MoGreen

Software Design Document

Version 2

February 19, 2019

Project Sponsor:
Ellen Vaughan and Brock Brothers

Team Mentor:
Ana Paula Chaves Steinmacher

Team Members:
Cassie Graham, Jennie Ryckman, Chase Mosteller, and Justin Shaner

NAU
NORTHERN ARIZONA UNIVERSITY

Table of Contents

1. Introduction

The NAU campus strives to be as green and sustainable as possible, but as the NAU student population grows so do instances of overflowing trash bins, full dumpsters, and campus litter. Basic maintenance problems occur more often as well, like broken sprinklers and blocked trash cans. There are few and poor mechanisms in place for students and community members to engage in caring for their campus themselves and alerting the appropriate facility managers for problems outside the scope of what any community member can handle.

This becomes more problematic as the size of the NAU community grows, as has been identified by Ellen Vaughan and Brock Brothers who lead the efforts for sustainability on the NAU campus. Ms. Vaughan was the manager of the Office of Sustainability here at NAU. Mr. Brothers is a supervisor and coordinator for NAU's Moving and Recycling. Everyday there are issues Mr. Brothers and his team deal with, but generally he and his team will not know if there is an issue unless they come across it in person and visually see the blockage, or someone reports it.

Currently, students can report minor issues by calling NAU Facilities or filling out a form using their website. Nearly everything done to keep the campus clean and maintained is done manually. There is not a system in place to accurately track when areas around NAU have been cleaned and there are not many tools for the NAU community to get involved. They are limited to calling or reporting via a work request made online, a feature which most are unaware of. Either way, the current reporting method is outdated, unintuitive, and most students will not make the time to report minor issues.

Our Capstone project is the "Clean My Campus" mobile application. This application will be developed by our four-person team. The mobile application proposed will make reporting minor issues easier for students and the general community and allow NAU facilities to receive and track these reports faster. It will also allow students and community members to take initiative in engaging with their communities by personally cleaning up litter. This application will help further NAU's sustainability efforts and keep our campus clean and green.

The key functional requirements for our mobile application are:

- The ability to view a map with designated "zones", colored to indicate their status
- Make quick reports about maintenance or litter problems that affects the color of zones
- Make full, more detailed reports about issues that require staff attention

The key functional requirements for our companion web application are:

- The ability to view a list of tasks that must be handled by staff, and all types of reports
- The ability to view a heatmap display of report traffic data, filterable by time period
- The ability to designate administrators who can access the web portal
- The ability for users to register accounts

Our key performance requirements are simplicity, usability, and reliability. Both the mobile application and web portal will have a limited number of pages for increased navigability. Average users will be able to make quick reports in under a minute, full reports in less than two and a full report with a photo included in less than 3. Our application is expected to be able to handle a minimum of 30,000 requests at once, which is enough for every NAU student, employee, and faculty member, plus some room for visitors.

Our environmental requirements describe our technical constraints, which are fairly few with regard to our minimal viable product (MVP). We are essentially constrained to using Google Maps since it has few viable competitors. Though Google Maps is our best option, we will be hard-pressed to handle any issues we may have with it since there are few comparable replacements.

Other environmental requirements for our extended solution include the development of an iOS compatible application, and the implementation of gamification features. These are less significant than for our MVP, but we must be mindful of them as we develop our MVP. For gamification features, we have to be careful to designate database space for the creation of user teams and design our report mechanisms so that it will be straightforward to assign some sort of point values to user actions. As far as a non-Android implementation goes, we do not anticipate having the time to implement any and recommend that a future group do so. However, there may be some environments where the languages our project is created in is reusable. Other OS implementations may share the same database and will be compatible with our existing web portal.

Following is a specification of the technologies and tools we'll be using to implement this project.

2. Implementation Overview

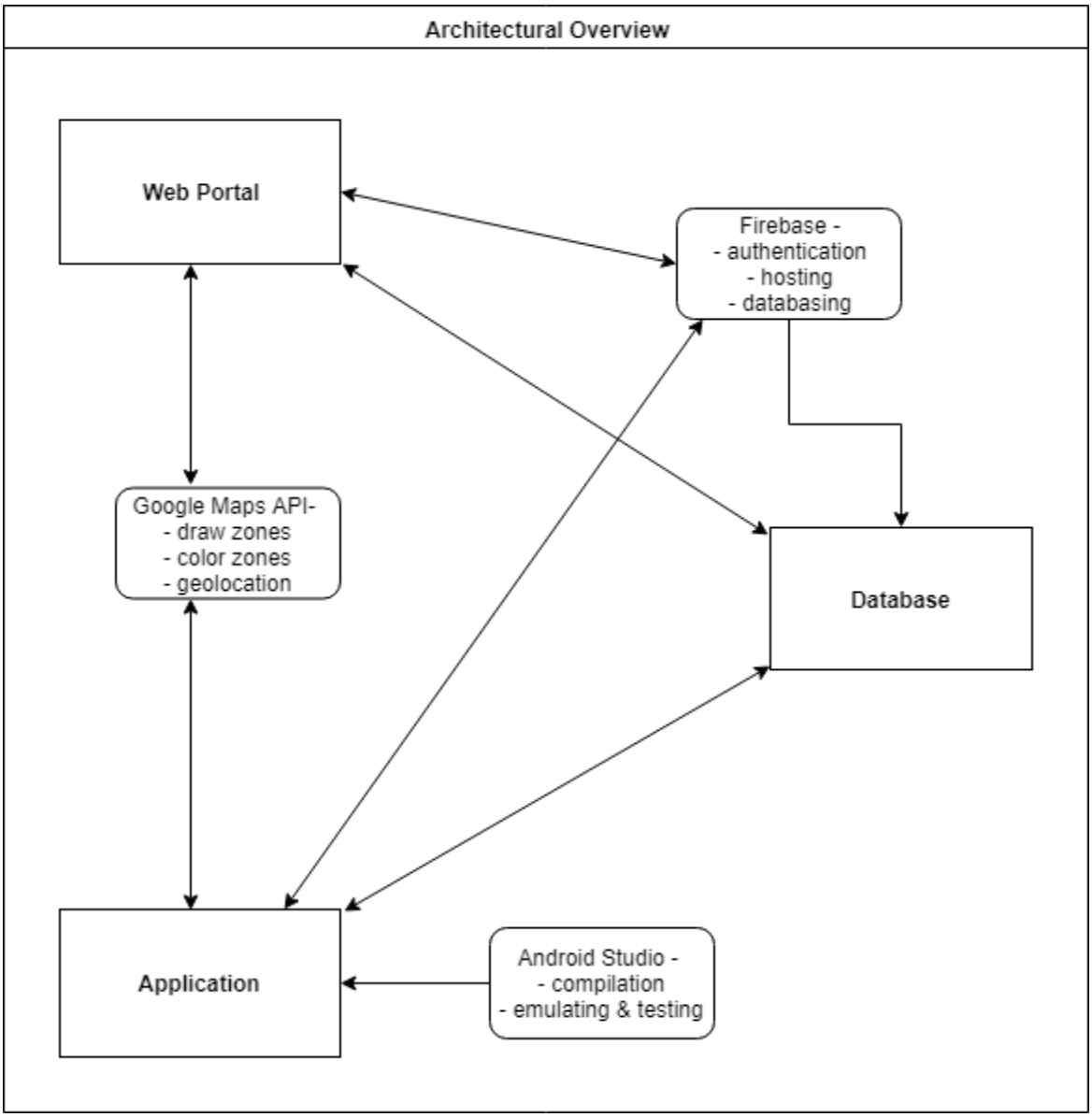
The software we create will allow a user to make maintenance and litter reports in defined areas for their campus, include the ability for users to sign up both on their mobile phone and through the web portal, and have an area for administrators to manage reports and zones. This provides an easy and interactive way for students to contribute to the cleanliness of their campus, and a solution for campus workers to know the status of campus areas. We will be using Android Studio, Firebase and Google Maps and utilizing the provided APIs. Here we distinguish types of users as "reporters" and "administrators," where "users" may refer to either.

The main point of user interaction for this project is the Android application. The application is in Java and will make use of multiple classes built in for Android development from Android Studio to create activities which control user interaction.

Any information that we need to store or manage will be handled by Firebase. Firebase has many tools that allow us to easily manage our database, host our website, and authenticate users. Firebase has built-in functions that allow users to register and sign in with authentication, letting us allow users to sign in using their Google accounts or email addresses. Firebase can also manage a user's session on the mobile application or website, which is useful for handling data between pages or pulling and using data from a user's profile.

Google Maps will be the main API that we will be using for implementing all map functionality, the main feature of our software. Through the use of its many built in classes, we will be able to create a map on both the website and on the mobile application. Both ends will be using a dynamic map, which will allow other unrelated locations to be viewed. We will also make use of the Polygons API to draw out zones marking different areas as clean or in need of attention.

3. Architectural Overview



The main method of communication between our web server and our application is Firebase, which hosts our web server, methods of authentication, and database. The Google Maps API is used by both the website and the Android application to display a map. The website uses this API by adding the ability to create polygons, display report data by a heatmap, and by displaying zones. The application uses this API purely to display the user's location and any zone added by an administrator in the web portal.

The website has three main tasks, only usable or viewable by administrators:

- Creation of zones and storage of coordinates
- Display and edit employee tasks
- Interpretation of report data

The purpose of the website is to be a portal where administrators can view the status of all zones, create new zones, view reports and view a heatmap of the collected reports. Reporters will not be able to access the internal features of the website. A tool will be provided to draw and store zones that users will interact with on the application, under the assumption that a campus employee is more familiar with the social areas and typical traffic on their campus than some algorithm we design. Administrators will be able to add, view or modify tasks for themselves or employees to complete.

The application also has four main tasks:

- Allow the user to create full reports or quick reports
- Display employee tasks for administrators
- Display the user's location
- Interpret and display zones

As any user successfully logs into the application, they will able to see a map with loaded zones. If the logging in user is an administrator, they will be able to see an extra menu item allowing them to view tasks. Included with a button, every user will be able to display their location by tapping a button on the map, helping them locate zones nearby. By tapping a zone, a user will be able to create a full report or a quick report by selecting a button on the appearing pop-up.

Both ends share these tasks:

- User registration and authentication
- Allow the user to change their password

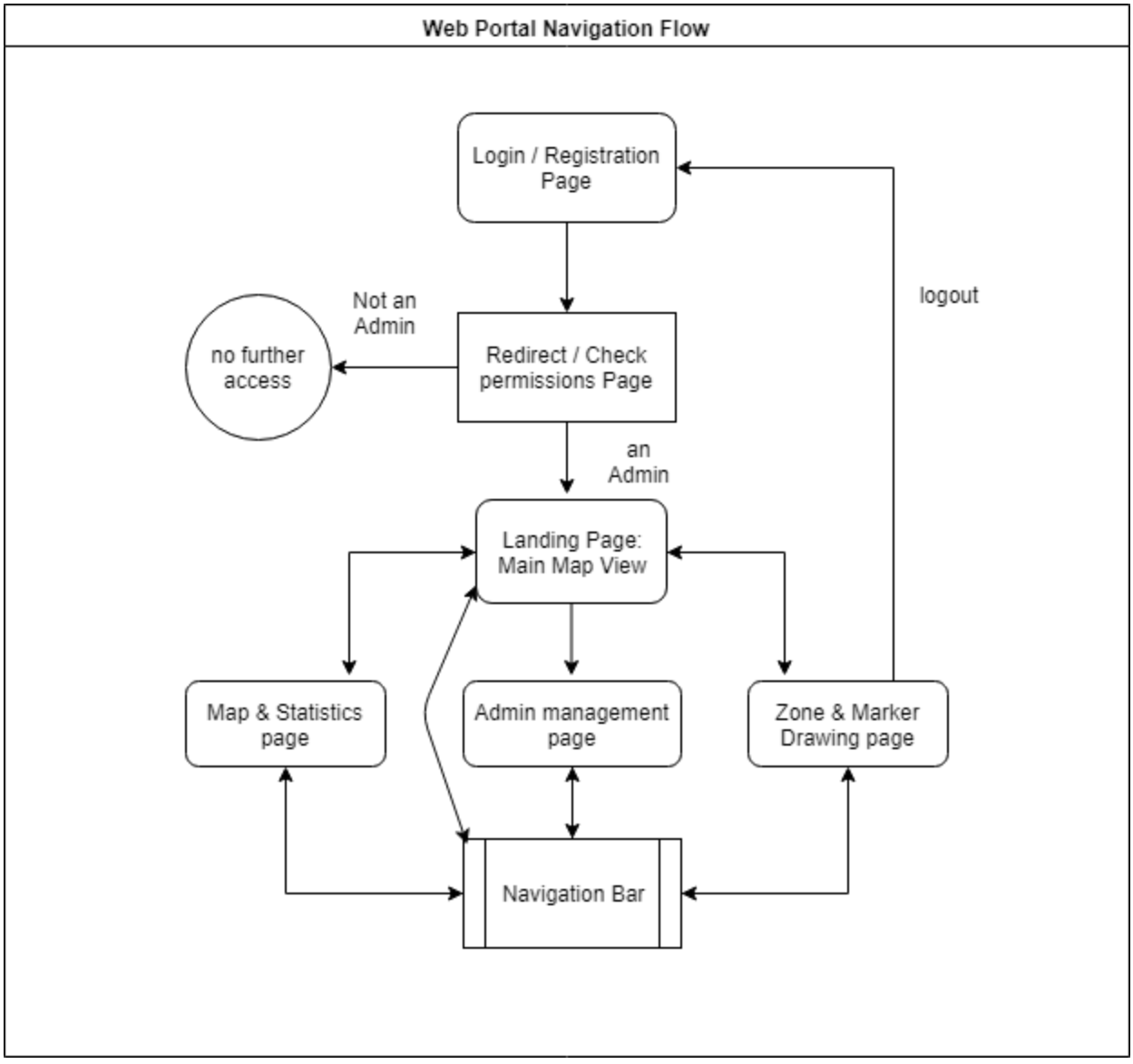
Both the mobile application and the web portal will allow users to create accounts with their Google account or using an email address. There will be an area to change a user's password or reset a lost password.

The information that will be transferred from each component of our software will be zone coordinates, user data, contents of reports and images. This will all be handled with Firebase using built in authentication, a live database, and file storage. When a report is created on the application, the information is sent to our database, which is then pulled and displayed on request with our website. Zones can be created on the website, which will be stored in our database and displayed on both ends on request.

4. Module and Interface Descriptions

Our entire project has three main components: the web portal, the application, and the database they share. The website and application each have a diagram of their navigation flows, and is broken up into modules. Each module has its own UML diagram.

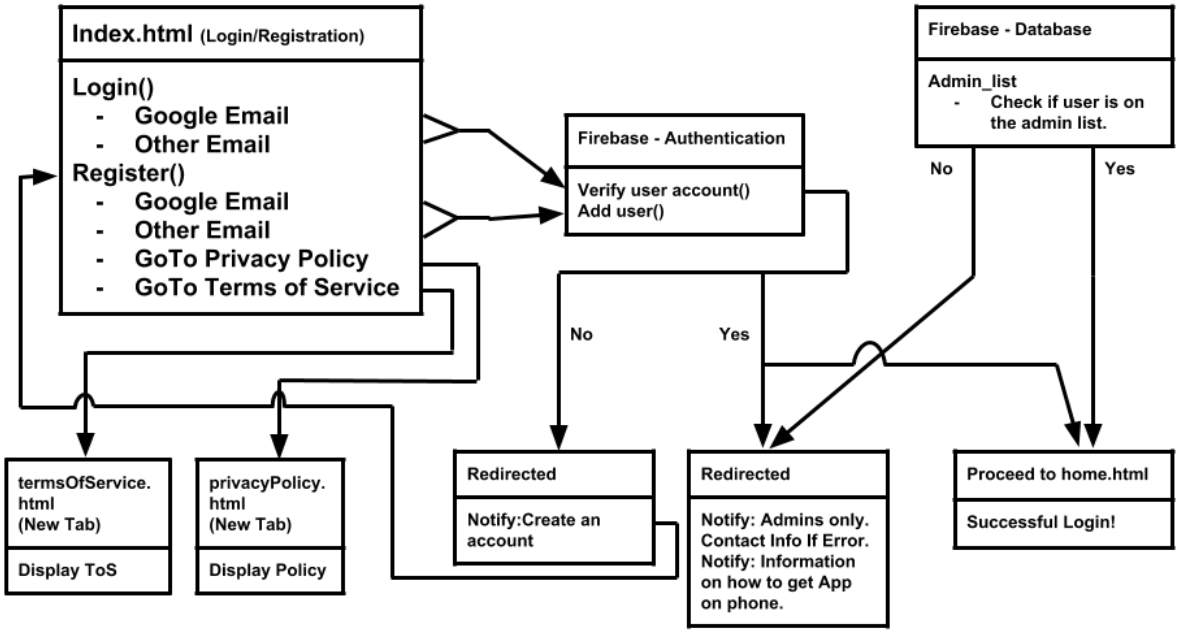
4.1 Website



(Add description and title to this image)

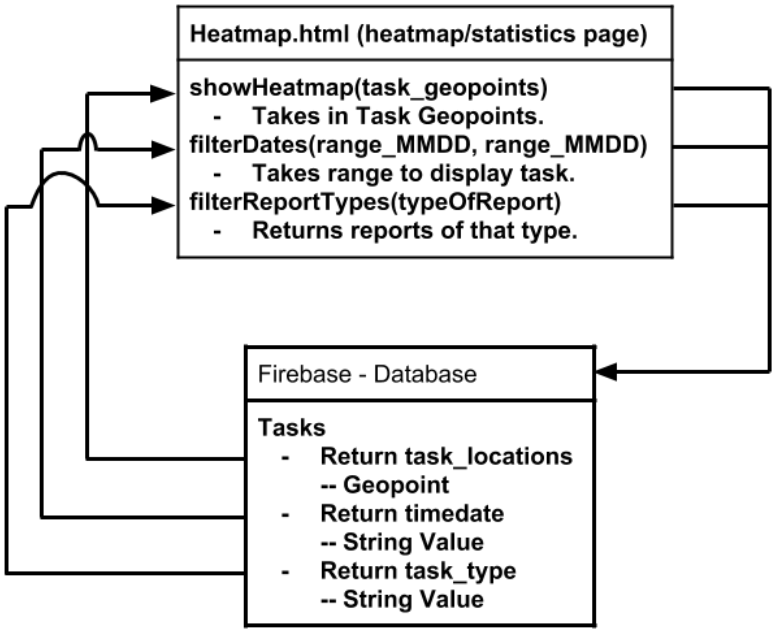
4.1.1 Login & Registration Page

This web page allows users to create accounts and log in to the web portal. Only users with administrative permissions will be able to log in to the web portal, though any user can register. If a non-admin user tries to log in, they will be redirected to a page that denies them access. This page has Google and email authentication, allowing users to create accounts using their existing Google or email accounts.



4.1.2 Map & Statistics Page

This web page allows administrators to view a heatmap sourced from report data from the application. They can filter time periods to edit the data reflected in the heatmap. They may also view reports in a list view, also with the ability to filter by time period.



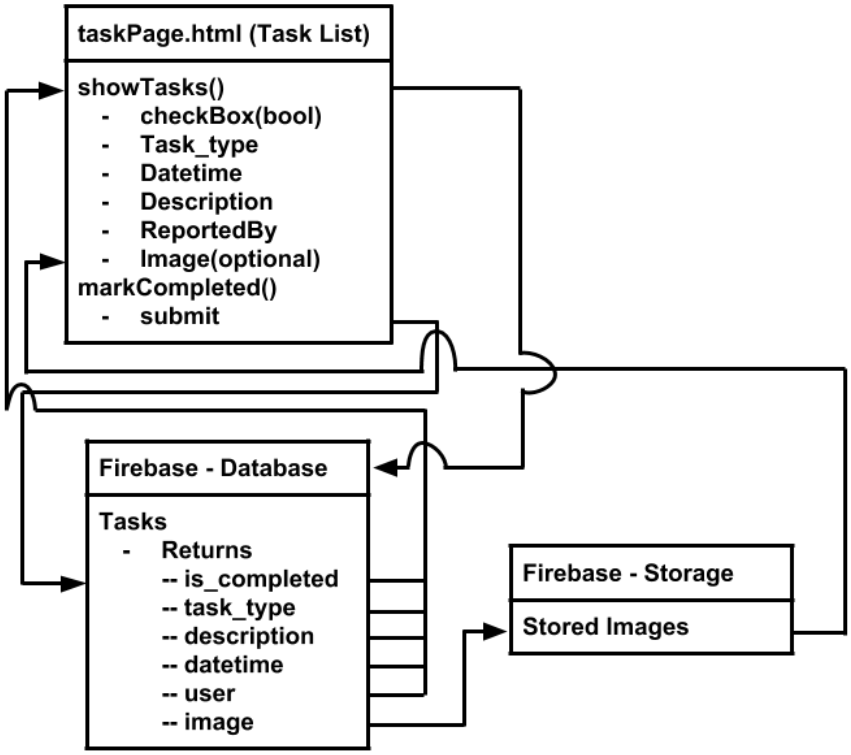
4.1.3 Task List Page

This page will be a where all tasks are viewable by the administrators. Task's will be able to be marked as completed by admins when the tasks have been handled. The tasks page will be automatically updated in real time as tasks are created by user's using the mobile application. The primary information stored will be the user who created the task, timestamp, description of problem, photo (optional), Boolean value representing completion, and GeoPoint of where the task is located. A sorting feature is possible but at this time in development, not a priority and will be evaluated at a later date. UI is not a priority as well so limiting the number of tasks shown at a given time is another feature we will investigate before we deliver to our clients. The basic format of what will be show will resemble the following:

[ ] Task Type: Timestamp: Task Description: Reported By: Image Link

A check box will start each task, when checked, this task should be removed from the active list. The task will remain in a completed task group for use in statistics. The timestamp will have its information parsed to show the format of MM/DD HH:SS. The task description will wrap if required and at the end will be a link to a provided image, though images are

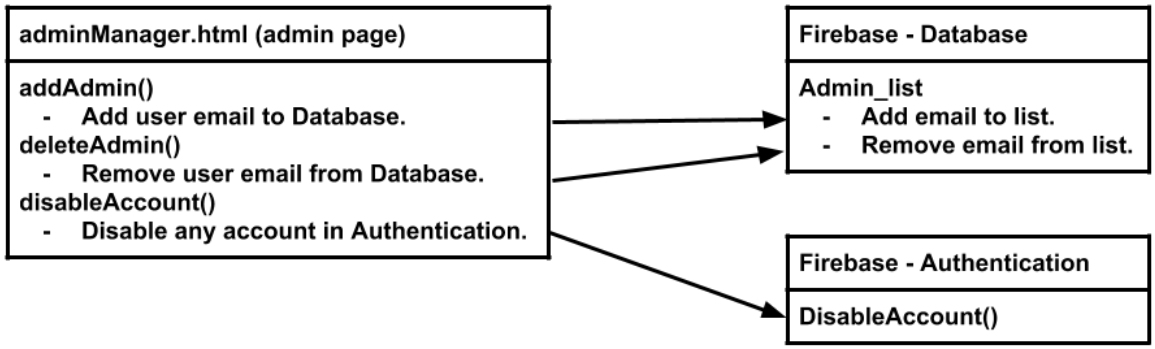
optional. The other data stored such as user and GeoPoint will be used for statistical purposes.



4.1.4 Admin Page

This webpage will handle sensitive information regarding who is an administrator and the management of zones. There will be a main account, most likely an account linked to NAU Facilities email, so that there will always be at least one admin at any given time. It will be possible to add an admin through Firebase but that should not be done by the client for any reason and only by a project developer in extreme circumstances. This administrative website, in general, will be the only interface that users should be using to interact with the database. Admins will be able to add Google Auth user emails to an admin list and that is the process of authorizing an account to use the administrative site. This webpage will allow for the removal of an admin from the admin list as well.

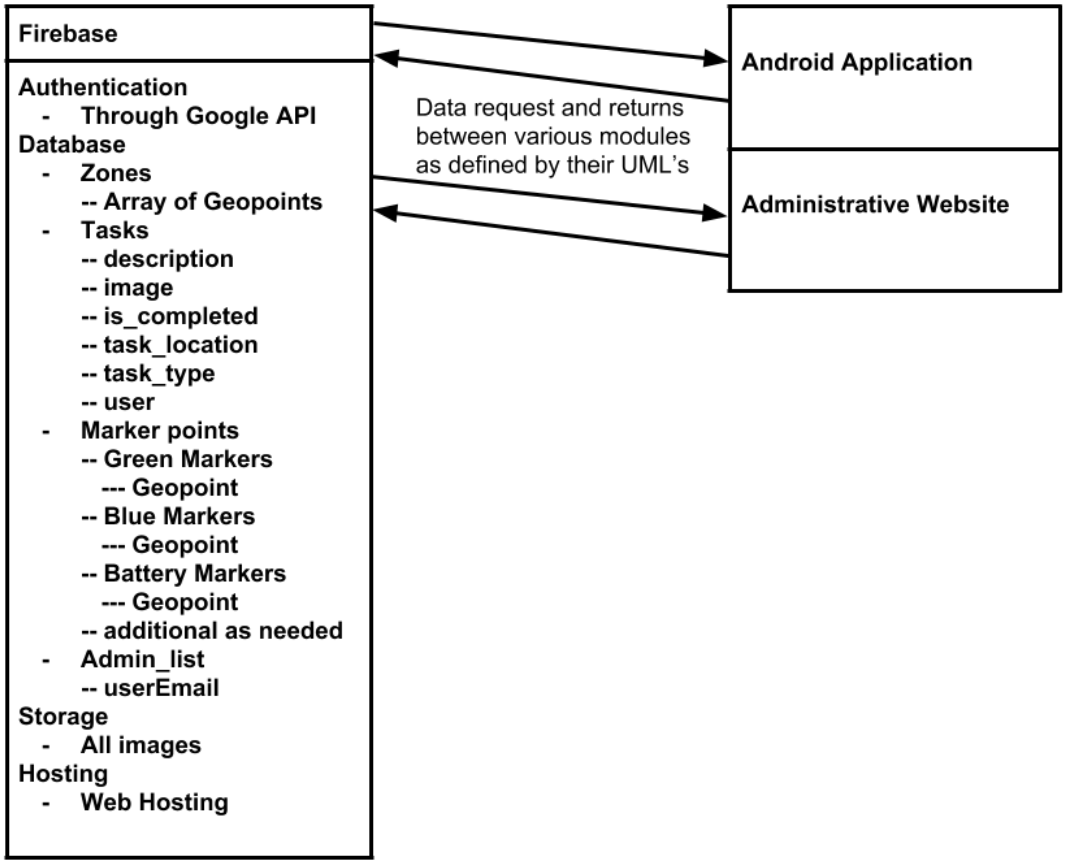
Creating zones easily by administrators is important since the campus is changing all the time. The ability to add, remove, or modify existing zones is a feature needed for the maintainability of the “Clean My Campus” application. This feature is straightforward and based on a simple Google Map where the admins can add a zone by clicking an “add zone” button then click the points where the zone is desired. The polygon created can easily be enabled to be modified at any time. The zones will also be able to be deleted when needed. This is also where icons will be managed as well. Placing or removing icons that represent dumpsters, glass recycling, normal recycling, or even the Bigbelly locations will be simple and requires just selecting the icon you wish to place then clicking on their locations on the map. Accurate locations of their physical location will require the client’s active participation in maintaining the current whereabouts of those services.



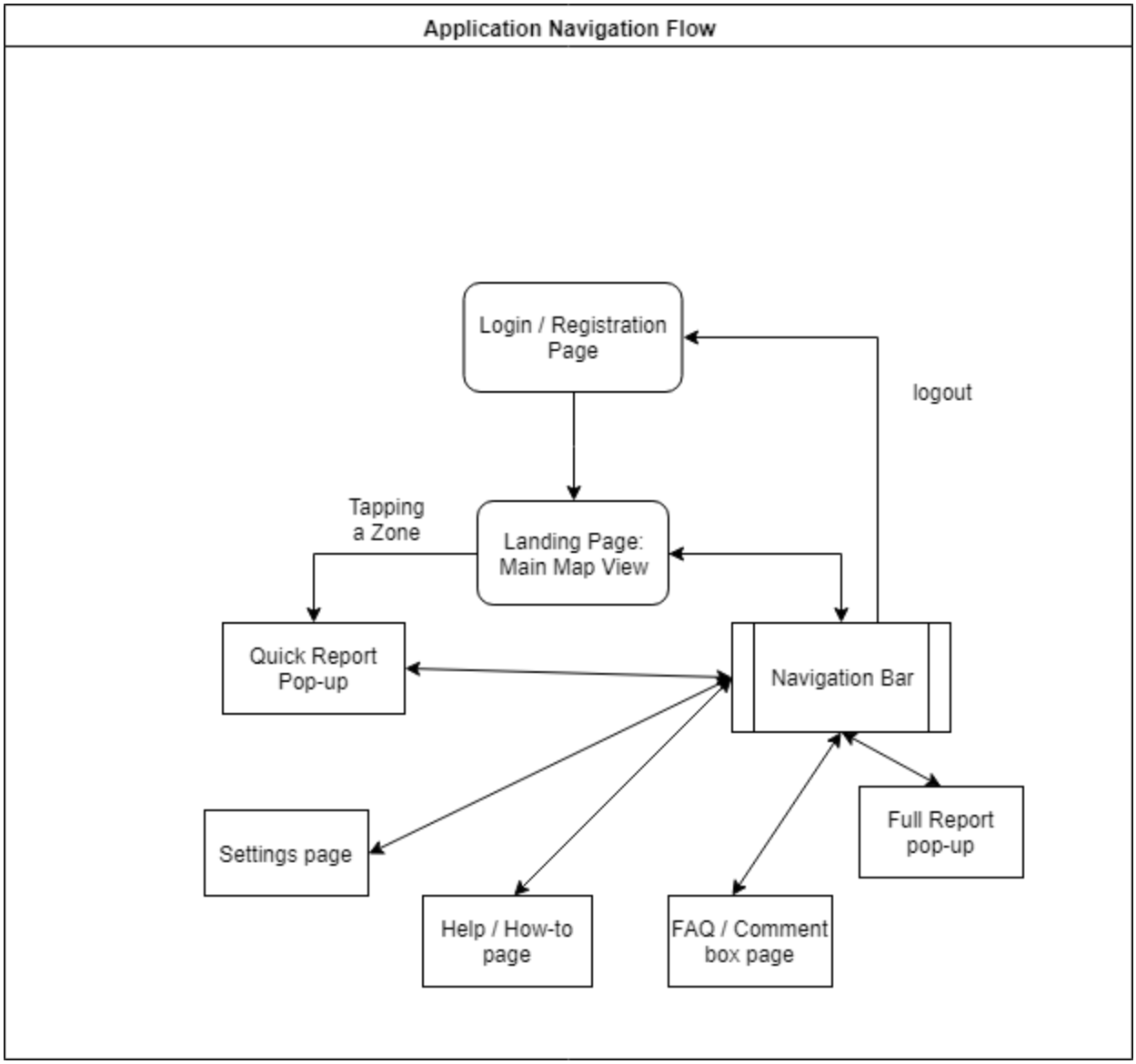
4.2 Database

The database we are using is self-scaling and should never have any down time since it is maintained by Google. The data we are handling is simple and the database has predefined types of data we can use to meet our needs. One of the more useful types is the GeoPoint type which allows our application to store latitude and longitude points easily for use in our administrative website and Android application. We are still determining what

information is required to be stored but the goal is to have no unnecessary information stored, all data to have a purpose, and minimize redundancy. The database is highly capable of being modified for our needs and allows for data to be stored nearly anyway we see fit. As development continues, the way in which data is stored and what is stored will be optimized as needed. This optimization will become a focus nearing the completion of our minimum viable product.



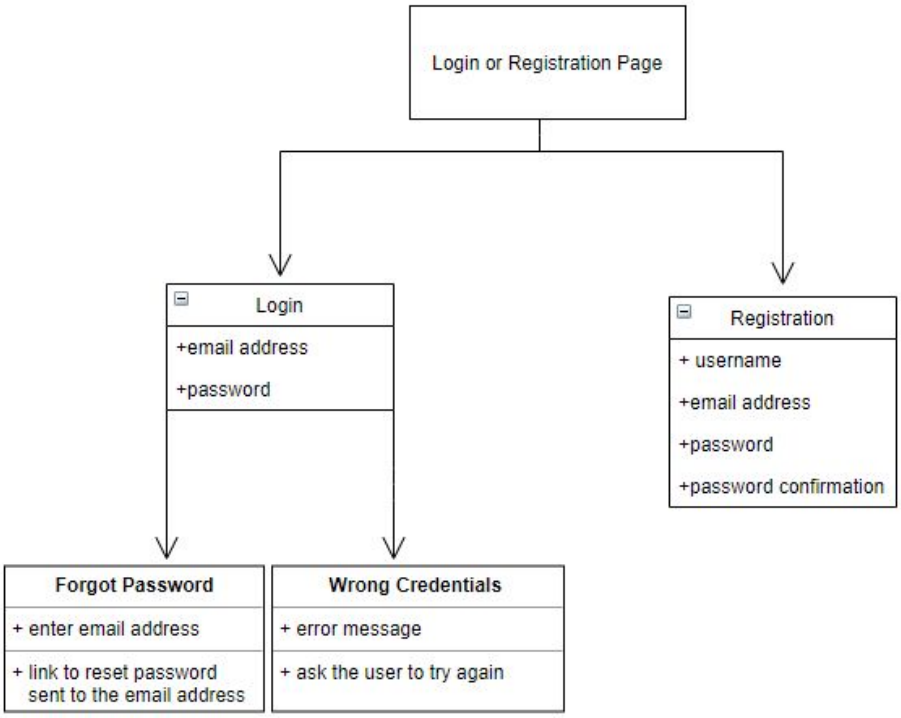
4.3 Application



add title and description

4.3.1 Login Page

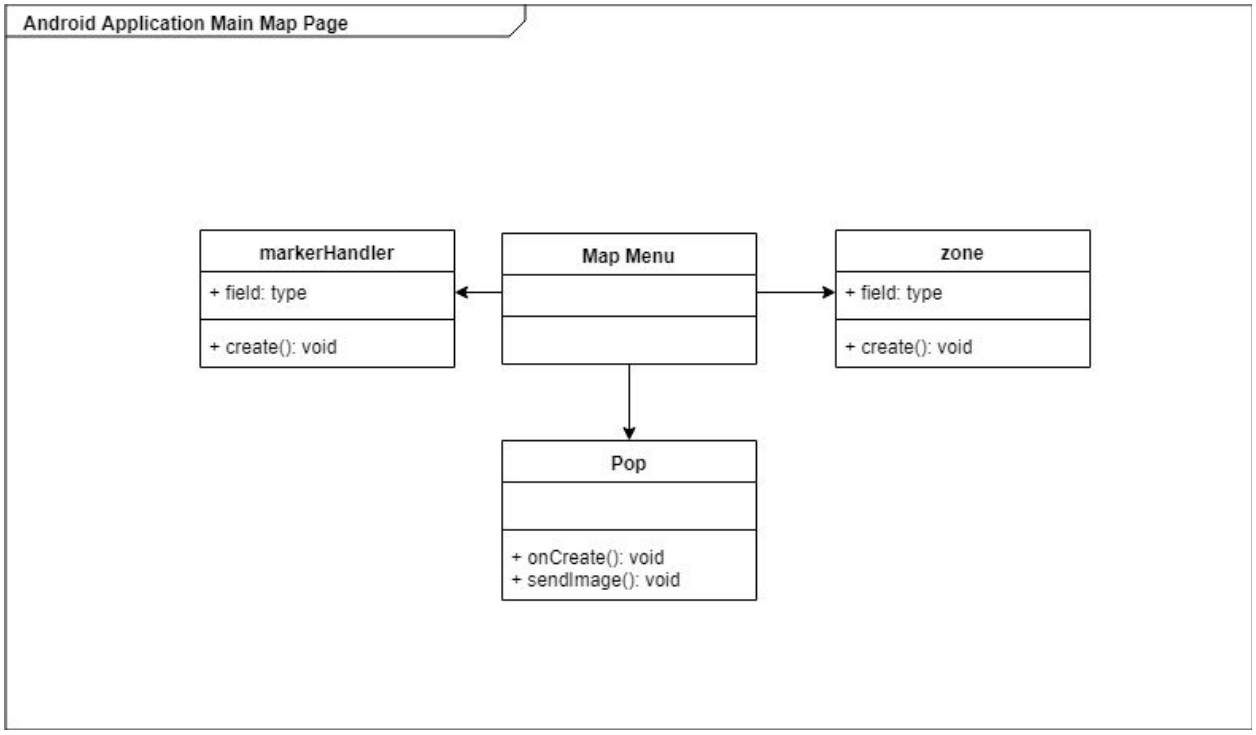
When opening the application, the user will only be given an option to register an account or login with an existing account. The registration page will ask the user for a username, email address, password, and a password field confirmation. The login page will ask the user for a email address and password in order to log into the application. If any of the credentials entered is incorrect an error message will pop up. In the login page there will be a link the user can click if they forget their password. The user will have to enter their email address in the field and a link to resetting the account's password will be sent to that email.



<title and number>

4.3.2 Main Map Page

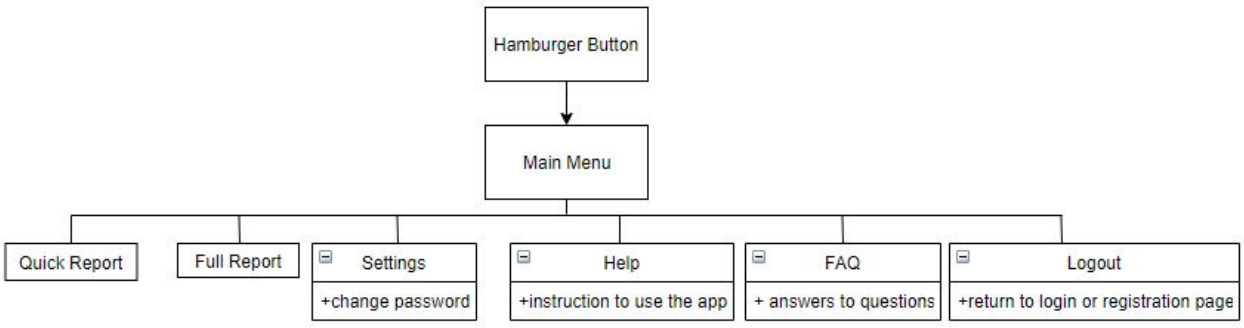
The main page of the application will contain the map and any zone that has been placed. From this interface, users will be able to view locations and submit a report based on their current location. The zones will display colors based on the current cleanliness factor of the zone, based solely on crowdsourced reports. No report that requires a campus employee to address will affect the color of the zone.



<title and number>

4.3.3 Main Menu

The main menu can be accessed by clicking on the hamburger button. In the main menu the user can go to their account settings and change their existing password. The “help” button will allow the user to write a message regarding the application and the message will be sent to an admin for response. The full report and Quick report options will allow the user to create sendable reports. The logout button will redirect the user to the login page.

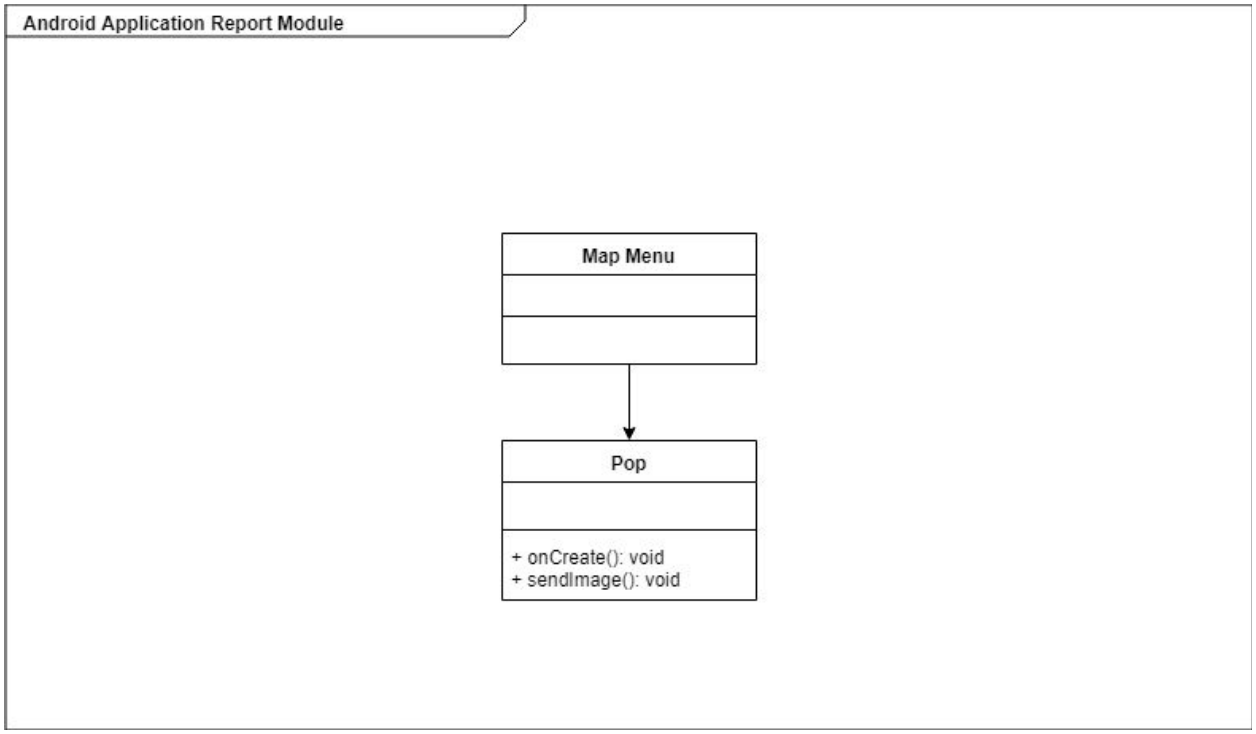


<title and number>

4.3.4 Full Report Module

When a user selects a zone, a pop-up will be displayed allowing the user to submit a report. This report will consist of a description of the situation and an optional picture. If the

report does not have a description, they will be notified that the report is incomplete and that the requested fields must be filled in.

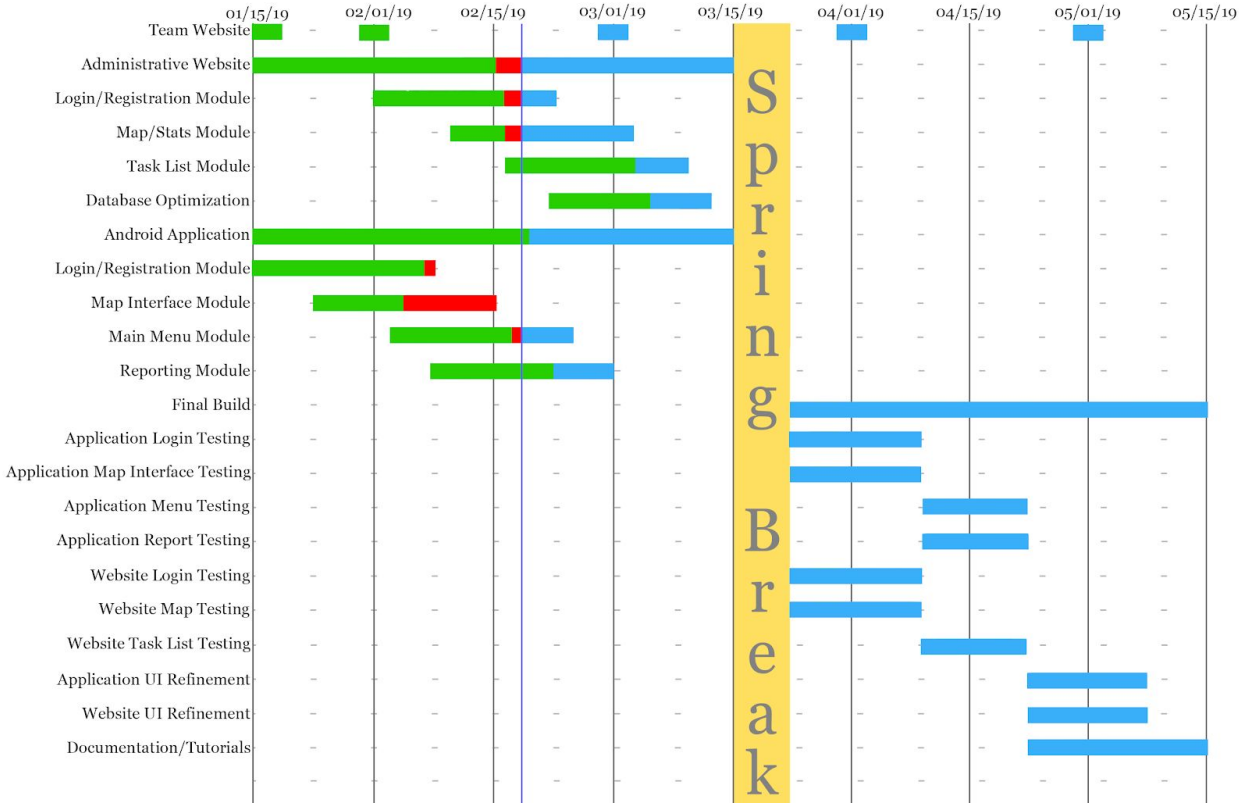


5. Implementation Plan

Our team is developing the “Clean My Campus” application in pairs. We have both an administrative website and an Android application to develop in parallel to get the minimum viable product (MVP) completed before the end of the semester. The administrative website is where the application administrators can manage tasks sent for them to complete, view the NAU campus based on client defined zones, edit those zones, manage admin accounts, and view statistics. The Android application is where most users will be interacting with. They will be able to view the zones on a map of the NAU campus and make various reports of issues they come across.

As it turns out, developing in parallel of both the administrative website and Android application became natural as either part required nearly the same steps in development. We have started by implementing authentication and login features using Google Auth. This feature is convenient, secure, and compatible with the website, application, and database. Once verification and authorization have been implemented, modifying and adding data to our database should be secured and only possible by authorized users. Authorized users being those who have signed up with our service. Once again, both application and website will be displaying the NAU campus map with overlay of zones. This map display feature will be implemented nearly the same in both part but adjusted for their own environments. The application will have to compensate for smaller screen size and user experience while the administrative display can be directed to admin specific functions. At this point, we start to diverge in what the application and website are doing. The application will be creating tasks and uploading to the database with its various information while the website will display those tasks for the admins to complete.

The website has a few more implementations that the application does not have. Those being statistics page and admin/zone management page. These are smaller modules and will require manipulation of the database but with everything else implemented prior, they should not take much time to create. Once we start nearing completion of our MVP, we will move into debug mode and testing. We will create documentation and user manuals for the use of our application and website once we feel all the minimum requirements are met. Our final considerations will be improving user interfaces throughout the website and application. By the end of the semester we will have a completed viable product to deliver to our clients.


01/15/19
02/01/19
02/15/19
03/01/19
03/15/19
04/01/19
04/15/19
05/01/19
05/15/19
Team Website
Administrative Website
Login/Registration Module
Map/Stats Module
Task List Module
Database Optimization
Android Application
Login/Registration Module
Map Interface Module
Main Menu Module
Reporting Module
Final Build
Application Login Testing
Application Map Interface Testing
Application Menu Testing
Application Report Testing
Website Login Testing
Website Map Testing
Website Task List Testing
Application UI Refinement
Website UI Refinement
Documentation/Tutorials
Spring Break

6. Conclusion

The current way to report issues on campus is rarely used and unmaintainable. This application will act as a tool for getting students and faculty more involved with keeping their campus clean and taking responsibility for the places that are like home to them. Our team and clients envision this application being a template that other campuses can use to motivate their communities to get more involved. We would also like to create more awareness amongst students and generate a sense of responsibility towards sustainability efforts.

We are developing a mobile application that will help encourage NAU students, staff, and faculty to actively engage in maintaining the campus. Cleaning litter on campus, fixing broken fixtures, and clearing trash bins all help contribute to NAU's sustainability efforts. The software we are building will assist NAU staff by sending reports and tracking statistics based on user reports.

This Software Design document is an overview of how we plan to implement and design our project. Detailed descriptions and UML diagrams are included for the modules and interfaces as guidelines. This will provide a clear guide for features that need to be implemented for the MVP for our sponsors, and how we intend to do it.