

Software Design Document

For

The Arqive

Prepared by: *Cesar Ayala, Songtao Bu, Alejandro Ceballos, Bryan Chan, Erik Donovan-Blood, Kennard Lim, Jorge Mata, Calvin Mateo, Misael Ortega, Jonathan Saldivar, Dustin Shin*

California State University, Los Angeles – The Arqive

Tables Of Content

Tables Of Content	2
Revision History	6
1. Introduction	7
1.1 Purpose	7
1.2 Document Conventions	7
1.3 Intended Audience and Reading Suggestions	7
1.4 System Overview	7
2. Design Considerations	8
2.1 Goals and Guidelines	8
2.2 Assumptions and Dependencies	8
2.3 General Constraints	9
2.4 Development Methods	9
3. Architectural Strategies	10
4. System Architecture	11
5. Policies and Tactics	13
5.1 Choice of which specific products used	13
5.2 Plans for ensuring requirements traceability	13
5.3 Plans for testing the software	13
6. Detailed System Design	14
6.1 User Account Requirements	14
6.1.1 Responsibilities	14
6.1.2 Constraints	14
6.1.3 Composition	14
6.1.4 Uses/Interactions	14
6.1.5 Resources	15
6.1.6 Interface/Exports	15
6.2 Gamification Requirements	15
6.2.1 Responsibilities	15
6.2.2 Constraints	15
6.2.3 Composition	15
6.2.4 Uses/Interactions	15
6.2.5 Resources	15
6.2.6 Interface/Exports	16
6.3 AR/VR Requirements	16
6.3.1 Responsibilities	16
6.3.2 Constraints	16

6.3.3 Composition	16
6.2.4 Uses/Interactions	16
6.3.5 Resources	16
6.3.6 Interface/Exports	16
6.4 Map Requirements	17
6.4.1 Responsibilities	17
6.4.2 Constraints	17
6.3.3 Composition	17
6.4.4 Uses/Interactions	17
6.4.5 Resources	17
6.4.6 Interface/Exports	17
6.5 Story Requirements	18
6.5.1 Responsibilities	18
6.5.2 Constraints	18
6.5.3 Composition	18
6.5.4 Uses/Interactions	18
6.5.5 Resources	18
6.5.6 Interface/Exports	18
6.6 Platform Requirements	19
6.6.1 Responsibilities	19
6.6.2 Constraints	19
6.6.3 Composition	19
6.6.4 Uses/Interactions	19
6.6.5 Resources	19
6.6.6 Interface/Exports	19
6.7 Multimedia Requirements	19
6.7.1 Responsibilities	19
6.7.2 Constraints	19
6.7.3 Composition	20
6.7.4 Uses/Interactions	20
6.7.5 Resources	20
6.7.6 Interface/Exports	20
6.8 Security Requirements	20
6.8.1 Responsibilities	20
6.8.2 Constraints	20
6.8.3 Composition	20
6.8.4 Uses/Interactions	21
6.8.5 Resources	21
6.8.6 Interface/Exports	21
6.9 User Role Requirements	21
6.9.1 Responsibilities	21

6.9.2 Constraints	21
6.9.3 Composition	21
6.9.4 Uses/Interactions	21
6.9.5 Resources	21
6.9.6 Interface/Exports	22
6.10 FAQ Requirements	22
6.10.1 Responsibilities	22
6.10.2 Constraints	22
6.10.3 Composition	22
6.10.4 Uses/Interactions	22
6.10.5 Resources	22
6.10.6 Interface/Exports	22
6.11 About Requirements	23
6.11.1 Responsibilities	23
6.11.2 Constraints	23
6.11.3 Composition	23
6.11.4 Uses/Interactions	23
6.11.5 Resources	23
6.11.6 Interface/Exports	23
6.12 Resources Requirements	24
6.12.1 Responsibilities	24
Provides users with access to various valuable resources. 6.12.2 Constraints	24
6.12.3 Composition	24
6.12.4 Uses/Interactions	24
6.12.5 Resources	24
6.12.6 Interface/Exports	24
6.13 Accessibility Requirements	24
6.13.1 Responsibilities	24
6.13.2 Constraints	24
6.13.3 Composition	25
6.13.4 Uses/Interactions	25
6.13.5 Resources	25
6.13.6 Interface/Exports	25
6.14 About us Requirements	25
6.14.1 Responsibilities	25
6.14.2 Constraints	25
6.14.3 Composition	25
6.14.4 Uses/Interactions	26
6.14.5 Resources	26
6.14.6 Interface/Exports	26

7. Detailed Lower level Component Design	27
8. Database Design	27
9. User Interface	29
9.1 Overview of User Interface	29
9.2 Screen Frameworks or Images	30
9.3 User Interface Flow Model	30
10. Requirements Validation and Verification	32
11. Glossary	32
12. References	34

Revision History

Name	Date	Reason For Changes	Version
First Rough Draft	11/11/2022	First document with template from previous versions	1.0
Second Draft			
Third Draft			

1. Introduction

1.1 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.

1.2 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

1.3 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.

1.4 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 preserves 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.

2. Design Considerations

2.1 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 archive's users
- 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
- 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 archive website, according to the guidelines provided.

2.2 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

2.3 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 users experience.

2.4 Development Methods

The agile development method was used in the design of this project. Members of the team handled bug fixes and deployed the changes after resolving them. To implement new features to The arqive, we take a waterfall development method by merging and refactoring the front-end and back-end code to 1 database. After multiple changes and implementations to the applications, the team then demonstrates the new changes to the sponsors to provide feedback and approval.

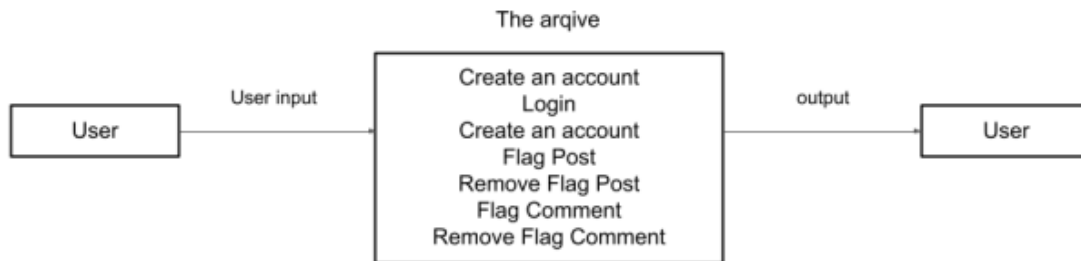
3. Architectural Strategies

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.

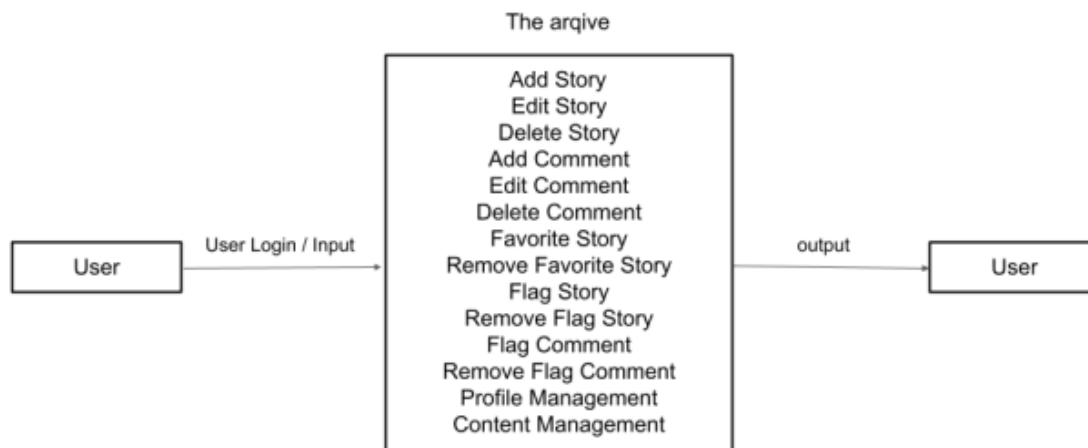
4. System Architecture

DFD Level 0

User Not Logged In



User Logged In



DFD LEVEL 1



5. Policies and Tactics

5.1 Choice of which specific products used

PostgreSQL, Digital Ocean, ReactJS, React Native, Django REST, OpenstreetMap

5.2 Plans for ensuring requirements traceability

Each developer will create test cases that will map to requirements on any effort they are working on.

5.3 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.

6. Detailed System Design

6.1 User Account Requirements

6.1.1 Responsibilities

Separate the users into regular users, anonymous users, moderators, and administrators with unique roles.

6.1.2 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.

6.1.3 Composition

No current subcomponents.

6.1.4 Uses/Interactions

The user's role allows them to create or modify 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.

6.1.5 Resources

Modification of the database of pins is dependent on the user's role.

6.1.6 Interface/Exports

No current exports

6.2 Gamification Requirements

6.2.1 Responsibilities

To encourage users to be active on the app.

6.2.2 Constraints

No constraints

6.2.3 Composition

No current subcomponents

6.2.4 Uses/Interactions

The gamification feature will encourage users in using the app by utilizing a reward system. Interacting with the app will provide the user account with points that can accumulate into rewards in the form of badges or tokens.

6.2.5 Resources

No current resources

6.2.6 Interface/Exports

No current exports.

6.3 AR/VR Requirements

6.3.1 Responsibilities

To encourage users to be interactive on the app.

6.3.2 Constraints

We must assume that users have a phone capable of using AR features and know to look around with a camera to view posts.

6.3.3 Composition

No current subcomponents

6.3.4 Uses/Interactions

The user's experience with AR will encourage users to interact with stories by walking around the places where they were posted. Especially for resource and historical posts.

6.3.5 Resources

No current resources

6.3.6 Interface/Exports

No current exports.

6.4 Map Requirements

6.4.1 Responsibilities

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

6.4.2 Constraints

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

6.3.3 Composition

No current subcomponents.

6.4.4 Uses/Interactions

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

6.4.5 Resources

Digital Ocean PostgreSQL Development Database
OpenStreetMap

6.4.6 Interface/Exports

No current exports.

6.5 Story Requirements

6.5.1 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.

6.5.2 Constraints

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

6.5.3 Composition

Favoriting a post

6.5.4 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.

6.5.5 Resources

Digital Ocean PostgreSQL Development Database.

6.5.6 Interface/Exports

No current exports

6.6 Platform Requirements

6.6.1 Responsibilities

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

6.6.2 Constraints

Not all available browsers will be covered.

6.6.3 Composition

No current composition.

6.6.4 Uses/Interactions

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

6.6.5 Resources

No specific resources.

6.6.6 Interface/Exports

No current exports.

6.7 Multimedia Requirements

6.7.1 Responsibilities

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

6.7.2 Constraints

The file size must be reasonable.

6.7.3 Composition

No current composition.

6.7.4 Uses/Interactions

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

6.7.5 Resources

No specific resources.

6.7.6 Interface/Exports

It is part of the pin creation process.

6.8 Security Requirements

6.8.1 Responsibilities

Protect the site, its data and users.

6.8.2 Constraints

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

6.8.3 Composition

No current composition.

6.8.4 Uses/Interactions

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

6.8.5 Resources

No specific resources.

6.8.6 Interface/Exports

No current exports.

6.9 User Role Requirements

6.9.1 Responsibilities

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

6.9.2 Constraints

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

6.9.3 Composition

No current composition.

6.9.4 Uses/Interactions

Allows users to have specific privileges within the application.

6.9.5 Resources

Database

6.9.6 Interface/Exports

No current exports

6.10 FAQ Requirements

6.10.1 Responsibilities

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

6.10.2 Constraints

Limited to what administrators post.

6.10.3 Composition

No current composition.

6.10.4 Uses/Interactions

Allows Administrators to post FAQs and users to read them.

6.10.5 Resources

Database

6.10.6 Interface/Exports

No current exports.

6.11 About Requirements

6.11.1 Responsibilities

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

6.11.2 Constraints

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

6.11.3 Composition

No current composition.

6.11.4 Uses/Interactions

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

6.11.5 Resources

Database

6.11.6 Interface/Exports

No current exports.

6.12 Resources Requirements

6.12.1 Responsibilities

Provides users with access to various valuable resources.

6.12.2 Constraints

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

6.12.3 Composition

No current composition.

6.12.4 Uses/Interactions

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

6.12.5 Resources

No current resources.

6.12.6 Interface/Exports

No current exports.

6.13 Accessibility Requirements

6.13.1 Responsibilities

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

6.13.2 Constraints

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

6.13.3 Composition

No current composition.

6.13.4 Uses/Interactions

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

6.13.5 Resources

UserWay Widget

6.13.6 Interface/Exports

No current exports.

6.14 About us Requirements

6.14.1 Responsibilities

Allows users to contact us directly through email.

6.14.2 Constraints

No current constraints.

6.14.3 Composition

No current composition.

6.14.4 Uses/Interactions

The user will input a message, and if they want a response, they could provide an email (optional).

6.14.5 Resources

SMTP, Django Documentation

6.14.6 Interface/Exports

A simple and secure way to contact us

7. Detailed Lower level Component Design

Refer to Section 4 of this Document.

8. Database Design



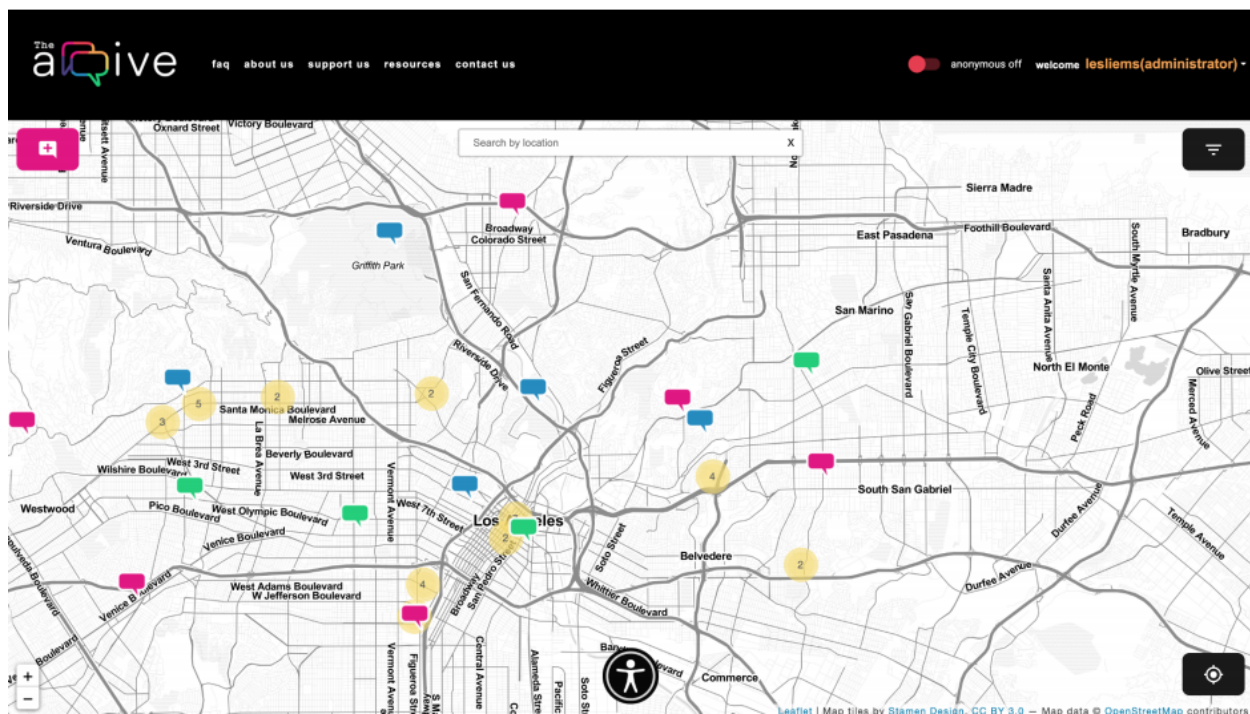
9. User Interface

9.1 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 #thearque on other social media.

9.2 Screen Frameworks or Images



add a story

address

locality (city, township, etc.)

region (state, province, etc.)

country

postcode

title
*please enter a story title

category

description
*please enter a story description

File Edit View Insert Format

"The Princess" and "The Duchess"

griffith park los angeles ca

Posted by: Anonymous

03/17/1968 - no end date

Two drag queens known as "The Princess" and "The Duchess" held a St. Patrick's Day party at Griffith Park, a popular cruising spot and a frequent target of police activity. More than 200 gay men socialized through the day.

Originally a place to protest police entrapment, this event led to a "gay-in" which took place on May 30, 1968.

9.3 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

10. Requirements Validation and Verification

Not applicable at this time.

11. 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

12. References

Brad Appleton

<http://www.bradapp.net>

https://www.cs.purdue.edu/homes/cs307/ExampleDocs/DesignTemplate_Fall08.doc

The archive's 2022 Development Team (SRS)

<https://csns.cysun.org/department/cs/project/resource/view?projectId=7913655&resourceId=7928311>