Archiving Archives

Craig Stevenson
Computer Science
University of Cape Town
Rondebosch, Cape Town, South Africa
stvcra005@myuct.ac.za

Alex Olivier
Computer Science
University of Cape Town
Rondebosch, Cape Town, South Africa
olvale005@myuct.ac.za

Callum Fraser
Computer Science
University of Cape Town
Rondebosch, Cape Town, South Africa
frscal004@myuct.ac.za

KEYWORDS

Archiving, Archives, digital libraries, data preservation, scraping, versioning

1 PROJECT DESCRIPTION

Digital archives are crucial in preserving digital items of the past and present for use in the future. However, they are complex to build. A digital archive must provide a way of adding, organising, cataloguing, and easily retrieving large amounts of information through a user-interface [1].Work on digital archives has become of great importance over the last two decades [2]. Any improper steps in the process of archiving, as archiving is an active state, could lead to a devastating loss to a complete archive [3]. Initially, work started out converting traditional library materials to a digital form for safekeeping, however recently the number of items 'born' digitally have increased [4]. Therefore, no second redundant hardcopy would be available to regenerate an item if it were to be lost. Digital archives do not only face concerns around the underlying technology but also on the organisations maintaining them, as they have great liberty in the methods and accuracy of the preservation applied [5]. This results in the existence of the archive depending on continued funding and commitment of its maintaining organisation.

It could be argued that the extensive number of tools created and research into digital libraries has solved the archiving problem [6]. Yet, we still face problems around failures in current systems for under-resourced environments. Problems such as archive distribution without network access and the failure to create trust in their ability to preserve information, as they can shut down due to a lack of funding [7]. This project is focused on addressing some of the concerns around digital archiving by conducting research around the creation of a new breed of digital archives that preserve other digital archives to establish trust, versioning, and long-term persistence of data through the public conservation of other archives. In practice this will involve the building and then experimenting on of a system that creates a single archive, using archival snapshots of multiple already available archives. To achieve such functionality, this will first involve the development of specially created archival scrapers. Each scraper will be specifically designed for scraping the contents of archives made with a particular popular archiving tool and converting this data to usable archive snapshots. Research with scrapers will be important in establishing how much data can be extracted from current archives, without compromising accuracy or the original user experience. These usable archive snapshots will then need to be ingested into an archiving system. The archiving system for this project will build on top of an open-source archiving tool called Simple DL, which focuses on being versatile and easily extensible. Simple DL can perform most of the heavy lifting work for archiving but will require a few additional extensions to be made to support all intended features. Finally, the development of a unique user-interface will be required as it will be important in conveying the different between this project and a normal archive and will have the single biggest impact on user experience.

2 Problem Statement

2.1 Research Questions

The overall research question seeks to understand whether multiple archives that were created with different archiving tools can be accumulated into one and successfully preserved using Simple DL. To effectively answer this question, it is broken into three sub-questions, which fundamentally tackle different components.

1) Administration/Management Tool and User Interface:
    a. Can users and admins effectively interact with an archive of archives, given its higher order structure?
2) External Archive Collections:
    a. To what degree can the data of archives, created using popular archiving tools, be scraped to produce preservable Archive Snapshots?
3) Ingestion & Archiving system:
    a. Can Archiving Snapshots be ingested without data loss into Simple DL, which is extended with versioning?

b. Can Simple DL be extended to support versioning, complex item manipulation, and Archiving Snapshots?

3 PROCEDURES & METHODS

3.1 System Design

Over time more technology develops, and an archive needs to keep up [8]. A typical problem of digital archives is that in the long run the underlying systems that power them can become unsupported, which presents security risks, and results in the software becoming incompatible with modern hardware running it [9]. For this reason, it is crucial that archives be able to evolve and maintain the accessibility and preservation of their stored digital objects [8]. Fedora, an archiving system, has previously attempted to address this problem of extensibility using a service-oriented architecture (SOA) [10]. This SOA system makes components reusable and interoperable via service interfaces. However, a simpler method has been presented by Mayo et al. [9] using microservices that combine multiple independent applications. By applying this modular approach, parts of the system can get replaced by newer systems that have more active development, without compromising the overall system or without needing in-depth knowledge of the codebase [11]. Microservices, a variant of SOA, provides the perfect architecture for building an archive of archives, using separate components developed by different people, while using different frameworks and tools for multiple parts is a main feature of the architecture [12].

Our implementation of archiving archives will be structured in a three-layer architecture approach. Each layer carries services that are solely responsible for a core set of duties. The layers are also developed to supply adjacent layers with appropriate functionality [13]. This functionality is provided via an API interface between services. This is important since this is a research project and is susceptible to changes. It also provides our team with three well defined layers, to subsequently split and work on individually, while having the ability to test individual component's functionality. Figure 1 graphically portrays our approach.

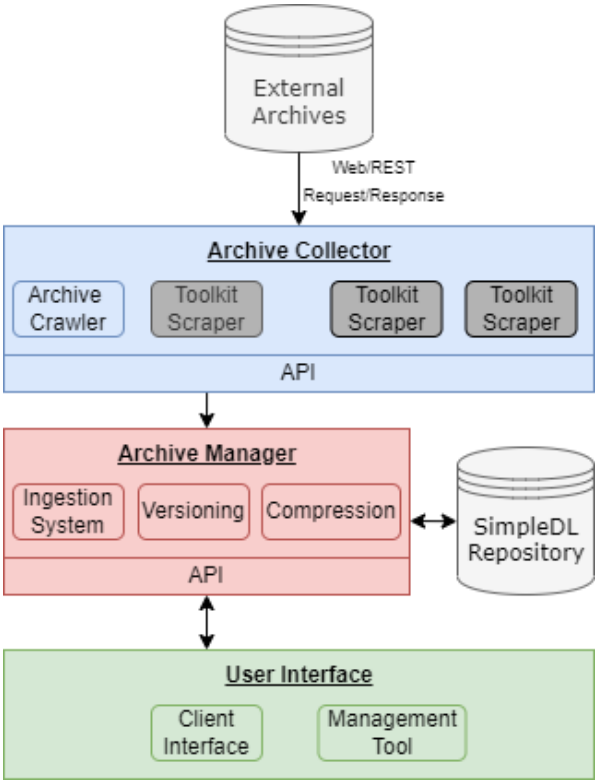


Figure 1: System architecture, layers, and services

3.2 Development Platform

Referring to our system architecture in Figure 1, the first layer is the Archive Collector. External archives will be extracted by scraping an archives website. Firstly, it will determine the tool used to create the archive, by searching the HTML header, and secondly it will extract the digitals object by HTTP GET requests. We will try support a few popular toolkits such as: DSPACE, EPRINTS and AtoM. The range of support will depend on complexities found and the time constraints. We will use Beautiful Soup, a Python package, to parse the HTML while searching for relevant metadata on digital objects and the digital objects themselves.

Secondly, our middle layer comprises of two components. Firstly, we will be building our application on top of the open-source Simple DL toolkit. It is a core software toolkit and will act as one of our main components [6]. It is responsible for the ingestion and pre-processing of digital objects, creating a structured flat file format repository, offline indexer, and JavaScript Query engine. The second is the Archive Manager providing functionality that oversees the archive. This component handles all requests to the Simple DL repository, facilitates requests to the Archive Collector for adding new archives to the collection and facilities compression and versioning functionality. We will use Python to implement this component.

The third layer, the view, including the main user interface for browsing and searching as well as managing the archive, will be using a web application relying on HTML, JavaScript, and CSS. Built out using Dart and Flutter, we leverage the capabilities of Flutter as it is an open-source development kit capable of creating cross platform applications. This is useful for providing access to as many devices as possible and can be used to create an interface suitable for non-technical users [7].

3.3 Implementation Strategy

We plan to foster an agile software development strategy. Our first reasoning is that they are well suited to small and medium scaled projects [14]. They also provide leeway to continuous improvements to code quality and architecture. This syncs well with the project's strict deadlines, increasing our likelihood of a functional prototype at the deadline, which can be used for evaluation and reporting. This will also help solve integration problems since continuous integration will occur at the end of each iteration. We will further make use of our weekly meetings with our supervisor, as a feedback session, to initiate changes and start new sprints.

3.4 Expected Challenges

It is an expected challenge of the research team to develop a deep and robust understanding of all aspects of digital archiving, which could create a challenge during development. Digital library creation and archiving is still relatively new to the research team, which means we are not too familiar with the approaches/paradigms used in the creation of such structures. This has somewhat been remedied by background reading performed while doing the literature review but may still present challenges.

Versioning and the indexing of all types of files is currently not supported by Simple DL. Implementing such a feature as an extension to the tool will present an interesting challenge to Archive Manager aspect of the project. The team member implementing the system has no previous experience with implementing this type of functionality and will thus require ongoing learning and extent of implementation may depend on this.

The short time available to complete the whole project may present a challenge. This project involves many steps: familiarising oneself on digital archiving; designing a system; separating development segments; integrating and testing. This must all be done in the span of eight weeks, which may require sacrificing aspects of the scope or being unable to complete certain features to their full potential.

Documentation is known to always be a challenge in developing complex software programs. Thus, creating clear and well documented APIs will be a time-consuming and important aspect of the project that will be needed to ensure that there is an ease of integration between each part of the project.

Archive scraping could be difficult due to the increasing dynamic approaches provided by scripts [15]. Certain data could be hidden by JavaScript, expecting user interaction to view it. Pagination of sites can also seriously affect the speed of scrapers.

Experimentation is key to our research project and may present one of the biggest challenges. The research team has little to no experience conducting it and may thus struggle to perform it timeously. Getting the experimentation right will require enough time and participants (with knowledge of digital libraries) to perform user-based testing. Both may be hard to come by. Managing this challenge well will be integral to the success of the project.

3.5 Evaluation Techniques

Each of the three components of the project will require their own experiments to evaluate them.

The scraper will be evaluated based on its performance, which will depend on its speed and resource efficiency, and the accuracy of its extractions when compared to the original archive. The accuracy is particularly important as it determines how much can be persisted from the original archive and whether it can provide trust for users. Performance matters as it can have impact on the costliness of solution and the frequency with which scraping can occur. We will be comparing toolkit specific scrapers against each other, as well as the results of scraping multiple libraries with individual scrapers. We will be using white box testing with unit testing and examination of generated data with predefined metrics such as item counts. While programming timing and resource reports can examine the performance of scraping with different threaded models.

The versioning extension to Simple DL will be evaluated with feature comparisons against the original tool, automated unit testing with mock data to check accuracy, and human evaluation to confirm the accuracy of the feature. These experiments will aim to verify that the feature works as stated and leads to accurate and actionable persistence of data.

The archive manager will be evaluated by its ability to integrate with the other parts of the project. These experiments will be important for showing the feasibility of using our architecture for future archives of archives. Experiments will be designed once the separate parts of the system are completed and will make use of integration

testing and user examination to check completeness and accuracy of integration.

User interface evaluation will involve a lot of outside participants. Potential participants will be broken into two groups, experienced archive users and inexperienced archive users. Ideally both groups would have a larger enough sample to be significant of around 15 participants. Inexperienced archive users will include any people from the general populous who have interest in occasionally accessing an archive and will be used to judge the normal user interface. Experienced archive users will come from people interested in archives from research lists, such as the SA Institutional Repository list and members of the UCT Digital Library research group, as well as staff from the UCT Computer Science department who have experience with digital archives. Experienced archive participants will be able to test both the administration and normal user interface of the system. Tests will require users to complete tasks to check the intuition, usability, and high order experience of interaction with a hosted version of the system. At the end a survey will be given to each user to record their perception, satisfaction, and additional feedback. The exact specifications of these experiments will be decided on later.

4 ETHICAL, PROFESSIONAL & LEGAL ISSUES

There are ethical issues that must be considered when accessing and storing other people's archives. That is why there will be no scraping of other people's content without permission from the managing party. Thus, the work will only comprise of test archives that have been generated or will only use real archives where permission has been granted by the owning authority to archive it. Additionally, there will be ethical considerations around getting users to test out the different aspects of the system and ensuring they provide permission to be part of any studies and know exactly what information about them is being stored and what the work is being used for.

Legally it is important to not represent the work or archives of others as our own. Thus, libraries, work, and archives used that belong to other people will be indicated as such. The license of the content will be shown as well as who the owner is of all archives stored and where the original work can be viewed if it is available. This relates to intellectual property, and it has been decided that all work produced will be made open-source, which is particularly important for extensions done to the already open-source Simple DL code repository.

5 RELATED WORK

While no Archive of Archives exists there is previous work that attempts to tackle some of the same problems or provides the foundation that our work can extend. A few tools exist that have core features that could be useful in creating a large new archive, such as ones that provide offline support and easy transportability. For instance, Greenstone is a service that provides a means of distributing a library on a CD-ROM [11]. Unfortunately, it requires a software installation to access collections, which could be problematic for different system distributions and present dependency issues [6]. Alternatively, Simple DL provides a means of pre-generating system independent digital libraries from assets presented in flat file formats, without the need for special installed software dependencies. Additionally, it differs itself from other tools by prioritising simplicity, focusing on static data, and providing offline support [7]. This toolkit is well suited for the purposes of archiving archives as it can facilitate preservation without active management and reduces the need for additional software installation. It can be easily extended since most of its data is stored as flat files with simple XML indexing. For these reasons Simple DL will be used as a key part of this project.

When it comes to large, well-established archives the Internet Archive is one of the most impressive. It is a massive 25-year-old web archive that preserves petabytes of data while being managed by a tiny team of under 10 people [16]. This archive is very popular, and it is estimated that the archive handles more than 10 TB of data per day. A core feature of the Internet Archive is its Way Back Machine that provides an access tool with the ability to retrieve stored web pages through URL search [17]. The Internet Archive's Way Back Machine presents data storage, access, and versioning. Versioning provides an interesting feature that can be applied to our own work. However, a downside of the Internet Archive that could be improved is its lack of an offline implementation or any client capabilities such as a way of browsing its collections or a means of redistributing.

Another interesting software that can influence our work is the popular online code repository GitHub. GitHub stores millions of repositories of versioned code that are created and tracked with the Git software [18]. Like an archive, this versioned code is intended to be stored safely for extended periods of time and provide backups if needed. Additionally, Git and subsequently GitHub can track and store a vast number of different file types and allow repositories of these to be easily copied to new local environments, making distribution easy. The copying, modifying, and sharing of stored repositories is a key feature of GitHub, with forking being the creation of new repositories from another one [19]. Forking allows users to have their own independent versions of another's code at a point in time They can then modify it to their own liking, and even send it back to the original as an improvement. In certain ways GitHub has key similarities to the archiving of another's archives as we will have to store and represent different files and versions of another's work and represent it in an intuitive way to the consumer. Thus,

the GitHub UI could inspire the one created from this project.

6 ANTICIPATED OUTCOMES

6.1 System

We expect the software produced in this work to be fully functional and to work as a single integrated system. The system should be hostable and runnable in an offline environment. It should be able to run on the Linux system as a fully self-contained offline system, with greater support of devices and systems when hosted. The UI should be able to run on all major platforms.

6.2 Expected Impact of Project

In combination this work is expected to result in the successful creation of an archive of archives that will produce a haven for information with long term preservation.

It is expected that:

- The project leads to the creation of a new UI for displaying an archive of archives that has positive test results indicating it properly conveys the system to its users, particularly the higher order structure of it.
- The projects work helps to extend the Simple DL archiving tool so that it may support versioning and complex objects. Both features are then able to be used for new archives created with the tool.
- An archive of archives is created during the project with snapshots of multiple archives that were previously built using more than one different archiving tool.
- As part of the project, there is the successful development of repeatable and easily controllable web-scrappers that can efficiently scrape archives that have been created by popular archive tools. These scrappers provide experimentation results that indicate the accuracy with which different archives may be scrapped.
- Through integration the project's work show that it is possible for effective preservation using an archive of archive, which indicates that it can work effectively in practice and address some of the current concerns around archiving.

6.3 Key Success Factors

The expected key success factor are as follows:

- A simple and intuitive user interface that allows extensive functionality.
- Users can interpret the higher order functionality of the archive of archives.
- Efficient and accurate scrapers that work on archives made with at least three different popular archiving tools.
- Archiving snapshots are successful ingested into an archive.
- Extensive versioning of multiple different archives.

7 PROJECT PLAN

7.1 Risks

Risk Condition	Consequence	Probability (out of 10)	Impact (out of 10)	Mitigation	Monitoring	Management
Archive management denying us permission to scrape their archive	Not being able to get key test data for our archive	2	3	Have backup archives that do not require permission such as UCT archives	Track responses and dates	Perform regular communication with the management organisations and assure them of ethical date usage
Failure to meet certain deadlines/features due to underestimating time for	Having to exclude certain features/testing	1	10	Ensure clear planning at the start, low	Updates asked for during team	Schedule regular team and supervisor

certain deliverables	time			scope, overestimate times	and supervisor meetings	meetings
Integration problems between our three development segments	Increased time needing to be spent on project and code rewriting	7	5	Ensure a simple and robust architecture is chosen and meet regularly with the person whose part you will be interfacing with	Continuously discuss API's as they are implemented at team meetings	Clear documentation of API methods
Hardware/development resources break down such as computers or no/slow internet speeds	No work able to be done by affected member(s)	1	5	Storing online backups of all work done, allowing any resource breakdowns to be recovered from.	Track the progress and backing up of all group members' work	Regular backing up of work
Ineffective communication or arguments amongst the research team members	Breakdown in the team's ability to function impacting the output of the work	2	7	Separate work is done to minimise the effect of one person's contributions to another, reducing the chance that one person's opinion of the work upsets the other. Take all arguments that cannot be resolved after one day to the supervisor to moderate.	Constantly check the happiness of all group members	Tackle arguments ahead of time and prevent them from festering. Allow the more passionate, greater impacted, or more technically knowledgeable member to have the final say around a particular issue.
Team member(s) not delivering on schedule or leaving Honours.	Incomplete work leading to poor integration or knock-on delays.	4	10	Continually monitor the progress of all members and plan for integration workarounds including the use of mock data and APIs	Follow up on team member progress on a bi-weekly basis	Make every part of the system independent and have member responsible for having mock data for their section. Assign extra time for integration and delays. Try and

						get ahead of development work (start and end early).

7.2

Timeline

Ethics Proposal/Application – 10 June
Software Feasibility Demonstration 25 July

Software Feasibility: 11th June – 25th July
User Test Version for testing: 26th July – 15th August

Time for Planning Experiments

- User Recruiting: 1st – 15th August
- Tasks & Questions Prep: 13th -15th August

User Testing: 15th -16th August

Conducting Experiments – 2 weeks
Write Up – 4 weeks
Paper Draft: 16th – Tue 23 August
Final Submission 24th August– 2nd September
Code Submission – 5 Sep
Poster – 4 October
Website – 10 Oct
School of IT Showcase- TBD

Check appendix A for the project GANNT chart.

7.3 Resources Required

Hardware:

- Flash Drive/Hard Drive
- Personal Computers

Software:

- Linux
- Simple DL Toolkit
- Microsoft Teams
- Microsoft Word
- Discord

Languages

- Python
- Dart

Digital Resources:

- Access to Archives, E-Journals

High Speed Internet connectivity

7.4 Deliverables

Deliverable:	**Date:**
Ethics Proposal	11 July 2022
Final Project Proposal	15 July 2022
Software Feasibility Demonstration	25 July 2022
Final Paper Draft	23 August 2022
Final Paper	2 September 2022
Final Code	5 September 2022
Final Demonstration	19 September 2022
Poster	3 October 2022
Website	10 October 2022
School of IT Showcase	TBD

7.5 Milestones

Milestone:	**Date:**
User Test Version Complete	15 August 2022
Task & Questions Prepared for User Test	15 August 2022
Evaluation and Testing Complete	16 August 2022

7.6 Work Allocation

The work has been broken down into separate viable projects that will be completed by each member while collating together to answer the overarching research question and attempt to create an archive of archives.

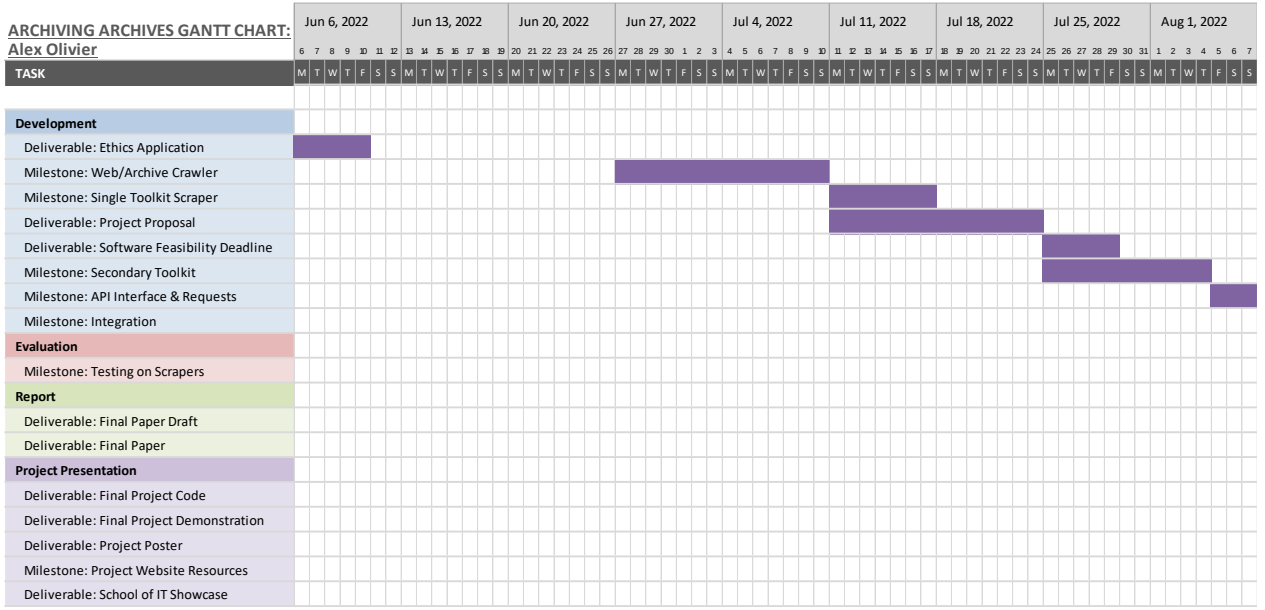
Callum will work on the User Interface for both administrators and general users.

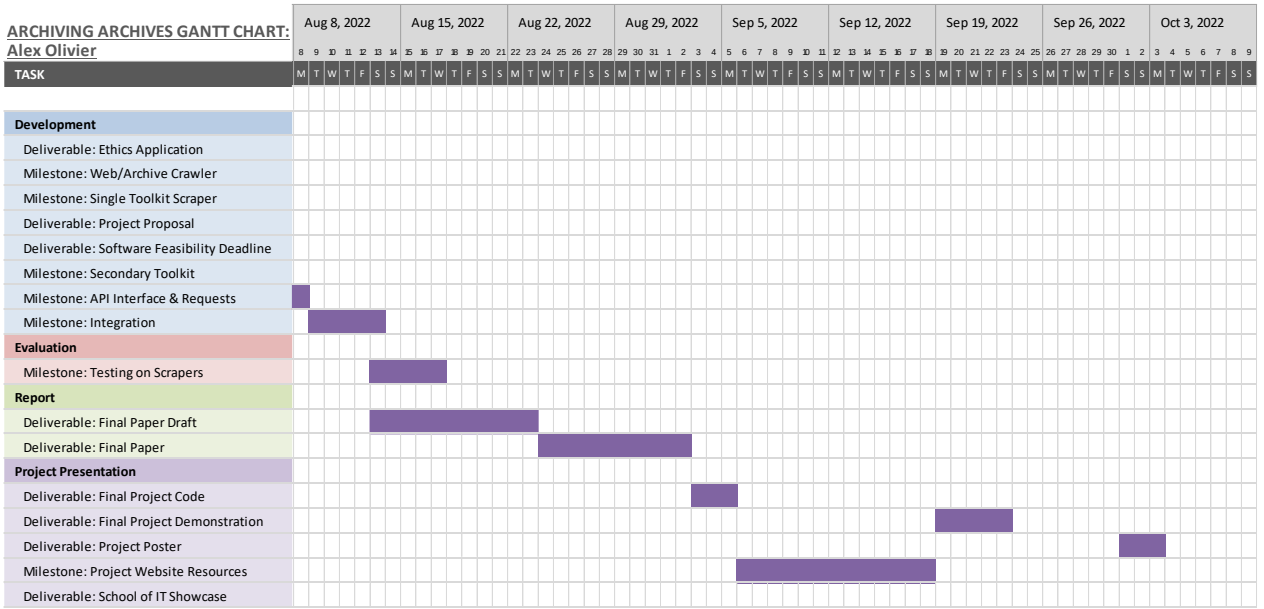
Alex will develop the collection system for scraping archives.

Archiving Archives

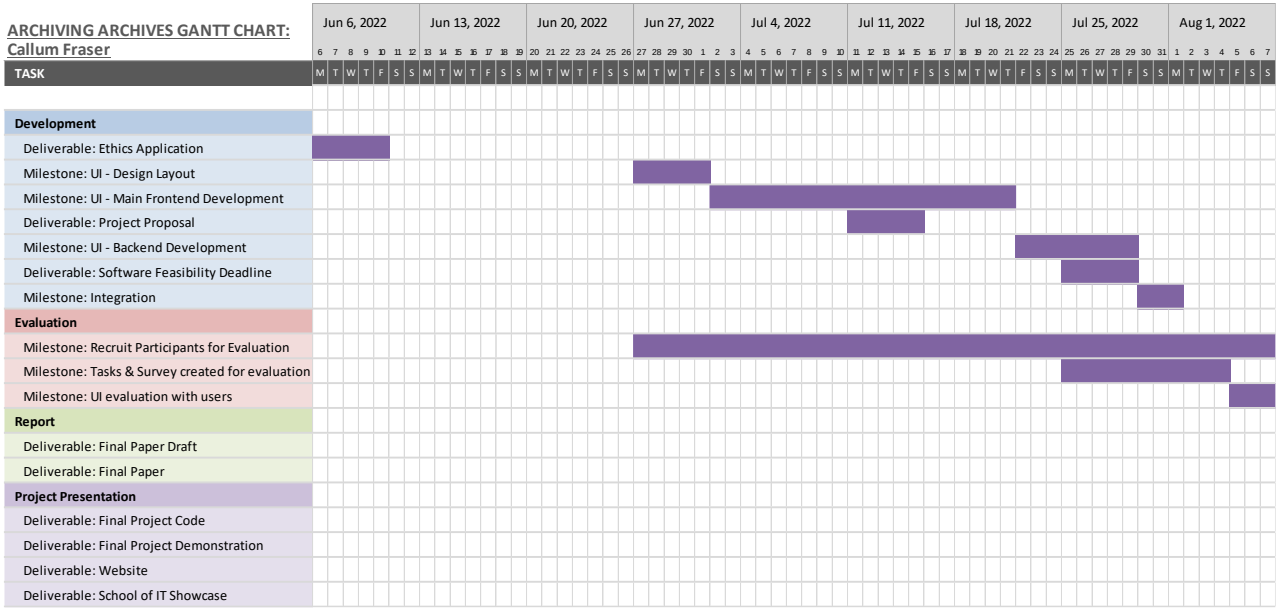
Craig will perform extension to the Simple DL system and manage all ingestion and archiving, as well as provide assistance in building scrapers for certain archiving tools.

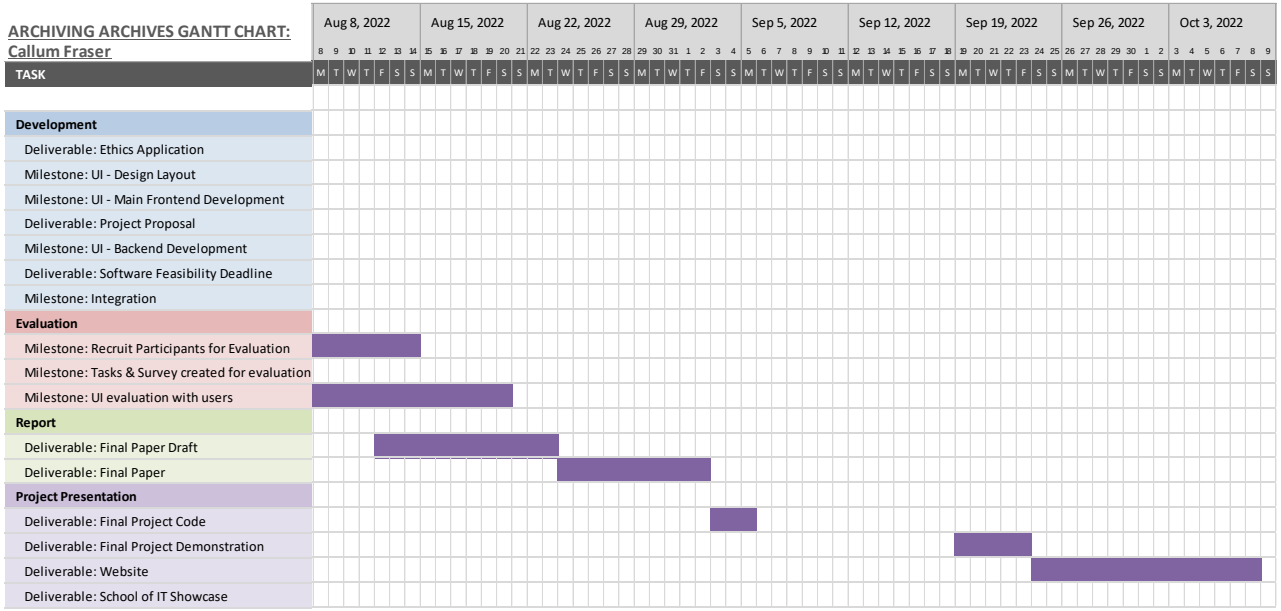
Appendix A

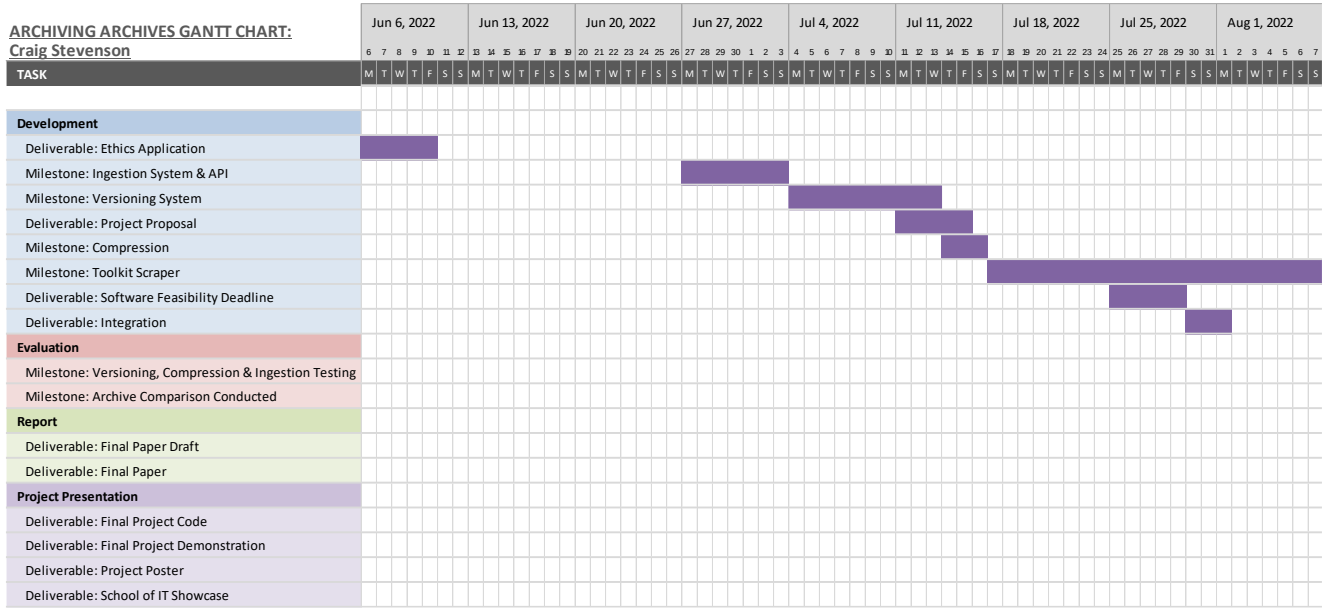

ARCHIVING ARCHIVES GANTT CHART:
Alex Olivier
TASK
Jun 6, 2022
Jun 13, 2022
Jun 20, 2022
Jun 27, 2022
Jul 4, 2022
Jul 11, 2022
Jul 18, 2022
Jul 25, 2022
Aug 1, 2022
Development
Deliverable: Ethics Application
Milestone: Web/Archive Crawler
Milestone: Single Toolkit Scraper
Deliverable: Project Proposal
Deliverable: Software Feasibility Deadline
Milestone: Secondary Toolkit
Milestone: API Interface & Requests
Milestone: Integration
Evaluation
Milestone: Testing on Scrapers
Report
Deliverable: Final Paper Draft
Deliverable: Final Paper
Project Presentation
Deliverable: Final Project Code
Deliverable: Final Project Demonstration
Deliverable: Project Poster
Milestone: Project Website Resources
Deliverable: School of IT Showcase

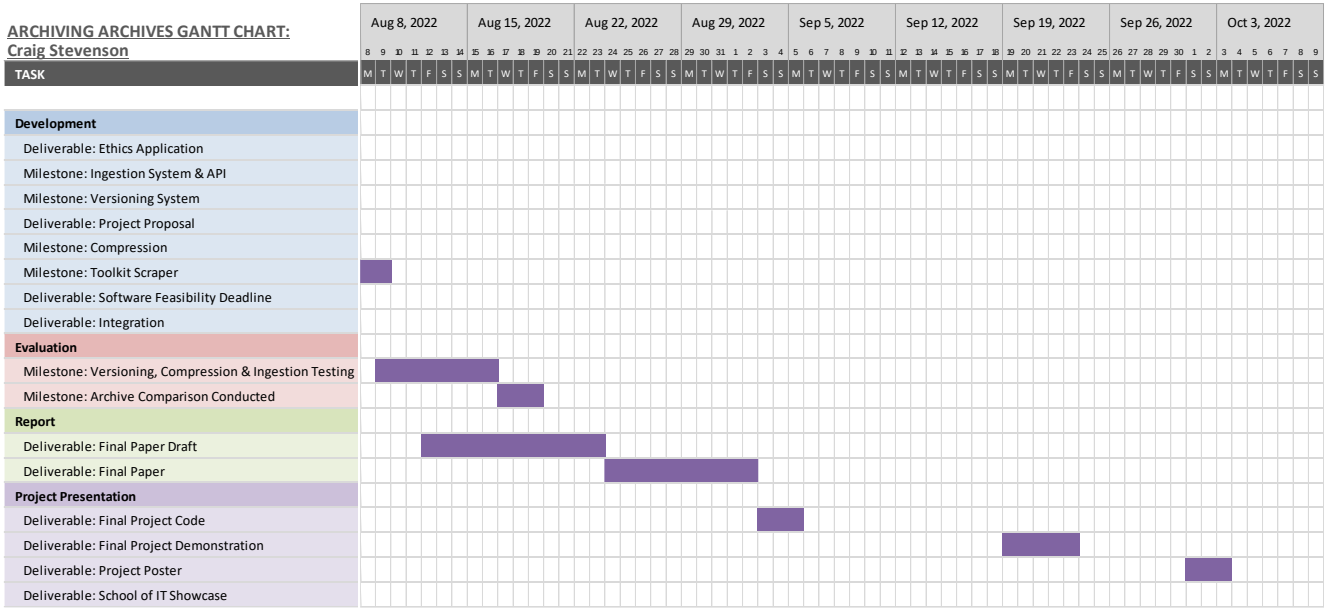

ARCHIVING ARCHIVES GANTT CHART:
Alex Olivier
TASK
Aug 8, 2022
Aug 15, 2022
Aug 22, 2022
Aug 29, 2022
Sep 5, 2022
Sep 12, 2022
Sep 19, 2022
Sep 26, 2022
Oct 3, 2022
Development
Deliverable: Ethics Application
Milestone: Web/Archive Crawler
Milestone: Single Toolkit Scraper
Deliverable: Project Proposal
Deliverable: Software Feasibility Deadline
Milestone: Secondary Toolkit
Milestone: API Interface & Requests
Milestone: Integration
Evaluation
Milestone: Testing on Scrapers
Report
Deliverable: Final Paper Draft
Deliverable: Final Paper
Project Presentation
Deliverable: Final Project Code
Deliverable: Final Project Demonstration
Deliverable: Project Poster
Milestone: Project Website Resources
Deliverable: School of IT Showcase

Archiving Archives






ARCHIVING ARCHIVES GANTT CHART:
Craig Stevenson
TASK
Jun 6, 2022
Jun 13, 2022
Jun 20, 2022
Jun 27, 2022
Jul 4, 2022
Jul 11, 2022
Jul 18, 2022
Jul 25, 2022
Aug 1, 2022
Development
Deliverable: Ethics Application
Milestone: Ingestion System & API
Milestone: Versioning System
Deliverable: Project Proposal
Milestone: Compression
Milestone: Toolkit Scraper
Deliverable: Software Feasibility Deadline
Deliverable: Integration
Evaluation
Milestone: Versioning, Compression & Ingestion Testing
Milestone: Archive Comparison Conducted
Report
Deliverable: Final Paper Draft
Deliverable: Final Paper
Project Presentation
Deliverable: Final Project Code
Deliverable: Final Project Demonstration
Deliverable: Project Poster
Deliverable: School of IT Showcase


ARCHIVING ARCHIVES GANTT CHART:
Craig Stevenson
TASK
Aug 8, 2022
Aug 15, 2022
Aug 22, 2022
Aug 29, 2022
Sep 5, 2022
Sep 12, 2022
Sep 19, 2022
Sep 26, 2022
Oct 3, 2022
Development
Deliverable: Ethics Application
Milestone: Ingestion System & API
Milestone: Versioning System
Deliverable: Project Proposal
Milestone: Compression
Milestone: Toolkit Scraper
Deliverable: Software Feasibility Deadline
Deliverable: Integration
Evaluation
Milestone: Versioning, Compression & Ingestion Testing
Milestone: Archive Comparison Conducted
Report
Deliverable: Final Paper Draft
Deliverable: Final Paper
Project Presentation
Deliverable: Final Project Code
Deliverable: Final Project Demonstration
Deliverable: Project Poster
Deliverable: School of IT Showcase

References

[1] R. Pandey, "Digital Library Architecture," Indian Statistical Institute, March 2003. [Online]. Available: http://dlissu.pbworks.com/w/file/fetch/44829234/B_architecture.pdf.

[2] A. Kumar, R. Chavez and N. Schwertner, "Architecting an Extensible Digital Repository," 2004. [Online].

[3] T. Owens, "The Theory and Craft of Digital Preservation," Johns Hopkins University Press, 2018. [Online]. Available: https://osf.io/preprints/lissa/5cpjt/download.

[4] D. Gerrard, J. Mooney and D. Thompson, "Digital Preservation at Big Data Scales: Proposing a step-change in preservation system architectures," DOI: 10.1108/LHT-06-2017-0122, November 2017. [Online].

[5] I. Carbajal and M. Caswell, "Critical Digital Archives: A Review from Archival Studies," The American Historical Review, Volume 126, pp 1101-1120, 2021. [Online]. Available: https://doi.org/10.1093/ahr/rhab359.

[6] H. Suleman, "Simple DL: A toolkit to create simple digital libraries," University of Cape Town, South Africa, 2021. [Online].

[7] R. Jantz and M. Giarlo, "Digital Preservation," Architecture and Technology for Trusted Digital Repositories. Volume 11 No. 6, 2005. [Online].

[8] D. Yadav, "OVERVIEW, OPPORTUNITIES AND CHALLENGES IN CREATING DIGITAL ARCHIVE AND PRESERVATION," International Journal of Digital Library Services. Volume 6, 2016. [Online]. Available: http://dcac.du.ac.in/documents/E-Resource/2020/Metrial/31NehaJingala1.pdf.

[9] C. Mayo, A. Jazairi, P. Walker and L. Gaudreau, "BC Digitized Collections: Towards a Microservices-based Solution to an Intractable Repository Problem," Issue 44, 2019. [Online]. Available: https://journal.code4lib.org/articles/14445.

[10] C. Lagoze, S. Payette, E. Shin and C. Wilper, "Fedora: An Architecture for Complex Objects and their Relationships," International Journal on Digital Libraries. pp 124-138. DOI: 10.1007/s00799-005-0130-3, 2006. [Online].

[11] H. Sastry and L. Reddy, "Digital Repository Software Packages: An extended architecture for image handling in open source packages," 2010. [Online]. Available: https://www.researchgate.net/publication/279466681_Digital_Repository_Software_Packages_An_extended_architecture_for_image_handling_in_open_source_packages.

[12] M. Amaral, D. Carrera and I. Mohomed, "Performance Evaluation of Microservices Architectures using Containers," 2015. [Online].

[13] A. Kumar, R. Saigal, R. Chavez and N. Schwertner, "Architecting an Extensible Digital Repository," Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries. DOI: 10.1109/JCDL.2004.239994, September 2004. [Online]. Available: https://dca.lib.tufts.edu/dl/publications/pubs/jcdl2004-tufts.pdf.

[14] L. Cao, K. Mohan, P. Xu and B. Ramesh, "A framework for adapting agile development methodologies," European Journal of Information Systems, [Online]. Available: 10.1057/ejis.2009.26.

[15] P. Meschenmoser, N. Meuschke, M. Hotz and B. Gipp, "Scraping Scientific Web Repositories: Challenges and Solutions for Automated Content Extraction," DOI: 10.1045/september2016-meschenmoser, 2016. [Online].

[16] E. Jaffe and S. Kirkpatrick, "Architecture of the internet archive," The Israeli Experimental Systems Conference. DOI: 10.1145/1534530.1534545, 2009. [Online].

[17] Z. Fernando, I. Marenzi, W. Neijdl and R. Kalyani, "ArchiveWeb: Collaboratively Extending and Exploring Web Archive Collections," International Conference on Theory and Practice of Digital Libraries. DOI: 10.1007/978-3-319-43997-6_9, 2016. [Online].

[18] J. Blischak, E. Davenport and G. Wilson, "A Quick Introduction to Version Control with Git and GitHub," PLoS Computational Biology. DOI:10.1371/journal.pcbi.1004668, 2016. [Online].

[19] J. Jiang, D. Lo, J. He and X. Xia, "Why and how developers fork what from whom in GitHub," Chinese distribution service of audiovisual products in international copy right trade. DOI:10.1007/s10664-016-9436-6, 2017. [Online].

[20] H. Liu, F.-C. Kuo and T. Chen, "Teaching an End-User Testing Methodology," Centre for Software Analysis and Testing, Swinburne University of Technology, Australia. DOI: 10.1109/CSEET.2010.28, 2010. [Online].

[21] I. Harms and W. Schweibenz, "Usability Engineering Methods for the Web," Fachrichtung Informationswissenschaf, 2000. [Online]. Available: https://publikationen.sulb.uni-saarland.de/bitstream/20.500.11880/25651/1/06247h3.pdf.