Software Requirement Specification

Where is the Power

Introduction

Where is the power, is a mobile and desktop app that navigates users through these dark times in South Africa. Loadshedding is inconveniencing a lot of South Africans, especially on the roads. Where is the power aims to assist South Africans avoid traffic, plan their day, find areas that have electricity, inform communities of surprise power outages in local areas and more.

By means of a map, areas will be coloured in on the map to indicate the status of loadshedding and any other power related issues. Navigating routes that are not impacted by loadshedding is a must, especially when trying to get home. Statistics will be used to identify what the areas' average uptime is and more.

User Story

Normal user

- This user interacts with the maps.
- Able to navigate through roads, where loadshedding is not happening.
- View the statistics of areas around them.
- See reports(e.g. Stolen cables) that are happening in Areas.

Registered User

- Everything a normal user can do.
- This user can report problems in an area, while a normal user can not.
- Receives push-notifications (mobile).

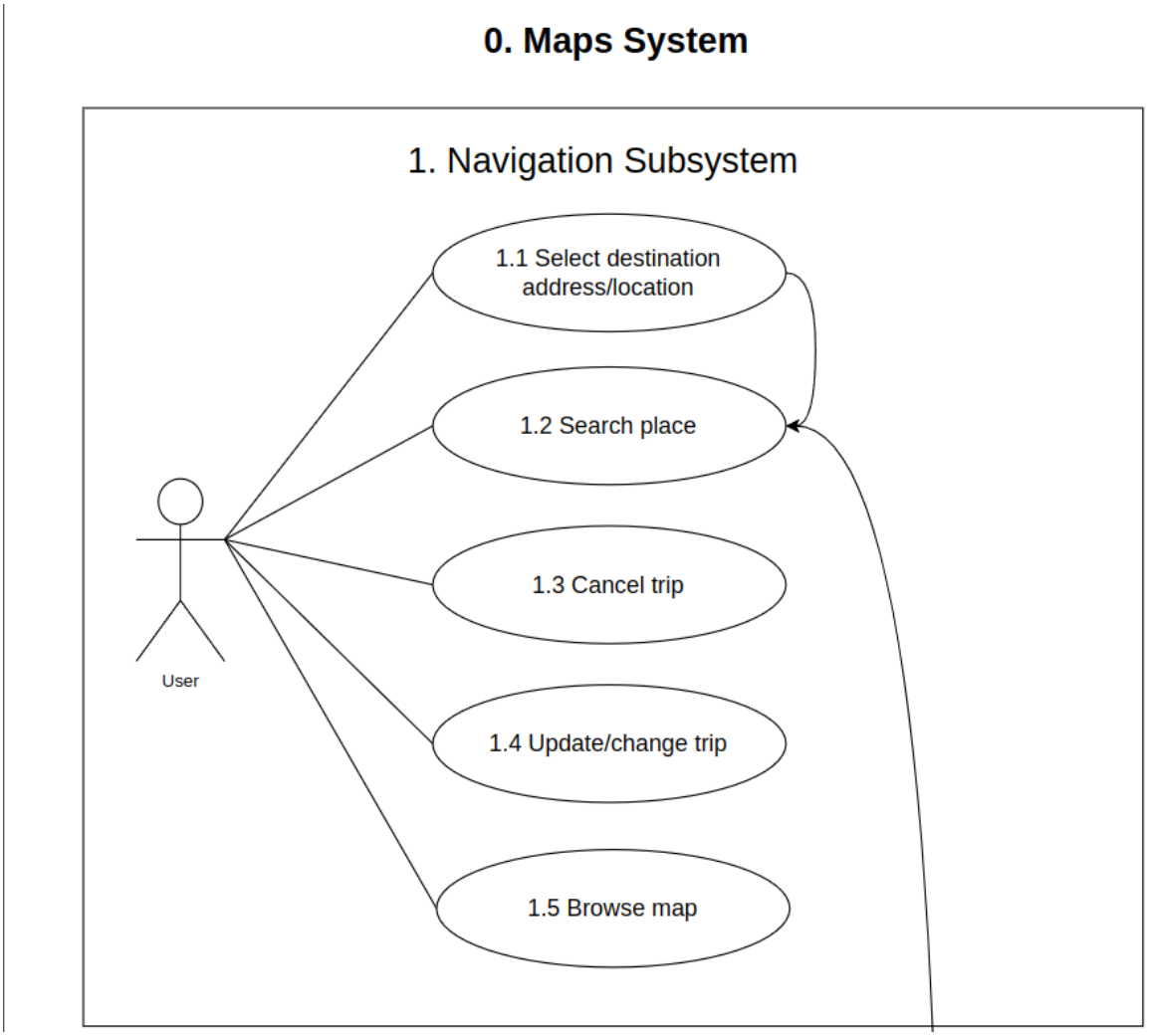
Example user Stories

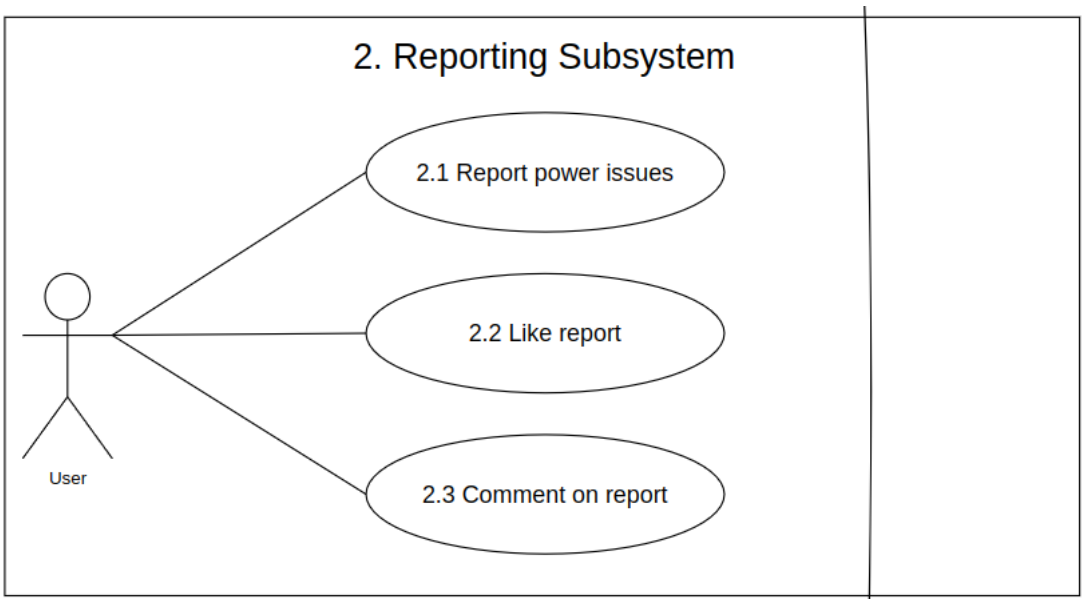
- As a a commuter, who has to travel from one place to another, I want an application that finds the optimal route despite load shedding, so that I don’t have to manually look at the load shedding areas and schedules and plan out my route manually. Given the destination location, and the time of travel, the user may find the optimum route, avoiding load shedding. When the user clicks navigate, then they are shown a map with a search bar at the top, then the user can search for a place.
- As a member of the community, I want to make awareness to any cables stolen, substations, blown or any other related power outage reasons, so that I can inform others and they can plan out their routes with more insight using the app. Given the user has something to report, when they click the report button, then they can select the type of report issue and fill in other details and send off a report.
- As a member of the community, I want to know the stages of load shedding for a particular area,so that I don’t have to use a separate app to do so. Given the user’s

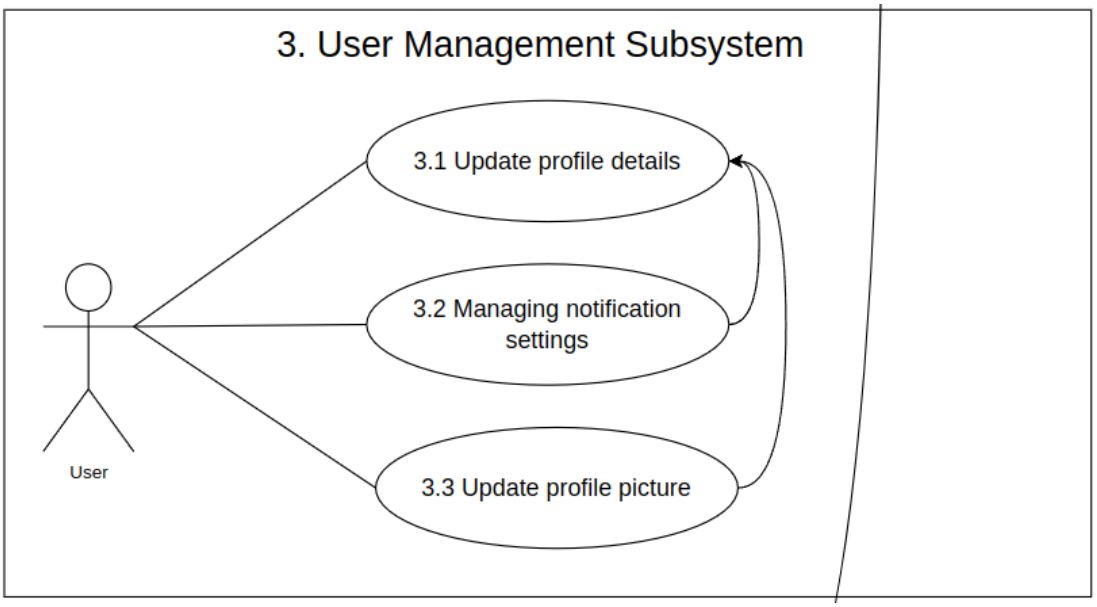
desire to view the schedules, when the user clicks the schedule tab, then the user will be shown all the schedules for a particular area they wish to see.
- As a South African, I would like to see on my map which areas have loadshedding so I can plan my trip.
- As a South African, I would like to report on issues in my area regarding stolen cables such that I can inform my community on outages
- As a user I would like to save places on the map so that I can easily access them

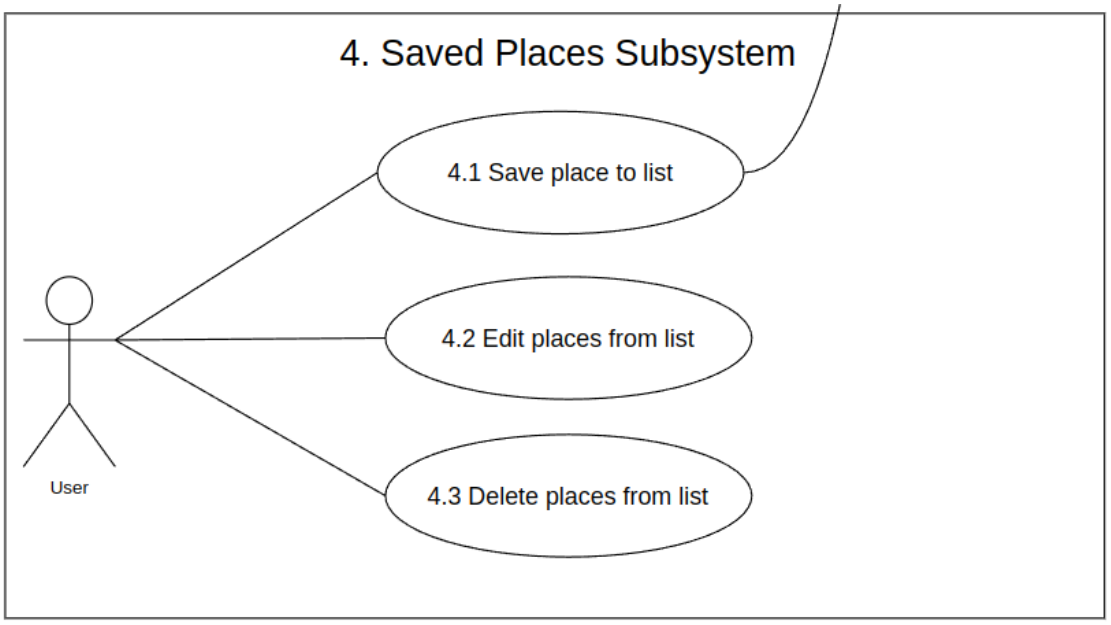
Functional Requirements

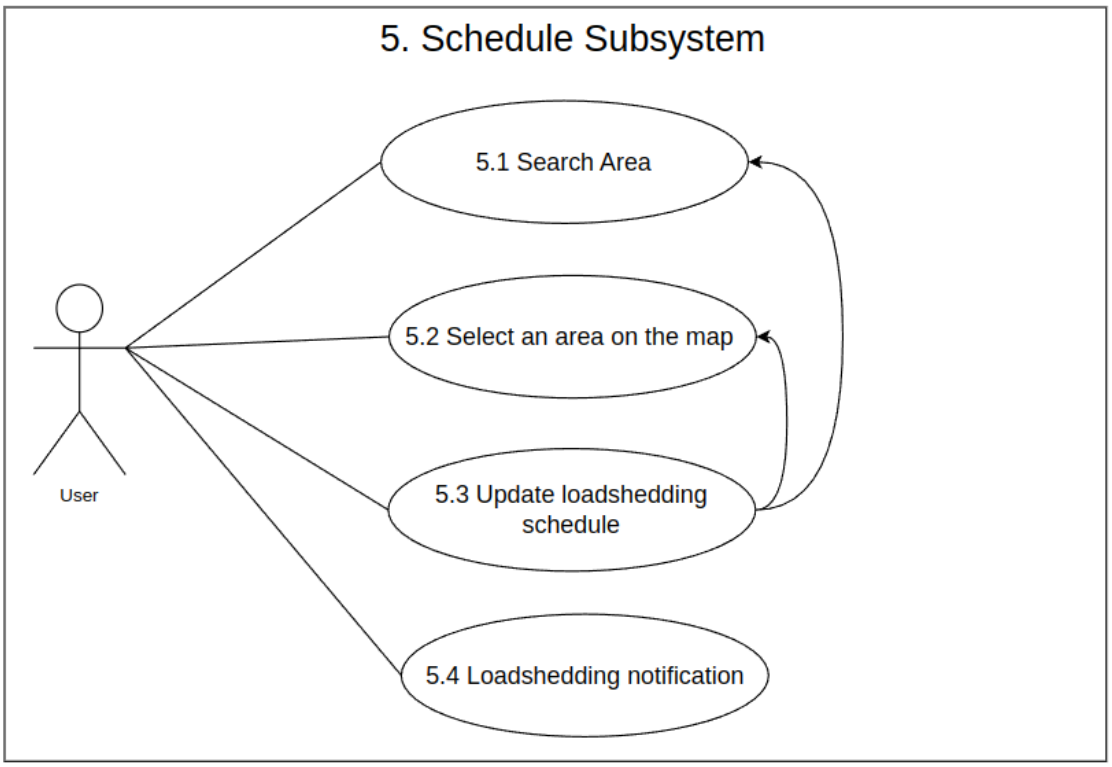
Use Cases

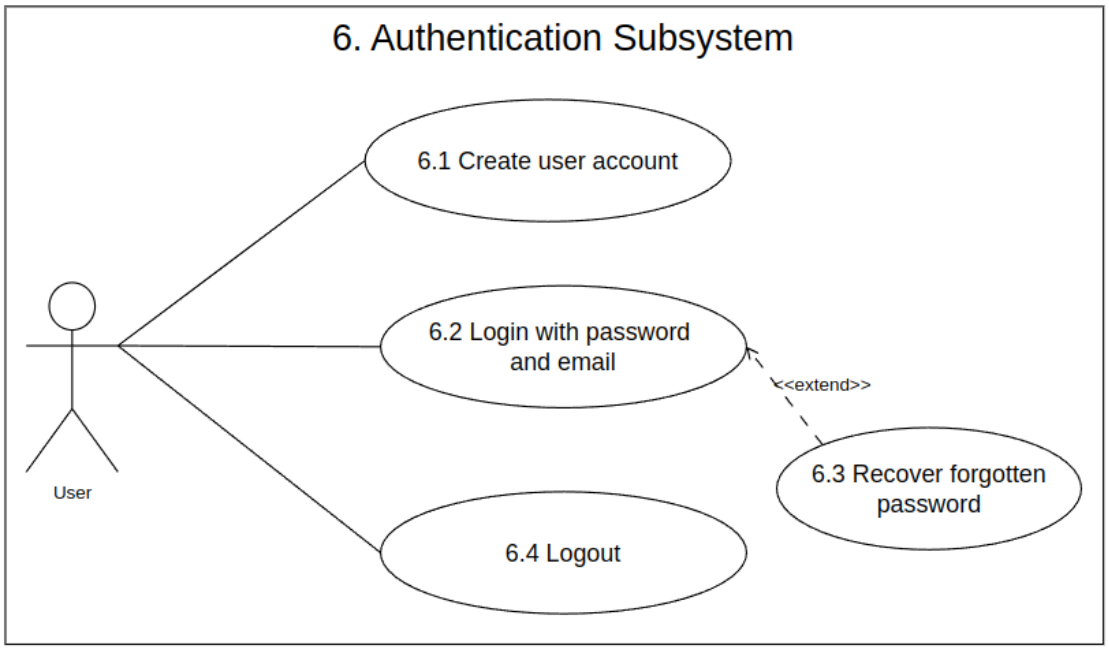



2. Reporting Subsystem
2.1 Report power issues
2.2 Like report
2.3 Comment on report
User


3. User Management Subsystem
3.1 Update profile details
3.2 Managing notification settings
3.3 Update profile picture
User


4. Saved Places Subsystem
4.1 Save place to list
4.2 Edit places from list
4.3 Delete places from list
User


5. Schedule Subsystem
5.1 Search Area
5.2 Select an area on the map
5.3 Update loadshedding schedule
5.4 Loadshedding notification
User


6. Authentication Subsystem
6.1 Create user account
6.2 Login with password and email
<<extend>>
6.3 Recover forgotten password
6.4 Logout
User

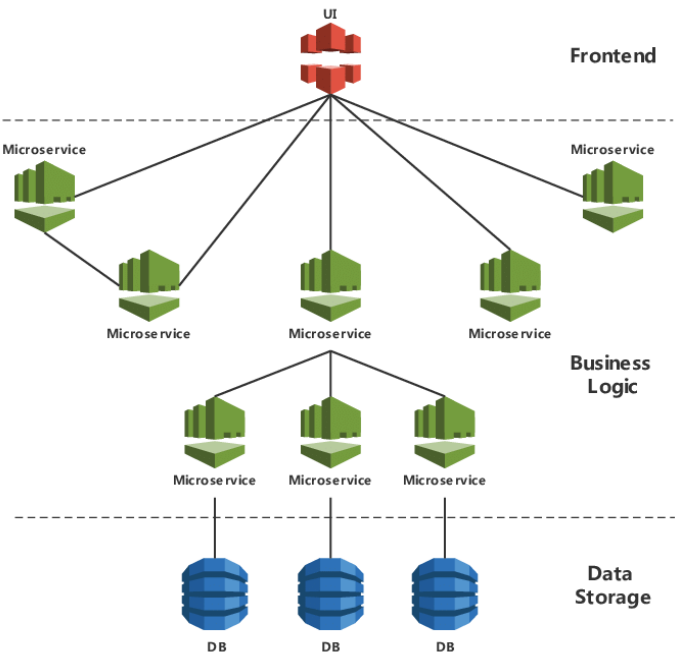
Requirements

- Open maps with loadshedding colours overlay
    - Colour overlay
    - Getting load shedding times for areas
    - Average uptime of an area
    - Click on area
    - Display load shedding INFO
    - Transition colour area for when load shedding
- Report power issues for an area
    - Report area down
    - Show where power is having trouble unrelated to Load shedding
    - A report is valid for 30 minutes and requires a new report
- Navigation
    - ETA based on loadshedding
    - Route alterations to avoid load shedding
    - Plan trip
    - Cancel trip
    - Updating trip
- Statistics
    - Display Daily loadshedding graph
    - Display weekly loadshedding graph

Architectural Patterns

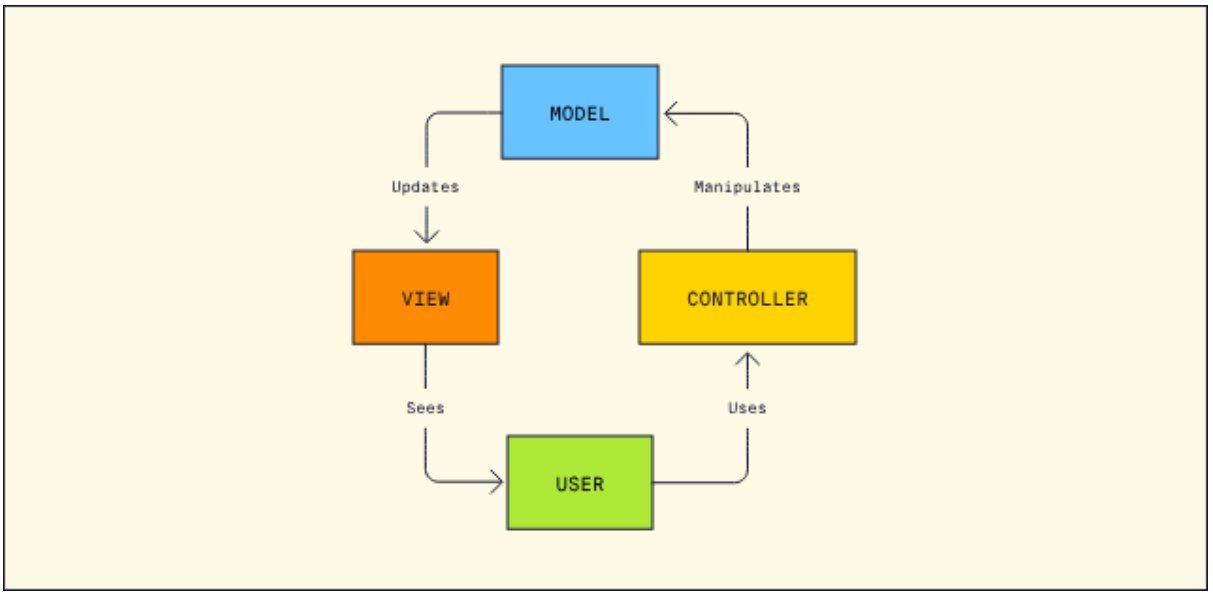
Microservices

The application is divided into microservices, each responsible for specific functionalities



Mode-View-Controller

MVC is mainly used for ease of control when working with GUIs (Frontend related). This will decompose the frontend by having models, views and controllers. This decomposition allows for views and controllers to be reusable and easily testable.



Multitier Architecture

Multitier allows the system to be broken up into tiers. The three main tiers the application will be broken up into will be the presentation tier, logic tier and data tier. This Architecture allows for technologies to be swapped out easily at different tiers. This is great for future proofing and scaling where is the power app as new technologies arise.

Architectural Quality Requirements

Mobility

The application should work anywhere and at anytime with a connected network. The app should be able to work on a desktop and/or a mobile environment.

Usability

The app should be intuitive enough so that the user can easily use it with out the need of a tutorial. The task of selecting a place to navigate to should be easy for the user to do in a minimal number of steps. What does green overlay mean (no loadshedding), Design choices will determine the usability factor.

Reusability

Code should be written in a modular manner, so that code can be reused later in the architecture if need by. This will remove the need for redundancy.

Efficiency

The system should be able to load data without costing to much Processing time on the user side. Load times for data should be optimized by sending small packets of data at time.

Availability

An uptime of at least 90% is expected of the system.

Credibility

The user should be able to rely on the application to give accurate routes and accurate information on load shedding schedules. The system should check the loadshedding times from verified sources frequently (Every 30 min)

Security

The system should not be exposing data so easily. Sensitive data should be encrypted (Registered users)

Scalability

The system should easily be able to pick up more areas (easily) as more web scrapers get loadshedding data from different municipalities. Data should be normalised so that there is consistency when scaling up.

Architectural Constraints

Limit locations to the city of Tshwane

Due to the manual labour and domain knowledge required for spatial mapping, the system will cater for a sample size of South Africa. Time is also a factor as the durationg of the capstone project is limited.

Mapping data to spatial data

Not all the data we have will be represented because of the lack of spatial data. This leads to following point: suburb extensions would be excluded.

Exclude suburb extensions.

Example: instead of showing the load shedding schedules for Menlopark extension 3, we would only show Menlopark as a whole.

Out of date Data

All spatial map data used in the system uses a census that was conducted in 2011 (12 years dated).

Technology choices

Figma

To design wireframes and mockups for planning purposes.

Angular Ionic

Develop Front-End for mobile and desktop web apps

Rust Rocket

API, Reason for using Rust Rocket is guaranteed type safety and speed.

MongoDB

Database to store everything needed.

AWS (Amazon Web Services)

Will do all our hosting needs on a virtual Linux machine.

GitHub

Used for version controlling the repository.

GitHub Project Board

For project management, issue tracking and user stories.

Architectural Design and Pattern