UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

# CS/IT Honours Final Paper 2020

Title: EXHIBIT: A Tool for Digital Presentations of Cultural Heritage Artefacts

Author: Ceara Mullins

Project Abbreviation: EXHIBIT

Supervisor(s): Professor Hussein Suleman

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

# EXHIBIT: A Tool for Digital Presentations of Cultural Heritage Artefacts

Ceara Mullins
Department of Computer Science
University of Cape Town
Private Bag X3,
Rondebosch, 7701
South Africa
MLLCEA001@myuct.ac.za

## ABSTRACT

As technology advances, the method of storing archival material has adapted to favouring digital databases. These archival databases have revealed a desire of users to view and exhibit the contents in a digital format that will simulate exhibits and allow archivists to share materials and allow other users with limited archival knowledge to access the digitised cultural artefacts. In this paper, we describe the development of a tool designed to populate digital exhibitions using local and archival content and the ability to view and distribute them. EXHIBIT is a development project and was created using Agile principles and Feature Driven Development. Usability tests and a heuristic evaluation of the system were completed to assess the project's effectiveness. The Usability survey was also used to identify particular features for future work.

## KEYWORDS

Digital exhibition, museum toolkit, curator, historical archives, usability

## 1 INTRODUCTION

The Five Hundred Year Archive [1], The Digital Bleek and Lloyd Collection [2] and The African Rock Art Digital Archive [24] are three examples of digitised archives of historical media. These three archives are rich in African cultural heritage available online for access by the public. EXHIBIT is a system that is designed which would allow users to create digital exhibitions while incorporating their own content along with access to digital artefacts from these types of archives.

### 1.1 Problem Statement

Existing museum toolkits used for generating digital exhibitions are not easily accessible for users who are not archivists and the content is generally dependent on a users' access to archival content. These tools also typically expect a higher level of digital literacy than most average users have, as the tools are designed as research projects for researchers, professional archivists and museum personnel. There is no cross section between a tool that provides access to archival material that is rich in historical data and a tool that is easily usable and accessible for users with lower levels of digital literacy. The importance of addressing this desire will help users to share their cultural histories and express their heritage with access to digitised archival material, while also having the ability to upload their own heritage artefacts.

### 1.2 Solution Outline

EXHIBIT is a software system that is comprised of two major parts: template creation and exhibition population. Templates are made up of textboxes and image boxes that can be styled by the user. After a template is saved, the template is then accessible by all other users. These templates can then be populated with available archival material or through the users' own content via local uploads. Populated exhibits and templates are downloadable in PDF and HTML formats for accessibility and to make them easily shareable. A browse page is developed to allow users to explore existing templates and exhibits for viewing, downloading and commenting on. Table 1 shows the separation of work of the EXHIBIT project. The project was designed for a broad spectrum of users, including high school students, museum curators, archivists and researchers.

**Table 1: Feature assignment**

| Developer | Features |
|---|---|
| Aa'isha Dout | Template Creation, browse, commenting |
| Ceara Mullins | Exhibition Population, archive, viewing |

### 1.3 Report Structure

This paper will look at related works in this field, their design and services. It will provide an overview of the software methodologies that were used for the development of the EXHIBIT project. It will then discuss the design of the exhibit population system and how the design was achieved through requirements gathering and prototyping, along with the generation of other design artefacts. Furthermore, it will elaborate on the testing methods used to evaluate the system and will analyse the results to identify what the successes and failures of the project were to determine the project's outcome. It will also touch on some aspects of the project which can be improved or further developed as future work.

## 2 BACKGROUND

The existing toolkits in this field have been designed with archive managers, museum curators and academic professionals in mind. As such, they are often integrated with a content management system to manage a museum's archive of historical artefacts. EXHIBIT is designed to address a larger audience interested in creating digital exhibitions, primarily focusing on users with limited digital literacy and without access to museum archival content of their own. The toolkits discussed in this section are open-source software.

### 2.1 Museum Toolkits

In this section three different museum toolkits designed for the creation of digital exhibitions and content management are discussed and compared.

#### *2.1.1 The Toolkits*

**MOVIO** (MOstre Virtuali Online) [16] is free and open-source software available for download both on desktop and mobile systems. It is useful for archival professionals because of its support for metadata standards and integrated content management. However, it requires a higher degree of IT knowledge as it is more targeted toward museum professionals and researchers, with access to archive systems. It also is not available online, so users without devices to support the download cannot access the system.

**VAES** (Virtual Archival Exhibition System) [21] is a Java and XML-based exhibition creation suite. It is a free open-source software system with a direct-manipulation interface. It also supports the Dublin Core metadata standard, with support for four types of artefacts, (text, photographs, video and audio). There are three approaches to exhibition creation in the VAES system. These three methods provide variety for users who are either unfamiliar or experienced with the system. VAES exports its completed exhibitions in a Web viewing format.

**Omeka** [15] is a free and open-source software available for download. It provides a paid version for users who require extra storage and supports metadata standards with customisable cataloguing for users with specific requirements. Omeka also provides the ability to extend its functionality through the use of plugins. It too supports four media types, (text, photographs, video and audio), with an additional fifth type that is a user customisable object. Omeka also provides a dedicated hosting service called Omeka.net if downloading the software is not an option for their users.

#### *2.1.2 Comparison of the Tools*

EXHIBIT's primary aim is to provide a service that is easily usable, accessible and learnable for users with lower levels of digital literacy and access to archival materials to still be able to create engaging digital exhibitions. The comparison of the tools mentioned (MOVIO, VAES and Omeka) will focus on five areas as there exists no consistent standard for comparison of these types of tools. The areas of comparison of interest for this project are Accessibility, Usability, Learnability, Cost and Metadata support.

**Accessibility** –All of the tools mentioned are available on desktop Web Browsers. MOVIO is the most accessible tool as it is available on desktop and mobile for viewing. These systems all require the user to download the application, with the exception of Omeka providing both a downloadable service, as well as a Web-hosted service called, Omeka.net.

**Usability** – These tools are essentially usable by museum curators as they make use of the exhibition creation in conjunction with content management systems linked to their museum archive. However, most of these tools do require a medium to high-level of computer expertise. They require knowledge of downloading services and system setup and interaction with hierarchies. Most of the systems organise exhibit elements by hierarchies. MOVIO and VAES were both developed for research purposes and therefore both have a bias towards researchers, academic professionals and museum curators accessing the system. Omeka is used more for commercial use and therefore has reviews that inform that it is regarded as an easily usable system. [15] It also provides templates called "Themes" this allows users unfamiliar with the process of template creation to easily start exhibition creation.

**Learnability** – MOVIO and VAES, as mentioned, are both research projects and are therefore intended for more professional and digitally literate users. Omeka as a commercial product does have support and extensive documentation for users to rely on and make use of to better learn the system. VAES does make use of a direct-manipulation interface to allow users some affordances and therefore supports a wider range of digitally literate users.

**Cost** – All of the tools are open-source and free. Omeka does have a paid service should the user require more than the standard amount of storage. Omeka also has the capacity for plugins and, depending on the creator of the plugin, these could incur an extra charge.

**Metadata support** – All of the tools support metadata standards.

#### *2.1.3 Summary of Related Work*

After the comparison of the tools and the preferred requirements for usability and accessibility, Exhibition population was determined to have an interface similar to that of VAES. The use of direct-manipulation where applicable will help make the tool usable and accessible for users with lower levels of digital literacy. Omeka is a tool that is favourable in design and documentation. It provides opportunities for users to learn and understand the system regardless of their digital literacy. Making use of tutorials videos, tooltips and help documentation will assist in making Exhibition population an accessible and learnable system. Export types for the digital exhibits will be limited to PDF and HTML. This is to make it accessible as a viewing format as not all users are capable of supporting augmented reality (AR) and virtual reality (VR) viewing platforms. This is also supported by the results of the Requirements Gathering survey that was completed in the initial phase of the project's development. Responses revealed that PDF and HTML files for exhibit viewing are favourable as they are easily shareable and familiar formats for most users, and they are easily learnable because of the support that exists for them. AR and VR viewing formats would be excessive for more casual users.

# 3 DESIGN AND IMPLEMENTATION

## 3.1 Software Development Methodology

In the initial phase of the project design, a more rigorous project methodology was suggested: Agile Scrum. However, this was far too restrictive given the small team of two members and the COVID-19 pandemic. It was decided to switch the methodology to a more flexible approach, which allowed the team members to work effectively and manage the software development. The approach chosen was Agile practices with Feature Driven Development.

### *3.1.1 Agile Methodology*

Agile methodology is a flexible project management guide to allow developers to work in iterative cycles while accounting for changing circumstances without compromising the development of the project. It was chosen for this flexibility that would allow the developers to make progress on development and adapt to any changing requirements. [17]

### *3.1.2 Feature Driven Development*

Feature Driven Development was chosen because of the short life-cycle of the project timeline. [20] It allowed the project members to rapidly develop a software prototype with working features for testing. Features were able to be delivered within 2 – 10 days on average, compared to Scrum that dedicates 2 – 4 weeks for feature development. The methodology relies on five steps. [3] The lifecycle of Feature Driven Development as used by this project is seen in Figure 1.

**Step 1: Develop an overall software model.** A high-level overview of the system is developed to determine was is in the scope of development.

**Step 2: Build a features list.** This step identifies from the system overview the specific client services that need to be developed for the system to function. This included a navigation page, template creation, exhibition population, and a browse page.

**Step 3: Plan by feature.** This step required analysis of the features and assigning tasks to the developers. The developers discuss together the evaluation of features. Online video meetings with the supervisor and team members were held to assign tasks and ensure progress on development. The assignment of features to project members is displayed in Table 1.

**Step 4: Design by feature.** Design by feature was achieved through cognitive walkthroughs completed by the developers simulating the users. Paper prototypes were also created as artefacts. These prototypes were explored by 3 Honours HCI students in a Wizard of Oz study to assess the flow and design of the system. From the paper prototype, along with feedback, a digital prototype was created. This prototype informed the software prototype.

**Step 5: Build by feature.** This step is where development happened. Classes and methods were implemented in order to achieve the functionality of the feature.

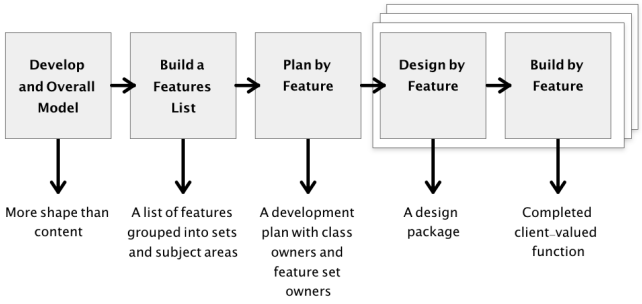


**Figure 1: FDD Lifecycle**

## 3.2 Requirements Gathering

Requirements gathering was completed in the form of surveys, emails and meetings with the project supervisor. Meetings with the supervisor were particularly important given that this development project is related to a larger system. Requirements gathering was completed in the first phase of the project development during the environment setup phase. A Requirements Gathering survey was created and sent to three archivists and two other students to understand the requirements for the features being developed. This section identifies how the process was completed and discusses the responses from the participants.

### *3.2.1 Survey Questions*

The Requirements Gathering questionnaire starts with a question to discover the participants' position (i.e. what type of user they are); this was used to inform the responses to the survey. For example, students, compared to archivists, are less likely to be familiar with pre-existing dedicated exhibition toolkits, or they might use these tools less frequently. These factors need to be considered as they could influence how a system responded to the rest of the questions and how they are likely to use the system. The next question asks how frequently they are required to create digital exhibitions; this allows us to understand how robust the features need to be. The next question discusses for what purpose they are usually creating the exhibition; this informs the type of content expected to be used in the exhibits and how they are used. Another question asks the participants where they retrieve the media that they incorporate into the digital exhibitions (i.e. whether or not it is local, from online sources such as results of Google searches, or if they use media from an archive that they have access to, or a combination of these options). This will allow us to understand where users are likely to be retrieving their media from. Another question asks what digital presentation creation tools they are familiar with; this helps us identify what features they are familiar with and how those features work. The next question looks at what kinds of features from the tools they already use and are used most often, followed by an option to list any features they would like to see. Then the survey inquires as to what method they use to share the digital exhibition, this gives us insight into the preferable file download formats.

### *3.2.2 Survey Responses*

The Requirements Gathering survey revealed that most digital exhibitions are created for educational purposes. The participants all seem to have a combination of preferred material for their exhibitions, some having access to archives, others using online

sources and some using their own uploads. Microsoft PowerPoint is the most well-known presentation tool, followed by Google Slides. All of the participants selected the available features listed as options, such as changing font, colours, size and orientation of elements, selecting multiple elements, downloading and sharing exhibitions. Some interesting features were listed in the suggested features section that were considered and implemented where possible and left for future work if not included in the current scope. The final question revealed PDF to be the most popular file download type.

## 3.3 Development Architecture

The project's architecture is structured as a Dynamic Web Applications with four layers: the Browser layer, Application and Logic layer, Database Connection layer and the Database layer. This architecture is displayed in Figure 2.

**The Front-End** is developed using HTML and Bootstrap CSS. It provides a clean and uniform appearance for the Web Application and helps keep the application appearance consistent across different Web browsers.

**The Client Services** are developed using HTML and JavaScript and the Python CGI module. These tools are used to create an application that allows a user to interact with the system and based on their interactions generate new content for the HTML pages. The Web application is dynamic because the application involves some redirection based on the users' interactions with different features of the system after starting on a landing page. Ajax was used to fetch and store server information.

**Database Management** consisted of storing data about populated exhibits, templates and archive data in JSON files. These files are used to populate exhibits and render templates for viewing and displaying them on a browse page so that other users can view them.

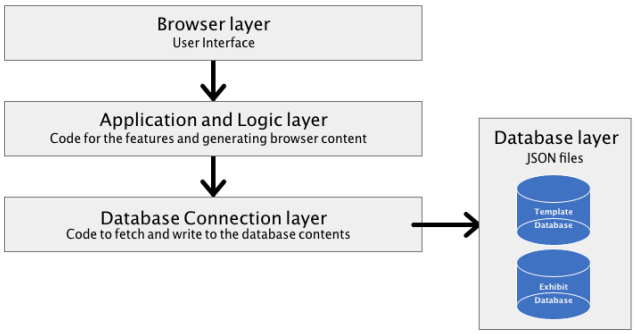


**Figure 2: High Level Overview of Exhibition population system architecture**

## 3.4 High Level System Architecture

The architecture of exhibition population is similar to a widget based Web application. [23] It relies on Ajax calls to query different Web services provided by the exhibition population system. These services include uploading local images, saving and downloading exhibits, viewing exhibits and displaying them on the browse page. Figure 3 is an example of a widget based Web application structure, similar to the architecture of exhibition population.

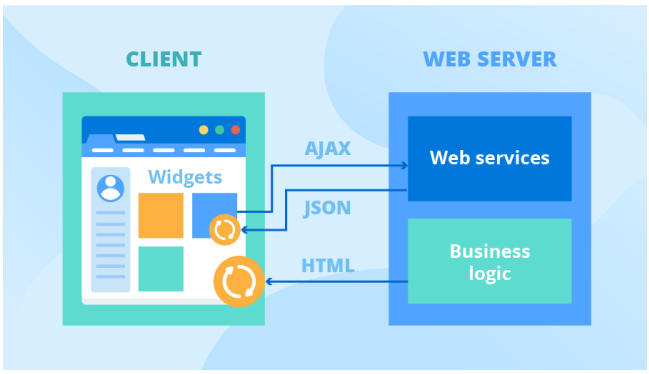


**Figure 3: A high level system architecture overview of exhibit population**

The development environment of the project was an Amazon Web Services EC2 Ubuntu instance. [4] The Apache server was setup with a Python CGI module [12] to handle server side code and generate code for the client browser.

The exhibit population system loads template data from the corresponding JSON file based on the template ID. The exhibit, at this point, is the empty template with the navigation bar at the top of the screen. This can be seen in Figure 4.

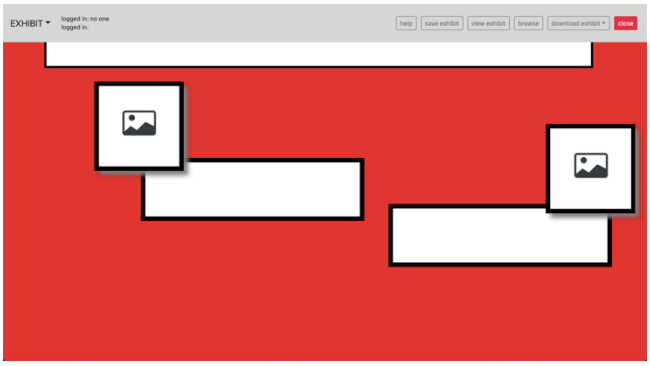

**Figure 4: The exhibition population system when opening a template to begin populating**

After opening a template to begin populating, a user can hover over the image box to bring up options to insert either personal image via a local upload, or an archive image. Figure 5 shows the form interface for a locally uploaded image.

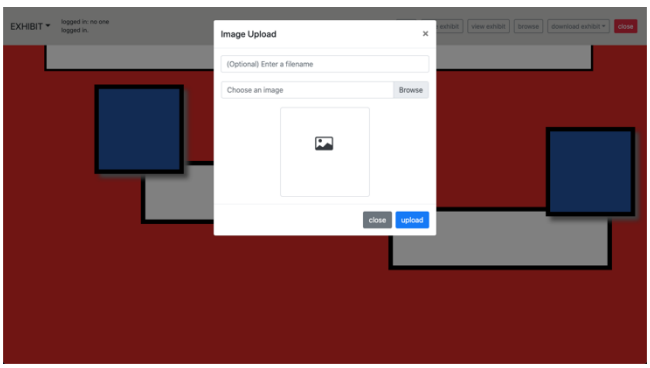


**Figure 5: The exhibition population system local upload option**

A user can populate their exhibit with text via a text editor. The user clicks on the option available to bring up the text editor, this brings up the CKEditor tool. The toolbar of CKEditor has been customised according to the user requirements survey feedback. This includes styling options and tools to copy formatting, include links and other options.

A browse view of the archive content is also available. This allows the users to personalise their tray with content from the archive. Figure 6 displays archival content in the form of cards that can be incorporated in the exhibition.

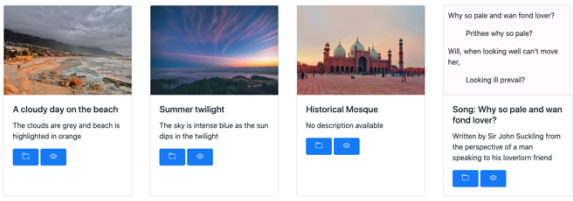


**Figure 6: Cards of archival content that can be included in the exhibition**

Once the exhibit is populated with content, the user can save the exhibit. The save option is kept on the navigation bar. Once clicked it brings up a modal with a form for the exhibit details. This includes the creators of the exhibit, a title and description for the exhibit and a cover image. This information is all stored in the exhibit JSON database, along with all of the information of the content that is populating the exhibit. The exhibit once stored in the database, can be viewed by clicking the view option in the navigation bar, and displays as a card on the browse page. Figure 7 is an example of an exhibition created using the exhibition population system.

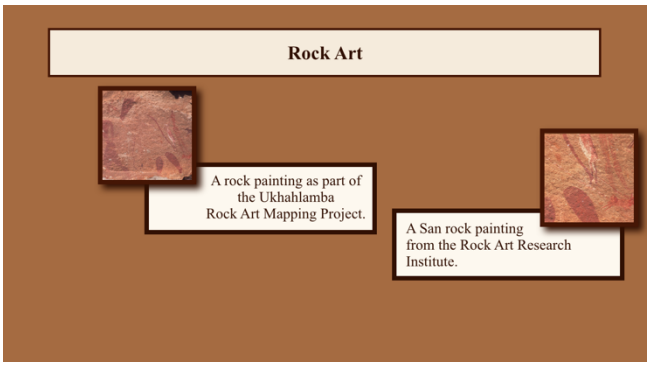


**Figure 7: A completed exhibit using the exhibit population system**

From the exhibit population stage the user can download the exhibit as an HTML or PDF file. It is also viewable from the Browse page. The exhibition population system incorporated with the archival material makes for an easily accessible and usable system to create digital exhibitions.

## 3.5 Design Methodology

This project was built using iterative design processes. Agile methodology was the focus of development to account for evolving requirements based on user feedback and to allow the developers to adapt to changing circumstances. It was designed with Human Computer Interaction (HCI) principles and iterative, user-centred design. This type of development focuses on a design phase, followed by a create phase, and then feedback phase, which is repeated in cycles. This helps to continuously improve the system. This also requires various design artefacts to be created to assist with the development of the system.

### *3.5.1 Iteration One*

The Requirements Gathering survey was conducted during iteration one, along with weekly meetings with the project supervisor to ensure an understanding of certain requirements or expectations of the system in terms of the larger project. This is when HCI artefacts were created, such as user personas and use cases in order to inform the prototyping phase. Personas are profiles created based on user archetypes: curator, student. These personas are used to help the developers understand the users' motivations and struggles in order to create a better prototype and product. An example of a persona used in this project can be seen in Figure 8. This persona was created using an online tool called Hubspot. [18]

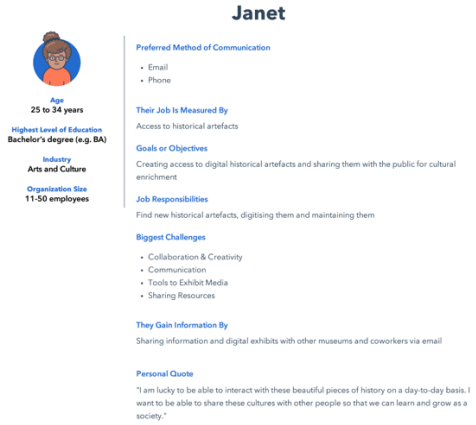


**Figure 8: A persona of one of the user archetypes of EXHIBT – a museum curator**

After the Requirements Gathering phase, the prototypes were developed. The first prototype developed was a paper prototype, as seen in Figure 11. These prototypes were designed quickly with minimal focus on precision, with more attention put on how the system will operate and transition, to gather an understanding of the flow of the project. These prototypes were developed individually by each team member on their respective feature focus. Using the created paper prototypes, the systems were evaluated through Cognitive Walkthroughs and Wizard of Oz evaluations, conducted by three HCI Honours students. Cognitive Walkthroughs were informed through the use of personas and use cases that were drawn up to ensure a user-centred approach to the testing of the prototype. As part of the development project, many Unified Modelling Language (UML) artefacts were generated, one of these being an activity diagram of the exhibit population

process. This along with other UML artefacts can be accessed on the Exhibit population website. Figure 9 is an example of the activity diagram UML artefact. It showcases how a user would interact with the system and the system's corresponding responses based on this user interaction. Figure 10 is the use case diagram for the exhibit population system, another of the UML artefacts generated as part of the development project lifecycle and design process.

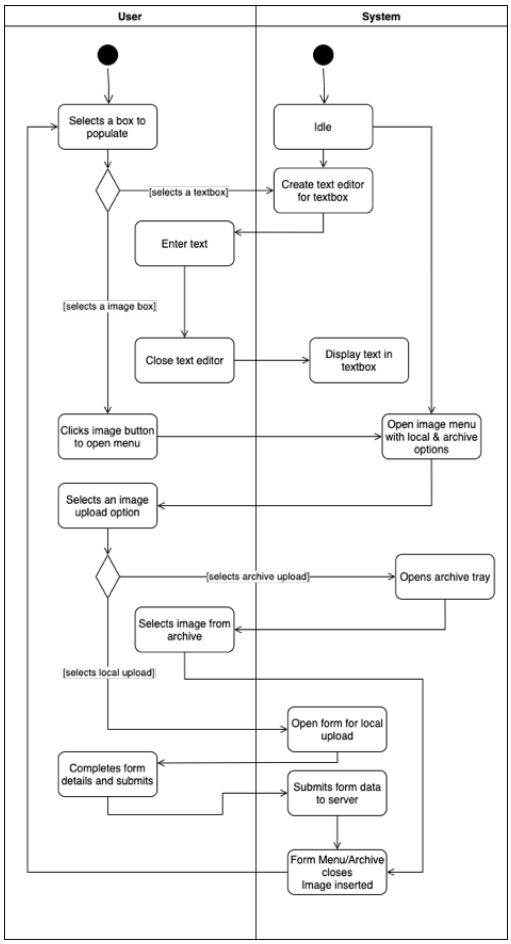


**Figure 9: Activity diagram documenting the exhibit population process**

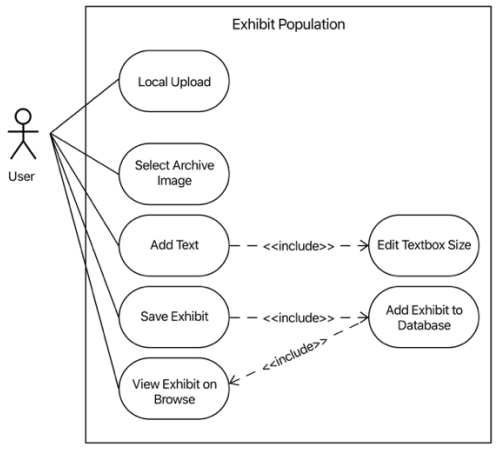


**Figure 10: Use case diagram of the exhibit population stage**

### *3.5.2 Iteration Two*

These paper prototypes provided further insights and allowed the developers to create a second prototype. The new digital prototype was designed by the team members through video conferencing and using online collaborative whiteboard services. It encompassed the system as a whole and was created using Adobe XD. This digital prototype was the design that was used to create the software prototype, which was created with minimal functionality using Feature Driven Development. The minimal functionality included a majority of the features, but low-level versions of these features, without extensive Bootstrap CSS. The software prototype was tested for User Acceptance Testing by the developers. This software prototype was used in the usability study that was conducted by a variety of participants including students, researchers and museum archivists. These participants were asked to complete an online survey after completing a series of tasks that were specified by the developers. The survey included Likert scale questions and questions specific to each feature of the Web application.

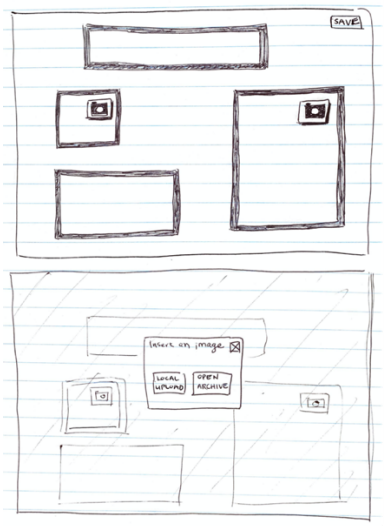


**Figure 11: Opening the image menu for exhibition population**

### *3.5.3 Iteration Three*

After completing the system usability survey and exploring the results, the prototype was resolved of any major bugs or errors the users faced and some suggested features were implemented to improve the usability of the system. The final development of the project was completed in this phase. All styling and remaining Bootstrap was added to give the final system a professional appearance that was pleasant to interact with.

## 4 TEST DESIGN AND METHODS

### 4.1 User Acceptance Testing

User acceptance tests were conducted by the developers while using personas and use cases developed in the first iteration. This was due to time constraints of the project, awaiting further ethical clearance and due to the COVID-19 pandemic. At this point the user personas created in the design phase were particularly useful in understanding the system from the perspective of the user. The

acceptance testing phase was used to determine whether or not the software met the functional requirements listed in the requirements gathering phase of the project. The features were graded as passable when the action could be completed successfully without the occurrence of a critical error. Table 2 displays the features of the exhibition population phase and its results during the user acceptance testing phase.

### 4.2 Usability Testing

A usability survey was completed by 12 participants of varying occupations and groups. The participants were made up of students and varying museum curators and digital archive managers. Participants were chosen randomly from a pool of suggested participants from the project supervisor. Students were randomly selected from a pool created by the project members. The survey was conducted online and followed by online video conferencing meetings with participants that opted to be interviewed for further information regarding their testing experience with the system. Table 3 in the results section displays the responses from the Likert scale portion of the survey.

The survey questions focused on testing navigation, how the users interacted with elements and if the interaction was intuitive. It also checked for any important suggested missing features. The flow of Web app and accessibility was also discussed in the survey. A large portion of the survey also included a Likert usability questionnaire. Likert usability questions are scaled questions ranging from strongly disagree to strongly agree. The scale used in this survey had an alternating scale with option 1 to 5. This meant that 1 on odd questions meant strongly disagree and then on even question 1 referred to strongly agree. This was swapped in the interviews to have a consistent scale as to avoid any issues with users misreading the scale. The exhibition population stage questions focused on how users understood the process of populating the exhibits and what functions were intuitive versus which needed more work to be effectively usable.

### 4.3 User Interviews

Some testers were contacted for more in-depth explanations regarding their experience of the system. The interviews were conducted via online video conferencing. The questions covered the interviewees access to digital archival content and the types of exhibitions they create, (i.e. content versus narrative-driven). The next questions focused on feedback from the interviewee, followed by general feedback from all the participants. The final questions covered the Likert scale again, but this time with a strictly positive scale. This meant that 1 referred to strongly disagree and 5 strongly agree throughout the questions. The Likert questions were asked again to account for the original version in the survey that had an alternating scale.

### 4.4 System Testing

Automated testing via the Jasmine framework was used in order to assess the system and to determine that it behaves as expected. These tests checked for accuracy and to ensure that the core of the project is functional.

### 4.5 Heuristic Evaluation

The heuristic evaluation was completed via discount usability. This was completed by three HCI Honours students and their feedback in conjunction with feedback identified in the usability survey was used to address critical areas of the project that needed more work or added features to make them effective. The usability survey responses were analysed to identify common issues that users mentioned as hotspots for errors and in the interviews, it was discussed with the participants the best approach to deal with these issues that were identified.

### 4.6 Speed Test

The online service Dareboost [11] was used to simulate a speed test of a user accessing the exhibit population system. The service provides a percentage score and lists “issues” as items that can be fixed to improve speed immediately, and “improvements” as less-critical items that can be fixed to help improve speed, but not as significantly as the “issues”. The service provides a report of the test with a video clip of the loading procedure, as well as with the information of the simulated user, (e.g. the browser they are using, and location they are accessing from).

## 5 RESULTS AND DISCUSSIONS

### 5.1 User Acceptance Test Results

The user acceptance test results are in Table 2. The features for the exhibit population stage all worked as intended in the first software prototype. There were no critical errors that warranted any failures for any of the features. All features were tested and provided acceptable results that allowed for the prototype to proceed to the next phase of development. The next phase included some basic styling of the working features to allow for the user testing to commence. It was later discovered that while all features were essentially working, some features had bugs that resulted in errors. These were discovered in user testing. These bugs that were not originally found in user acceptance testing could be attributed to the fact that the users testing the features were Computer Science Honours students and therefore, have extensive understanding of computer systems. A limiting factor of this research was the inability to meet with the actual users in-person for more extensive testing and waiting for ethical clearance for the project.

**Table 2: Acceptance tests completed by the developer**

| ID | CRITERIA | RESULT |
|---|---|---|
| 1 | Exhibition Population | PASS |
| 1.1 | All features of the template are loaded | PASS |
| 1.2 | Template features have the correct styling | PASS |
| 1.3 | Image boxes accept local uploads | PASS |
| 1.4 | Image boxes accept archive uploads | PASS |
| 1.5 | Textboxes are editable | PASS |
| 1.6 | Viewing an exhibit shows all elements correctly | PASS |
| 1.7 | Can change the filename of a local upload | PASS |
| 1.8 | Enter exhibit name and creator name | PASS |
| 1.9 | Upload an exhibit cover image | PASS |
| 1.10 | Save an exhibit | PASS |
| 2 | Exhibit Browse | PASS |
| 2.1 | View an exhibit | PASS |
| 2.2 | Comment on an exhibit | PASS |
| 2.3 | Saved exhibit is visible on browse page | PASS |
| 2.4 | Exhibit is downloadable as PDF and HTML | PASS |

## 5.2 System Usability Survey Results

The system usability test was conducted via an online survey. The results were from a number of participants including museum curators, archivists, researchers and students. The first question asked participants how frequently they would make use of such a system, the results showed that archivists scored more likely than students. The system usability survey asked participants on an alternating scale. This was changed in the interview portion to a strictly positive scale to account for any unintentional bias.

The system usability survey results are in Table 3. These results are used to calculate a system usability score. The score for the exhibition population project is 73, and is therefore, considered better than the industry standard of 68; therefore, the usability of the system is above average and rates in the adjective rating of Good. [21]

**Table 3: System Usability survey responses**

| | SYSTEM USABILITY SCALE QUESTIONS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **PARTICIPANTS** | | | | | | | | | | |
| **1** | 3 | 2 | 5 | 1 | 3 | 1 | 4 | 1 | 5 | 1 |
| **2** | 2 | 2 | 4 | 3 | 4 | 2 | 5 | 2 | 2 | 4 |
| **3** | 3 | 1 | 5 | 1 | 4 | 2 | 5 | 1 | 5 | 1 |
| **4** | 3 | 2 | 2 | 1 | 5 | 2 | 4 | 4 | 3 | 1 |
| **5** | 3 | 2 | 4 | 1 | 4 | 2 | 5 | 2 | 4 | 1 |
| **6** | 2 | 2 | 3 | 3 | 3 | 3 | 5 | 4 | 3 | 2 |
| **7** | 4 | 2 | 4 | 1 | 4 | 2 | 5 | 2 | 4 | 1 |
| **8** | 5 | 1 | 3 | 1 | 5 | 3 | 5 | 3 | 3 | 1 |
| **9** | 5 | 4 | 1 | 1 | 2 | 4 | 4 | 5 | 2 | 1 |
| **10** | 3 | 4 | 4 | 4 | 3 | 2 | 2 | 4 | 2 | 4 |
| **11** | 3 | 4 | 3 | 2 | 3 | 1 | 5 | 2 | 5 | 2 |
| **12** | 3 | 5 | 5 | 1 | 3 | 1 | 5 | 1 | 4 | 5 |

## 5.3 Heuristic Evaluation Results

Jakob Nielsen developed a Heuristic evaluation [19] to test the usability of a system. This involves measuring aspects of the system to analyse which areas of the project require the most attention.

### *5.3.1 Visibility of System Status*

Feedback from the participants was positive for most of the project. However, saving exhibits was deemed not to be informative enough. The status of the save was unclear, the alert that informed successful save took too long to appear. This meant that users resubmitted their exhibit numerous times, resulting in multiple uploads of the same exhibit to the browse page. During interviews with the participants we discussed more informative methods of communicating the save status and it was deemed that a progress bar would be more intuitive to indicate the status of the save.

### *5.3.2 Match Between System and Real World*

Majority of feedback indicated that users were comfortable with the design of the system. Most felt that they could easily interact with the elements without any barriers of discovering an elements function.

### *5.3.3 User Control and Freedom*

User discovered a critical error with the textboxes proving to be a major issue as they covered many important menu options. Options such as saving the exhibit and accessing the image menu on image boxes. This was resolved with a navigation bar that could be opened and closed and image menu options being brought into view when hovering over the image box. Text editors are also sized in accordance to the textbox that generated them.

### *5.3.4 Consistency and Standards*

Most users were satisfied with the level of consistency in the design. More consistency with the rest of the system was achieved

through the introduction of a navigation bar. The style of the system is consistent by using Bootstrap CSS. Web development standards are achieved via using Bootstrap, making use of their predefined classes for modals, navigation bars and toasts.

### *5.3.5 Error Prevention*

Error messaging regarding saving exhibits was deemed inadequate by the users. More detailed error messages were created to mitigate this.

### *5.3.6 Recognition over Recall*

A mixture of icons and text is used for the project. Text is used were standard icons are not universally known. For example, adding a local image or archive upload on the image box uses the words instead of icons because the words are more intuitive; however, for downloading, the generic down arrow icon was used because this is used throughout browsers and is intuitive.

### *5.3.7 Flexibility and Efficiency of Use*

In the usability survey, the users were asked which default of text editor sizing they would prefer, some users suggested that a default is necessary, but options to edit the size of the text editor and the textbox to accommodate entered text should be available.

### *5.3.8 Aesthetic and Minimalist Design*

Majority of participants commented on the clean design of the population menu and styling. Using Bootstrap CSS was favourable because of their clean design, styling options and allowing a standard design across the whole system.

### *5.3.9 User Error Recovery*

Many users commented on this aspect as a failure of the system. Users would encounter an error or problem, such as with the text editor blocking menu options, and would then be unable to recover from these errors. Text editor and box resizing offered a chance of error recovery. More informative save status updates helped to prevent errors from occurring in the first place. An option to delete an exhibit from browse, provided you are the creator, was also provided as a form of user error recovery.

### *5.3.10 Help and Documentation*

One user commented on the lack of help documentation. After consulting with user in the interviews, it was deemed that tooltips and a help button with clips of how to interact with elements was sufficient for a new user to start interacting with the system.

## 5.4 Speed Test Results

The speed test identified nine areas for improvement, these changes included how images were loaded into the browser to help save bytes of data, how redirecting could be improved and some reminders of internet Webpage standards, such as having a description tag for the Webpage. A snapshot of the report can be seen in Figure 12. The overall percentage score of the exhibit population system was 71% pre-changes, which is above average. However, it should also be noted that a large portion of the project is dependent on JavaScript to generate code for the Webpage and Dareboost [11] does put emphasis on minimising JavaScript code where possible, that is unavoidable in this case.


**Figure 12: Results of the Dareboost speed test**

# 6 LIMITATIONS

At the time of the project, in accordance with the COVID-19 pandemic guidelines and regulations as stipulated by the country, in-person meetings were not an option. This severely impacted the project as communications between members, the supervisor and all participants were strictly online via email and video conferencing. Aspects such as the user acceptance testing had to be completed initially by the project developers, these tests would have been better conducted by participants, but this was not possible with the pandemic and restrictions at the time. Approval for the project's ethical clearance took a large portion of the project's timeline which was limiting. User requirements gathering was limited to persons outside of the UCT system as the team had to wait for clearance to access UCT Staff and Students. This meant that the pool of responses was less than desired. As a result of the online limitations, the usability survey was not able to be discussed in person with participants and so, some participants mistakenly selected incorrect values for the Likert scale portion.

# 7 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

Ethical clearance for the project was received from the UCT Faculty of Science Research Ethics Committee. In order to interview UCT Students and Staff further ethical clearance was received from the Executive Director of Student Affairs and the Executive Director of Human Resources. Ethical clearance forms can be accessed on the project website for review. In order for the users to participate in the study, all had to read an informed voluntary consent form and agree to it before accessing the survey. In the survey there was a voluntary question which asked participants to provide their email to be contacted for interviews. The interview was in order to obtain further information regarding the feedback provided in the survey. Participants were again reminded of the voluntary consent form and their ability to withdraw from the study.

# 8 CONCLUSIONS

In this paper, a system for populating templates with archival and local content in order to create digital exhibitions was presented. The HCI focus of the project and Agile methodology was implemented to create the system over various iterations by approaching the system with Feature Driven Development. Throughout the design process, many design artefacts were generated, these artefacts were created after some requirements

gathering and after meetings discussing the system with the project supervisor. All of this resulted in a dynamic Web application being built using HTML, JavaScript, the Python CGI Module technologies and styled using Bootstrap CSS. The application was developed on an Apache server via an EC2 Instance set up with Amazon Web Services.

The system was tested extensively using user acceptance testing, automated testing, a speed test, a system usability test and a heuristic evaluation. The system usability test and heuristic evaluation were the most insightful tests as they provided feedback directly from the types of users expected for the system. All errors have since been corrected with some suggested features incorporated into the project to make it more functional and others left for future work as they could not fall within the scope of the project.

The general response from users showed that there is significant interest in such a tool that allows users to interact with archival material whilst populating exhibitions. The results of the usability test, along with the feedback from interviews and the heuristic evaluation showed that the exhibition population was successfully implemented. Exhibit population on the whole was successfully implemented according the results achieved throughout the process and attributed to the methodologies and technologies used to make the tool possible.

## 9 FUTURE WORK

Some of the responses from the usability survey expressed areas that could be left for future work as they did not fall within the scope of the project. These were aspects that could be developed more in-depth and fully, such as more advanced search functionality and more full-featured user accounts systems to view their exhibits, or colleagues' exhibits. Live collaboration (such as in Google Docs), to be able to populate an exhibit collaboratively. The ability to make your exhibits private, so that either only the creator, or specified users can have viewing access. More full-featured metadata viewing and incorporation. Another user expressed interest in adding referencing to text or image sources. Currently in text sources, the user can provide a link and create a reference themselves, but this as an incorporated feature was suggested with interest by the participants.

## 10 ACKNOWLEDGEMENTS

I would like to thank Dr Hussein Suleman for his dedicated support and guidance throughout the duration of this project, his expertise was invaluable. I would also like to thank my teammate, Aa'isha Dout, without whom this project would not have been possible. Her encouragement was greatly appreciated. I would also like to thank the museum curators, archivists and students that participated in this study. Your feedback was especially helpful and valued.

## REFERENCES


[1] [n.d.]. *The Five Hundred Year Archive.* Retrieved May 16, 2020 from http://www.apc.uct.ac.za/apc/research/projects/ve-hundred-year-archive

[2] [n.d.]. *THE DIGITAL BLEEK AND LLOYD.* Retrieved May 16, 2020 from http://lloydbleekcollection.cs.uct.ac.za/

[3] Agile Modelling. 2020. *Feature Driven Development and Agile Modelling.* Retrieved September 10, 2020 from http://agilemodeling.com/essays/fdd.htm

[4] Inc. Amazon Web Services. 2020. *Web Hosting.* Retrieved June 02, 2020 from https://aws.amazon.com/websites/

[5] Daniel Biella, Wolfram Luther, and Nelson Baloian. 2010. *Beyond the ARCO standard.* In Proceedings of the 16th International Conference on Virtual Systems and Multimedia. IEEE, 184–191. https://ieeexplore.ieee.org/document/5665985

[6] John Brooke. 1986. *System usability scale (SUS): A quick-and-dirty method of system evaluation user information. In Usability Evaluation In Industry.* Taylor & Francis, Reading, UK, 1–7.

[7] Inc. Bubble Group. 2020. *Getting started.* Retrieved May 20, 2020 from https://bubble.io/documentation

[8] Carrd. 2020. *Documentation: General.* Retrieved May 20, 2020 from https://carrd.co/docs/general

[9] John P Chin, Virginia A Diehl, and Kent L Norman. 1988. *Development of an instrument measuring user satisfaction of the human-computer interface.* In Proceedings of the SIGCHI conference on Human factors in computing systems. 213–218.

[10] Gobinda Chowdhury. 2009. *From digital libraries to digital preservation research: the importance of users and context.* Journal of Documentation 66, 2 (July 2009), 207–223. https://tefkos.comminfo.rutgers.edu/Courses/Zadar/Readings/Chowdhury%20dig%20pres%20J%20Doc%202009.pdf

[11] Dareboost. 2020. *Website Speed Test and Website Analysis.* Retrieved September 10, 2020 from https://www.dareboost.com/en

[12] Python Software Foundation. 2020. *cgi — Common Gateway Interface support.* Retrieved July 9, 2020 from https://docs.python.org/2/library/cgi.html

[13] The Apache Software Foundation. 2020. *Apache Solr.* Retrieved May 16, 2020 from https://lucene.apache.org/solr/

[14] Paula Goodale, Mark Hall, Kate Fernie MDR, Phil Archer, Francesca Tavanti, and Paul Clough. 2011. D 1.1 *User Requirements Analysis.* PATHS Project (2011).

[15] Juliet Hardesty. 2014. *Exhibiting library collections online: Omeka in context.* New Library World 115, 3 (March 2014), 75–86. http://dx.doi.org/10.1108/NLW-01-2014-0013

[16] Sam Habibi Minelli, Maria Teresa Natale, Paolo Ongaro, Marzia Piccininno, Rubino Saccoccio, and Daniele Ugoletti. 2014. *MOVIO: A toolkit for creating curated digital exhibitions.* Procedia Computer Science 38 (January 2014), 28–33. https://www.sciencedirect.com/science/article/pii/S1877050914013660/pdf?md5=ef95993470fa0f6eb980ae581c5d5bce&pid=1-s2.0-S1877050914013660- main.pdf&_valck=1

[17] Dinnie Muslihat. 2020. *Agile Methodology: An Overview.* Retrieved May 22, 2020 from https://zenkit.com/en/blog/agile-methodology-an-overview/

[18] Hubspot. 2020. *Hubspot Tools: Make My Persona.* Retrieved May 14, 2020 from https://www.hubspot.com/make-my-persona

[19] Jakob Nielsen. 2012. *How Many Test Users in a Usability Study?* Retrieved May 12, 2020 from https://www.nngroup.com/articles/how-many-test-users/#:~:text=If%20you%20want%20a%20single,using%20many%20more%20test%20participants.

[20] Product Plan. 2020. *Feature Driven Development (FDD).* Retrieved September 10, 2020 from https://www.productplan.com/glossary/feature-driven-development/

[21] UIUX Trend. 2020. *Measuring and Interpreting System Usability Scale (SUS).* Retrieved September 10, 2020 from https://uiuxtrend.com/measuring-system-usability-scale-sus/#interpretation

[22] Ruan Yang, Chennupati K Ramaiah, and Schubert Foo. 2007. *Virtual Archival Exhibition System: An authoring tool for developing web-based virtual exhibitions.* In International Conference on Dublin Core and Metadata Applications. Dublin Core Metadata Initiative (DCMI), 96–105. http://dcpapers.dublincore.org/pubs/article/viewFile/870/866

[23] Anastasia Yaskevich. 2017. *Web application architecture: Components, models and types.* Retrieved September 15, 2020 from https://www.scnsoft.com/blog/web-application-architecture

[24] [n.d.]. *The African Rock Art Digital Archive.* Retrieved May 16, 2020 from http://www.sarada.co.za/#/library/