Software Design Document

for

the arqive

Version 3.0 approved

Prepared by Randy Arruda Jr., Balarama Carter, Richard Cruz-Silva, Abram Flores, Carlos Larios-Solis, Khang Le, Brandon Lee, Casandra Pahed, Evelyn Ramirez

California State University, Los Angeles - the arqive

December 11, 2020

Revision History

Name	Date	Reason For Changes	Version
First rough draft	11/18/19	First Document for Review and Approval	1
Second Draft	11/28/19	Second draft for approval	2
Third Draft	3/21/20	Third Draft for Approval	3
Fourth Draft	12/11/20	Submission for Review	4

1. Introduction

1.1 Purpose

This document outlines the Software Design Specifications as part of the design plan and specifications for the functionality of the arqive. The document will expand on the functionality described by the features in the Software Requirements Specifications (SRS). Each feature discussed will describe the existing functionality of the arqive, as well as additional features to be implemented. The scope of this document takes the features as outlined in the SRS and expands each of them to include the design issues. The features are described by the names given in the descriptions and diagrams in the SRS.

1.2 Document Conventions

The SDD uses the following conventions:

SDD	Software Design Document
SRS	Software Requirement Specification
User	The individual using the the arqive website

1.3 Intended Audience and Reading Suggestions

The main users of this SDD are the software developers. Project managers and Webmasters running the website need to be able to know the website's capabilities. Testers will need to know what they are testing against. Documentation writers will need to see if the document is written correctly.

1.4 System Overview

The web app is intended to connect the LGBTQ+ community through sharing safe places they have personally visited. The relationship between the users and the web app starts by letting the user make an account to use the arqive properly, although people can post anonymously. The main page’s map will be filled with vetted user content that will allow others to see what happened during the user’s time there.

2. Design Considerations

2.1 Assumptions and Dependencies

- OpenStreetMap
- OpenLayers Javascript Library
- UserWay Widget (free)
- TinyMCE
- Recaptcha API
- User must have basic knowledge of the internet
- Computer and/or mobile device with a web browser
- iOS and/or Android device (if using the mobile application)

2.2 General Constraints

- Slow internet connections may negatively impact user experience
- Privacy for the users may not always be guaranteed
- Javascript must be enabled in the user's web browser

2.3 Goals and Guidelines

- The arqive's main goal is to provide members of the LGBTQ+ community with a platform where they can share their stories, experiences, and other meaningful resources available to the LGBTQ+ population
- Security and protection from malevolent agents that might cause harm to the arqive's users
- Integrating other social media to enhance user engagement
- Automated content moderation
- Maintaining the current arqive website
- The mobile apps should implement additional functionality and be created with the established style of the arqive website, according to the guidelines provided by a team of graphic designers

2.4 Development Methods

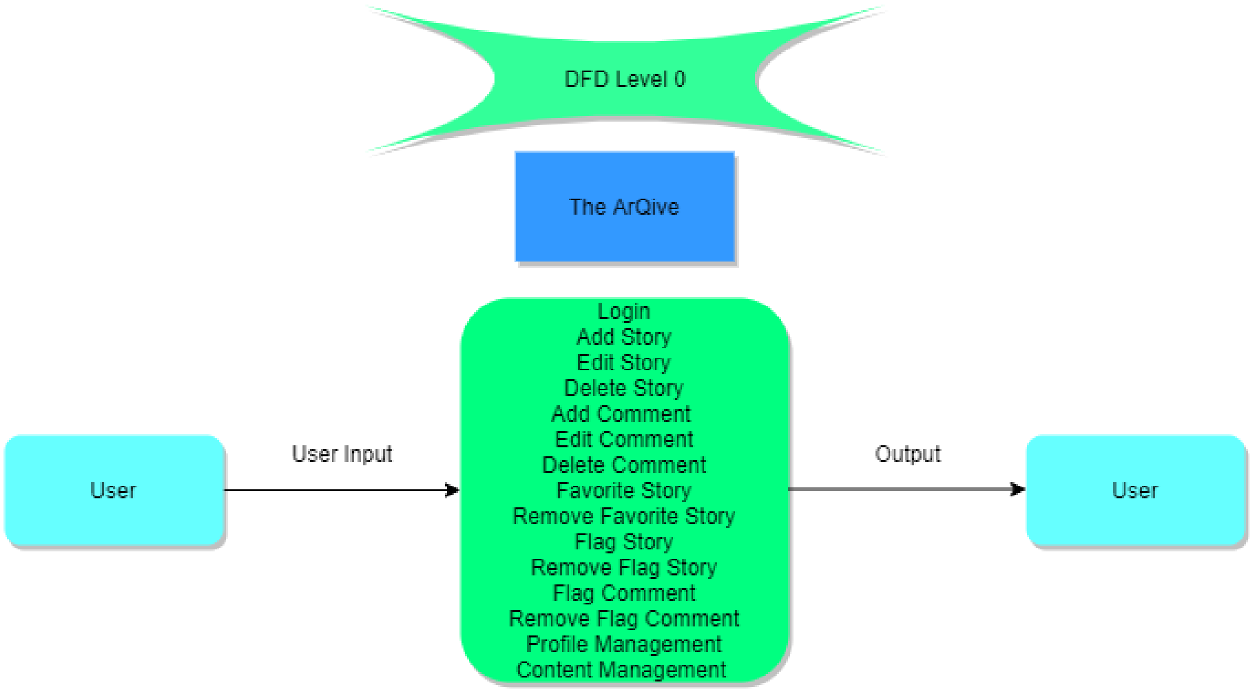
The method used to design this software is similar to that of the Agile Development method. Throughout the development process, members of the team developed small portions of each required functionality until the entire requirement was successfully fulfilled. We then merged the new code with the current code base and ensured that nothing broke. Finally, we would demonstrate the new functionality to the project sponsors where they would provide feedback and approval.

3. Architectural Strategies

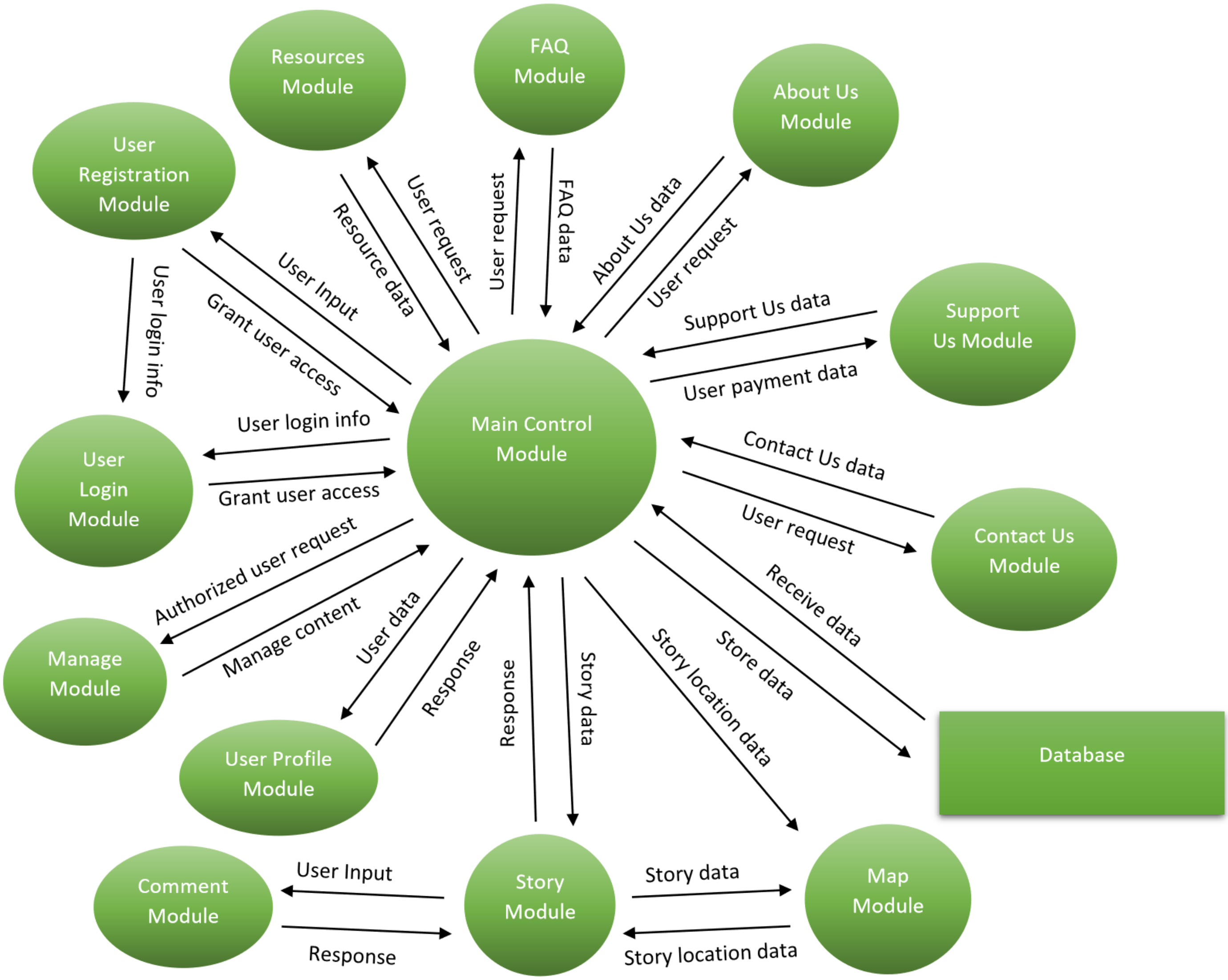
Using Python and Javascript along with Django and React, we were able to make our website flexible and responsive. Since Django and React are established technologies, there is a large library of resources that we were able to pull from to enhance functionality with minimal developer time. Since our largest effort was the development of the mobile apps, we used React Native to cross compile them for both iOS and Android. By using a framework such as React Native, we were able to minimize code management efforts by only having one codebase for the mobile apps. Since React Native is similar to React, we are able to mirror functionality across the mobile apps and the website app with minimal effort.

4. System Architecture

Please refer to section 3 of the SRS



DFD LEVEL 1


Resources Module
FAQ Module
About Us Module
User Registration Module
Support Us Module
User Login Module
Main Control Module
Contact Us Module
Manage Module
User Profile Module
Database
Comment Module
Story Module
Map Module
User request
Resource data
User request
FAQ data
About Us data
User request
User Input
Grant user access
User login info
Support Us data
User payment data
User login info
Grant user access
Contact Us data
User request
Receive data
Store data
Authorized user request
Manage content
User data
Response
Response
Story data
Story location data
User Input
Response
Story data
Story location data

5. Policies and Tactics

5.1 Choice of which specific products used

PostgreSQL, Github, Digital Ocean

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

While building the website we are testing along the way. Each developer will create their own test cases while a second developer will ensure requirement verification.

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.

6.1.3 Compositions

X

6.1.4 Uses/Interactions

The user’s role allows them to create or modify posts based on their role.

6.1.5 Resources

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

6.1.6 Interface/Exports

6.2 Gamification Requirements

6.2.1 N/A

6.3 Map Requirements

6.3.1 Responsibilities

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

6.3.2 Constraints

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

6.3.3 Compositions

X

6.3.4 Uses/Interactions

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

6.3.5 Resources

Digital Ocean PostgreSQL Development Database

6.3.6 Interface/Exports

6.4 Story Requirements

6.4.1 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.

6.4.2 Constraints

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

6.4.3 Compositions

Favoriting a post.

6.4.4 Uses/Interactions

Users will able to view the story page

6.4.5 Resources

Digital Ocean PostgreSQL Development Database

6.4.6 Interface/Exports

6.5 Platform Requirements

6.5.1 Responsibilities

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

6.5.2 Constraints

There are some browsers that will be overlooked

6.5.3 Compositions

6.5.4 Uses/Interactions

6.5.5 Resources

6.5.6 Interface/Exports

6.6 Multimedia Requirements

6.6.1 Responsibilities

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

6.6.2 Constraints

The file size must be reasonable.

6.6.3 Compositions

X

6.6.4 Uses/Interactions

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

6.6.5 Resources

X

6.6.6 Interface/Exports

It may be a part of the pin creation process.

6.7 Security Requirements

6.7.1 Responsibilities

The security will protect the site and its data.

6.7.2 Constraints

Attacks are constantly evolving, so our security must too.

6.7.3 Compositions

X

6.7.4 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.

6.7.5 Resources

X

6.7.6 Interface/Exports

X

6.8 User Role Requirements

6.8.1 Responsibilities

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

6.8.2 Constraints

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

6.8.3 Compositions

6.8.4 Uses/Interactions

Allows users to have specific privileges within the application.

6.8.5 Resources

Database

6.8.6 Interface/Exports

6.9 FAQ Requirements

6.9.1 Responsibilities

Provide users with information regarding the site.

6.9.2 Constraints

Limited to what Administrators post.

6.9.3 Compositions

6.9.4 Uses/Interactions

Allows Administrators to post FAQs and Users to read them.

6.9.5 Resources

Database

6.9.6 Interface/Exports

6.10 About Requirements

6.10.1 Responsibilities

Provide users with information on the mission of the arqive.

6.10.2 Constraints

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

6.10.3 Compositions

6.10.4 Uses/Interactions

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

6.10.5 Resources

Database

6.10.6 Interface/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 Compositions

X

6.12.4 Uses/Interactions

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

6.12.5 Resources

X

6.12.6 Interface/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 functionalitites.

6.13.3 Compositions

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

6.14 Contact Us Requirements

6.14.1 Responsibilities

Allows users to contact us directly through email.

6.14.2 Constraints

X

6.14.3 Compositions

X

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

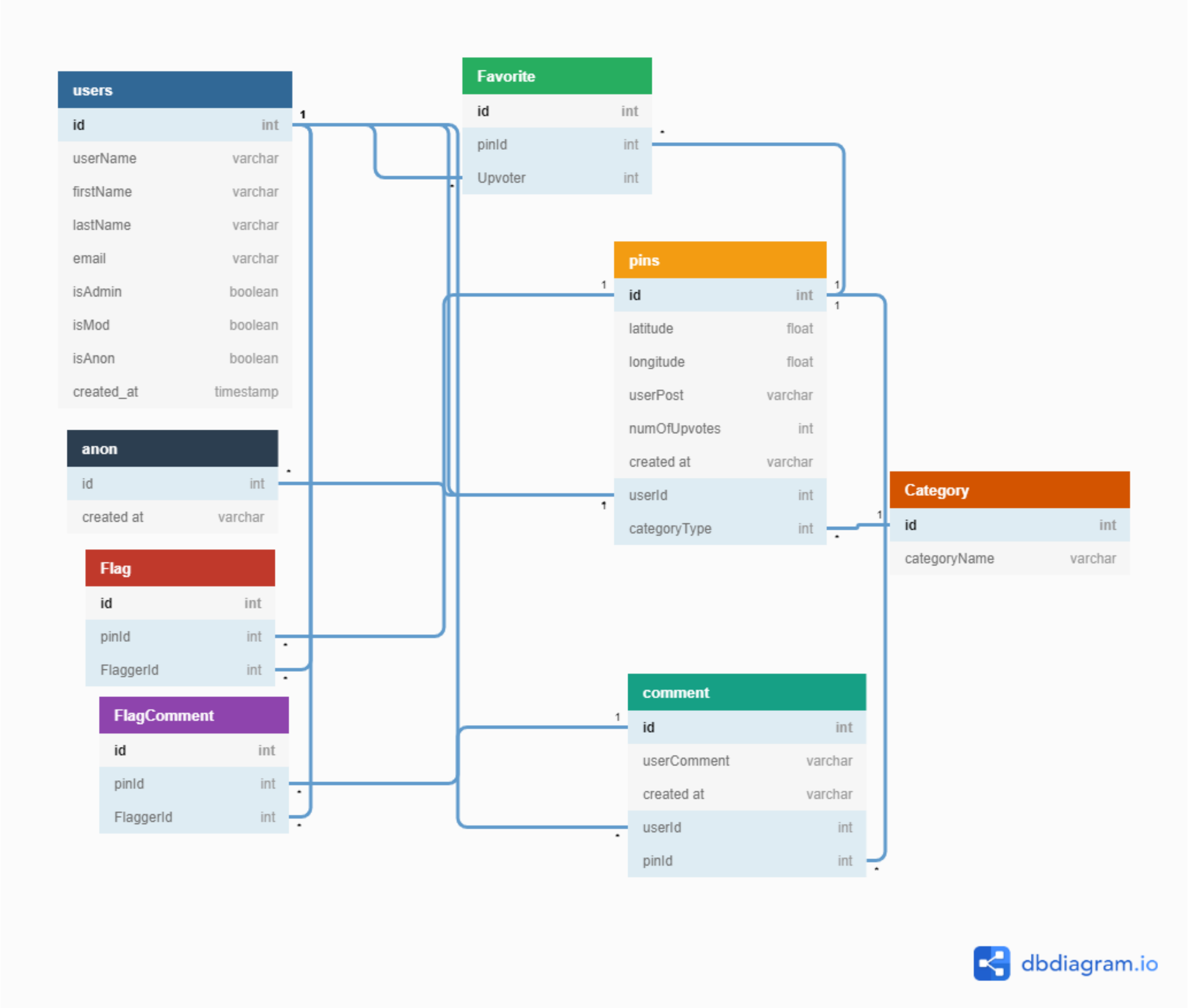
6.14.6 Interface/Exports

A simple and secure way to contact us

7. Detailed Lower-level Component Design

Refer to Section 4 of this Document.

8. Database Design



users	
id	int
userName	varchar
firstName	varchar
lastName	varchar
email	varchar
isAdmin	boolean
isMod	boolean
isAnon	boolean
created_at	timestamp

Favorite	
id	int
pinId	int
Upvoter	int

pins	
id	int
latitude	float
longitude	float
userPost	varchar
numOfUpvotes	int
created at	varchar
userId	int
categoryType	int

anon	
id	int
created at	varchar

Category	
id	int
categoryName	varchar

Flag	
id	int
pinId	int
FlaggerId	int

FlagComment	
id	int
pinId	int
FlaggerId	int

comment	
id	int
userComment	varchar
created at	varchar
userId	int
pinId	int

9. User Interface

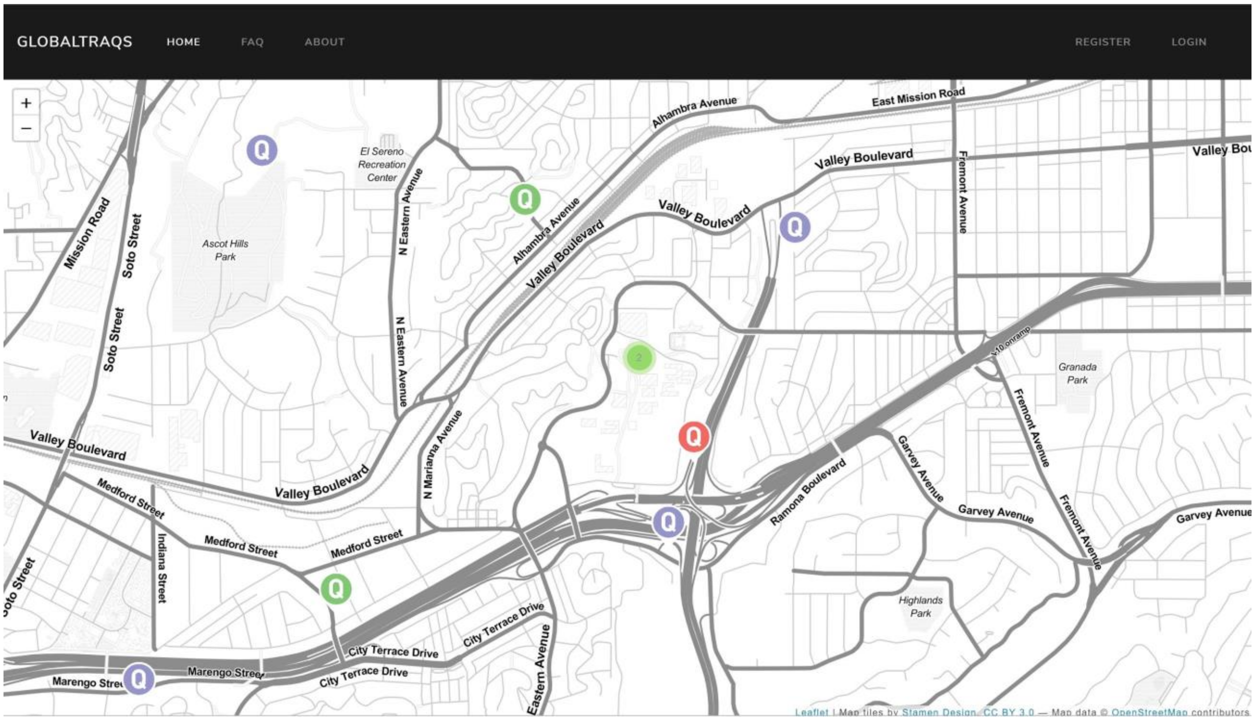
9.1 Overview of User Interface

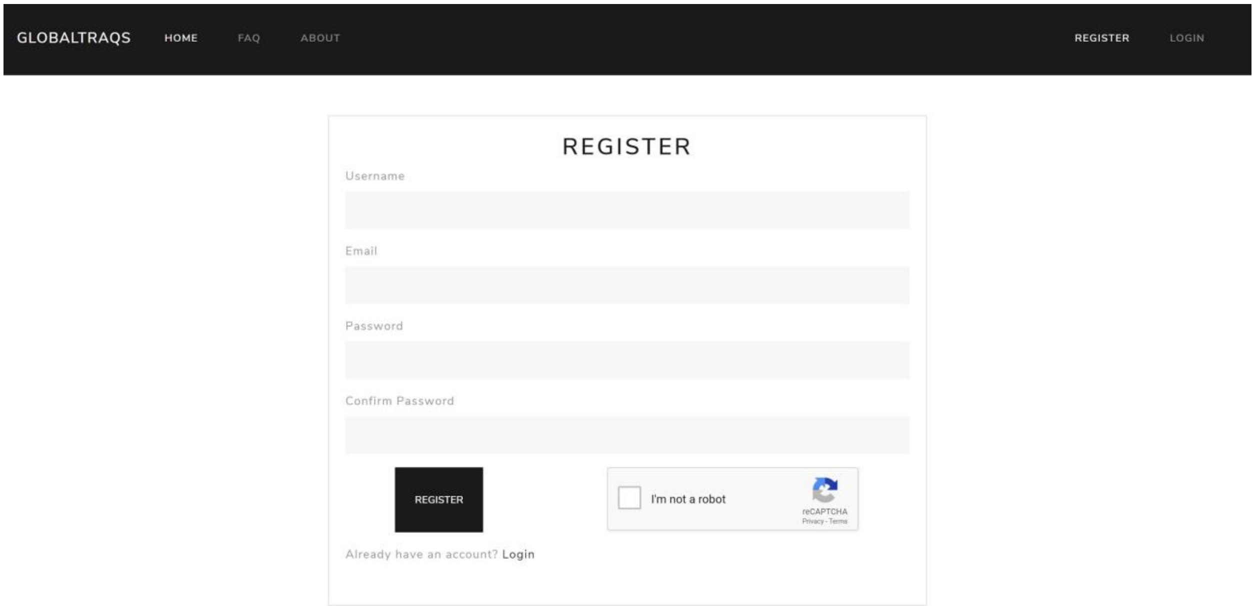
The user will access the main content of the site by navigating to the homepage (https://thearqive.com). Within this page, users can post pins on the map by category (personal, historical, or community). Users can also access other stories by navigating around the map and clicking on pins, which will 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 the arqive email. Lastly, within the Support Us page, users can support the website by sharing the #thearqive on other social media.

9.2 Screen Frameworks or Images





REGISTER

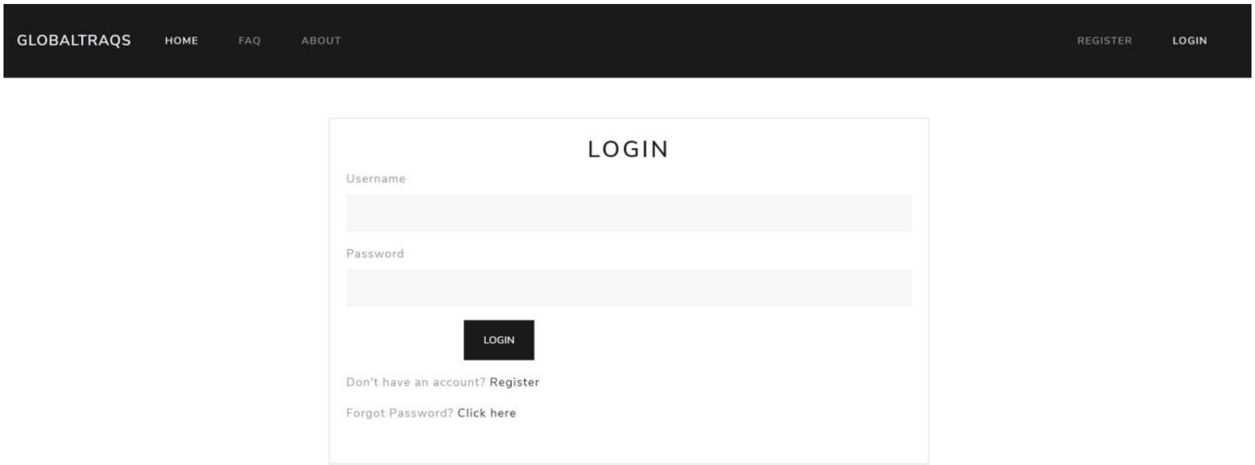
Username

Email

Password

Confirm Password

Already have an account? Login



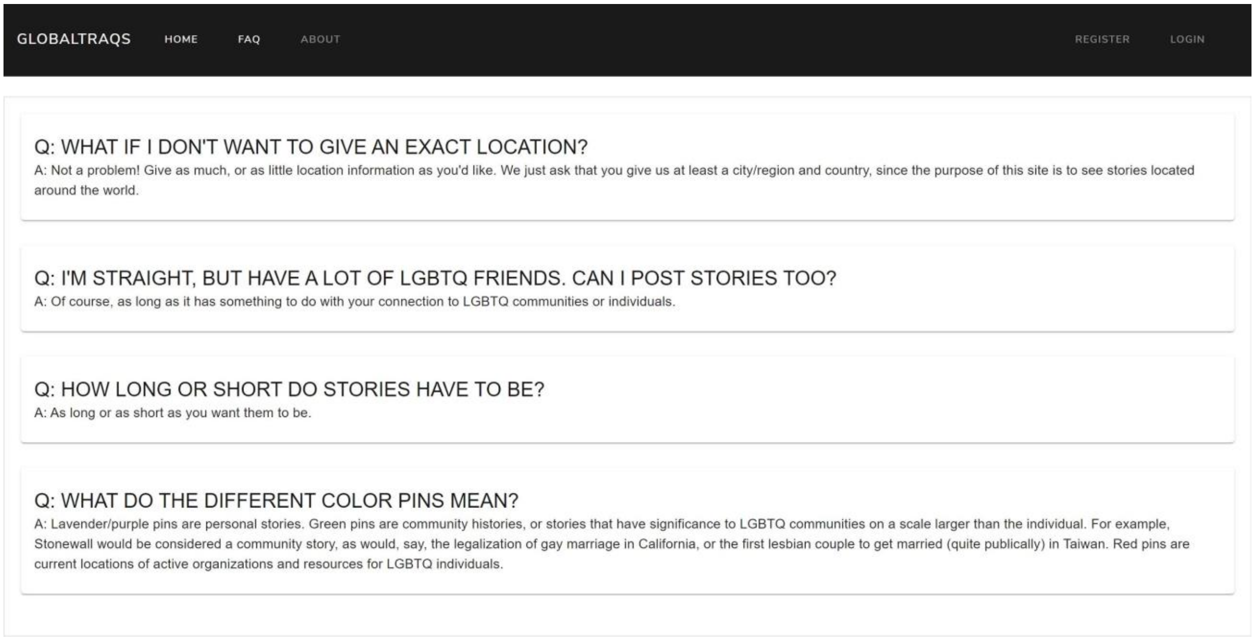
LOGIN

Username

Password

Don't have an account? Register

Forgot Password? Click here



Q: WHAT IF I DON'T WANT TO GIVE AN EXACT LOCATION?
A: Not a problem! Give as much, or as little location information as you'd like. We just ask that you give us at least a city/region and country, since the purpose of this site is to see stories located around the world.

Q: I'M STRAIGHT, BUT HAVE A LOT OF LGBTQ FRIENDS. CAN I POST STORIES TOO?
A: Of course, as long as it has something to do with your connection to LGBTQ communities or individuals.

Q: HOW LONG OR SHORT DO STORIES HAVE TO BE?
A: As long or as short as you want them to be.

Q: WHAT DO THE DIFFERENT COLOR PINS MEAN?
A: Lavender/purple pins are personal stories. Green pins are community histories, or stories that have significance to LGBTQ communities on a scale larger than the individual. For example, Stonewall would be considered a community story, as would, say, the legalization of gay marriage in California, or the first lesbian couple to get married (quite publically) in Taiwan. Red pins are current locations of active organizations and resources for LGBTQ individuals.

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

Refer to SRS 1.4

12. References

Brad Appleton <brad@bradapp.net> http://www.bradapp.net

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