Software Design Document

For

**The Arqive**

**Version 1.2**

**8 December 2023**

**Sponsors: Zachary Vernon and Cynthia Wang Advisor: John Hurley**

**Prepared by:** *Juan Alcobas, Hak Beak, Ryan Goshorn, Melissa Hernandez, Thien Ho, Channing Jou, Raul Martinez Sanchez, Daniel Rodas, Stephanie Tan, Antonio Vazquez Bravo*

**California State University, Los Angeles – The Arqive**

[**Revision History 5**](#_heading=h.30j0zll)

[**1. Introduction 6**](#_heading=h.1fob9te)

[1.1 Purpose 6](#_heading=h.3znysh7)

[1.2 Document Conventions 6](#_heading=h.2et92p0)

[1.3 Intended Audience and Reading Suggestions 6](#_heading=h.tyjcwt)

[1.4 System Overview 6](#_heading=h.3dy6vkm)

[**2. Design Considerations 8**](#_heading=h.1t3h5sf)

[2.1 Assumptions and Dependencies 8](#_heading=)

[2.2 General Constraints 8](#_heading=)

[2.3 Goals and Guidelines 8](#_heading=)

[2.4 Development Methods 9](#_heading=h.4d34og8)

[**3. Architectural Strategies 10**](#_heading=h.2s8eyo1)

[**4. System Architecture 11**](#_heading=h.17dp8vu)

[**5. Policies and Tactics 14**](#_heading=h.3rdcrjn)

[5.1 Choice of which specific products used 14](#_heading=h.26in1rg)

[5.2 Plans for ensuring requirements traceability 14](#_heading=h.lnxbz9)

[5.3 Plans for testing the software 14](#_heading=h.35nkun2)

[**6. Detailed System Design 15**](#_heading=h.1ksv4uv)

[6.1 User Account Requirements 15](#_heading=h.44sinio)

[6.1.1 Responsibilities 15](#_heading=h.2jxsxqh)

[6.1.2 Constraints 15](#_heading=h.z337ya)

[6.1.3 Composition 15](#_heading=h.3j2qqm3)

[6.1.4 Uses/Interactions 15](#_heading=h.1y810tw)

[6.1.5 Resources 16](#_heading=h.4i7ojhp)

[6.1.6 Interface/Exports 16](#_heading=h.2xcytpi)

[6.2 Map Requirements 16](#_heading=h.1hmsyys)

[6.2.1 Responsibilities 16](#_heading=h.uhnx3kres68p)

[6.2.2 Constraints 16](#_heading=h.351l2sneua4v)

[6.2.3 Composition 16](#_heading=h.awxqtqgvc790)

[6.2.4 Uses/Interactions 16](#_heading=h.men5fgwdt032)

[6.2.5 Resources 16](#_heading=h.u5vpqwofyat7)

[6.2.6 Interface/Exports 17](#_heading=h.49tn0zbhl44v)

[6.3 Story Requirements 18](#_heading=h.2u6wntf)

[6.3.1 Responsibilities 18](#_heading=h.19c6y18)

[6.3.2 Constraints 18](#_heading=h.3tbugp1)

[6.3.3 Composition 18](#_heading=h.28h4qwu)

[6.3.4 Uses/Interactions 18](#_heading=h.nmf14n)

[6.3.5 Resources 18](#_heading=h.37m2jsg)

[6.3.6 Interface/Exports 18](#_heading=h.1mrcu09)

[6.4 Platform Requirements 19](#_heading=h.46r0co2)

[6.4.1 Responsibilities 19](#_heading=h.2lwamvv)

[6.4.2 Constraints 19](#_heading=h.111kx3o)

[6.4.3 Composition 19](#_heading=h.3l18frh)

[6.4.4 Uses/Interactions 19](#_heading=h.206ipza)

[6.4.5 Resources 19](#_heading=h.4k668n3)

[6.4.6 Interface/Exports 19](#_heading=h.2zbgiuw)

[6.5 Multimedia Requirements 19](#_heading=h.1egqt2p)

[6.5.1 Responsibilities 19](#_heading=h.3ygebqi)

[6.5.2 Constraints 20](#_heading=h.2dlolyb)

[6.5.3 Composition 20](#_heading=h.sqyw64)

[6.5.4 Uses/Interactions 20](#_heading=h.3cqmetx)

[6.5.5 Resources 20](#_heading=h.1rvwp1q)

[6.5.6 Interface/Exports 20](#_heading=h.4bvk7pj)

[6.6 Security Requirements 20](#_heading=h.2r0uhxc)

[6.6.1 Responsibilities 20](#_heading=h.1664s55)

[6.6.2 Constraints 20](#_heading=h.3q5sasy)

[6.6.3 Composition 20](#_heading=h.25b2l0r)

[6.6.4 Uses/Interactions 21](#_heading=h.kgcv8k)

[6.6.5 Resources 21](#_heading=h.34g0dwd)

[6.6.6 Interface/Exports 21](#_heading=h.1jlao46)

[6.7 User Role Requirements 21](#_heading=h.43ky6rz)

[6.7.1 Responsibilities 21](#_heading=h.2iq8gzs)

[6.7.2 Constraints 21](#_heading=h.xvir7l)

[6.7.3 Composition 21](#_heading=h.3hv69ve)

[6.7.4 Uses/Interactions 21](#_heading=h.1x0gk37)

[6.7.5 Resources 22](#_heading=h.4h042r0)

[6.7.6 Interface/Exports 22](#_heading=h.2w5ecyt)

[6.8 FAQ Requirements 22](#_heading=h.1baon6m)

[6.8.1 Responsibilities 22](#_heading=h.3vac5uf)

[6.8.2 Constraints 22](#_heading=h.2afmg28)

[6.8.3 Composition 22](#_heading=h.pkwqa1)

[6.8.4 Uses/Interactions 22](#_heading=h.39kk8xu)

[6.8.5 Resources 22](#_heading=h.1opuj5n)

[6.8.6 Interface/Exports 22](#_heading=h.48pi1tg)

[6.9 About Us Requirements 23](#_heading=h.2nusc19)

[6.9.1 Responsibilities 23](#_heading=h.1302m92)

[6.9.2 Constraints 23](#_heading=h.3mzq4wv)

[6.9.3 Composition 23](#_heading=h.2250f4o)

[6.9.4 Uses/Interactions 23](#_heading=h.haapch)

[6.9.5 Resources 23](#_heading=h.319y80a)

[6.9.6 Interface/Exports 23](#_heading=h.1gf8i83)

[6.10 Resources Requirements 24](#_heading=h.40ew0vw)

[6.10.1 Responsibilities 24](#_heading=h.2fk6b3p)

[6.10.2 Constraints 24](#_heading=h.lr1bxsic1xtd)

[6.10.3 Composition 24](#_heading=h.upglbi)

[6.10.4 Uses/Interactions 24](#_heading=h.3ep43zb)

[6.10.5 Resources 24](#_heading=h.1tuee74)

[6.10.6 Interface/Exports 24](#_heading=h.4du1wux)

[6.11 Accessibility Requirements 24](#_heading=h.2szc72q)

[6.11.1 Responsibilities 24](#_heading=h.184mhaj)

[6.11.2 Constraints 25](#_heading=h.3s49zyc)

[6.11.3 Composition 25](#_heading=h.279ka65)

[6.11.4 Uses/Interactions 25](#_heading=h.meukdy)

[6.11.5 Resources 25](#_heading=h.36ei31r)

[6.11.6 Interface/Exports 25](#_heading=h.1ljsd9k)

[6.12 Credits Requirements 25](#_heading=h.tx6s7yw3640z)

[6.12.1 Responsibilities 25](#_heading=h.faef82c5f5ie)

[6.12.2 Uses/Interactions 26](#_heading=h.ahwua4eiwlxt)

[6.12.3 Resources 26](#_heading=h.had0wy92trpo)

[6.12.4 Interface/Exports 26](#_heading=h.z6287gal3a78)

[**7. Detailed Lower level Component Design 27**](#_heading=h.1d96cc0)

[**8. Database Design 28**](#_heading=h.3x8tuzt)

[**9. User Interface 29**](#_heading=h.2ce457m)

[9.1 Overview of User Interface 29](#_heading=h.rjefff)

[9.2 Screen Frameworks or Images 30](#_heading=h.3bj1y38)

[9.3 User Interface Flow Model 31](#_heading=h.1qoc8b1)

[**10. Requirements Validation and Verification 33**](#_heading=h.4anzqyu)

[**11. Glossary 34**](#_heading=h.2pta16n)

[**12. References 35**](#_heading=h.14ykbeg)

# Revision History

| Name | Date | Reason For Changes | Version |
| --- | --- | --- | --- |
| First Rough Draft | November 9, 2023 | First document with no specific details | 1.0 |
| Second Rough Draft | December 5, 2023 | Document added with templates from previous versions, and other details | 1.1 |
| Third Draft | December 8, 2023 | Finalized formatting; additional content for blocking and nsfw tagging | 1.2 |
|  |  |  |  |

# Introduction

## Purpose

This document will provide the Software Design Specifications as part of the design plan for improving functionality and adding on features to The arqive. This document will expand on the functionality and features as described in the Software Requirements Specification (SRS). Each functionality that has been improved will be described in detail as well as additional features to be implemented. Overall, this document will expand on features the SRS presents along with design issues encountered.

For reference, the arqive is an LGBTQ+ website and mobile application where anyone can post stories that either happened in history or stories that have happened to them. Users can mark the location of their story using pins pointing to where it took place. The arqive gives people the ability to mark their place in the world and share their stories!

## Document Conventions

| SDD | Software Design Document |
| --- | --- |
| SRS | Software Requirements Specification |
| User | The individual using The arqive website and/or mobile app |
| Administrator | Users with permission to other functions of the website and/or mobile app, not available to regular users |

## Intended Audience and Reading Suggestions

This document is intended for Software Developers to have an understanding of the design behind the basic functionalities and implemented features that the SRS describes.

## System Overview

The arqive is a digital online storytelling map for LGBTQ+ stories that seeks to provide the full range of queer stories and geolocates them to digitally preserve the posts. Users have a safe platform where they can share personal, historical, and community stories, as well as have access to information about safe spaces. It is currently available as a web and mobile application.

# Design Considerations

## Assumptions and Dependencies

* OpenStreetMap
* ReactJS Framework
* User must be sharing their location
* Mobile device or Computer with a web browser
* iOS and/or Android device

## General Constraints

* Privacy for users is not always guaranteed.
* Too many story pins on the map may cause a slower load in for the user.
* Javascript must be enabled in the user’s web browser.
* Slow internet connection may affect the user’s experience.

## Goals and Guidelines

* Provide a platform for the LGBTQ+ community to post and share stories, experiences and have access to resources.
* Security and protection from malevolent agents that might cause harm to the arqive’s users.
* Filter out inappropriate content.
* Implement gamification to encourage and reward users for their activity on the app and/or web application. Have no ranking or

competitive system that other social media platforms reward users for.

* Highlight and display local stories.
* Block unwanted content.
* Ability for administrators and moderators to moderate flagged posts with the help of integrated content moderation.
* The mobile apps should implement additional functionality and be created with the established style of the arqive website, according to the guidelines provided.

## Development Methods

The development team followed the agile development method to design the additions to this project. The team was split into two to focus on different features. Both had direct communication with their sponsors. As each task was given, groups would divide into sub groups to create, implement, and merge the needed code into the back-end or front-end databases shared by the group. After the changes and implementations to the applications, the team would demonstrate the change to the sponsors to provide feedback and approval.

# Architectural Strategies

The applications of The arqive were built using Python, JavaScript, Django and React which resulted in having a flexible and responsive website. In addition, Django and React are libraries that are able to provide a more enhanced functionality to the website. The mobile app was built prior using React Native which helps to focus on one code base for both iOS and Android devices. A new mobile app has been created while the desktop version exists to ensure that all updates are seamless. The mobile app has been created with the help of a paid outside consultant, where during the next cycle, the student developer team will take over mobile development.

# System Architecture

****



# Policies and Tactics

## Choice of which specific products used

PostgreSQL, Digital Ocean, ReactJS, React Native, Django REST, OpenstreetMap, Python, Scikit-learn, Ubuntu, AR.js, Docker, Expo, Yarn, GitHub, Postman

## Plans for ensuring requirements traceability

Each developer will create test cases that will map to requirements on any effort they are working on. Additionally, teammates are encouraged to create complete documentation of features added as well as apply their changes via commits to git.

## Plans for testing the software

Testing the website and mobile app will be done by the developers, sponsors and third-party users asked to use the program to identify potential bugs or UI improvements. Testing the software may occur throughout the developmental process of both the website and mobile app.

# Detailed System Design

## User Account Requirements

### Responsibilities

 Separate the users into regular users, anonymous users,

 moderators, and administrators with unique roles.

 Add a blocking feature to allow users to block any users/posts.

### Constraints

We must assume that the roles with more power do not abuse others with their status. We must also make sure that actions other than reading a post are restricted to registered users.

### Composition

No current subcomponents.

### Uses/Interactions

The user’s role allows them to create, modify, or relinquish posts based on their role. Admin’s have the same basic functions that users have but can moderate and delete any posts, comments and/or users that may have violated the code of conduct. Users who are not logged in may also interact with the app but have limited functionality as opposed to users

who are logged in and registered. These users may still view posts, flag and unflag posts and comments. All other basic functions require a login.

### Resources

Modification of the database of pins is dependent on the user’s

role.

### Interface/Exports

No current exports.

## Map Requirements

### Responsibilities

The map is able to show all pins from the database.

### Constraints

The size of the map is set to be 100% of the viewing screen’s size.

### Composition

No current subcomponents.

### Uses/Interactions

The map will interact with the database in order to show and store location based pins via form submission.

### Resources

Digital Ocean PostgreSQL Database OpenStreetMap.

### Interface/Exports

No current exports.

## Story Requirements

### Responsibilities

The story page shows all the data of the pin. It allows users to favorite, report, or comment on the post. Users can create anonymous posts to hide their identity. Users can also block users from these stories.

### Constraints

Content containing explicit material will be flagged for moderation. Moderators will then decide on the appropriate action for flagged content.

### Composition

Favoriting a post.

### Uses/Interactions

Registered and logged in users will be able to view, favorite, bookmark or report the stories posted. Users not registered or logged in will only be able to view posts, flag comments and flag posts.

### Resources

Digitalocean PostgreSQL Development Database.

### Interface/Exports

No current exports.

## Platform Requirements

### Responsibilities

This allows the user to access the site with different browsers like Microsoft Edge, Firefox, Chrome, etc.

### Constraints

Not all available browsers will be covered.

### Composition

No current composition.

### Uses/Interactions

Users will be able to access and use the platform via the web and on mobile.

### Resources

No specific resources.

### Interface/Exports

No current exports.

## Multimedia Requirements

### Responsibilities

Allows users to embed pictures, videos, and other media onto their posts.

### Constraints

The file size must be reasonable.

### Composition

No current composition.

### Uses/Interactions

Users can upload media about the pin to let other people view it.

### Resources

No specific resources.

### Interface/Exports

It is part of the pin creation process.

## Security Requirements

### Responsibilities

The security will protect the site, its data and users.

### Constraints

Security can not be fully guaranteed but the security implemented will still offer safety.

### Composition

No current composition.

### Uses/Interactions

Ideally, the security would not be used for anything major. The users will not interact with the security unless the user causes acts in a malicious manner.

### Resources

No specific resources.

### Interface/Exports

No current exports.

## User Role Requirements

### Responsibilities

User roles will give people a sense of membership and provide moderation of content.

### Constraints

Currently, user roles only include registered users, administrators, and moderators.

### Composition

No current composition.

### Uses/Interactions

Allows users to have specific privileges within the application.

### Resources

Database.

### Interface/Exports

No current exports.

## FAQ Requirements

### Responsibilities

Provide information regarding the site. This also includes informing the user of the terms and service.

### Constraints

Limited to what administrators post.

### Composition

No current composition.

### Uses/Interactions

Allows Administrators to post FAQs and users to read them.

### Resources

Database.

### Interface/Exports

No current exports.

## About Us Requirements

### Responsibilities

Provide users with information on the mission of the arqive and the teams that helped create it.

### Constraints

Administrators are the only ones allowed to post in the “About Us” section.

### Composition

No current composition.

### Uses/Interactions

Allows administrators to edit the “About Us” and users to read them.

### Resources

Database.

### Interface/Exports

No current exports.

## Resources Requirements

### Responsibilities

Provides users with access to various valuable resources.

### Constraints

We may not provide all available hotlines and websites are not enough.

### Composition

No current composition.

### Uses/Interactions

Hotlines and Websites that mainly aid LGBTQ+ individuals will be posted for users to use.

### Resources

No current resources.

### Interface/Exports

No current exports.

## Accessibility Requirements

### Responsibilities

Allow users with disabilities to use the web app to its full potential.

### Constraints

We are using UserWay Widget, so we do not have control over their functionalities.

### Composition

No current composition.

### Uses/Interactions

Users will be able to use the UserWay Widget which allows them to have text spoken aloud among other things.

### Resources

UserWay Widget.

### Interface/Exports

No current exports.

## Credits Requirements

### Responsibilities

 Gives credit to the developer teams for the Arqive.

### Uses/Interactions

Names of the developers, art director, art designers, public relations teams, and others who worked on the Arqive will be credited here.

### Resources

No current resources.

### Interface/Exports

No current resources.

# Detailed Lower level Component Design

Refer to Section 4 of this Document.

# Database Design



# User Interface

## Overview of User Interface

The user will access the main content of the site by navigating to the homepage (https://thearqive.com). Within this page, users can post pins on the map by category (personal, historical, or community). Users can also access other stories by navigating around the map and clicking on pins, which will direct them to the individualized story page of the selected pin. Inside this individualized story page, users can read more detailed information about the pin, i.e., the story author. If clicked, the author’s name leads to their user profile which displays their profile picture, name, biography, and all previous stories created. In the site header, users have the ability to click the site logo to navigate back to the homepage, click the search button to search the website, click the login/logout button to login or logout, or click the Register button to sign up on the site. Also located in the site header are links that take users to the About Us, FAQs, Help, Contact Us, or Support Us pages. In the About Us page, users can read about the arqive and its

mission. In the FAQs page, users can read frequently asked questions and their respective answers. Within the Help page, users can access useful information and resources that help support the LGBTQ+ community. In the Contact Us page, users can submit a message directly to the arqive email. Lastly, within the Support Us page, users can support the website by sharing the #thearqive on other social media.

## Screen Frameworks or Images







## User Interface Flow Model

Homepage → Story Page

Story Page → Story Page Story Page → User Profile Site Header → Login Login → Register

Login → Homepage Site Header → Register Register → Login Register → Homepage

Site Header → Registration Site Header → FAQ

Site Header → About Us Site Header → Contact Us Site Header → Help

Site Header → Support Us Site Header → Homepage Site Header → Profile Profile → Settings

# Requirements Validation and Verification

Not applicable at this time.

# Glossary

| AR | Augmented Reality |
| --- | --- |
| DFD | Data Flow Diagram |
| Digital Ocean | Cloud infrastructure provider |
| Django REST | Toolkit for building Web APIs in Python |
| LGBTQ+ | Lesbian, Gay, Bisexual, Transgender, Queer, and more |
| Openstreetmap | An editable geographic database of the world |
| React JS | An Open source front end framework for developing websites |
| React Native | Open source mobile application framework for iOS and Android devices |
| SciKit-learn | A Machine learning framework |
| SRS | Software Requirement Specification |

# References

Brad Appleton [http://www.bradapp.net](http://www.bradapp.net/)

Design Template <https://www.cs.purdue.edu/homes/cs307/ExampleDocs/DesignTemplate_Fall08.doc>

**The arqive’s 2022 Development Team (SRS)** [https://csns.cysun.org/department/cs/project/resource/vie](https://csns.cysun.org/department/cs/project/resource/view?projectId=7913655&resourceId=7928311) [w?projectId=7913655&resourceId=7928311](https://csns.cysun.org/department/cs/project/resource/view?projectId=7913655&resourceId=7928311)

**The arqive’s 2021 Development Team (SRS)** <https://ascent.cysun.org/project/resource/get/1461?inline=true>