WELCOME TO CHENEY APP

Senior Project and Capstone Report

Submitted by:

The A Team

NOLAN POSEY (nposey1@ewu.edu)

TIMOTHY NELSON (tnelson32@ewu.edu)

MATTHEW MATRICIANO (mmatriciano@ewu.edu)

TANNER STEPHENSON (tstephenson2@ewu.edu)

DANIEL PALMER (dpalmer10@ewu.edu)

Sanmeet Kaur

Sponsored by Maria Fell and Welcome to Cheney

Welcome to Cheney

Eastern Washington University
March 2024

ABSTRACT

Welcome to Cheney is a non-profit organization committed to fostering communication, connection, and action within the city of Cheney. Their primary purpose is to provide timely and accurate information to the residents of Cheney. Welcome to Cheney has tried utilizing other forms of social media such as Facebook and Instagram to share information, but their presence is being overshadowed amidst the noise on those platforms. Therefore, the intention of this project is to develop a mobile app with the sole purpose of being a reliable means of sharing important information with the residents of Cheney.

The information being shared on this mobile app will include but is not limited to: road closures, weather alerts, public events, school activities, jobs and volunteer opportunities. To prioritize accuracy of information, the app will feature tiered account levels, allowing reliable information directly from a verified source. Since only certain accounts can make posts, accounts without the ability to post will still be able to directly share information with admins. Using this strategy, admins will be able to curate information that is being shared in order to give the most accurate and credible information to the residents of Cheney.

This paper will be used to document our progress and methodologies in developing the Welcome To Cheney mobile app, addressing the organization's need for a centralized and reliable information sharing platform.

ACKNOWLEDGEMENT

We extend our deepest gratitude to Maria Fell, our esteemed sponsor, for entrusting us with this pivotal project. Her support and faith in our abilities have been the cornerstone of our progress and achievements. Maria's vision and dedication to the betterment of the Cheney community have inspired every phase of this project, motivating us to exceed expectations.

Our heartfelt thanks also go to each member of our project team. Their tireless work, innovative ideas, and collaborative spirit have been instrumental in bringing the Welcome to Cheney mobile app from concept to reality. This project has been a testament to their commitment, expertise, and relentless pursuit of excellence. Their ability to overcome challenges and adapt to changing demands has been nothing short of remarkable.

Special recognition is due to our teacher and mentor, Sanmeet Kaur. Her guidance has been invaluable, not just in the technical aspects of our project, but in instilling a sense of purpose and determination within us. Sanmeet's belief in our potential and her constant encouragement have pushed us to explore new heights and have laid the foundation for our success.

To all who have contributed, whether directly or indirectly, to this project: your input, feedback, and encouragement have been crucial. We are immensely thankful for the collective efforts that have made this project possible. It is our hope that the Welcome to Cheney mobile app will serve as a lasting contribution to the community, reflecting the dedication and spirit of everyone involved.

LIST OF TABLES

None Applicable

LIST OF FIGURES

LIST OF ABBREVIATIONS(GLOSSARY)

WTC: Welcome to Cheney

UI/UX (User Interface/User Experience) Refers to the design and usability of the app, emphasizing ease of use, accessibility, and engagement for all users.

QA (Quality Assurance): The process of testing the mobile app to ensure its quality, functionality, and performance meet the project requirements and user expectations.

Firebase: A platform by Google used for the backend development of the app, providing database, authentication, and hosting services.

Flutter: The open-source UI software development kit used for creating the app's frontend, allowing for the development of natively compiled applications for mobile, web, and desktop from a single codebase.

MySQL: A database management system utilized in the app's development for storing and retrieving data.

SFF (Specific Fillable Form): A feature within the app allowing viewers to submit information to admins for potential sharing within the app, enhancing community-driven content curation.

Gantt Chart: A project management tool used to illustrate the project schedule, showing the start and finish dates of the elemental tasks**.**

WBS (Work Breakdown Structure): A method of breaking down the project into smaller, more manageable sections, detailing the tasks and responsibilities required to complete the project.

Agile Development: A project management methodology adopted for the app development, characterized by iterative development, flexible responses to changes, and incremental delivery of the product.

PR (Pull Request): A feature of Github where you are able to post a request to the project where you want to merge your work from one branch into another, and it shows a difference view for review

TABLE OF CONTENTS

REFERENCES

[1] In Figure 29, “tags” in our system will be how we handle categorization and interests. Tags can be anything, from business, to age groups, to things like “sports” or “board gaming” and so on, and users will be able to specify all the tags that interest or apply to them, and that will create a personalized feed, and/or an option for sorting/filtering posts. Events, posts, and users will all have these tags, and while we considered an Enum at first, due to the open-ended nature of what a tag can be, we left it as a string. We will have a comprehensive bank of tags, but as new businesses come and go, they will automatically add/remove tags for them, and that’s how business notifications will work.

CHAPTER ONE

INTRODUCTION

1.1 - PROJECT OVERVIEW

"Welcome to Cheney" is a non-profit organization with a vital mission: to enhance community engagement and communication within the city of Cheney. Recognizing the challenges of disseminating timely and accurate information to residents through conventional social media channels, which often result in vital updates being lost amidst the digital noise, the organization has embarked on an ambitious project. The development of a dedicated mobile application aims to establish a direct, reliable communication channel to keep the Cheney community informed and connected.

To ensure the accuracy and reliability of the information distributed, the app will introduce tiered account levels. This feature is designed to enable direct information flow from verified sources, including local government, educational institutions, and recognized community organizations. While accounts with posting privileges will be limited to these verified entities, all users can interact with the content and, crucially, submit information to the app administrators. This approach ensures a participatory platform while maintaining the integrity and credibility of the information shared.

This document will serve as a living record of the project's development journey. It will detail the methodologies employed, challenges encountered, solutions implemented, and milestones achieved. Regular updates will ensure stakeholders are kept informed of the project's progress and any adjustments to the project scope or timelines.

1.2 - PROBLEM DEFINITION AND SCOPE

The city of Cheney faces a significant challenge in effective communication and information dissemination within the community. Traditional social media platforms, such as Facebook and Instagram, are currently used for this purpose. However, these channels are increasingly becoming oversaturated with content, causing essential local updates and information to be lost in the noise. This lack of a dedicated, reliable platform for community information sharing leads to misinformation, decreased community engagement, and missed opportunities for residents to participate in local activities and events.

Scope of the Problem

The problem extends across various dimensions of community life in Cheney, impacting areas such as:

- Public Safety: Residents miss out on critical updates about road closures, weather alerts, and emergency notifications.
- Community Engagement: Information about public events, community gatherings, and school activities fails to reach a wide audience, leading to lower participation rates.
- Economic Opportunities: Job postings and volunteer opportunities are not efficiently communicated, hindering economic development and community service efforts.
- Information Reliability: The current reliance on multiple, unverified sources for community news fosters an environment where misinformation can easily spread.

1.3 - ASSUMPTIONS AND CONSTRAINTS

Technical Assumptions

- Device Compatibility: The app will be developed to be compatible with the majority of smartphones and tablets, assuming users have access to devices capable of running modern mobile applications.
- Internet Accessibility: It is assumed that the target user base has regular access to the internet, enabling them to receive updates and notifications from the app in real-time.
- Digital Literacy: Users are presumed to have a basic level of digital literacy, allowing them to navigate and utilize the app's features effectively.

Community Engagement Assumptions

- User Adoption: A significant portion of the Cheney population will download and regularly use the app, making it a viable platform for community information dissemination.
- Content Generation: Local organizations, schools, government entities, and businesses will actively participate by providing timely and accurate information for app content.

Technical Constraints

- Resource Limitations: Limited budget and human resources may restrict the complexity of the app's features and the speed of development.
- Device Variability: The wide range of user devices and operating systems may affect the app's performance and appearance across different platforms.

Time Constraints

- Development Timeline: The project timeline may be constrained by funding, staffing levels, and other priorities within the organization, potentially impacting the launch date.

User Engagement Constraints

- Market Saturation: The proliferation of apps and digital tools competing for users' attention may impact the adoption and regular use of the Welcome to Cheney app.
- Digital Divide: Variations in internet access and digital literacy levels within the community may limit the app's reach and effectiveness.

Financial Constraints

- Ongoing Funding: Securing continuous funding for app updates, maintenance, and promotion poses a challenge, affecting long-term sustainability.

1.4 - OBJECTIVES

- **Enhance Community Engagement**: To create a centralized platform that significantly improves the way residents of Cheney engage with their community, fostering a sense of belonging and active participation in local events, activities, and initiatives.
- **Improve Information Dissemination**: To provide a reliable, efficient, and direct channel for disseminating timely and accurate information to Cheney residents, overcoming the limitations of existing social media platforms where local updates can be lost in the noise.
- **Support Local Economy and Opportunities**: To promote economic growth and community service by offering a dedicated space for local businesses, non-profits, and organizations to post job openings, volunteer opportunities, and community events, thereby facilitating local engagement and support.
- **Enhance Public Safety and Awareness**: To ensure the safety and well-being of the Cheney community by offering real-time alerts and updates on weather emergencies, road closures, and other critical public safety information.
- **Facilitate Direct Communication**: To establish a two-way communication channel between the city's residents and local authorities, organizations, and event organizers, enabling feedback, suggestions, and direct interaction with the community.
- **Ensure Accessibility and Inclusivity**: To design the app with a focus on accessibility, ensuring that all members of the community, regardless of age, ability, or technological proficiency, can access and benefit from the platform.
- **Promote Verified and Trustworthy Information**: To prioritize the accuracy and credibility of information shared through the app by implementing a system of tiered account levels for verified sources and a curation process for content.
- **Support Sustainability and Scalability**: To build the mobile app with future growth in mind, ensuring that it can be scaled and updated to meet evolving community needs and incorporate new technologies or features.

1.5 - METHODOLOGY USED

Phase 1: Project Initiation and Planning

- Stakeholder Engagement: Identify and engage with key stakeholders including the Welcome to Cheney Board and potential app users to gather initial requirements and expectations.
- Objective Setting: Clearly define project objectives based on stakeholder input and community needs.
- Resource Allocation: Allocate necessary resources, including budget, technology, and human resources, ensuring alignment with project scope and objectives.

Phase 2: Requirements Gathering

- Competitive Analysis: Analyze existing apps and platforms to identify best practices and potential gaps in serving community needs.
- Requirements Documentation: Compile and document detailed app requirements, including functional, non-functional, and technical specifications.

Phase 3: Design and Prototyping

- UI/UX Design: Design user interfaces and experiences that cater to all community members, emphasizing ease of use, accessibility, and engagement.
- Prototyping: Develop prototypes to visualize app functionality and layout, facilitating stakeholder and user feedback.
- Feedback Iteration: Iterate on design and prototypes based on feedback to refine and finalize the app design.

Phase 4: Development and Testing

- Agile Development: Adopt an agile development methodology, allowing for iterative development cycles with regular sprints to progressively build and refine the app.
- Quality Assurance: Implement comprehensive testing, including unit testing, integration testing, and user acceptance testing (UAT), to ensure app quality and functionality.
- Security and Compliance: Ensure the app meets all relevant security standards and privacy regulations to protect user data and ensure trustworthiness.

Phase 5: Deployment and Launch

- Beta Testing: Release a beta version of the app to a select group of users for real-world testing and feedback.

- Launch Preparation: Prepare marketing materials, app store listings, and support documentation in anticipation of the app launch.
- Official Launch: Launch the app to the public on the Apple app store and the Google Play store.

This methodology combines strategic planning, community engagement, agile development, and continuous improvement to ensure the Welcome to Cheney mobile app effectively meets the needs of the Cheney community while remaining adaptable to future changes.

1.6 - PROJECT OUTCOMES AND DELIVERABLES

Project Outcomes

- **Centralized Communication Platform**: The creation of a single, user-friendly mobile app that serves as the definitive source for local news, events, and emergency alerts for Cheney residents, effectively bridging the information gap in the community.
- **Enhanced Community Engagement**: An increase in community participation in local events and activities as a direct result of improved communication and accessibility of information.
- **Improved Public Safety**: An improvement in public safety and preparedness among Cheney residents through timely and reliable emergency alerts and safety information.
- **High User Adoption and Satisfaction:** A high rate of app adoption among Cheney residents, accompanied by positive feedback and high satisfaction rates regarding the app's usability, content quality, and overall impact on community life.

Project Deliverables

- **Mobile Application**: A fully functional mobile application available for both iOS and Android platforms, featuring:
    - User registration and profile management
    - Tiered account system for content creators and general users
    - Real-time alerts and notifications system
    - Listing of local events, job postings, and volunteer opportunities
    - Interactive features for community feedback and suggestions

- **Post-Launch Report**: An initial post-launch report detailing stakeholder feedback summary and recommendations for future improvements.

These outcomes and deliverables are designed to ensure the Welcome to Cheney mobile app effectively meets its objectives, providing a valuable resource for the Cheney community and setting a foundation for future enhancements and community-driven initiatives.

1.7 - NOVELTY OF WORK

Hyper-local Focus

Unlike generic social media platforms or broad communication apps, the Welcome to Cheney app is uniquely focused on the Cheney community. This hyper-local approach ensures that all content is relevant, timely, and directly beneficial to the residents, addressing the specific challenges and opportunities within Cheney.

Tiered Information System

The app introduces a novel tiered account system that differentiates between verified content creators (such as local organizations, schools, and businesses) and general users. This system prioritizes the accuracy and reliability of the information shared, directly addressing the issue of misinformation that plagues broader social media platforms.

Integrated Public Safety Features

Incorporating real-time alerts and emergency notifications, the app serves as a critical tool for enhancing public safety. Its ability to deliver urgent messages directly to residents' mobile devices represents a significant advancement in the community's emergency preparedness and response capabilities.

Community-Driven Content Curation

The app's content curation strategy is designed to be participatory, allowing residents to contribute to the information shared while maintaining high standards of accuracy and relevance through administrative oversight. This balance of open contribution and curated oversight is a unique feature that fosters a sense of ownership and trust among users.

Inclusive Design and Accessibility

Acknowledging the diversity within the Cheney community, the app is developed with a strong emphasis on inclusivity and accessibility. Features and design elements are carefully considered to ensure that the app is user-friendly for people of all ages, abilities, and tech-savviness, making it a truly community-wide resource.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 - USER STORIES & DIAGRAMS

(U1) As an admin I want to be able to create posts so that I can share information with the viewers on the app.

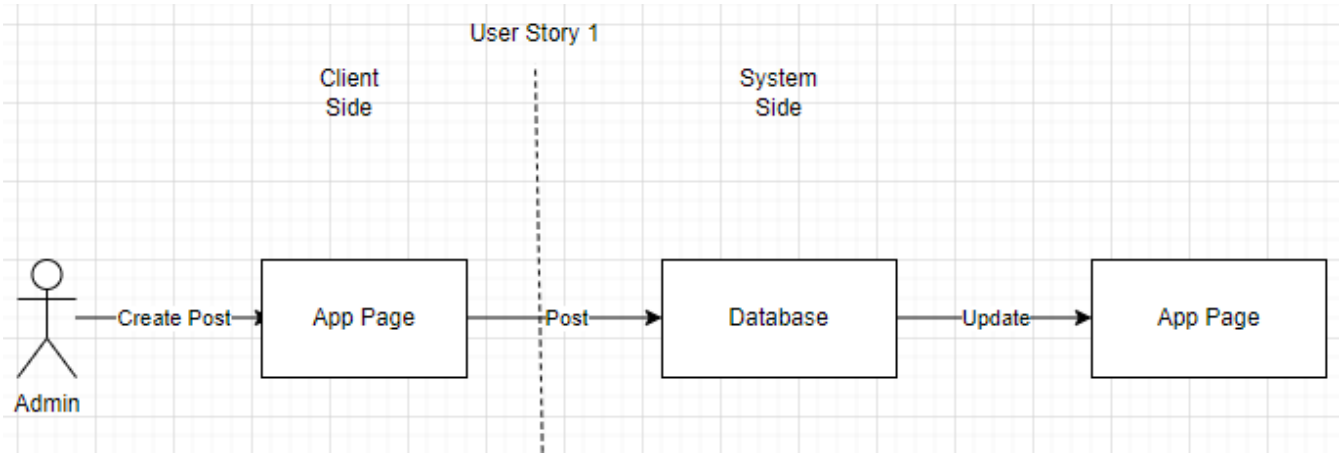


Figure 1: User Story 1 Diagram

(U2)As an admin I want to be able to edit posts I create so that information can be updated and/or corrected.

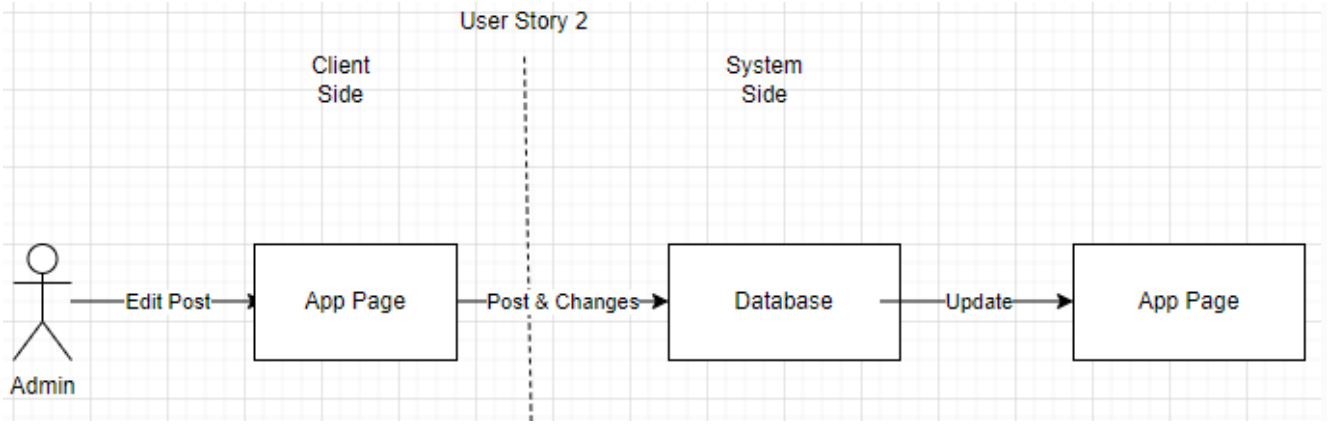


Figure 2: User Story 2 Diagram

(U3) As an admin I want to be able to delete posts so that I can remove information that is no longer relevant.

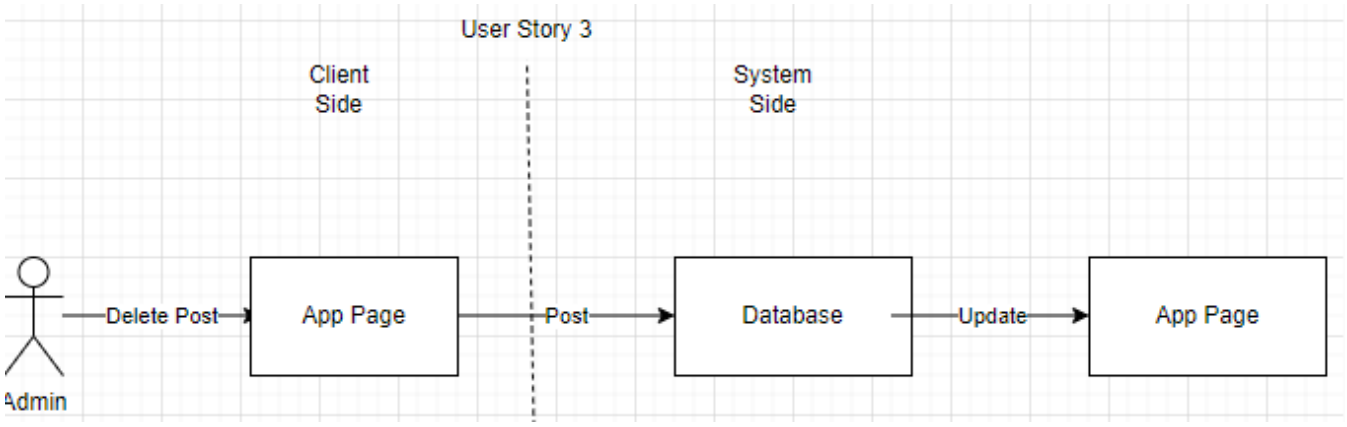


Figure 3: User Story 3 Diagram

(U4) As an admin I want to add tags to posts so that viewers who are interested in the tagged content will be notified when the post becomes public.

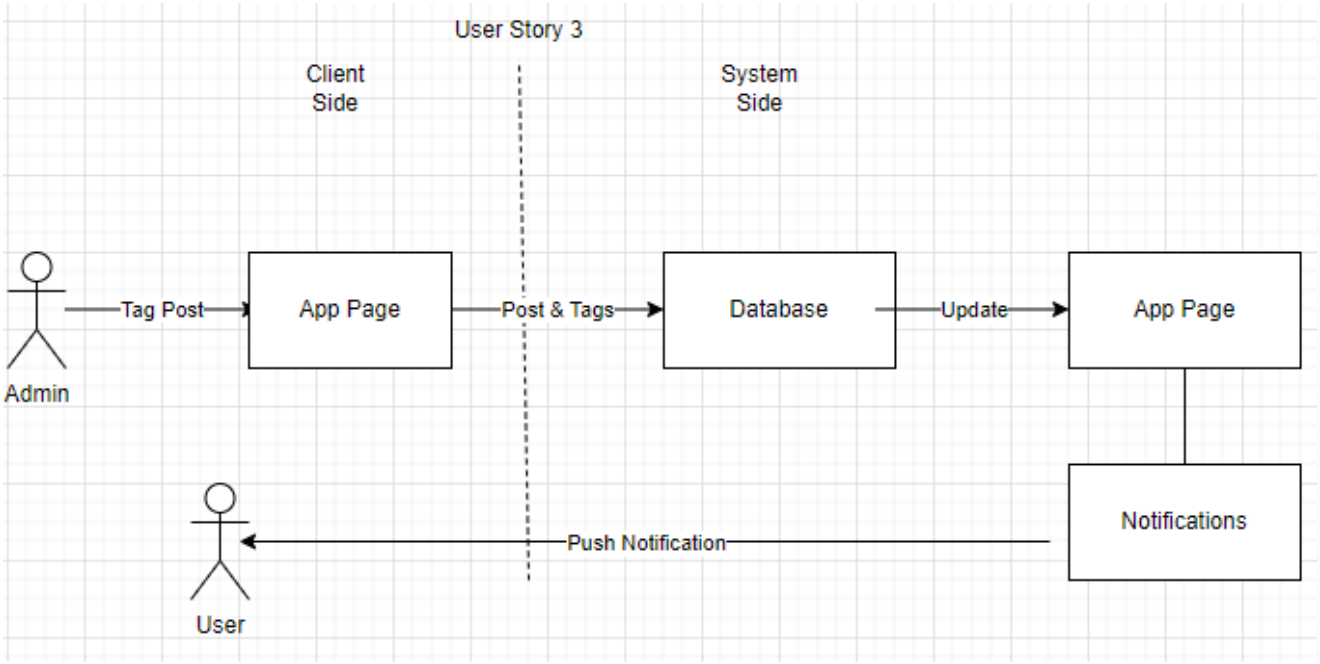


Figure 4: User Story 4 Diagram

(U5) As an admin I want users to be able to share whether they're interested in attending an event so that the expected turn out can be easily determined.

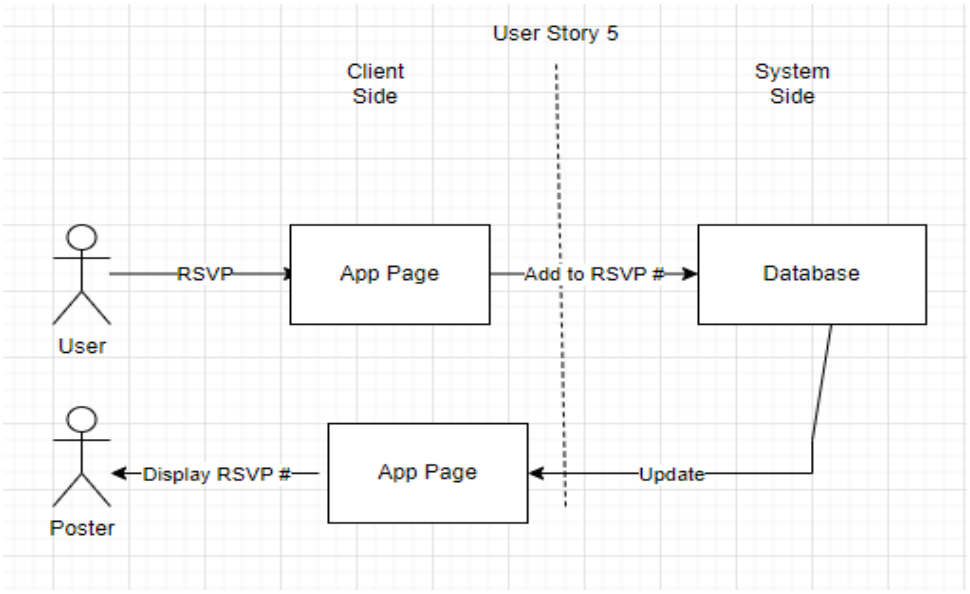


Figure 5: User Story 5 Diagram

(U6) As an admin I want to be able to approve poster accounts, so that I can ensure that accounts that can create and edit posts are vetted and legitimate.

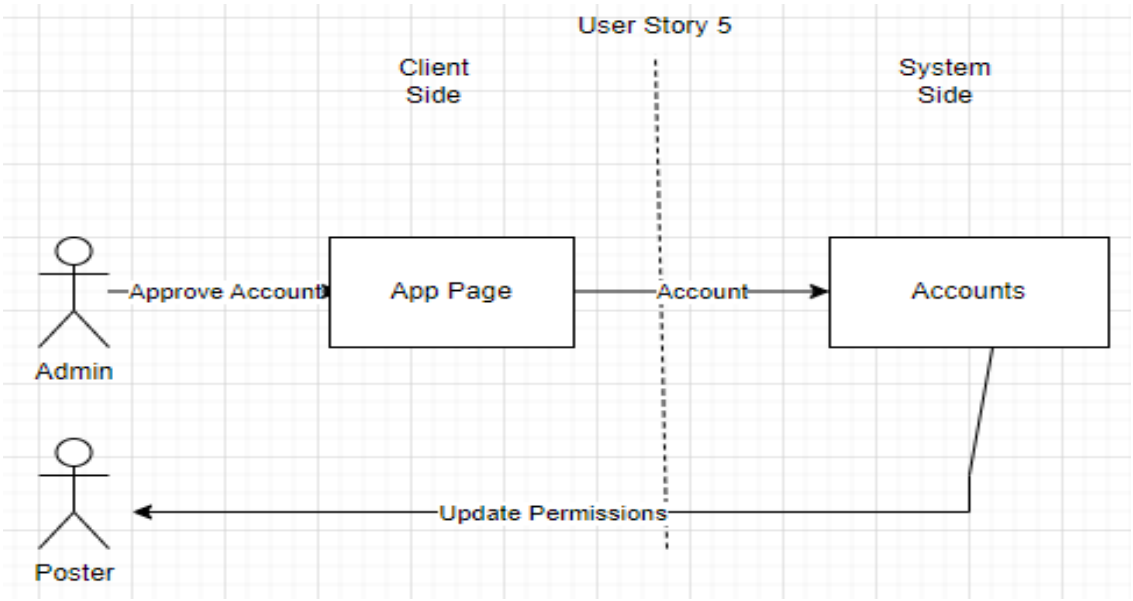


Figure 6: User Story 6 Diagram

(U7) As an admin I want to be able to delete other poster or viewer accounts so that I can maintain the integrity of the application.

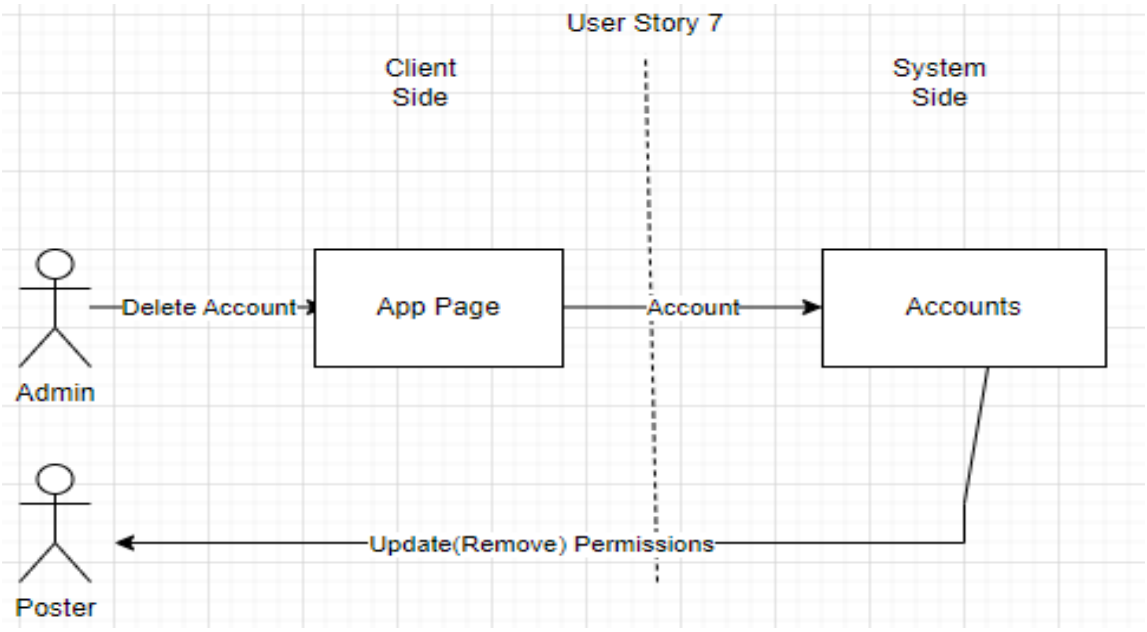


Figure 7: User Story 7 Diagram

(U8) As a viewer I want to be able to specify which tags interest me so that I can receive information relative to my interests.

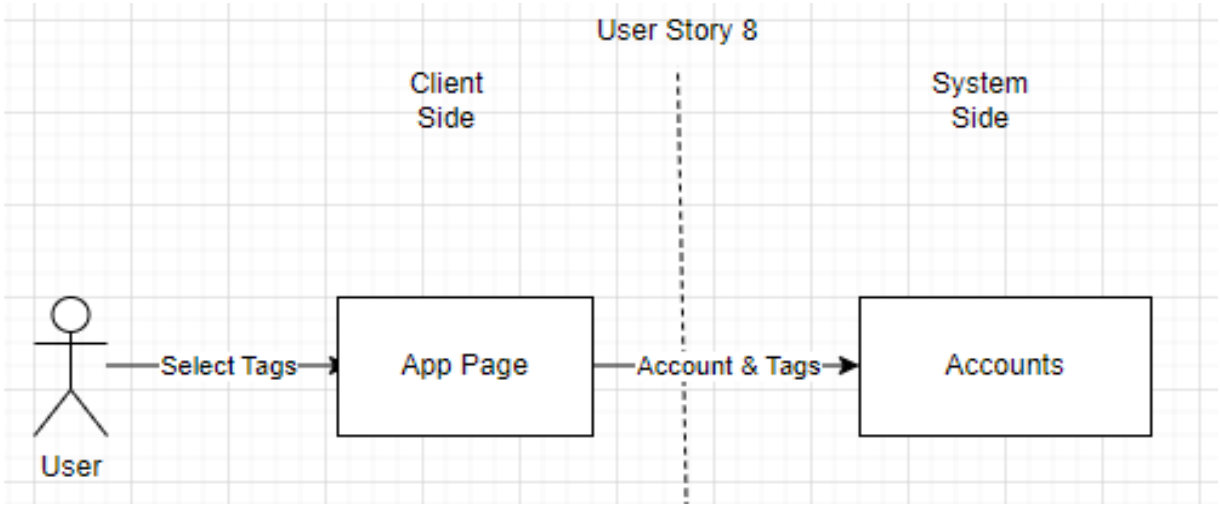


Figure 8: User Story 8 Diagram

(U9) As a viewer I want to be notified when a post regarding one of my interests is made public so that I can review that information as soon as possible.

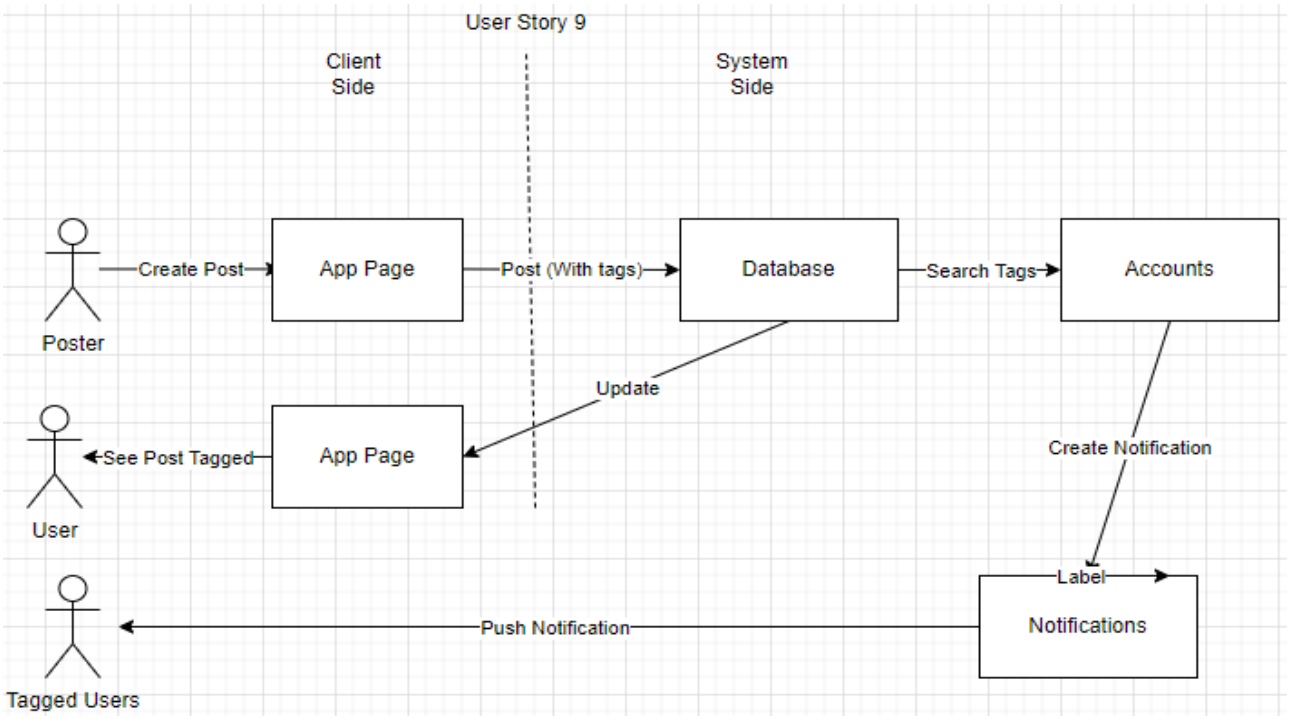


Figure 9: User Story 9 Diagram

(U10) As a user I want a personal feed so that I can browse all posts with tags relevant to my interests.

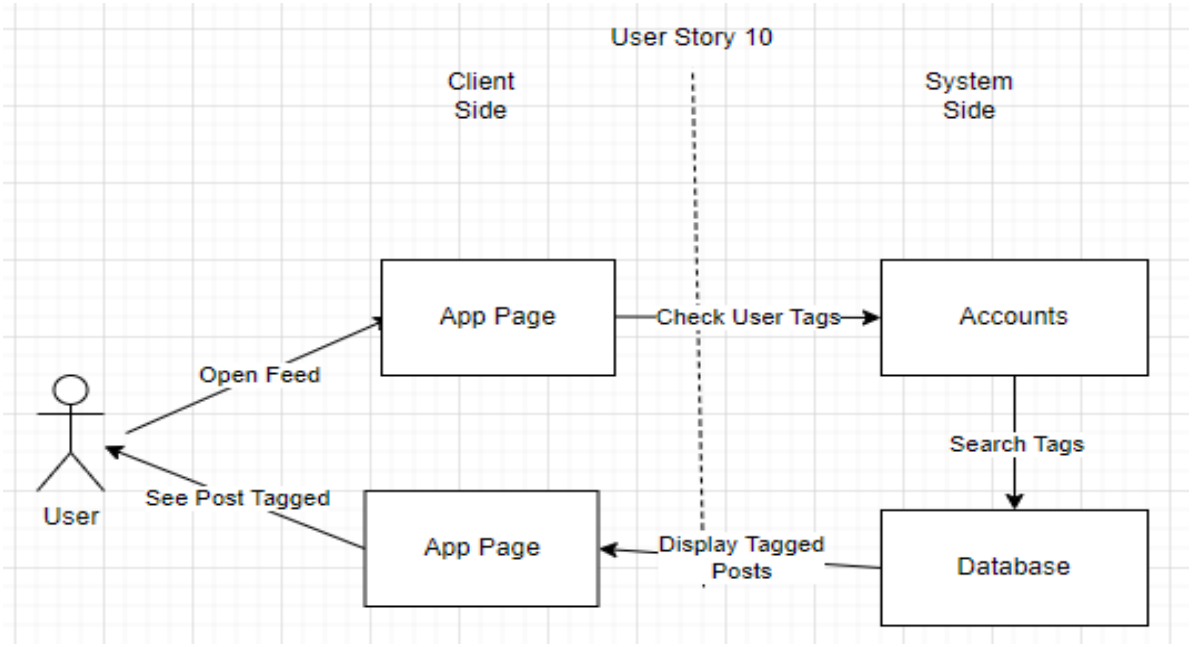


Figure 10: User Story 10 Diagram

(U11) As a user I want to be able to have access to a feed where I can see all posts, so that I can explore

information that I may not have known interests me.

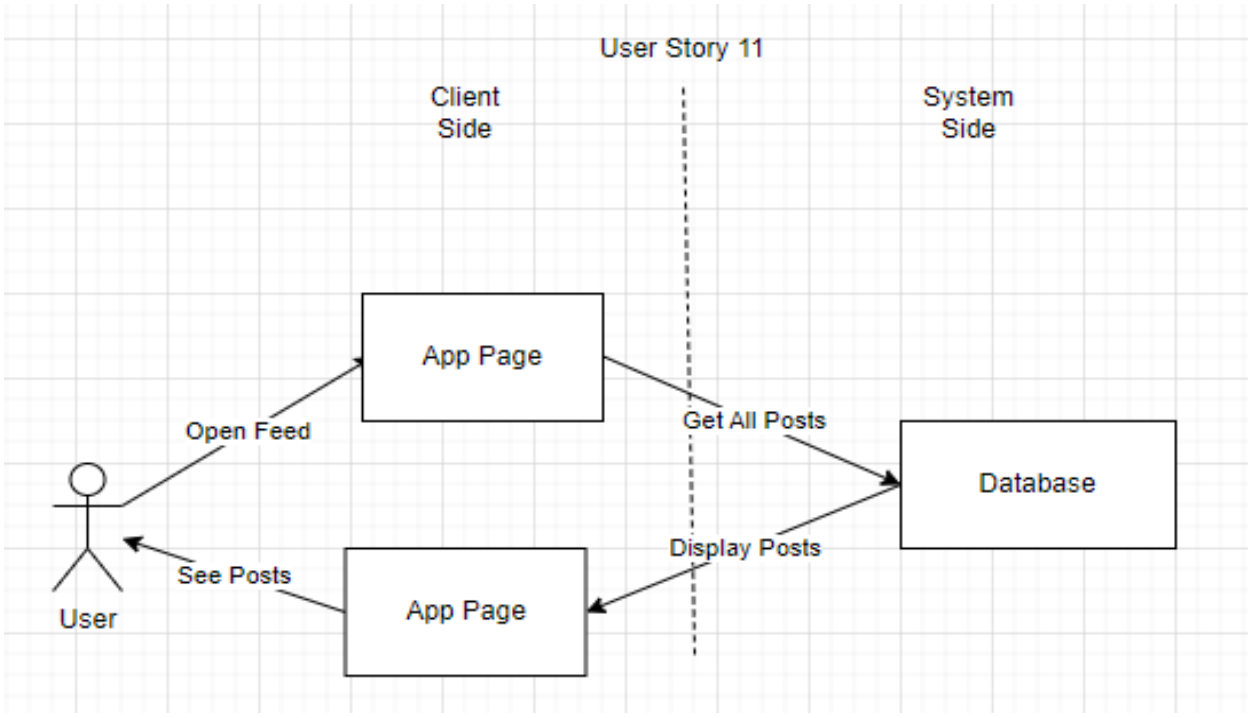


Figure 11: User Story 11 Diagram

(U12) As a user I want to be able to sort all posts by multiple metrics (most recent, tags, title, ect.) so that I can navigate all the posts more easily.

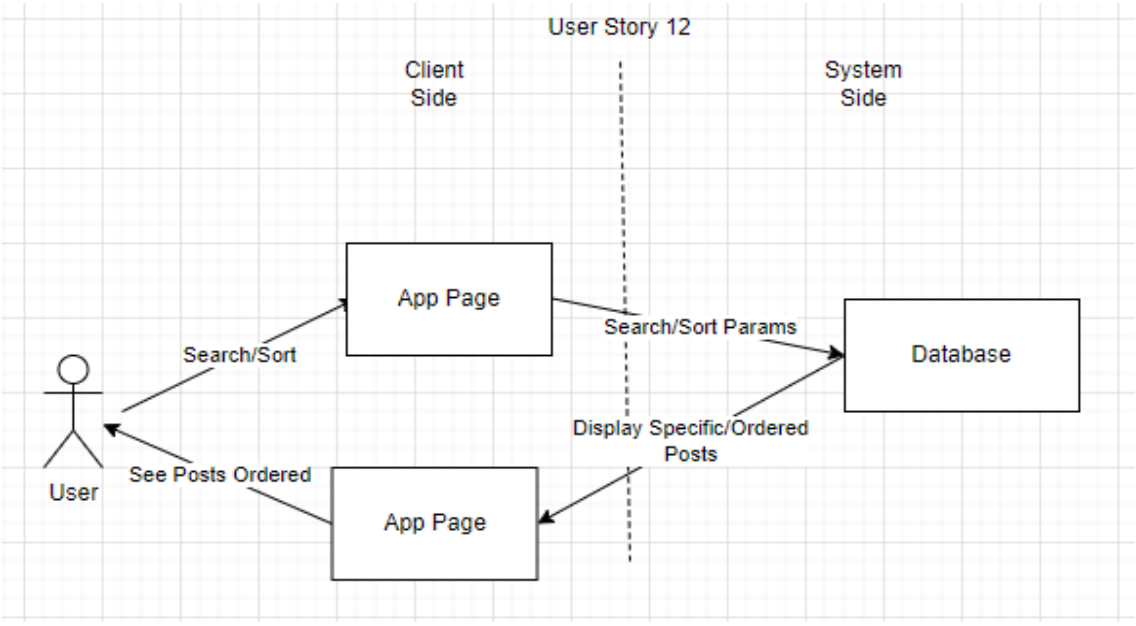


Figure 12: User Story 12 Diagram

(U13) As a viewer I want to be able to fill out a Specific Fillable Form (SFF) for admins to review so that they can share that information if they deem it appropriate to share with other viewers.

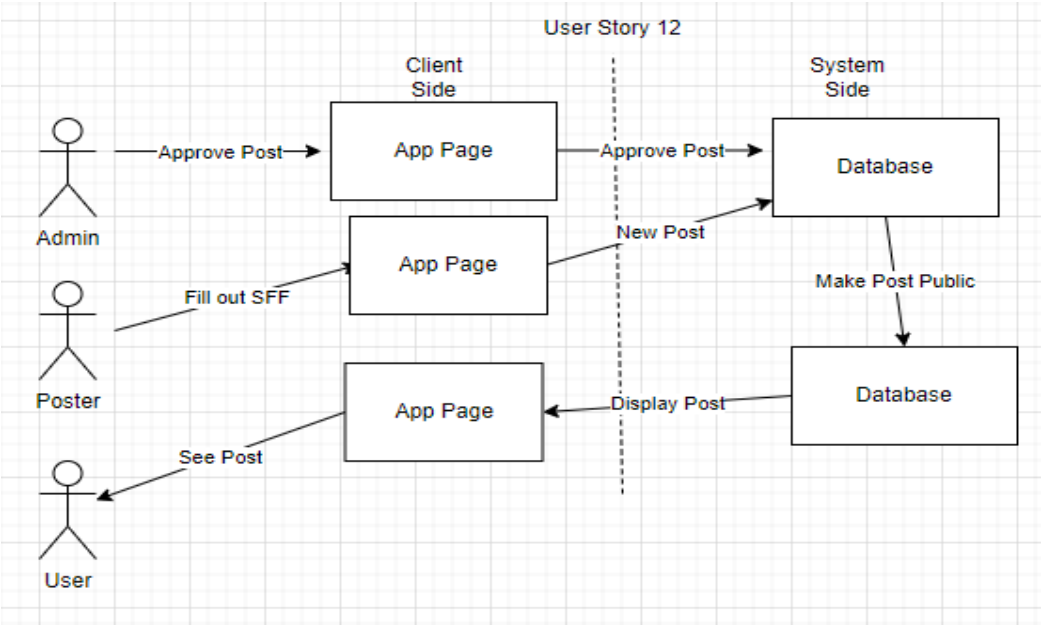


Figure 13: User Story 13 Diagram

(U14) As a user I want to be able to create an account using a unique username and a password so that I can have access to the application and its features.

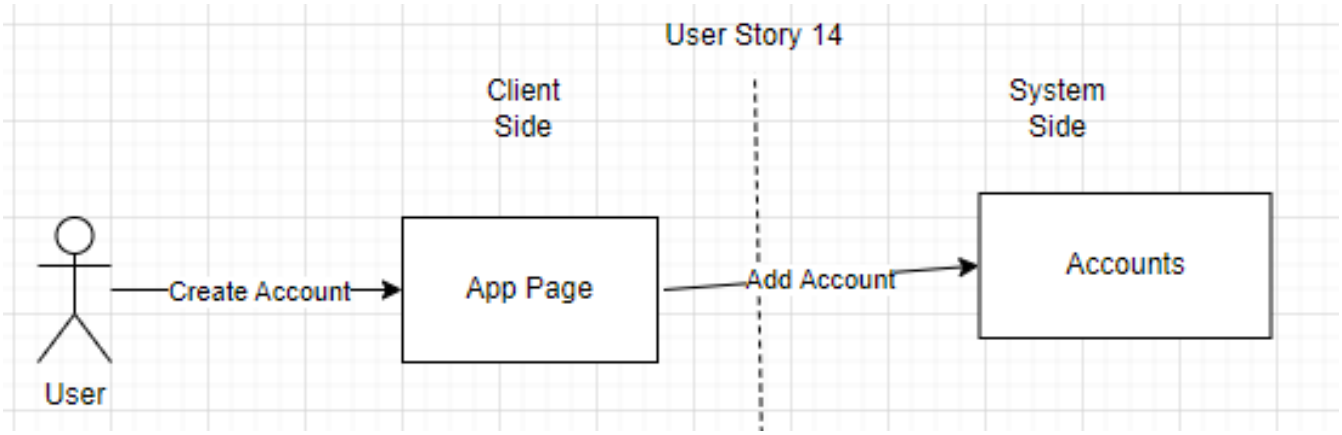


Figure 14: User Story 14 Diagram

(U15) As a user I want to be able to change my password so that I can maintain the security of my account.

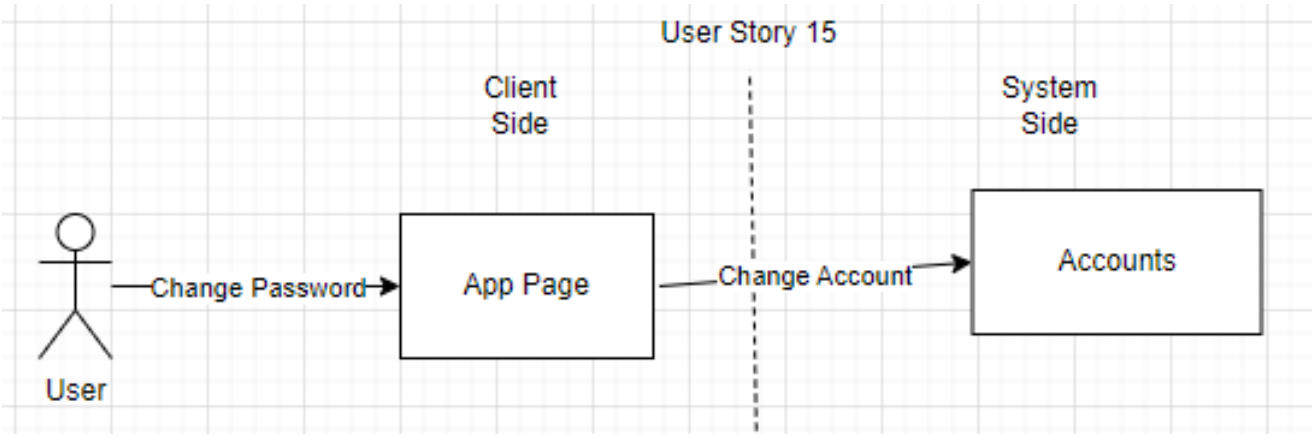


Figure 15: User Story 15 Diagram

(U16) As a user I want to be able to delete my own account so that I can remove my personal information from the application.

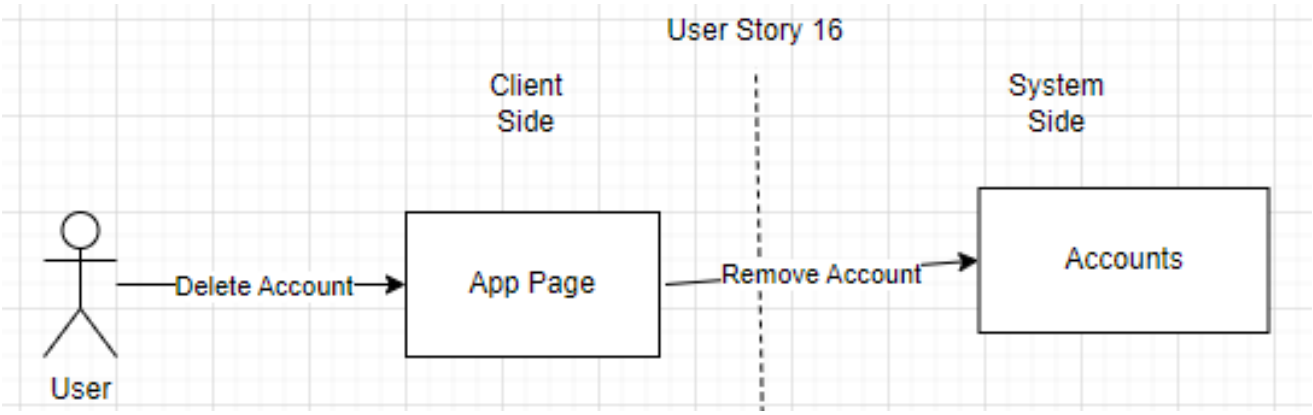


Figure 16: User Story 16 Diagram

(U17) As an approved poster I want to be able to create posts so that I can share information about my business or organization with the viewers.

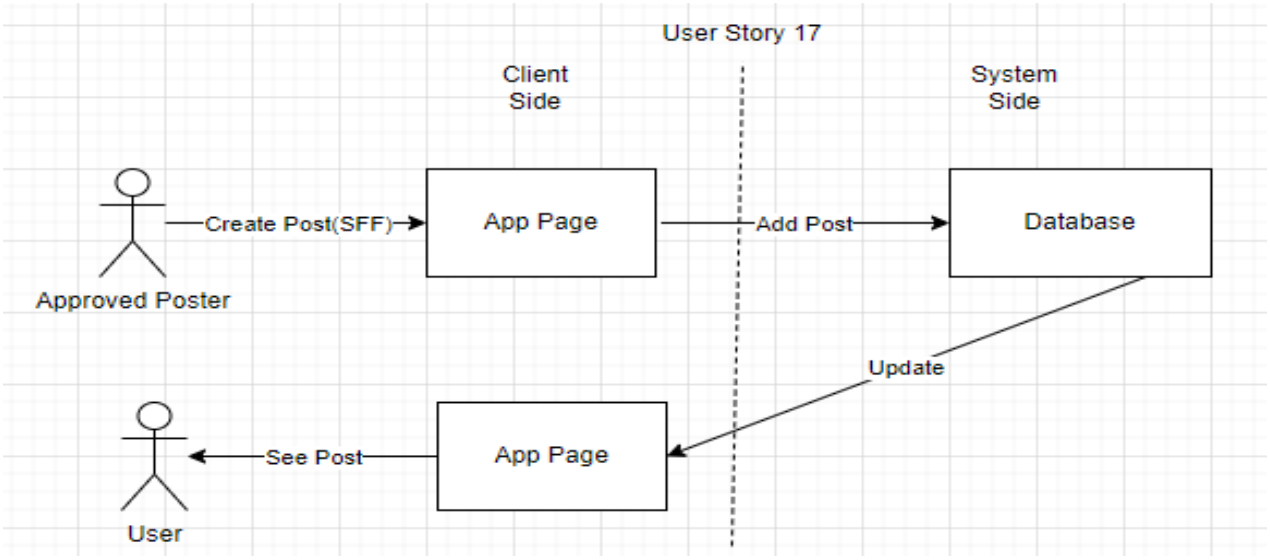


Figure 17: User Story 17 Diagram

(U18) As an approved poster I want to be able to edit my posts so that I can update the information being shared.

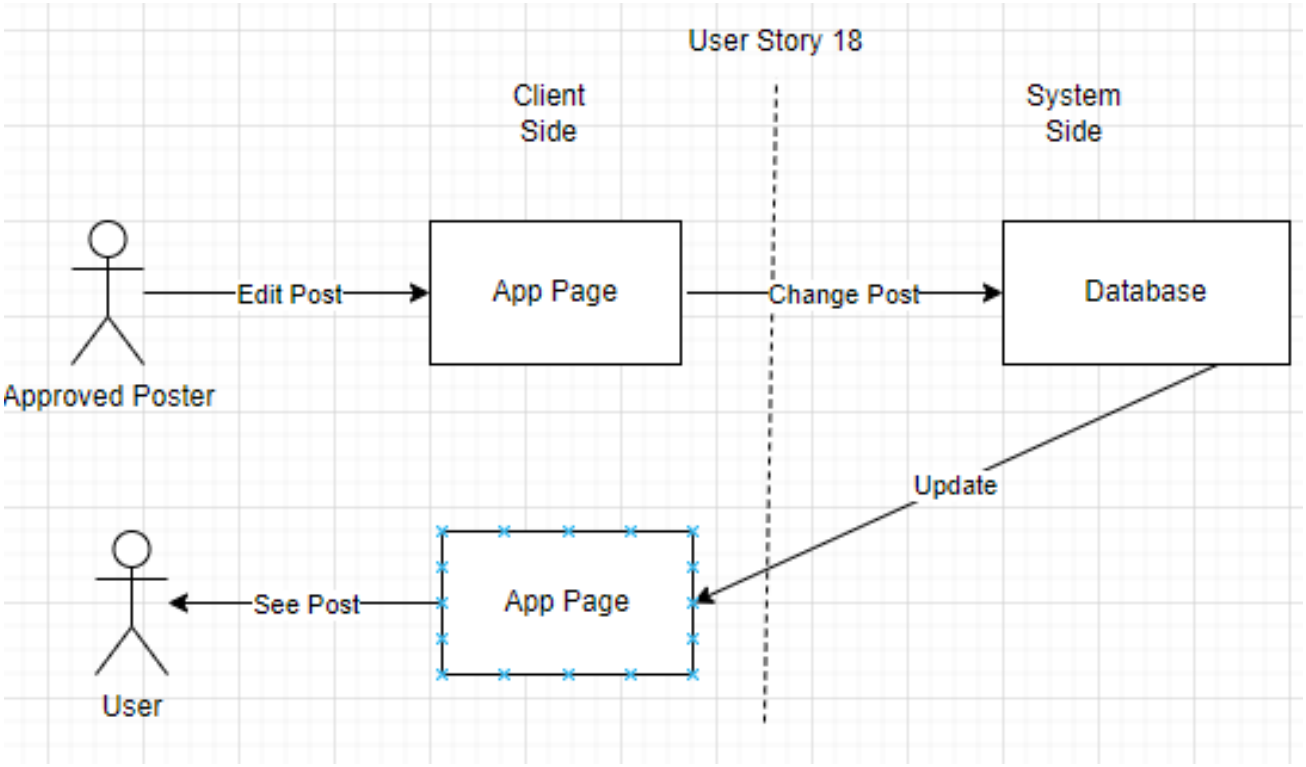


Figure 18: User Story 18 Diagram

(U19) As an approved poster I want to be able to delete my posts so that I can remove information that is no

longer relevant.

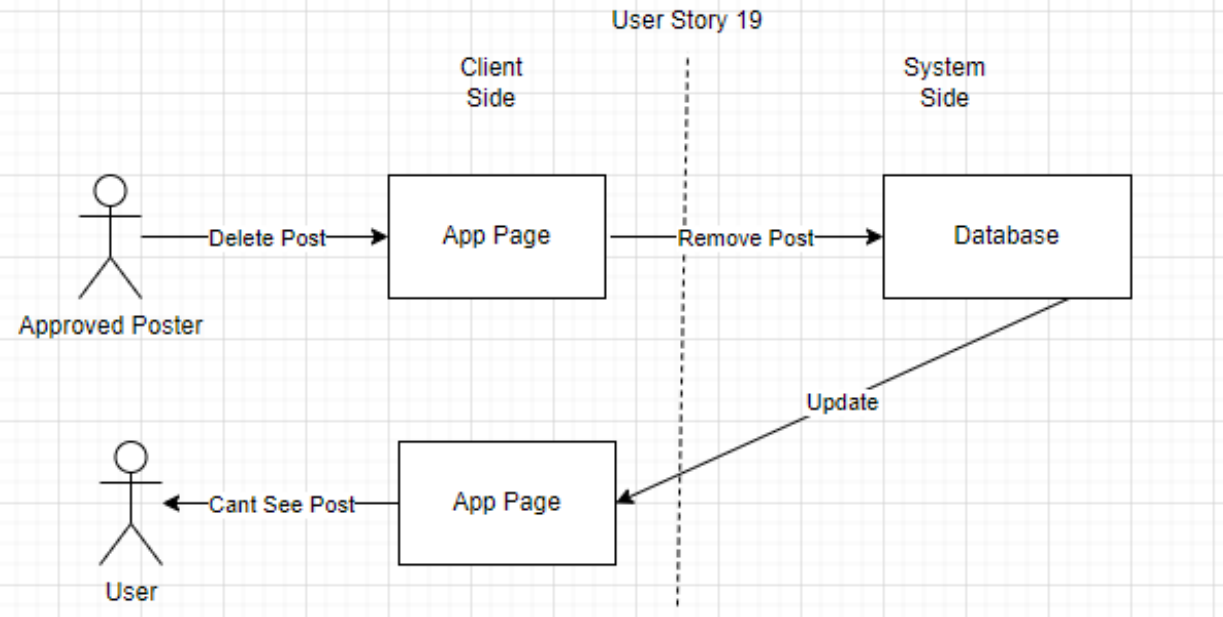


Figure 19: User Story 19 Diagram

2.2 - FUNCTIONAL REQUIREMENTS

R1(U1,2) Admins must be able to create posts and edit any post.

R2(U3) Admins must be able to delete any post.

R3(U4) Any user that can create posts must be able to add tags to their post.

R4(U5) Any post tagged with “Event” must have buttons to allow users to RSVP.

R5(U6) Admins must be able to upgrade/downgrade to the privileges of another account.

R6(U7) Admins must be able to delete other accounts.

R7(U8) Viewers must be able to choose pre-determined tags regarding information that interests them.

R8(U9) Users must be notified when a post that contains a tag they currently have selected is posted.

R9(U10) All Users must have a personal feed that will contain all recent posts with tags that they currently have selected.

R10(U11) All Users must have access to a global feed that contains all recent posts regardless of tags.

R11(U12) All Users must be able to sort posts by multiple metrics(most recent, tags, title, ect.)

R12(U13) Viewers must be able to fill out a Specific Fillable Form (SFF) to share possibly useful information to an Admin.

R13(U14) All Users must be able to create an account using a unique username and a password.

R14(U15) All Users must be able to change their password at any time.

R15(U16) All Users must be able to delete their own account.

R16(U17,18) Approved Posters must be able create posts and edit their own posts.

R17(19) Approved Posters must be able to delete their own posts.

2.3 - EXTERNAL INTERFACE REQUIREMENTS

User Interfaces

The system will include a login screen, page displaying posts, page displaying alerts, a contact us page, a calendar, jobs/ volunteer page, and an account settings page. At the bottom of the screen there will be a tab bar, allowing users to switch to different pages. The system will also contain a menu button in the top left corner, so the user can access account settings, contact us, and less popular pages.

The pages and interface will be as user friendly as possible, with an easy to read font and buttons large enough that are easy to press.

Hardware Interfaces

Storage: 200 MB or above

Software Interfaces

Operating System : Android, iOS

Database : Firebase, MySQL Database Server

Development Tool : Flutter

Communication Interfaces

The system should be connected to the internet

The system shall connect to the database

2.4 - NON-FUNCTIONAL REQUIREMENTS

Usability

The system will be simple and easy for anyone to use.

Reliability and Availability

The system should have minimal downtime on rare occasions.

Security

The system will encode user data and password to ensure privacy for users.

Maintainability

The system will require minimal maintenance that should not require a tech support group.

Scalability

The system will be able to handle new users and posts being added to the system.

Compatibility

The system will be compatible with both Android and iOS.

Performance

The system will be able to provide results to the user as fast as the user's connection and the database can process the query.

2.5 - USE CASE DIAGRAMS

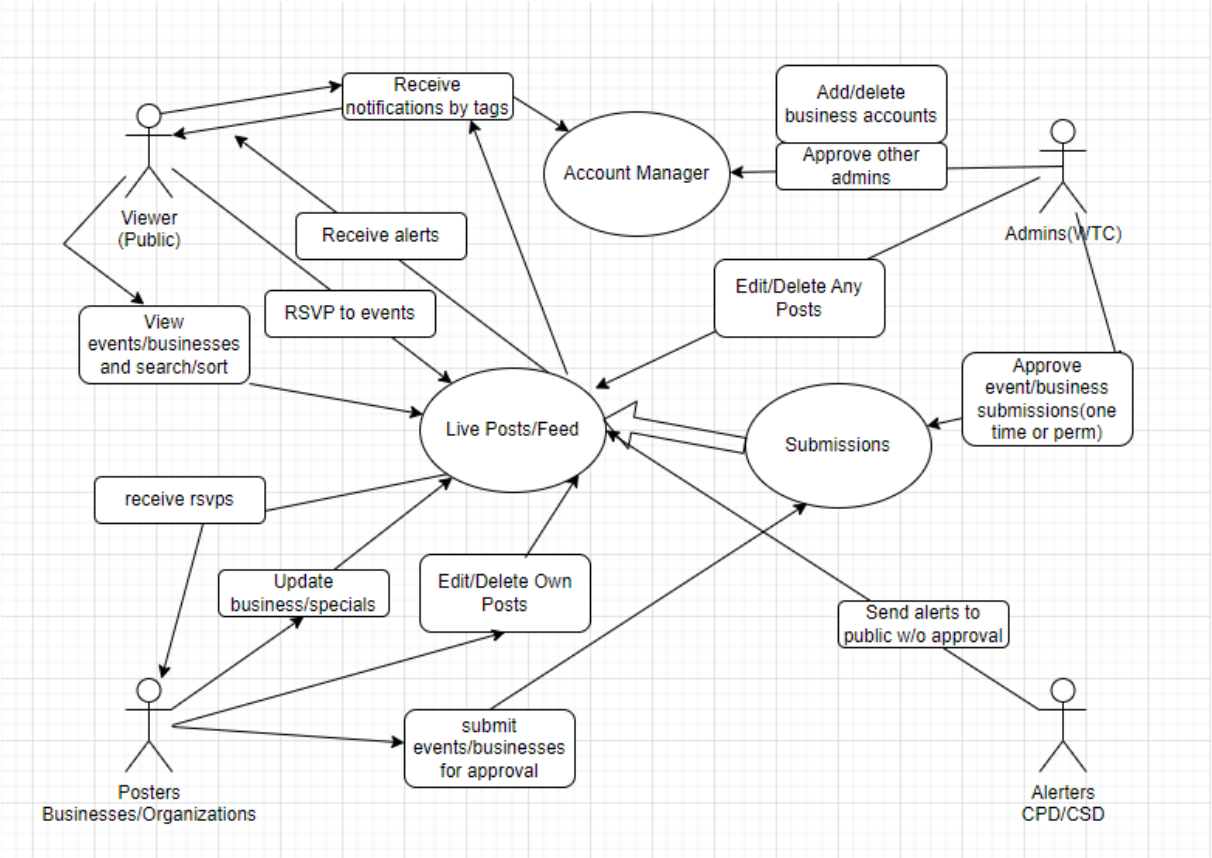


Figure 20: Use Case Diagram

CHAPTER 3

PROJECT PLANNING

3.1 - TEAM RESPONSIBILITIES

- (# of Team Members likely needed) Task - Specifics/Subtasks

- (5) Object Structures - Account, Different Accounts, Posts, Notifications

- (3)Front End/UI - Displays

- (2)Notification system

- (2)SFF Create post -> Approval

- (3)Sign In Page/Account management

- (3)Admin View

- (5)Pages - Business/Map, Event/Calendar, Settings/Account, Orgs/Activities, Jobs, Volunteer/Donate, contact us

- (2)Tag System

- (3)Database

- (5)Integration

3.2 - WORK BREAKDOWN STRUCTURE

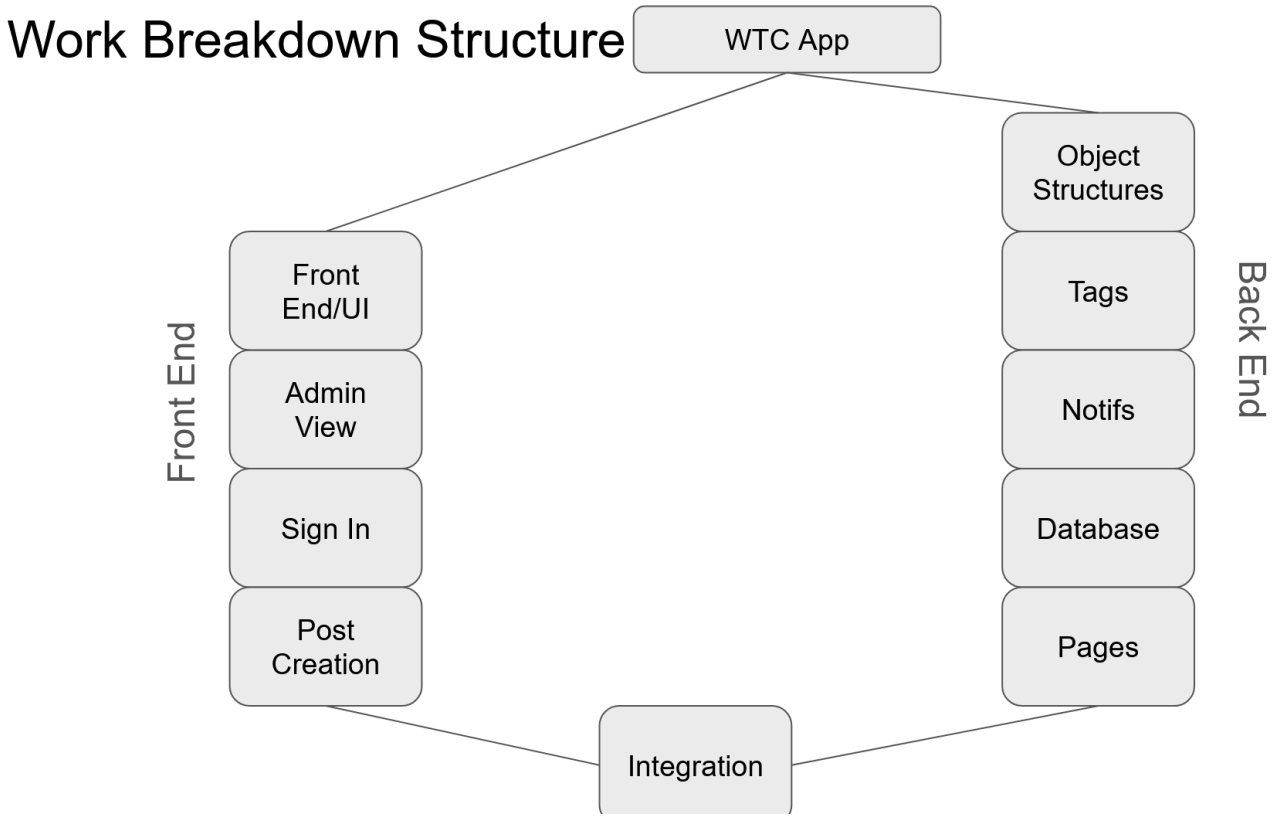


Figure 21: Work Breakdown Structure Diagram

3.3 - GANTT CHART

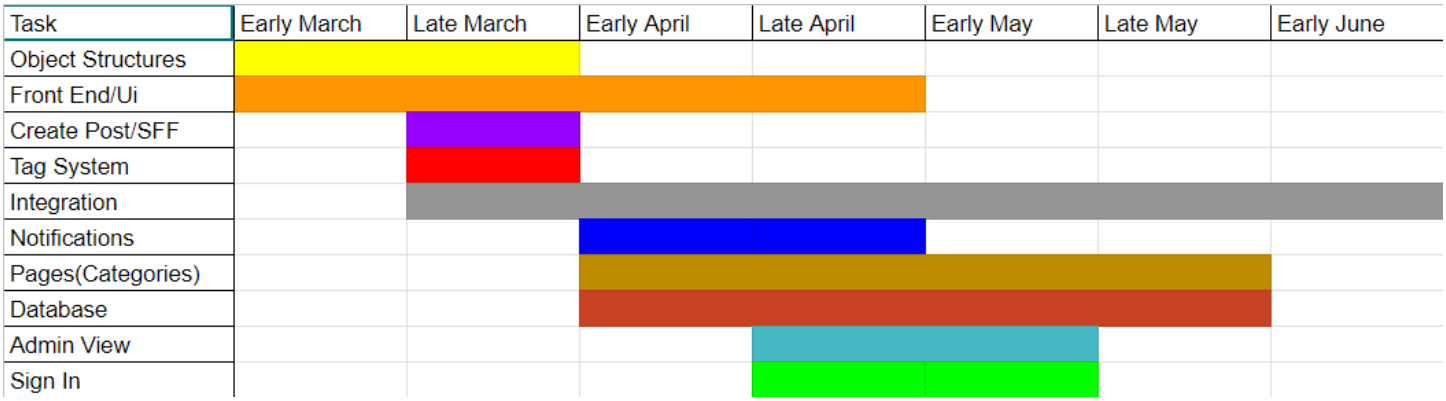


Figure 22: Gantt Chart

3.4 - PERSPECTIVE TOOLS AND TECHNOLOGIES

Frontend framework and language

- Flutter
- Dart programming language

Backend framework and language

- Firebase
- MySQL
- Java programming language

CHAPTER 4

DESIGN SPECIFICATIONS

Behavioral Architecture (Sequence Diagrams):

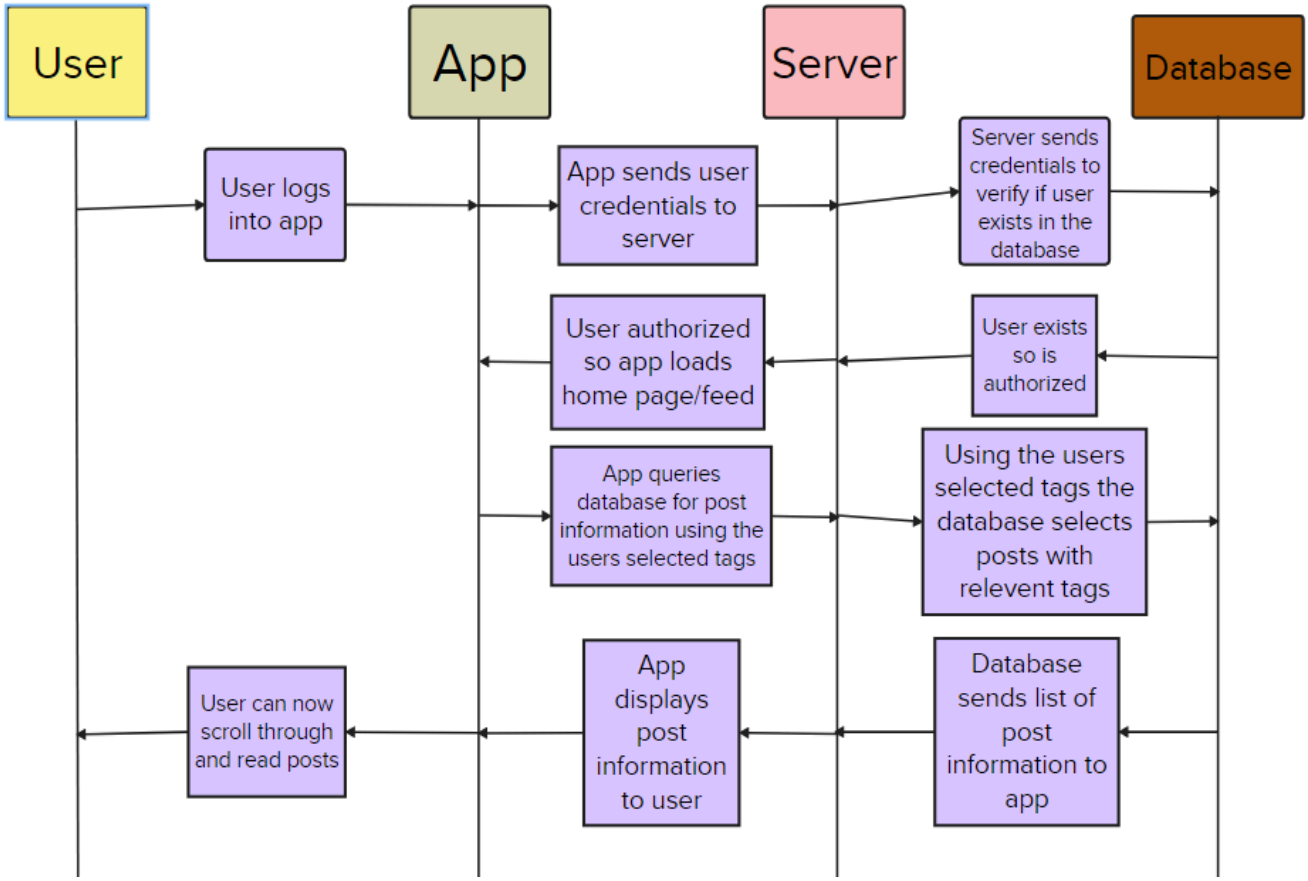


Figure 23: Sequence Diagram - Login

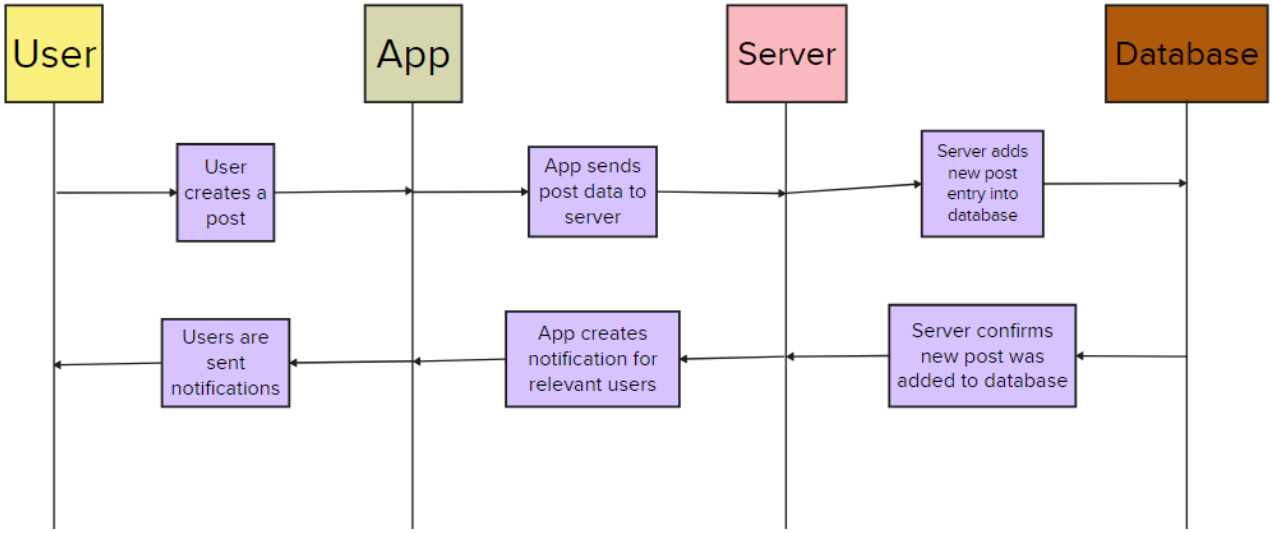


Figure 24: Sequence Diagram - Create Post

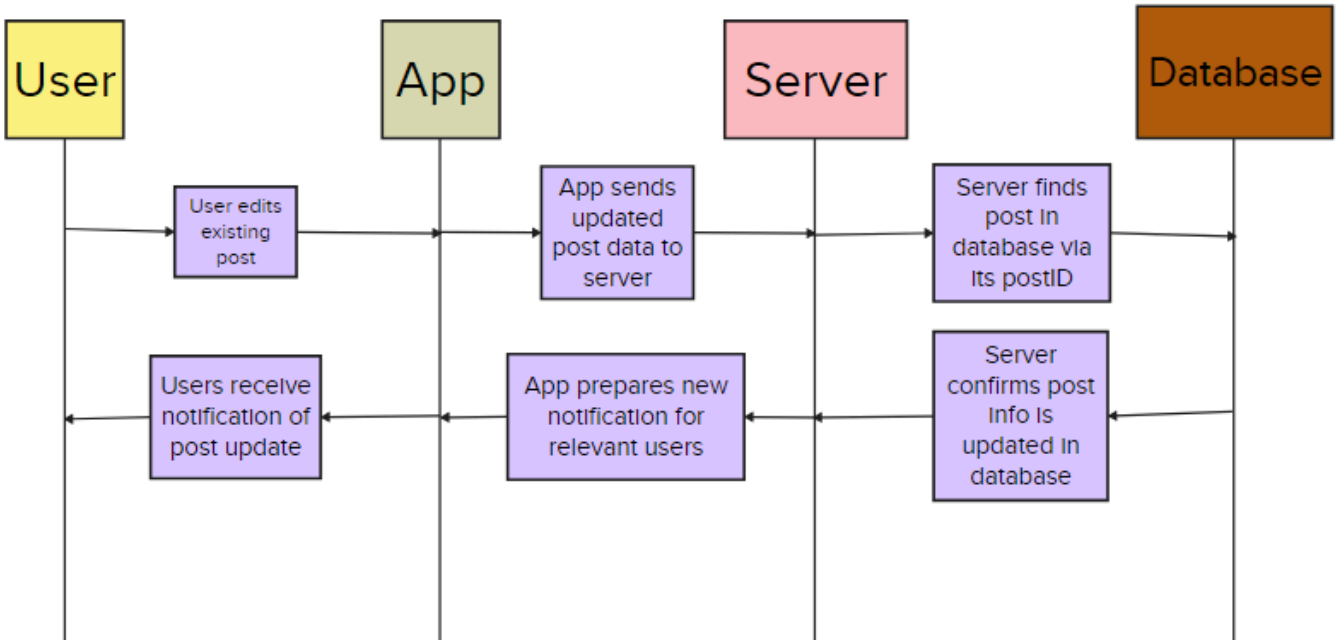


Figure 25: Sequence Diagram - Edit Post

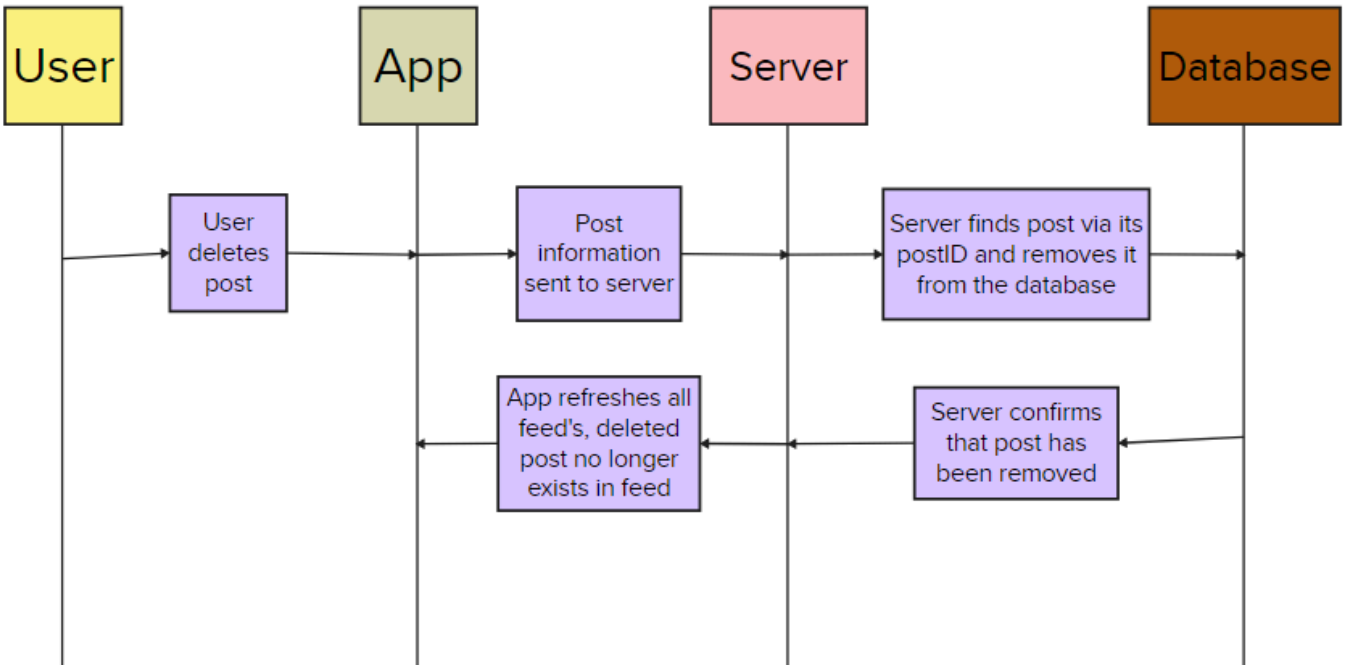


Figure 26: Sequence Diagram - Delete Post

Behavioral Architecture (Activity Diagrams):

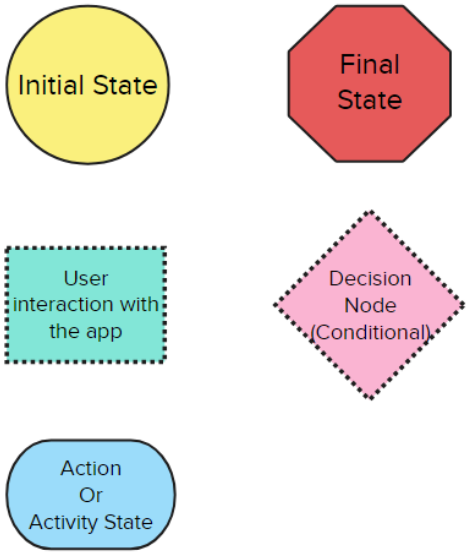


Figure 27: Activity Diagram Legend

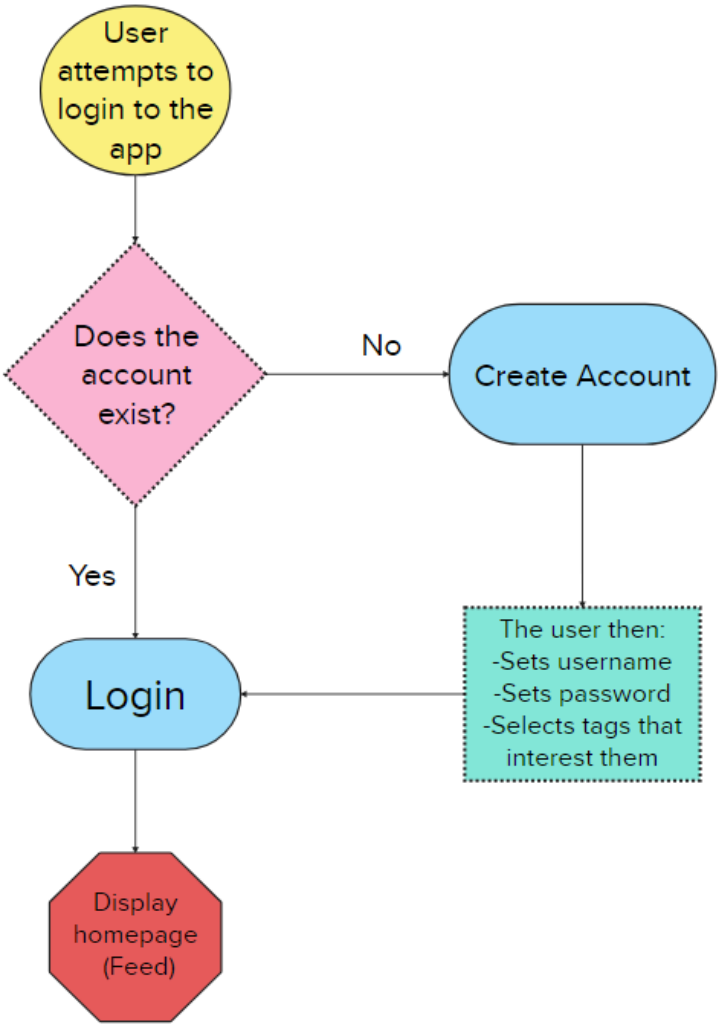


Figure 28: Activity Diagram 1 - Login

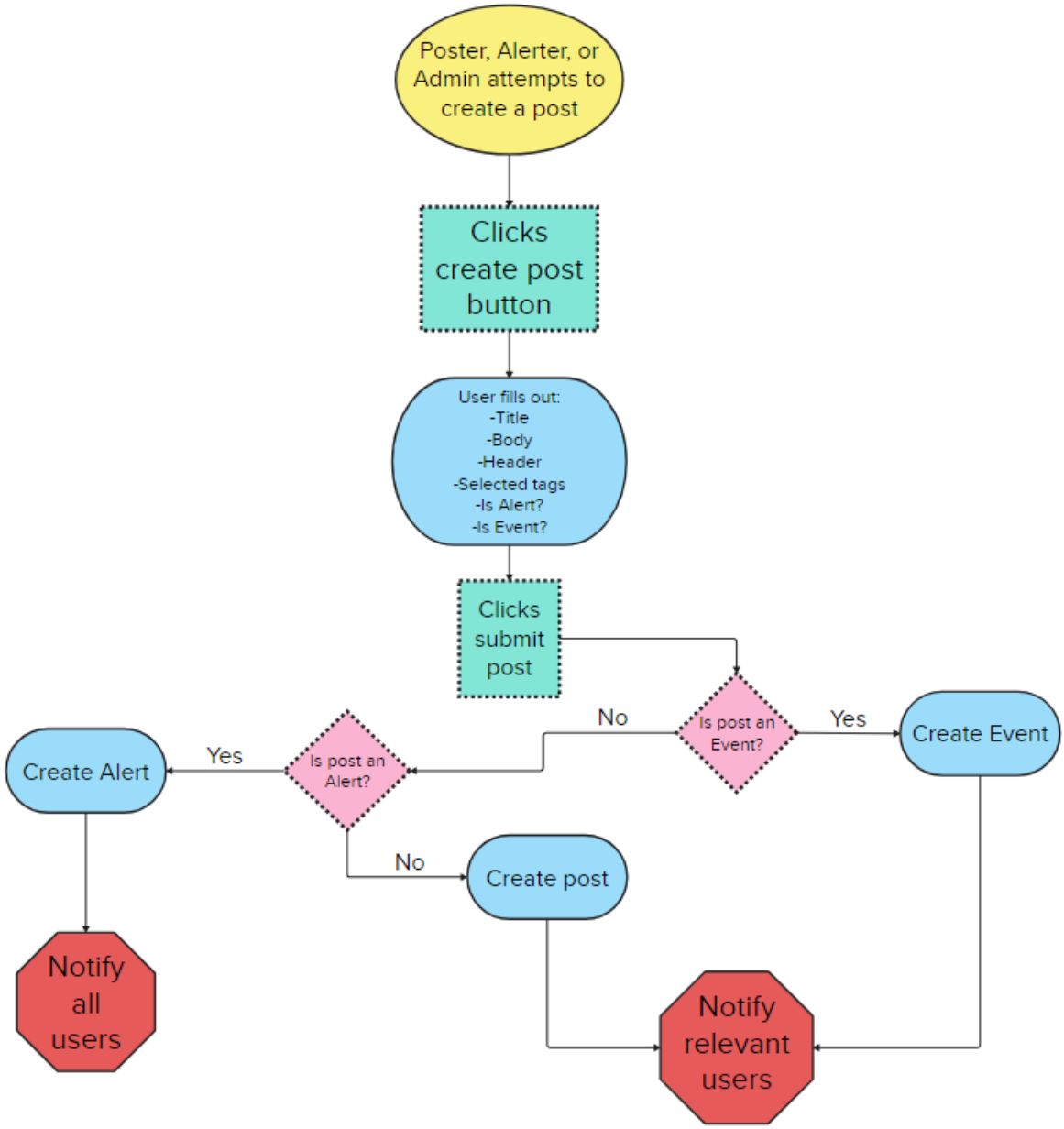


Figure 29: Activity Diagram 2 - Create Post

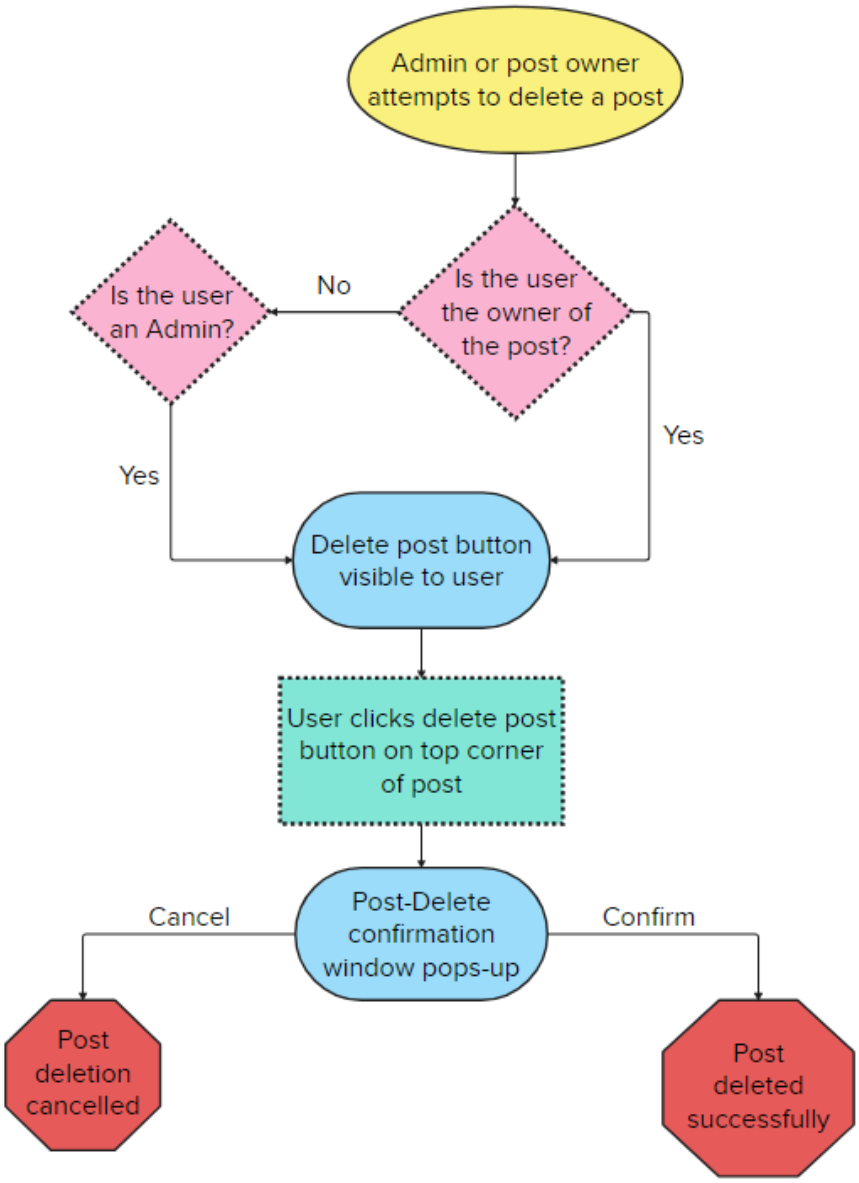


Figure 30: Activity Diagram 3 - Delete Post

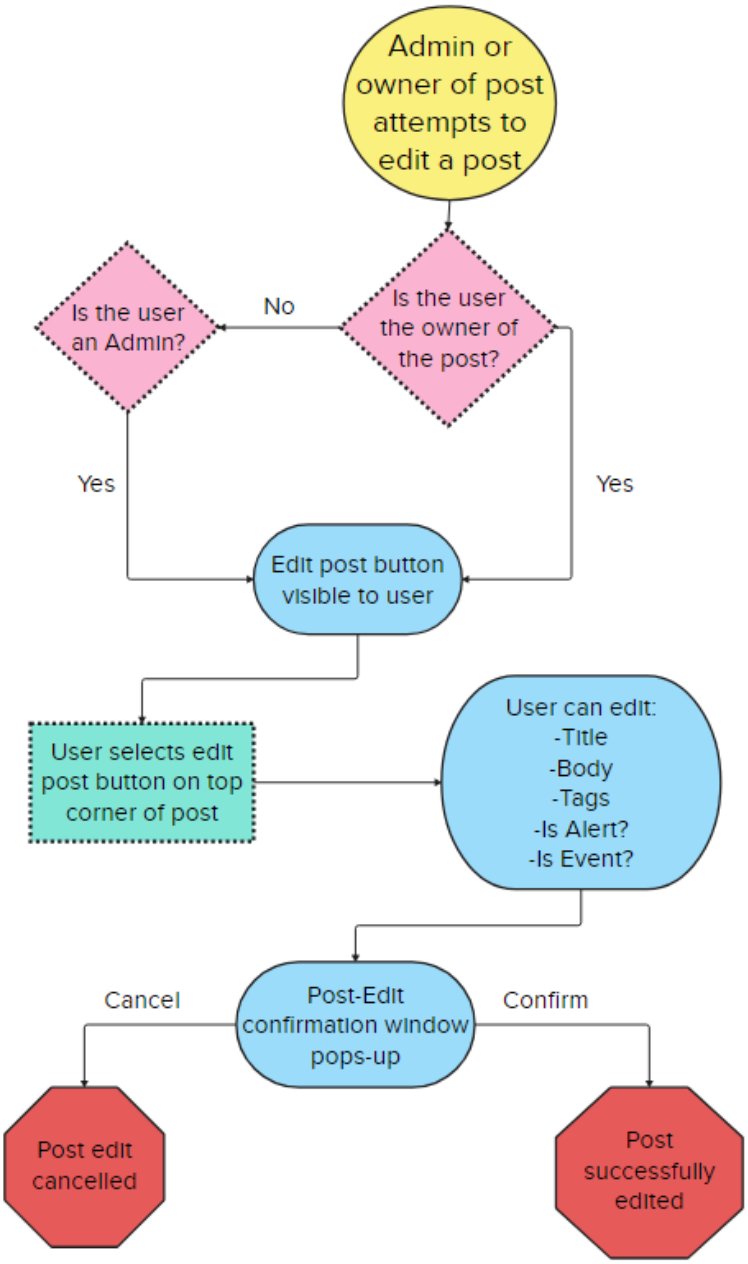


Figure 31: Activity Diagram 4 - Edit Post

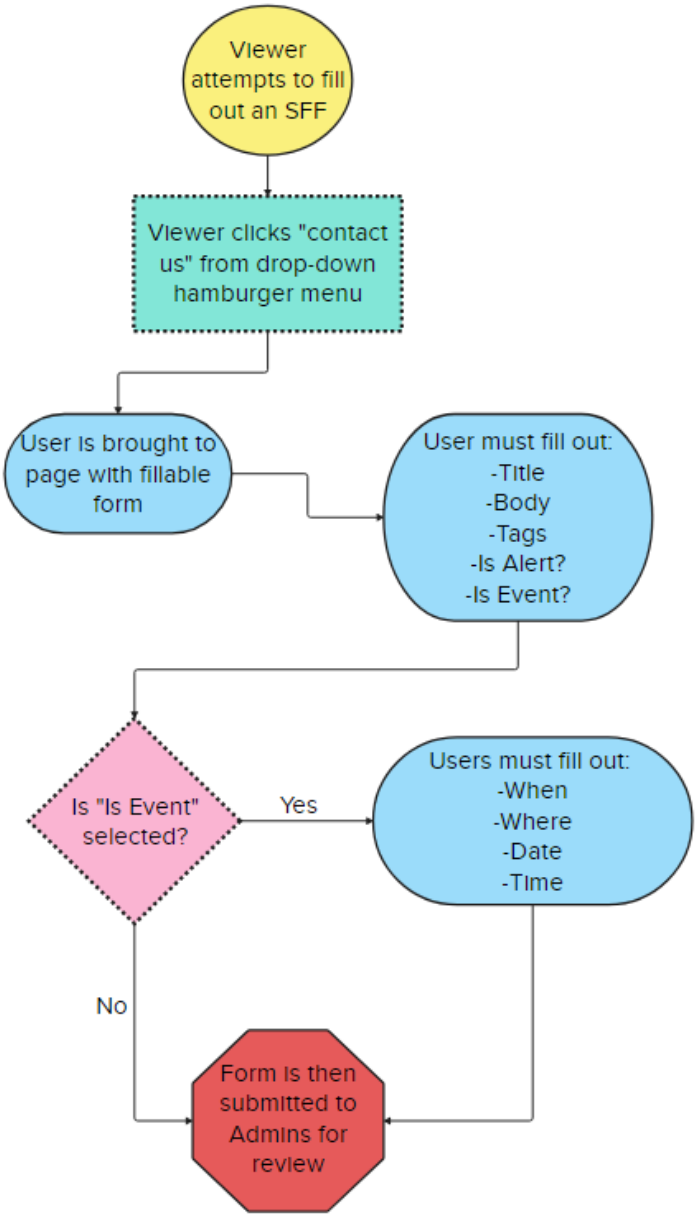


Figure 32: Activity Diagram 5 - SFF (Specific Fillable Form)

Structural Architecture (UML Class Diagrams):

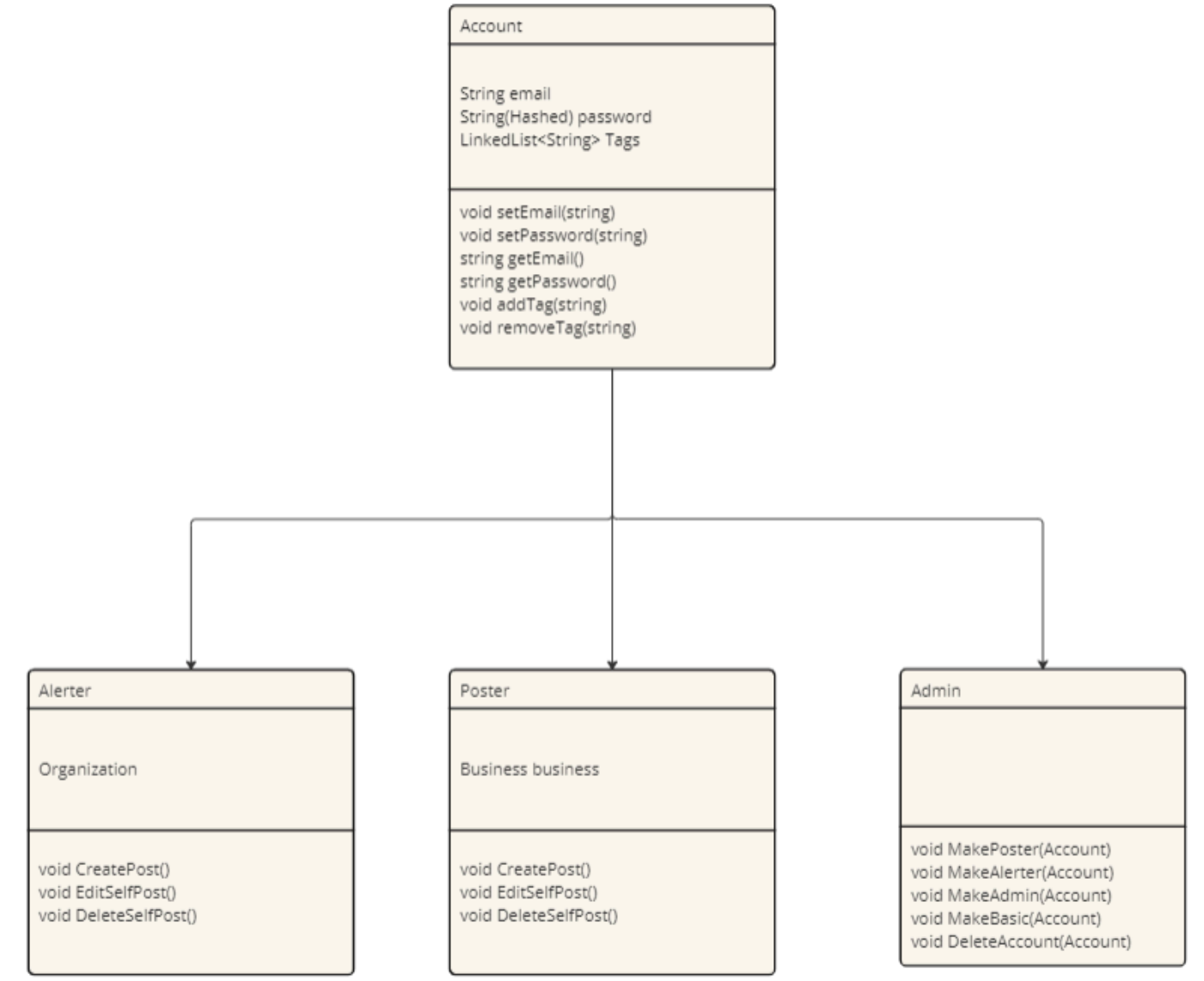


Figure 33: Account Object Diagram

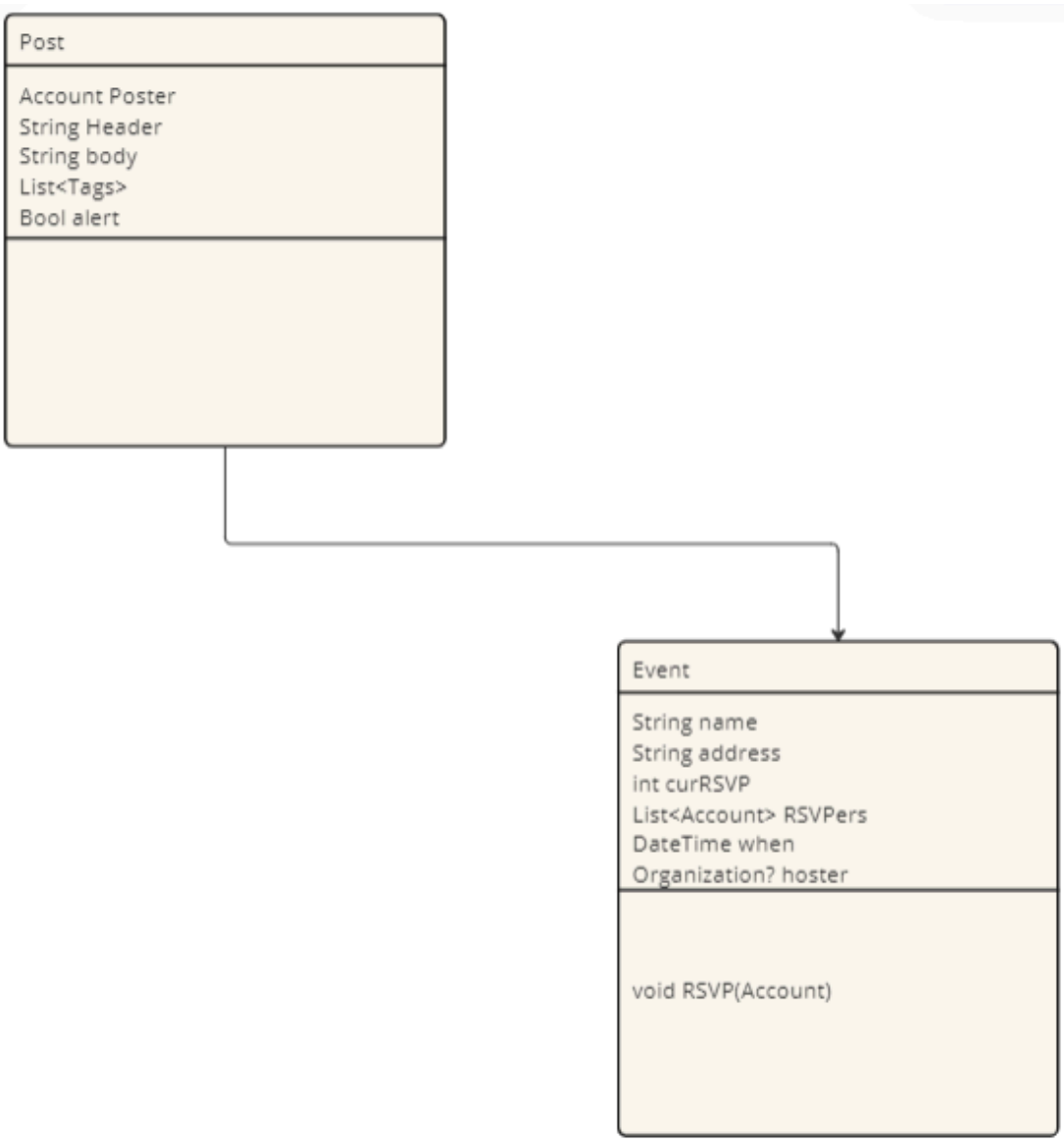


Figure 34: Post Object Diagram

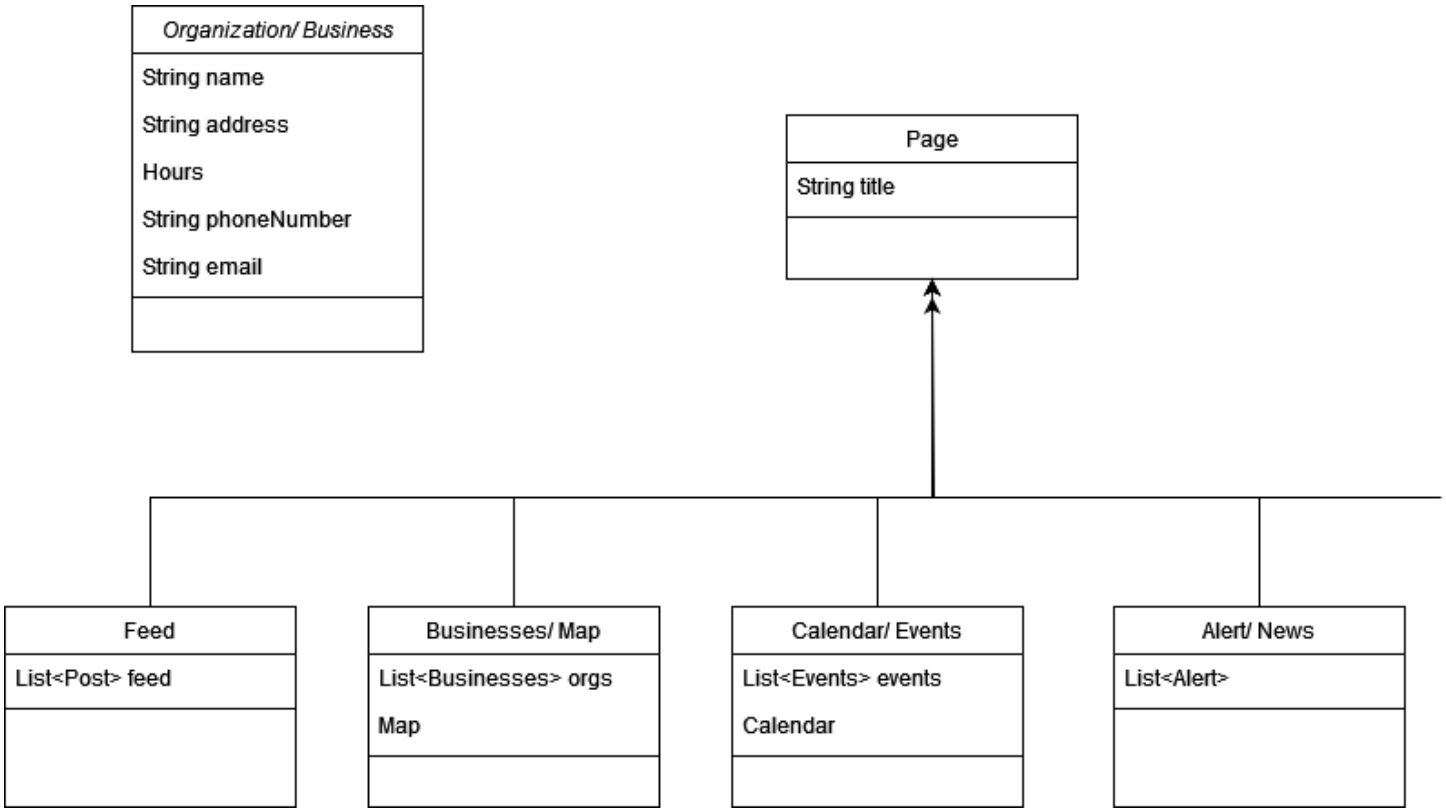


Figure 35: Pages Object Diagram 1

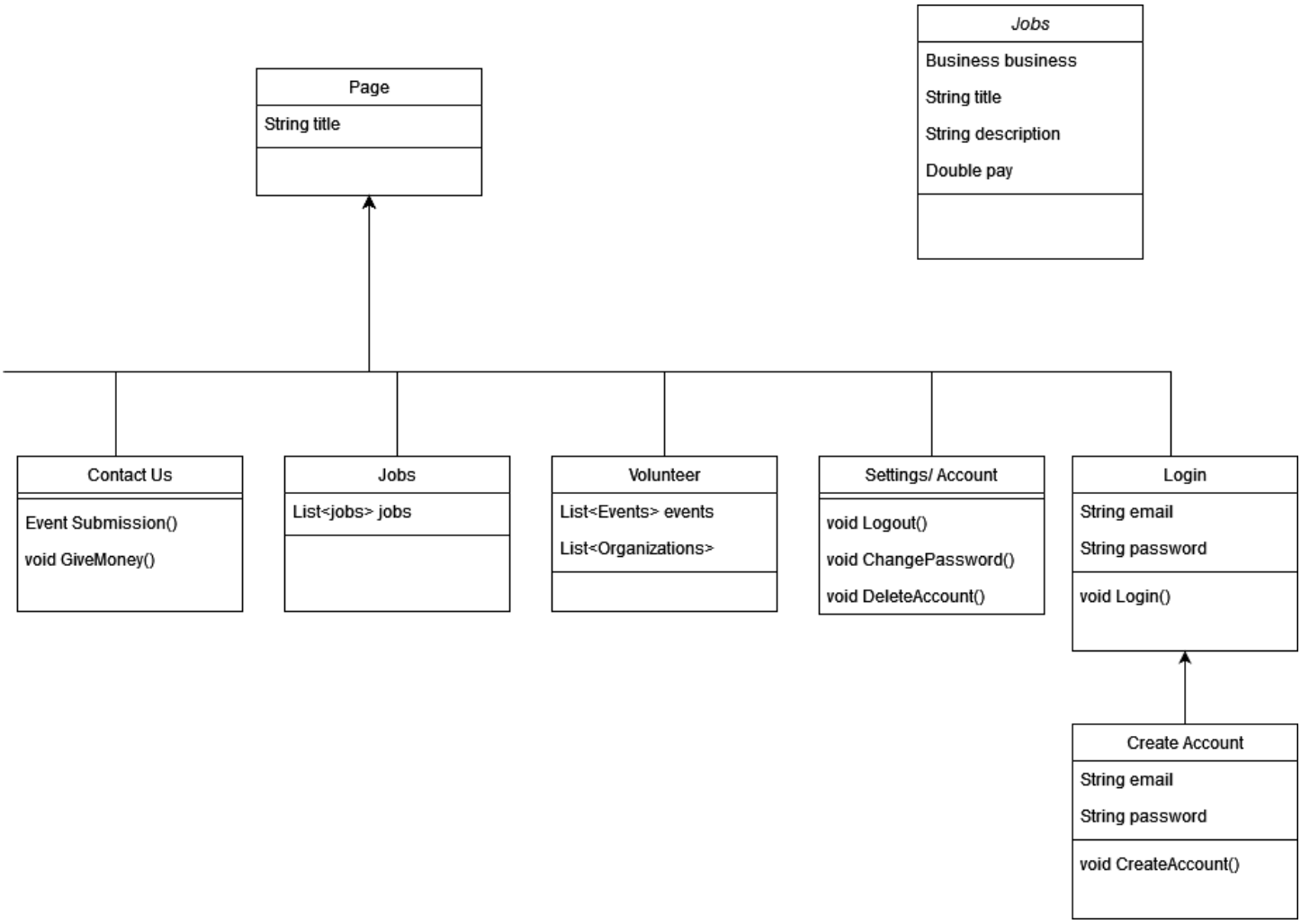


Figure 36: Pages Object Diagram 2

4.2 - DATA DESIGN

Database Design

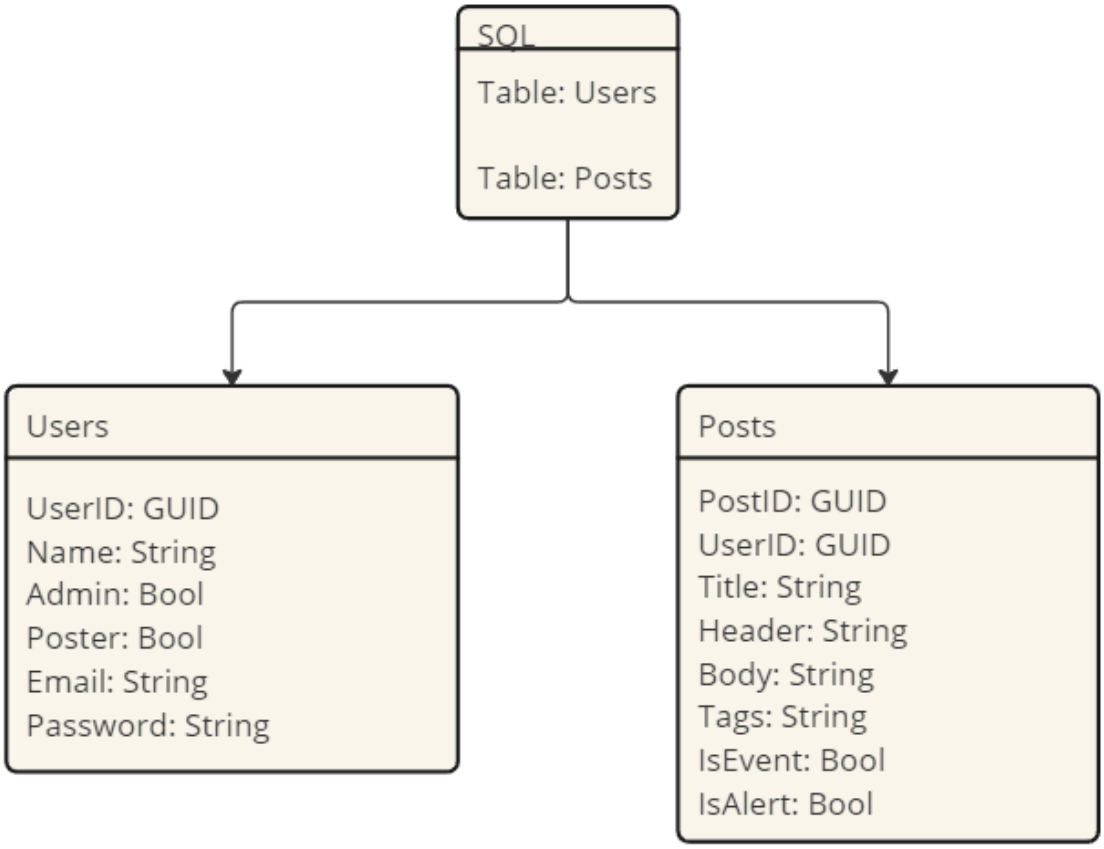


Figure 37: Database Design Diagram

4.3 - USER INTERFACE DESIGN

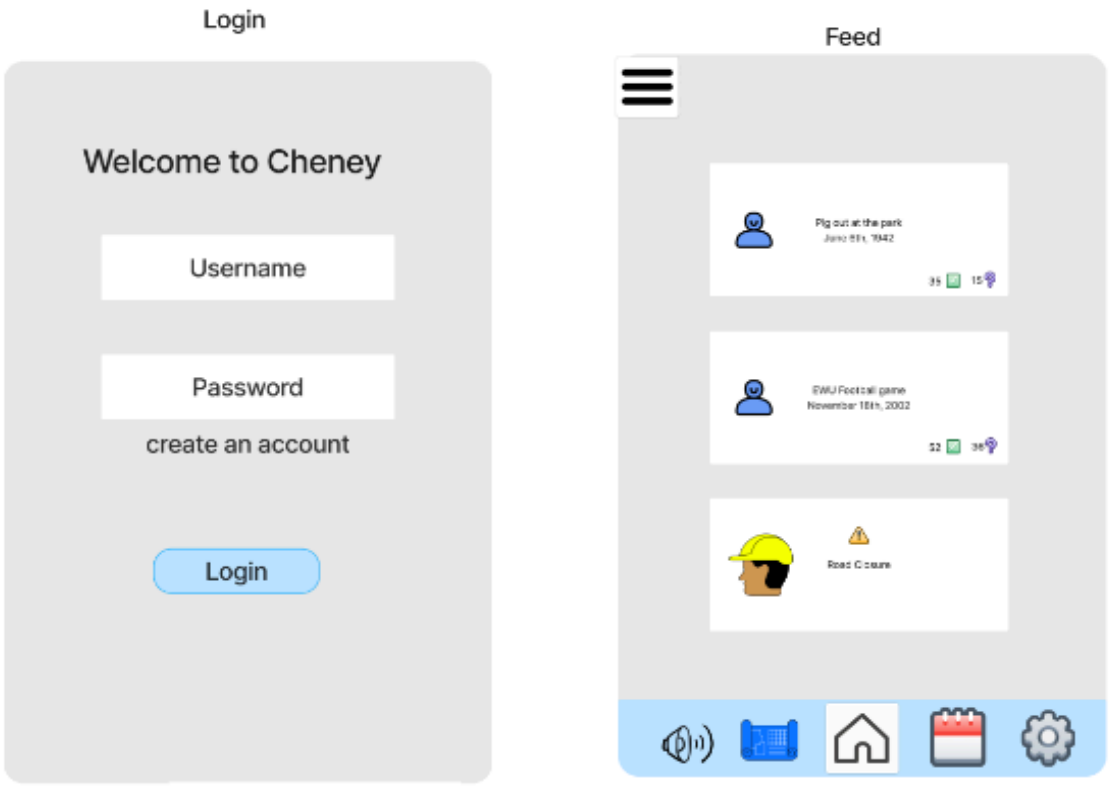


Figure 38: User Interface Mockup 1

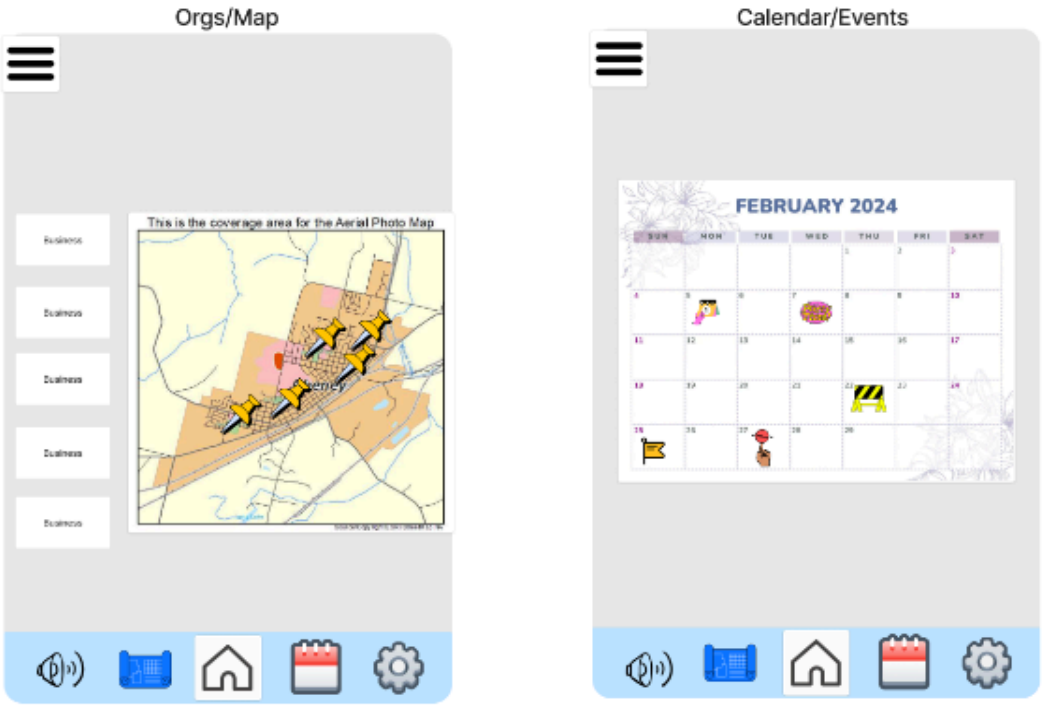


Figure 39: User Interface Mockup 2

Figure 40: User Interface Mockup 3

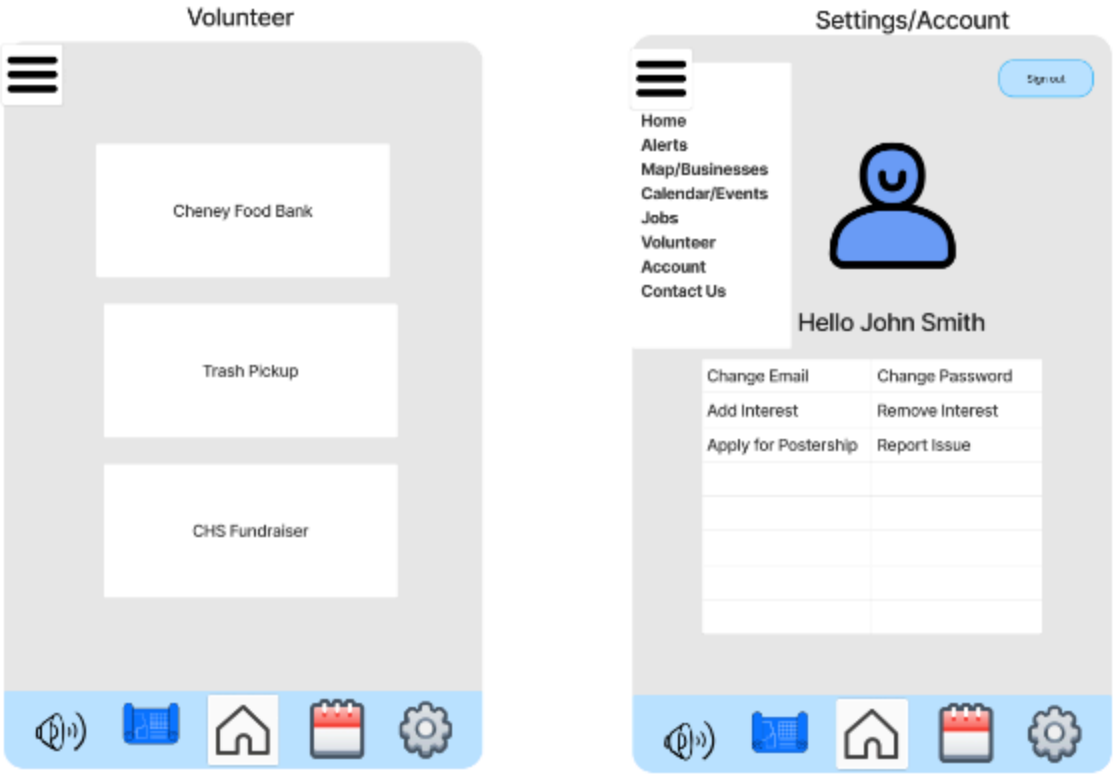


Figure 41: User Interface Mockup 4

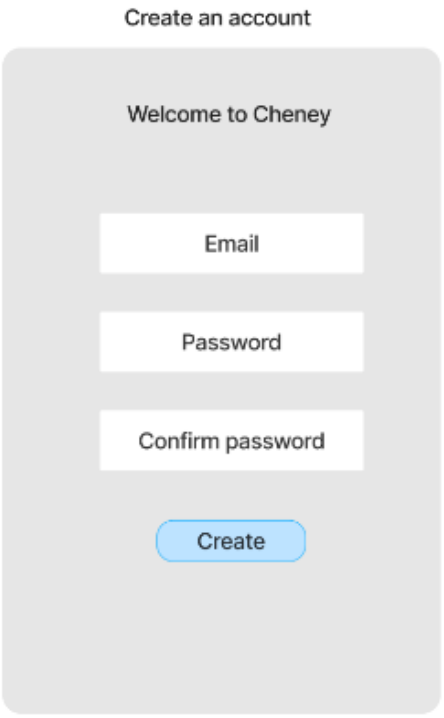
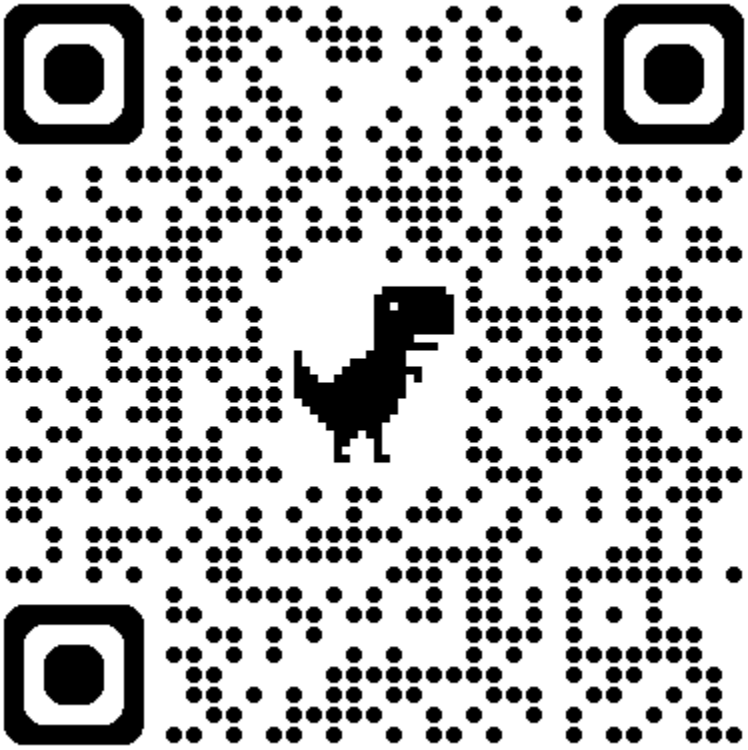


Figure 42: User Interface Mockup 5

Prototype:

https://www.figma.com/file/qmU7s8SlIUKPInZkhiznEE/Untitled?type=design&node-id=14%3A80&mode=design&t=sODwLvK7OHsPDCFR-1



CHAPTER 5

IMPLEMENTATION AND EXPERIMENTAL RESULTS

5.1 - EXPERIMENTAL SETUP

Hardware and Software Configuration:

- Platform: IOS & Android
- IDE - Visual Studio Code
- Front End: Flutter and Dart
- Backend: Firebase
- Version Control: Github
- Testing: Android Emulator
- OS: Windows, but one developer had Mac in addition

Network Infrastructure:

- Wifi
- Cloud functions via Firebase
- Internet connection to Firebase

Experimental Procedures:

- Derive task from client requirement or known bug/issue
- Update Kanban board and make new branch
- Complete codework to satisfy requirement
- Self Test
- Create PR and notify team
- Get feedback from team via PR
- Once feedback implemented or contested, merge branch into the working shared branch

Data Collection Instruments:

- Firebase logs for usage and cloud functions
- Feedback/output from Visual Studio Code
- Feedback/output from Github
- Peer feedback from PRs

5.2 - EXPERIMENTAL ANALYSIS

The objective of our experimental analysis was to fulfill the requirements of our client, ensure the app was functional and free of bugs, and to keep the upkeep cost as low as possible. We fulfilled the requirements by placing ourselves in an account matching the role/user of the user stories, and ensuring that from that perspective, we had the capability to complete the condition to satisfy the requirement. We ensured the app was bug free by testing each other's versions via Github and pull requests, and we often tested the app ourselves when using the app on our own during development. We also tried to keep the upkeep costs of Firebase low by analyzing the usage reports during our testing and estimating that usage rate applied to the population of Cheney, along with incorporating expected usage among the public.

The vast majority of our quality assurance was through peer review, but we also regularly met with our client, Maria, in order to get her feedback and to ensure that the app was becoming what she had wanted.

We fulfilled the vast majority of our requirements, some still in progress such as iOS integration. This was due to complications of acquiring a developer license with Apple, to which we tried to work out with the school; however, the school's solution did not work out in the end. We have taken matters into our own hands and have purchased a license for ourselves and we are currently waiting on approval to begin the integration process.

5.3 - WORKING OF THE PROJECT

5.3.1 - PROCEDURAL WORKFLOW

1) Derive task from requirement or bug
2) Create branch in github to separate work from current build
3) Code to fulfill the requirement or remove the bug
4) Self review to ensure the code works
5) Set up a pull request from the working branch to the current build
6) Someone else will review the pull request and either approve it or ask for changes
7) Repeat steps 3-6 until it is approved
8) Merge the pull request into the build and resolve any merge conflicts

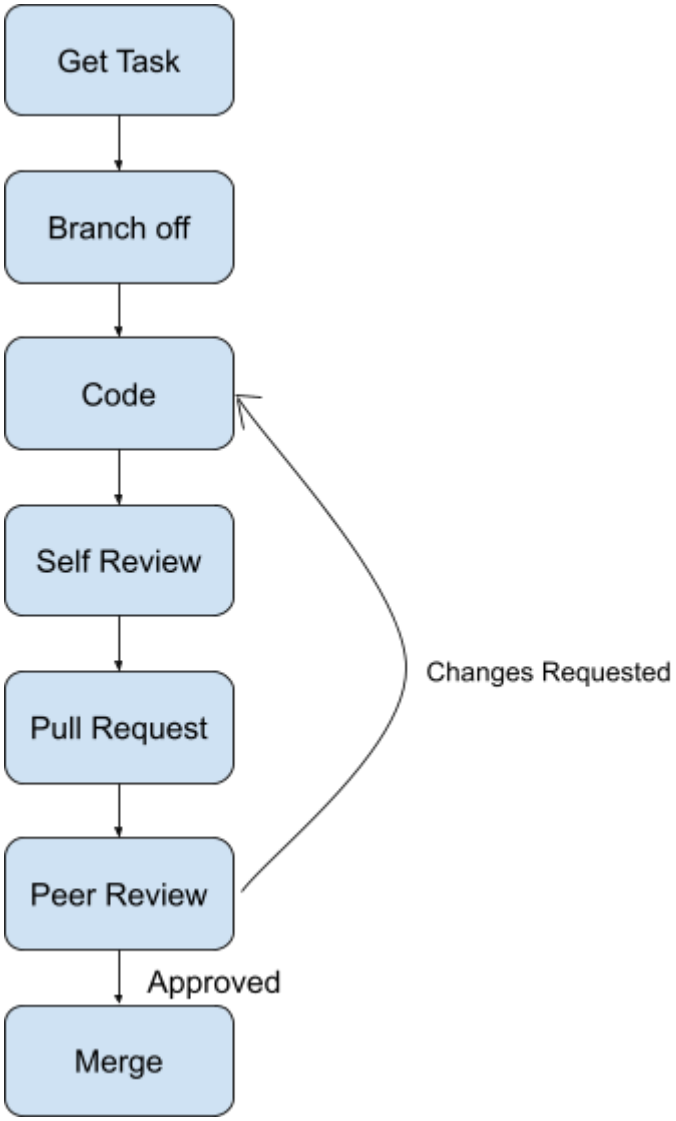


Figure 43: Procedural Workflow

5.3.2 - ALGORITHMIC APPROACHES USED

Data Management and Storage:

Firebase's Firestore was used for our data management and storage, the collections we had are:

- Users: Each has a(n) email, isBusiness, isPending, name, notification token, profile picture, saved posts, tags, tier, user ID, and username
- Tags: Each tag has a list of emails with the users who have the tag in their "tags" collection
- Businesses: Each has a(n) about, address, business hours, coordinates, email, name, pfp, and phone number
- Posts: Each has a(n) body, when created, header, interest count, post ID, tags, timestamp, title, type, and user
- Review Posts: Same as Posts, but ones that haven't been approved yet

User Authentication and Authorization:

Firebase's Authentication was used for our users, in connection to the users table in Firestore. Each user has an identifier (their email), and it supports email, google, or apple sign in. We originally planned to support all three, but due to the apple errors, we may revert to just email. Then each user also has when they were created, when they last signed in, and their user ID. Authentication supports email verification, password changes, SMS verification, and is encrypted via "HTTPS and logically isolates customer data".

Real-time Updates and Synchronization:

While we didn't use Firebase's Real Time Database, we opted to use Firestore because it was more simple, and it still updates very rapidly (within seconds), and the updating of our app wasn't that critical. Our pages use a stream builder, and use a classic pull-down-to-refresh to update the page, and any updates to the database are often reflected within a few seconds, as well as notifications from the cloud code

Cloud Functions and Server-side Logic:

We have 3 cloud functions, two of them aid in the process for deleting a user and removing them from our database, but the last one triggers when a post in the database is updated or a new one is created. It first grabs the post information from the posts collection, and then if it is an alert, it sends a notification to everyone, if it is a job, it doesn't send a notification, and if it is a post, it sends a notification to people if one of the post's tags are in that person's interested tags. Based on that, it queries the user collection to get the users and their notification tokens, and then sends the notifications.

Performance Optimization:

Since Firebase is the main cost of upkeep, our main objective was to keep the usage of that to a minimum. The main aspect of our Firebase usage is calls to Firestore to get/post new data, so our main method of reducing that was to minimize the amount of calls we make, by reducing the amount of lines of code that do said call, resulting in more complicated/length code, but less lines/statements. We also implemented manual refreshing of the pages, so that by not having it be "live", the amount of pulls from the database can be reduced

without worsening the user experience.

5.3.3 - PROJECT DEPLOYMENT

Google Play store:

1. Ensure all dependencies in your pubspec.yaml are up-to-date.
2. Build app in console with command “flutter build apk --release”
3. Create a new application in the Google Play Console.
4. Generate the release APK or App Bundle and upload it to the Google Play Console.
5. Fill in the necessary details like app description, screenshots, app icon, etc.
6. Submit the app for review.

iOS app store:

1. Ensure all dependencies in your pubspec.yaml are up-to-date.
2. Open the ios directory of your Flutter project in Xcode.
3. Provisioning Profile: Ensure you have a valid provisioning profile. This can be managed in the Apple Developer account.
4. App Version and Build: Update the version and build number in Xcode under the General tab.
5. Run the following command to build the release version of your app: “flutter build ios --release”.
6. In Xcode, select Product > Archive to create an archive of your app.
7. Once the archive is created, the Archive Organizer will open. Select the archive and click Distribute App.
8. App Store Connect: Ensure you have an Apple Developer account and have created an app record in App Store Connect.
9. Follow the instructions in Xcode to upload your app to App Store Connect.
10. Fill in the necessary details like app description, screenshots, app icon, etc.
11. Submit the app for review.

Tips:

- Testing: Test your app thoroughly on both platforms before submission.
- Compliance: Ensure your app complies with both Google and Apple's guidelines.
- Documentation: Keep track of all necessary documentation and requirements specific to each platform.

5.3.4 - SYSTEM SCREENSHOTS

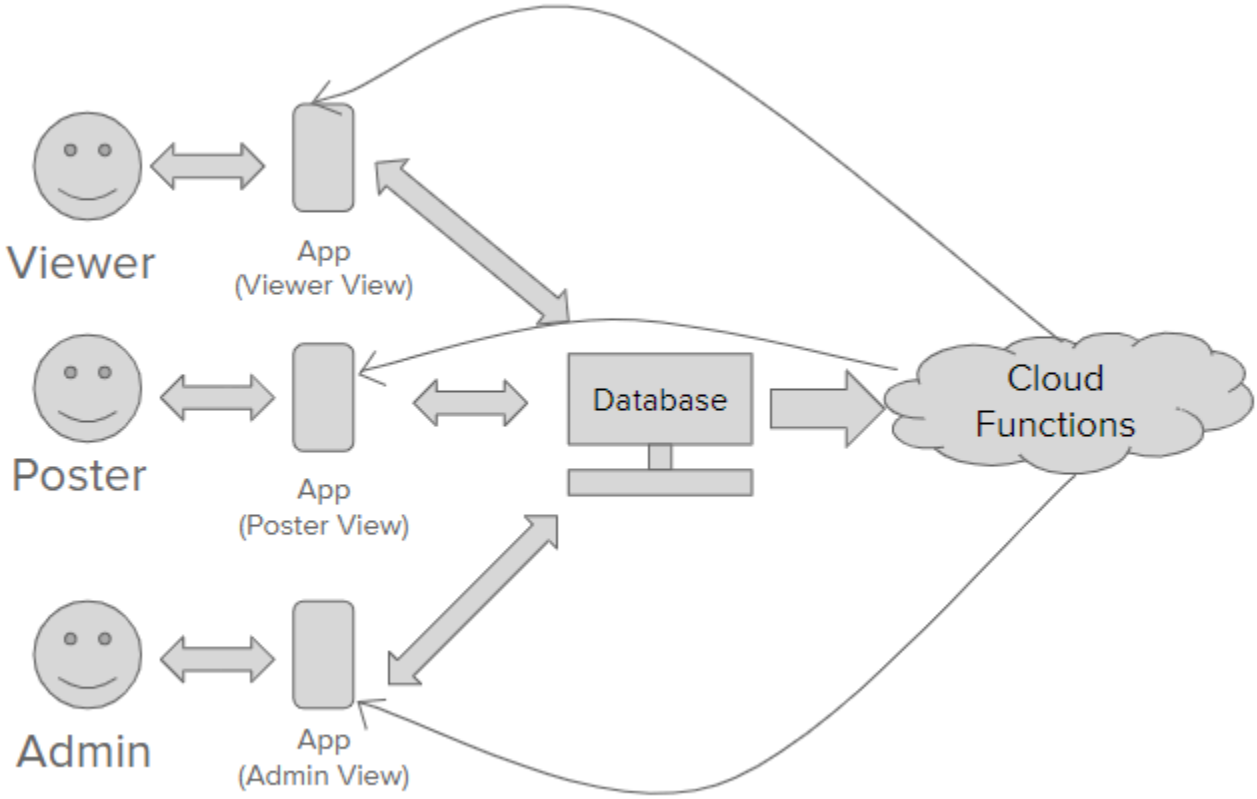


Figure 44: System Architecture Diagram

5.4 - TESTING PROCESS

5.4.1 - TESTING PROCESS

To test our features continuously, we do so via peer review on Github, by reviewing the pull requests on there which highlight the differences between the current build and what the developer wants to change. In the pull request, the requesting developer will also describe the purpose of the changes and what they're meant to do, as well as instructions on how to verify and test the changes. Then, once a peer review is completed and the reviewer approves, the pull request will be marked as approved and it will be applied to the current build.

5.4.2 - FEATURES TO BE TESTED

The features to be tested will essentially be the client requirements, as these requirements are what inspires the features of the app. But to reiterate them:

- Admins must be able to create posts and edit any post by filling out the specific fillable form or pressing the edit button on any post.
- Admins must be able to delete any post by pressing the delete button on any post.
- Any user that can create posts must be able to add tags to their post by filling out the specific fillable form.
- Any post tagged with "Event" must have buttons to allow users to RSVP.
- Admins must be able to upgrade/downgrade to the privileges of another account.
- Admins must be able to delete other accounts.
- Viewers must be able to choose pre-determined tags regarding information that interests them by going to account settings.
- Users must be notified when a post that contains a tag they currently have selected is posted via a push notification.
- All Users must have a personal feed on the home page that will contain all recent posts with tags that they currently have selected.
- All Users must have access to a global feed that contains all recent posts regardless of tags by deselecting all of the tags.
- All Users must be able to sort posts by multiple metrics(most recent, tags, title, etc.) on the home page
- Viewers must be able to fill out a Specific Fillable Form (SFF) to share

possibly useful information to an Admin.

- All Users must be able to create an account using a unique username and a password.
- All Users must be able to change their password at any time in the account settings.
- All Users must be able to delete their own account in the account settings.
- Approved Posters must be able create posts and edit their own posts by pressing the edit button on their post.
- Approved Posters must be able to delete their own posts by pressing the delete button on their post.

5.4.3 - TEST STRATEGY

We mainly practiced behavior-driven-testing due to the many user interface and visual components of our app. While this isn't as concrete as test-driven-development, it was the most practical and applicable to our scenario, and still satisfied all of our testing needs since the requirements given are mostly based on end behavior and not the internal workings. We derived our behavior driven testing from our requirements and user stories, and then testing the behavior through peer review of our code.

5.4.4 - TEST TECHNIQUES

Our technique when testing each other's code was mainly done by placing ourselves in the role of the story or requirement, i.e being a user, approved poster, or admin, and then attempting to complete the requirement ability, following the directions given by the person who made the code. The criteria for the code being approved is:

- The requirement should be fulfilled
- The code should be error and bug free
- The directions should be intuitive, where a common user could complete it without the instructions
- The code shouldn't undo the satisfaction of any other requirements
- The internal code should also be intuitive, efficient, and minimize cost when applicable

5.4.5 - TEST CASES

The test cases for most of the pull requests will be altering the account that is logged in, mainly to test the different tiers. The view for a lot of the features will vary between a Viewer, Poster, and Admin, mainly with more sections and features being visible to the latter roles. The other variation we tested was what tags the user had selected, to test things like notifications being based on only the selected tags, as well as edge cases such as if the user has no tags selected.

5.4.6 - TEST RESULTS

While the pull requests were sometimes not perfect, most often they were remedied with a few fixes,

and were caught either by the initial review, or later on in regular app use and flagged by our developers. As far as the requirements being fulfilled, nearly all of them were fulfilled and passed our tests, and the few that weren't were pardoned by our client, as they weren't as necessary and vital as the other requirements were. Our behavior-driven-testing via peer reviewed pull requests was overall successful, and we've ensured that our current build is always satisfactory.

5.6 - VALIDATION OF OBJECTIVES AND REQUIREMENTS

Of the requirements laid out in section 2.2, we fulfilled all but 2 of them for our client. Upon speaking with her though, she was okay with the fact that these were not completed, as they weren't as important as the other requirements, and were also covered by some of the other features of the app.

As far as our objectives of section 1.4, we will have to wait and see the actual impact and usage of the app, but we have completed everything necessary to allow and encourage these objectives to flourish.

To address these requirements and objectives individually:

- **R1(U1,2) Admins must be able to create posts and edit any post:** Admins have an edit button on every post, and if an admin fills out the specific fillable form, it is auto-approved and live.
- **R2(U3) Admins must be able to delete any post:** Admins have a delete button on every post.
- **R3(U4) Any user that can create posts must be able to add tags to their post:** Any user can fill out the specific fillable form, including tag selection, that then awaits admin approval before becoming public.
- **R4(U5) Any post tagged with "Event" must have buttons to allow users to RSVP:** Every even has two buttons, "Attending" and "Interested", and the names and count of those who have selected the button is recorded on the event.
- **R5(U6) Admins must be able to upgrade/downgrade to the privileges of another account:** Admins are able to search accounts and change their tier to/from Admin, Viewer, or Poster.
- **R6(U7) Admins must be able to delete other accounts:** On the account search previously mentioned, they are also able to delete the account.
- **R7(U8) Viewers must be able to choose pre-determined tags regarding information that interests them:** Under account settings, every user is able to edit their interested tags.
- **R8(U9) Users must be notified when a post that contains a tag they currently have selected is posted:** When a normal post including a tag that they have marked, a push notification is sent to their phone alerting them of that post.
- **R9(U10) All Users must have a personal feed that will contain all recent posts with tags that they currently have selected:** The home page shows all recent posts that are alerts, or that are posts with tags that they have marked.
- **R10(U11) All Users must have access to a global feed that contains all recent posts regardless of tags:** This requirement was not strictly fulfilled, but is possible if a user selects that they are interested in all of the tags.
- **R11(U12) All Users must be able to sort posts by multiple metrics(most recent, tags, title, ect.):**

This requirement was also not fulfilled, but was supplemented by the feature of being able to save posts with a button on every post, so that a user can return to specific posts without having to scroll. The home page is sorted by the current tags and most recent, but otherwise the sorting is not changeable.

- **R12(U13) Viewers must be able to fill out a Specific Fillable Form (SFF) to share possibly useful information to an Admin:** Users are able to fill out a specific fillable form to draft a post that gets shared to the admins.
- **R13(U14) All Users must be able to create an account using a unique username and a password:** This was fulfilled with Firebase Authentication, and are able to do so under "register" on the login page.
- **R14(U15) All Users must be able to change their password at any time:** Users are able to do so on the account settings page.
- **R15(U16) All Users must be able to delete their own account:** Users are able to do so on the account settings page.
- **R16(U17,18) Approved Posters must be able create posts and edit their own posts:** When a poster fills out the specific fillable form, it is auto-approved and made public, and when a poster sees a post that they have made/submitted, there is an edit button similar to the admins'.
- **R17(19) Approved Posters must be able to delete their own posts:** Similar to the admins, there is a delete button next to the edit button that only appears when viewing a post that the poster has made/submitted
- **Enhance Community Engagement**: Our app enhances community engagement by offering the opportunity to have a public source of information about local events.
- **Improve Information Dissemination**: Our app improves information dissemination by minimizing what is allowed on the app and strictly keeping it to only important information that is approved by admins or from approved sources.
- **Support Local Economy and Opportunities**: Our app supports the local economy and opportunities by giving local businesses a platform to share about events and information regarding their business, as well as a place to advertise any job postings that they are hiring for.
- **Enhance Public Safety and Awareness**: Our app enhances public safety and awareness by having an outlet for immediate and accurate updates from local authorities so that people can stay informed about any situations that may affect them.

- **Facilitate Direct Communication**: Our app facilitates direct communication by having a shared platform where sources can reach audiences that they both utilize, and the audience can be sure that the information is coming directly from the source.
- **Ensure Accessibility and Inclusivity**: Our app ensures accessibility and inclusivity by monitoring posts through approval and restriction to ensure that no hateful or harmful posts are made on our app.
- **Promote Verified and Trustworthy Information**: Our app promotes verified and trustworthy information by having posts that come directly from the source, and that are verified and approved by our admins as well.
- **Support Sustainability and Scalability**: Our app supports sustainability and scalability by ensuring that our app is bug and error free, and also minimizes costs and storage as much as possible in order for the app to remain easily kept up and usable.

CHAPTER 6

CONCLUSIONS AND FUTURE DIRECTIONS

6.1 - CONCLUSIONS

After six months of research, planning development, and review, our project is coming to a close. We embarked on an ambitious journey to create a mobile application designed to enhance community awareness and engagement. The primary goal of our app is to provide timely, relevant, and accurate information about local alerts and events, empowering residents to stay informed and connected with their neighbors.

From the inception of this project, we prioritized user-centric design and functionality. We researched various platforms and frameworks to fit our needs, and met with our client often to ensure that. Our iterative development process allowed us to adapt quickly to feedback and improve the app's performance and usability continuously.

The app serves as a centralized platform where users can receive critical alerts such as weather warnings, emergency notifications, and community announcements. Additionally, it promotes local engagement by highlighting upcoming events, public meetings, and cultural activities. By combining these functionalities, we aim to foster a more informed and cohesive community, which our client has struggled to do using available means.

Throughout the development phase, we faced various challenges, including technical issues, financial barriers, and time constraints. However, through collaboration, innovation, and perseverance, our team successfully navigated these obstacles. The culmination of our efforts is an app that not only meets but exceeds our initial objectives, which our client is satisfied with.

In conclusion, our application stands as a symbol of our hard work, and we are happy to have something to show for it. We believe that this tool will be instrumental in helping residents navigate their daily lives more effectively, stay aware of their environment, and participate actively in local events. We will stay in contact with whomever takes charge of the app, and we hope that Maria and future developers can make it better than we ever imagined.

6.2 - ENVIRONMENTAL, ECONOMIC, AND SOCIETAL BENEFITS

The development and deployment of our mobile app will help foster positive impacts across environmental, economic, and societal dimensions.

From an environmental standpoint, our app introduces a digital alternative to traditional paper-based methods of spreading local alerts and events. By minimizing the use of physical resources such as paper and ink, we contribute to reducing the carbon footprint associated with printing and distribution processes. We promote a paperless approach that aligns with sustainability goals, reducing waste and conserving natural resources.

On an economic level, the efficiency and accessibility offered by our app translate into tangible benefits for both users and local communities. By streamlining the communication process between businesses and users, we facilitate the dissemination of critical information in a cost-effective manner. This efficiency not only saves time and resources but also enhances productivity and engagement within the community. Moreover, by providing a platform for local events and alerts, our app contributes to the promotion of local businesses and initiatives, driving economic activity and fostering community growth.

Societally, our app serves as a catalyst for community engagement, empowerment, and resilience. By delivering timely and relevant information about local events and alerts, we empower users to stay informed and connected with their surroundings, fostering a sense of belonging and collective responsibility. This increased awareness and engagement contribute to building a stronger, more resilient community that is better equipped to respond to challenges and opportunities. Furthermore, by facilitating communication between admins, approved posters, and users, our app promotes transparency, inclusivity, and democratic participation, fostering a culture of collaboration and mutual support within the community.

In conclusion, our mobile app offers a multitude of environmental, economic, and societal benefits, ranging from sustainability and cost-effectiveness to community engagement and resilience. By harnessing the power of technology to streamline communication and promote information sharing, we strive to create positive impacts that extend beyond the digital realm, enriching the lives of individuals and communities alike.

6.3 - REFLECTIONS

Throughout the journey of developing our mobile app, our team encountered many challenges and opportunities for growth. During this experience, several key learnings stood out:

First, the collaborative nature of the project emphasized the importance of effective teamwork. Building a functioning public application from scratch required coordination and communication among team members. We had to learn to leverage each other's strengths, delegate tasks efficiently, and maintain open channels for feedback and support.

Secondly, the process of designing and implementing both front and back-end components provided valuable insights into the intricacies of app development. From user interface design to database management, we gained hands-on experience in translating concepts into real features while ensuring smooth functionality throughout the app.

Additionally, consulting with our client and maintaining a continuous feedback loop proved vital in aligning our development efforts with their vision and requirements. This iterative approach not only ensured our client’s satisfaction, but also fostered a deeper understanding of the importance of client collaboration in delivering successful outcomes.

Moreover, navigating the complexities of version control was a significant learning curve for our team. From mastering Github workflows to resolving conflicts, we gained proficiency in managing code changes effectively, which proved instrumental in maintaining project stability and scalability.

Finally, the process of getting our mobile app published on mainstream app stores introduced us to the regulatory and technical nuances of the digital marketplace. While Android was fairly streamlined, we hit many technical and financial barriers with Apple. From meeting submission guidelines to optimizing app performance, we learned to navigate these challenges with diligence and attention to detail.

In summary, our journey in developing this mobile app has been a profound learning experience, encompassing teamwork, technical proficiency, client collaboration, version control, and app deployment. These learnings not only enriched our individual skill sets but also strengthened our collective ability to tackle future projects with confidence and expertise.

6.4 - FUTURE WORK

While we managed to complete all of the core requirements, there is definitely always room for improvement. Some ideas that we've had, things we abandoned, or things that we were unable to complete were:

- Pictures with posts
- More cost/upkeep reduction
- Advertising - Either for app upkeep or local businesses to advertise
- Refactoring to make it applicable/reusable to other towns/cities
- Android notification icon
- Apple sign in
- Color Coding (Map pins, posts, tags, etc)
- Home feed with all posts
- Sorting/searching the posts

Going forward with Maria, we plan to transfer all ownership over to her, and while she may want to take the app as is, we've mentioned the possibility of offering this project over to the class of 2025 students to improve upon. We will also happily keep in contact with Maria to help with any questions or quick bug fixes, and a few of us are remaining in Cheney as well, and may end up using our own app. This has been a meaningful and fulfilling project, and we can't wait to see how Cheney likes it.