users also highlighted that the current system does not confirm when changes have been successful. The current system requires users to zip multiple files in order to do a batch upload which many users found to be frustrating. To improve the system, batch uploading, downloading and deleting should be possible. The tree-like structure will be removed with a new, directory-focused view. Each file will have a preview box to confirm that it is the right file the user is working with. The final product with the additional features will be tested by users using Nielsen's ten usability heuristics, usability testing, load testing and performance testing. Through a combination of agile methodologies and user-centered design, the project achieves improved system performance, increased user engagement, and a more accessible approach to building and managing digital libraries. Multiple iterations were incorporated while keeping users informed and allowing users to provide feedback. This allowed for improvements throughout the development process.

PAST TENSE

Additional points:

- Results from testing
- Future work
- Conclusions

?

KEYWORDS

Digital libraries, Simple DL, User Testing, File and Data Management, Nielsen's Ten Usability Heuristics

1 Introduction

Digital archives or digital libraries (DL) are collections of data items. They are used to store digitized resources in a safe and convenient way [1]. The Simple DL toolkit was designed with the goal of being as simple as possible while allowing users to access the digital libraries even when there is a computer system or network failure. This software toolkit was designed to assist users in the creation of digital libraries [2].

The Simple Archive Management project, also known as the ARCHMAN project, involves the design and implementation of an improved, usable user interface for administrators of the Simple DL toolkit. This paper will discuss the development of the file and data management of the administrative site. The paper will be divided into sections including a short review on related work, the design and implementation of the new and improved system, the results and a discussion of each, and finally, conclusions and future work.

This project was proposed by Emandulo staff. Emandulo is a project belonging to the Archives and Public Culture (APC) center [2]. It provides a platform for the general public to browse through collections of artefacts that document the culture and language of indigenous groups in South Africa. Simple DL was used in the creation of this repository. It enabled users to build and maintain the repository without prior technological expertise however, the administrative interface is not user-friendly. The aim of this project is to design and build a usable interface with additional functionality to improve the usability of the system. The interface should be simple enough for anyone to use but include all the necessary functionality for the management of files and data. ✓

✓

The improved system will be designed, prototyped and improved through MULTIPLE ~~numerous~~ iterations of user testing and development. Requirements gathering with Simple DL users and Emandulo staff will highlight the necessary improvements. Through a series of interviews and development cycles, an improved interface will be developed. It is important that this interface is usable and simple. Having a system that can perform the basic functionality for the management of data and files in the repository is important as it can improve productivity, simplify the management of files and data which can be a very difficult task since these repositories can have numerous items. Improving the interface will simplify tasks, speed up processes and allow the management of these archives to be a smoother process.

↑ PAST TENSE

✓ Keywords?

2 Related Work and Background

Digital libraries not only allow resources to be accessed remotely but they also offer a safe place to store records which could have historical importance. Digital libraries can store various types of resources for long periods of time while reducing the chances of these electronic records being stolen, lost, or possibly damaged. This can be advantageous for cases when the resources are rare, fragile or treasured [1] [3]. This section will shed light on some similar repository toolkits, highlight the importance and advantages of digital archives, and finally, discuss some familiar file and data management applications and their architecture.

2.1 Repository Toolkits

Simple DL is not the first toolkit developed for creating repositories. Some other toolkits include DSpace, EPrints and Greenstone. DSpace and EPrints are the most popular toolkits and are both web-based and have backend databases [2]. Greenstone requires installation of software to access collections, but it was designed to work both offline and online.

2.1.1 DSpace. DSpace represents a collaborative initiative involving the Massachusetts Institute of Technology (MIT) Libraries and Hewlett-Packard (HP) labs [4]. Operating as an open-source platform, it serves as a digital repository designed for housing content generated by different organizations and institutions for research and educational objectives. The communities within the system have collection of related content that are grouped together. Members of the communities can upload content directly via a Web interface or perform batch imports though a batch importer [5]. DSpace enables the capture of electronic material with corresponding metadata, allows for systematic listing of resources, and the preservation of digital data. It also assists in searching with the use of keywords and metadata. [6].

2.1.2 EPrints. EPrints was created by the University of Southampton in England. It is a free software application that allows the academic community to submit preprints, post prints and other academic publications via a web interface [4]. It uses conventional technology and runs entirely on Open-Source platforms. The administrator determines which metadata fields are stored for all resources and the administrative back end provides access to configuration options [7] [8]. EPrints makes use of MySQL, Oracle and PostgreSQL for the database management system. EPrints does not support bulk uploading.

2.1.3 Greenstone. Greenstone is an open-source software that was distributed under the GNU (a free and open-source operating system) General Public License (GPL) that is often used to develop and publish repositories online [6]. Although Greenstone collections can be transferred on CDROM ~~(Compact Disc Read Only Memory~~), assessing them requires the installation of software [2]. The University of Waikato's New Zealand Digital Library Project, UNESCO (United Nations Educational, Scientific and Cultural Organization), and the Human Info NGO worked together to build and distribute GSDL (Greenstone Digital Library). The software's purpose was to enable users build their own digital libraries. The Greenstone software builds data structures for searching and browsing from the provided resources as opposed to relying on any manual work [9].

2.1.2 Importance and Advantages. There are many advantages that come with digital archives. First, they do not require any physical space compared to a physical archive with physical files and document. Since all the resources are digitized, searching, browsing, organizing and managing these resources can be done effortlessly [10]. Digital archives also allow for remote access at any time, this also means that resources can be accessed by multiple users since the distribution does not depend on the number of copies available. Archives reduce the risk or wear and tear which consequently enables longer preservation and conservation since the original copy is never handled physically [1].

too strong

2.2 File and Data Management Applications

Management of resources in digital archives are important since these repositories could contain historically important documents. It is important that the structure of the archive is maintained, and files and folders are organized correctly. File and data management of electronic resources has been made possible through various platforms. Some commonly known platforms include Google Drive [4], Dropbox [5] and OneDrive [6]. Each of these platforms include different features but the basic functionality of storing and managing data are similar [3].

2.2.1 Google Drive. Google Drive allows users to store files in the cloud on Google's servers, share data, and sync files across devices. It also enables users to sign into several accounts and switch between them with ease. Additionally, it offers shared disks for customers to collaborate [7].

2.2.2 Dropbox. Dropbox, like Google Drive, allows users to sync, share and edit documents over the internet. File sharing and updates are possible since users connect to secure servers. This also allows for ease of accessibility [7].

2.2.3 OneDrive. By downloading the application, users can access their own files offline. Users can access their uploaded files from any device [7].

2.2.4 File and Data Management Platform Architecture. Each of the previously discussed applications are cloud storage systems. The basic services provided by each application are similar but differ in cost options for unlocking additional features and space [7]. The front end of these applications is responsible for user-access to files. It provides the user with an interface to interact with. The back end oversees user information security and is utilized by service providers. It is also where the physical data storage is located and can either be an internal protocol that makes use of particular or a traditional backend to physical disks [15].

The front end is connected to the backend via the internet. A cloud storage system requires a single data server that is connected to the internet. A client transfers files to the data server through the internet, where the data is saved. When a user needs information, they connect to the data server via a web-based interface. The server either returns the files to the user or grants them access to and manipulation of the server's files [16].

3 System Design and Implementation

This section will discuss the current functionality and user interface for the file and data management site of the Simple DL toolkit. It will also provide an overview of the approach taken and then dive deepener into the finer details of the steps taken in the approach. different word

3.1 The Current System

The current administrative site handles the files and data uploaded by administrators. The files used in the repository can contain audio files, portable document formats (PDFs), documents, videos, comma-separated value (CSV) files and many more. Although Simple DL can perform many actions such as uploading, deleting, creating and downloading files and folders, the interface is not usable and requires many additional steps for completion the of tasks. For example, to upload a zipped folder, the user is required to zip the folder and the upload it. This is because the current system automatically unzips folders. To upload multiple files, the user is required to zip the files and then upload them since there is no functionality to select multiple files for downloading, uploading or relocating. Some additional functionality that is completely missing from the system included a search bar, a sort-by function, a preview box that allows users to have a view of the file they are working with. Figure 1 below displays the current interface of the Simple DL administration site.

SOFTWARE DEPLOYMENT

3.2 Approach

This project required the administrative site to be rebuilt. In order to separate tasks and complete the project, the system was divided into three separate projects. This division enabled the team to use an agile approach on a higher level since each member could work on a specific part of the system independently of on another. Weekly meeting with our supervisor allowed for progress updates between iterations with users. The system architecture resulted in the model displayed in Figure 2. The editing of files is lightly connected to the file and data management system since each file in the system has metadata that should be editable. The goal is to create a more usable administrative interface with the file and data management functionality and interface being like Google Drive.

WHICH PART IS YOURS?

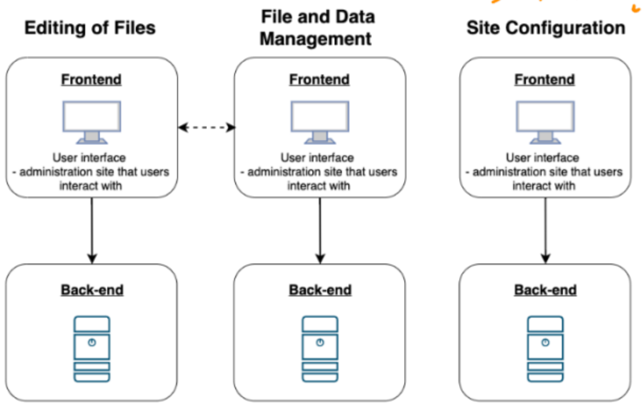
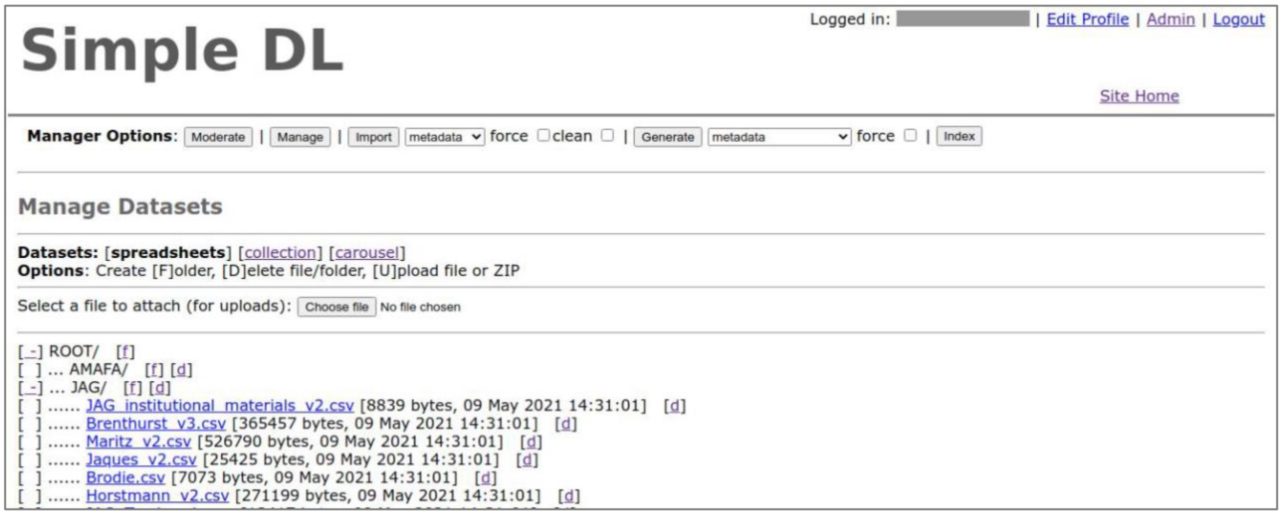


Figure 2: High-level system architecture



Simple DL

Logged in: [redacted] | Edit Profile | Admin | Logout

Site Home

Manager Options: Moderate | Manage | Import metadata force ☐ clean ☐ | Generate metadata force ☐ | Index

Manage Datasets

Datasets: [spreadsheets] [collection] [carousel]
Options: Create [F]older, [D]elete file/folder, [U]pload file or ZIP

Select a file to attach (for uploads): Choose file No file chosen

```
[-] ROOT/   [f]
[ ] ... AMAFA/   [f] [d]
[-] ... JAG/   [f] [d]
[ ] ...... JAG_institutional_materials_v2.csv [8839 bytes, 09 May 2021 14:31:01]   [d]
[ ] ...... Brenthurst_v3.csv [365457 bytes, 09 May 2021 14:31:01]   [d]
[ ] ...... Maritz_v2.csv [526790 bytes, 09 May 2021 14:31:01]   [d]
[ ] ...... Jaques_v2.csv [25425 bytes, 09 May 2021 14:31:01]   [d]
[ ] ...... Brodie.csv [7073 bytes, 09 May 2021 14:31:01]   [d]
[ ] ...... Horstmann_v2.csv [271199 bytes, 09 May 2021 14:31:01]   [d]
```

Figure 1: Current Simple DL administrative site

NEW HEADING?

NOT REALLY CLEAR

Before any coding or implementation could happen, the technologies of the project had to be understood. For this project Node.js, React.js, JavaScript and Express.js will be utilized. The reason for this selection was to use a more recent technology that allows components to be reused, hence the choice to use React. Since JavaScript was going to be used in the frontend with React and NodeJS is one of the most popular platforms for hosing and running servers along with React. The team chose NodeJS for the server side and React for the client side. NodeJS is used to build the server-side application while Express.js is used to handle the HTTP requests. It is used on top of Node.js web server functionality and includes helpful utilities to Node.js HTTP objects. Express.js also allows users to define application routes using URLs which was necessary for this system [17]. JSON (JavaScript Object Notation) is a text-based format used to represent structured data [18]. Since data had to be sent between the server and client, this was also a prerequisite. It is based on the syntax of JavaScript and is mainly used for transmitting data in web applications.

NOT SURE THIS IS USEFUL ✓

The first step was to find a course that could explain how React, Node and Express all connected. Since JavaScript, HTML and CSS were not very familiar languages, the next step was to build a basic landing page with some functionality. After building a responsive webpage and a small application using React, these skills had to be applied to the project.

3.2.1 Overview. For the development of the new file and data management part of the system, an agile approach was used. This was to ensure a more user-centered design could be achieved with the user requirements and feedback being the driving force of development. This was achieved through an iterative process that began with a small subset of requirements and then iteratively improved the system by enhanced versions until the system was complete. Iteration one included the development of a prototype which was used to get some basic functionality working and give the users an idea of what the system should achieve. During the first iteration the prototype was demonstrated to the project supervisor and a second evaluator. A second demonstration was done with Simple DL users (from the Emandulo project) for requirements gathering and to discuss some additional ideas. Iteration two included implementing the requirements from the users and developing a second, more functional system. This was used during user testing to evaluate the system and get any additional feedback from the users.

3.2.2 Iteration One. Before coding, a paper-based design was drawn to visualize what the system would look like. Initially, the prototype built from this design would just be a facelift of the previous system since user requirements was not complete yet. {insert image of first paper design} The ✓ system had the same tree-like structure but instead the links used for the functionality were replaced with buttons. The aim of this design was to get some basic functionality working and being able to display the directories and subfolders. This prototype was used in iteration one. After requirements gathering and some testing of the system, some changes had to be made. Initially, all the data of all the folders, subfolder and files were being returned to the frontend through a JSON response. The system could display all the folders and its contents through expansion buttons. However, most of the users preferred working in one directory at a time rather than a tree-like structure. This meant that the system had to be redesigned to suit their preferences.

3.2.3 Iteration Two. After the redesign, the process of fetching the folder structure had to be changed since this was not proving to be very efficient. The system was then redesigned for a second time and rebuilt to focus on one directory at a time but only fetch the contents of a folder upon the user's request. This required a GET request (an HTTP request used to request data form a specific resource) that could handle multiple subfolders. To handle the subfolder issue, a dynamic URL had to be used. The first attempt at using a dynamic URL included making use of a place holder that was placed at the end of the URL at the endpoint. This failed since there were cases when no subfolder was chosen, to account for this, a question mark was used. Although this was more successful, it failed if a subfolder had a subfolder. This implied that the URL endpoint had to be able to accept a dynamic path with any number of directories. After some research and experimentation with express routing, a viable solution was created. This included the use of an asterisk to allow for multiple subfolders and a question mark to account for the case when no subfolders were selected. This system was used during iteration two. The testing was conducted with the same Simple DL uses since expert feedback was necessary.

NOT CLEAR

3.2.4 Iteration Three. Iteration three included the implementation of the additional functionality gathered during testing and fixing any bugs or issues highlighted during testing. After the final improvements, the final set of usability tests were conducted with DL users who were recruited through email. These DL users are not specifically Simple DL users. This was to ensure the system was useable for any DL user and not only expert users.

EXPLAIN digital libraries lab at uct ✓

3.3 Requirements Gathering ✓

Requirements gathering took place online since THE majority of the Simple DL users were remote. Three users were recruited since they are current users of the Simple DL system and could provide expert input/opinion. The requirements were gathered through a series of interviews where the users were asked specific questions which were formulated beforehand based off what was missing from the current system and any additional functionality that could potentially make the system more usable. During the interviews the interviewee could offer any additional suggestions or feedback. The questions were aimed at the usability of the system. These included questions such as:

1. What are the biggest challenges you face while using the system?
2. What works well or what do you like about the system?
3. What do you find most frustrating about the system?

MOVE 4.1 - OUTCOME OF REQ. GATHERING TO HERE here
NO NEED TO SEPARATE AND REQ. ARE NOT RESULTS

— USE BULLET LIST OF REQS

MERGE 3.4 INTO TESTING SECTION AT END SINCE THESE ARE RESULTS. ALSO, LIST USER COMMENT EXPLICITLY ACCORDING TO NIELSEN'S HEURISTICS

list useres comments explicitly

4. What functionality do you think is missing from the system?
5. Do you prefer a tree-like structure of working in one directory at a time?

The aim of these questions was to encourage discussion in order to fully understand the issues and frustrations with the system. This semi-structured approach allowed for follow-up questions while following a set of pre-determined questions [17]. After the interviewee answered all the questions, the prototype was demonstrated. The goal of the demonstration was to give the user a basic idea of what the system would look

3.4 User Testing

The functionality discussed during requirements gathering was implemented to develop a more functional version of the first prototype. This version of the system was used for user testing with the Simple DL users. This was achieved by demonstrating the functionally through a series of task scenarios such as uploading multiple files or folders, deleting multiple files and folders, uploading zipped files without the system automatically unzipping the file, navigating through the system. The new system displayed files and folders in a directory-focused view as opposed to the tree-like structure since this is what all the users preferred. This directory-focused approached also allowed for files to be sorted according to name, type, size or the last modified date. The main goal of this iteration was to build the main structure of the directories, allow multiple uploads of files including zipped files and handling multiple deletions. Through the user testing, additional functionality was highlighted such as renaming folders, moving folders and files and allowing users to copy files and paste them into different directories. After demonstrating the system to the users, users could also request certain tasks to be completed. Since users were remote, they would instruct the interviewer and the interviewer would perform the task after which the user would offer feedback or suggestions.

TESTING SHOULD NOT LIST FEATURES

features ✓

3.4.1 Expert Feedback. The feedback for the functionality built in this iteration was positive but through the evaluation of Nielsen's ten usability heuristics, additional functionality was highlighted Through a questionnaire and a rating scale, the expert users rated each heuristic, these were recorded with the additional feedback and suggestion. The ratings were averaged and recorded in Table 1 in the supplementary resources. The main improvements suggested were including feedback or confirmation of when a task has been done successfully. For example, when deleting a file, alert the user that the file has been deleted successfully. One of the functionalities that the users requested to be included in the system was a button for the creation of folders. With this, they also suggested that an exit or cancel button be included in case they change their mind. The error prevention of the system also needed to be improved. During user testing one of the users asked what the consequence was if a file was uploaded to the system which already existed. To improve the flexibility and efficiency of use, one of the users suggested implementing a hierarchical overview of the system on the left so that users could move files

MOVE TABLE 1 HERE - DO NOT USE IN APPENDIX

from one directory to the next without multiple steps involved. Most of the users were impressed and satisfied with the minimalist design of the system, however, there was a suggestion to include a path so that the user was aware of which directory they were currently in. Since no error messages were displayed during the testing, this was also highlighted as an area of improvement to ensure error messages are clear and expressed in plain language with a suggestion as a solution. One user suggested that the help and documentation of the system be improved since it wasn't clear that the checkboxes were used for batch deletions. Instead, the user suggested that an information block be displayed so that they user knows what the checkboxes are for.

3.5 Refining the Requirements

Since the focus of this project is to improve the interface by making it more usable, the focus would be placed on quality, necessary functionality. After the requirements gathering and user testing, there was a long list of suggestions and additional functionality that could be implemented to improve the system. However, it was a priority to build a usable interface with reliable and necessary functionality. Two lists were written up. The nonnegotiable list included all the functionality that was essential for managing files and data, this included uploading, creating folders, downloading and deleting. The remaining functionality was added to the list for additional functionality which included functionality that could potentially improve the user experience but were not essential. ✓

WHAT IS THIS LIST? bulleted list

3.6 Usability Testing

For the last iteration, after implementing the additional features and fixing some errors, the system was ready for usability testing. This was done with the DL users. The users were asked to perform a series of tasks while providing feedback. Tasks included uploading a file or folder, downloading a file or folder, unzipping a folder, creating a folder, moving a file or folder, copying and pasting files or folder and renaming a folder. With each task, the user could provide feedback. After the usability testing, users were asked to rate the system based off Nielsen's ten usability heuristics. This was used to test the improvement of the system through quantitative analysis. {results will be attached in appendix}. {include feedback and necessary improvements}

* RESULTS MUST BE HERE AND MERGE 3.6 INTO TESTING AT END

3.7 Final Product

here and merge 3.6 into testing a the end

After iteration three, the final product was developed with the basic functionality to perform tasks such as uploading, creating, downloading and deleting files and folders. The system also focuses in on one directory at a time as opposed to the tree-like structure that was used before. Additionally, zipped files would only be unzipped if the user chooses to do so, zipped folder could also be uploaded without having to zip the contents first, bulk-uploading and downloading is possible, users can now select multiple files and folders and move them to another directory, folders can be renamed from the system directly, there is a preview box for users to confirm they are working with the correct file, files can be copied and pasted into different locations,

USE A LIST?

* WHERE ARE SYSTEM SCREENSHOTS? system screenshots

a sort functionality has been implemented for each files properties such as name, type, size and date last modified{add any other additional functionality here}.

ADD ITERATION HEADINGS

To improve ~~some of the Nielsen's~~ usability ~~heuristics,~~ exit and cancel buttons have been included for each action button (delete, unzip, upload, create, move, download, etc.). A search bar has been included for quick look-up of files and folders. Error messages have been included for clear communication to the user, and confirmation messages are implemented to double check with the user that they are sure they want to perform an action. This was included to reduce and diagnose errors. To offer a more comfortable and familiar feel, the system also includes a "current location" to clearly inform the user which directory they are currently working in. {any other functionality included}

ADD CONC NARRATIVE

USE CITE AND SUGGESTIONS

Apart from functionality, numerous visual aspects have been improved to contribute to the look and feel of the system. The biggest visible change was the layout of the folders. The folders and files are now displayed in rows for each directory. This eliminated the ~~tree-structure~~ of the previous design. The system also includes physical button for functionalities such as uploading, creating and deleting folders and files as opposed to the previous system which just had linked letters which represented each function. Any possible functions that can be applied to folder or files has been placed in a button located toward the top of the page. This reduces the user's memory load by making actions and options visible. The system also makes use of familiar phrases, term and symbols in order to match the system to the real world. A preview box has been implemented so users can have a quick view of the files they are working with before or after performing any kind of action.

To achieve the goal of the system being something like Google drive, the folders and files in a single directory was displayed through rows in a table that could be sorted by name, type, size or date last modified. The drag and drop functionality for uploading was also inspired from Google Drive. The system also makes use of familiar terms for functionality to improve the user experience and understanding. This also reduced the need for documentation and create a match between the system and the real world by following conventional layouts and making information appear in a natural and logical way. Placing all the functionality at the top of the page reduces the memory load of the user since all the action buttons are visible and easily retrievable. All the lines, colours and visual aspects of the system were kept simple to create a cleaner look and make the system more visually appealing.

NEED PICTURES!

4. Results and Discussion

This section will discuss all the results that were collected during ~~requirements gathering~~, user testing and usability testing. It will discuss the implemented functionality and why these were implemented.

THIS IS NOT RESULTS

✓ MERGE INTO EARLIER SECTION

earlier

4.1 Requirements Gathering

The requirement's gathering process was very insightful. The expert users had many suggestions for functionality and system improvements. The most common requirement was that the system displays directories one at a time rather than in a tree-like structure. The second most common requirement was that the system could handle batch uploads without having to zip the files into a folder. Many users also stated that the system required zipped folders to be zipped beforehand if the user wanted to upload a zipped folder. The users found this to be frustrating. Additional requirement included some way to confirm that the file the user is working with is the correct file. For example, before a user deleted the file, they would want to confirm it is the correct file that they are doing to delete. A preview window was suggested as a solution. Additionally, users wanted the option to select multiple folders and files before performing an action.

4.2 User Testing

WHICH ITERATION IS THIS?

During user testing some additional functionality was suggested and some features to improve the user experience. The functionality included being able to rename folders directly on the system, being able to copy and paste file or move files from one directory to another. Most of the users also suggest that a path to show the current directory would be very helpful. The use of breadcrumbs made this possible. Additionally, the system required more pop-up boxes to communicate with the user whether an action was successful or to confirm with the user that they are sure about the action before proceeding. A search box to find folders and files was also suggested. Some users also felt that the system required more exit buttons to cancel an action should they decide not to proceed. Error messages and error checking had to be improved to ensure error messages are clear and offer a solution to the user.

4.3 Usability Testing

- results from usability testing
- include how the users performed tasks, any patterns or concerns
- Include the user's opinion about the look and feel of the system

4. Conclusions

The final system is more usable, the users had positive feedback and the ratings for Nielsen's heuristics improved.

- What worked, what didn't work
- How would you have done things differently
- What was expected/unexpected
- How did Nielsen's heuristics improve and what were user's final thoughts
- Future work

ACKNOWLEDGMENTS

Thank you to Professor Hussein Suleman for his invaluable support and guidance throughout the duration of this project. Thank you to the Simple DL users and DL users who participated

in this project, your crucial insights, feedback and participation was valued and helpful.

REFERENCES

[1] D. Yadav, "Opportunities And Challenges In Creating Digital Archive And Preservation: An Overview," International Journal of Digital Library Services, vol. 6, no. 2, pp. 63-73, 2016.

[2] H. Suleman, "Simple DL: A toolkit to create simple digital libraries," in Proceedings of The 23rd International Conference on Asia-Pacific Digital Libraries, Online, 2021.

[3] A. Maxwell, "Digital archives and history research: feedback from an end-user," Library Review, vol. 59, no. 1, pp. 24-39, 2010.

[4] G. Biswas and D. Paul, "An evaluative study on the open source digital library softwares for institutional repository: Special reference to Dspace and greenstone digital library," African Journal of Library and Information Science, vol. 5, no. 6, pp. 001-010, 2019.

[5] R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan and M. Smith, "The DSpace Institutional Digital Repository System: Current Functionality," in Proceedings of the 2003 Joint Conference on Digital Libraries (JCDL'03) , Houston, 2003.

[6] L. Verma, "Comparative Analysis of Open Source Digital Library Softwares: A Case Study," DESIDOC Journal of Library & Information Technology, vol. 38, no. 5, pp. 361-368, 2018.

[7] M. A. U.-N. K K. Singh, "Emerging trends and technologies for digital transformation of libraries," IP Indian Journal of Library Science and Information Technology , vol. 4, no. 2, pp. 41-43, 2019.

[8] M. Castagné, "Institutional repository software comparison: DSpace, EPrints, Digital Commons, Islandora and Hydra," 14 August 2013. [Online]. Available: https://open.library.ubc.ca/soa/cIRcle/collections/graduateresearch/42591/items/1.0075768. [Accessed 24 August 2023].

[9] S. J. B. D. B. R. J. M. Ian H Witten, "Greenstone: a comprehensive open-source digital library software system," in In Proceedings of the fifth ACM conference on Digital libraries (DL '00), New York, 2000.

[10] S. J. P. S. Kumaravel, "Is Digital Archiving Simple ?," in INFLIBNET's Convention Proceedings , Imphal, 2004.

[11] "Google Drive," [Online]. Available: https://www.google.com/drive/#features. [Accessed 24 August 2023].

[12] "Get a Dropbox free account," [Online]. Available: https://www.dropbox.com/basic. [Accessed 24 August 2023].

[13] "OneDrive Personal Cloud Storage," [Online]. Available: https://www.microsoft.com/en-za/microsoft-365/onedrive/online-cloud-storage. [Accessed 24 August 2023].

[14] C. Ujjainkar, "Comparision Between Google Drive, One Drive And Dropbox," International Research Journal of Modernization in Engineering Technology and Science, vol. 4, no. 10, pp. 49-53, 2022.

[15] T. P. D. D. Ewa Ziemba, "Critical Success Factors for Adopting Electronic Document Management Systems in Government Units," in Proceedings of the of the 17th Conference on Computer Science and Intelligence Systems, Sofia, 2022.

[16] D. K. M. M. R. Dr.T.KamalaKannan, " Study on Cloud Storage and its Issues in Cloud Computing," International Journal of Management, Technology And Engineering, vol. 9, no. 1, pp. 976-981, 2019.

[17] Geeks for Geeks, "Express.js," Geeks for Geeks, 19 April 2023. [Online]. Available: https://www.geeksforgeeks.org/express-js/. [Accessed 27 August 2023].

[18] M. contributors, "Working with JSON," Mozilla, 3 July 2023. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON. [Accessed 27 August 2023].

[19] T. George, "Semi-Structured Interview | Definition, Guide & Examples," scribbr, 22 June 2023. [Online]. Available: https://www.scribbr.com/methodology/semi-structured-interview/. [Accessed 23 August 2023].

incomplete

INCOMPLETE

WHERE ARE SE ARTEFACTS? USE CASES? CLASS DIAGRAMS? ETC.

NEED PICTURES! need pictures

LESS NARRATIVE AND MORE CONCISE AND PRECISE INFORMATION

DISCUSS SIMPLEDL more

less narative and more concise and precise information

USE BETTER REFS

use better refs

Supplementary Resources

YOU DO NOT NEED TO EXPLAIN NIELSEN'S HEURISTICS

Heuristic	Description	Rating (Avg)
1. Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.	2
2. Match between system and the real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.	0
3. User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.	1
4. Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing.	0
5. Error prevention	Even better than good error messages are a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.	2
6. Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.	0
7. Flexibility and efficiency of use	Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.	0.66 ~ 1
8. Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.	1
9. Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.	4
10. Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.	1

Table 1: Average Ratings for Nielsen's Ten Usability Heuristics