MoGreen

Requirements Specification

Version 2

December 5, 2018

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

As part of the Fall 2018 semester, we are required to create a contract between our clients and Team MoGreen. By signing this document, the clients understand the key requirements that our team will implement in the development of the "Clean My Campus" mobile application.

Cassie Graham		
MoGreen Team Lead	Date	Signature
Brock Brothers		
Client	Date	Signature
Ellen Vaughan		
Client	Date	Signature

Table of Contents

1. Introduction

Surrounded by a ponderosa pine forest, the NAU campus strives to be as green and sustainable as possible. NAU has glass, paper, plastic, and battery recycling located throughout the campus. As NAU continues to grow its student population, instances of overflowing trash bins, full dumpsters, and campus litter rises as well. Basic maintenance needs occur more often as sprinkler heads are accidently broken by passersby and other minor issues.

This 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.

Currently, there are three main types of trash containers around NAU. There are the standard concrete types and the large dumpsters. The third type are the Bigbelly trash bins and there are over 300 of those across our campus. These trash bins are very useful since they are connected to a network and can send out notifications when full. They are also solar powered and can compact the garbage they collect. However, there are situations where the bin entrance can be blocked. The system is limited and cannot report such obstructions. Mr. Brothers and his team will not know if there is an issue unless they come across it and visually see the blockage or someone reports it.

The current way for a student to report minor issues is to call NAU Facilities or report it using their website. Nearly everything done in the effort to keep the campus clean and maintained is completed manually. There is not a system in place to accurately track when areas around NAU have been cleaned. 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. Either way, the reporting method is outdated and unintuitive and most students will not make the time to report minor issues.

The name of our project is "Clean My Campus" and it will be a mobile application. This application will be developed by our four-person team. The mobile application proposed will make reporting minor issues easier and allow NAU facilities to receive and track these reports faster. This application will help NAU's sustainability and keep our campus clean and green.

Ms. Vaughan is interested in having the mobile application developed to help sustainability, and not just for NAU. She envisions that NAU will be the pilot campus where the initial application is deployed. If successful, she will bring it to other campuses in California. Mr. Brothers is interested in the application for its use in helping he and his team maintain the campus in general. He also would like to have the application help identify trends and areas of high trash volume. Creating a sustainable campus is everyone's responsibility and this application will contribute to that effort. Team MoGreen looks forward to working on this application and embraces all challenges that may emerge during its development.

This document serves as a contract with our clients for creating their mobile application. Here we will discuss the problem statement and the solution envisioned for our clients. The three components of our project requirements (which are functional, performance, and environmental requirements) will be listed for our clients as an overview, as well as the potential risks that can occur while developing this application along with their mitigation. Finally, we will provide our project plan outlining our future steps towards developing the application.

2. Problem Statement

Currently students have no incentive to go out of their way to clean their campus. One might generally expect that a student on campus will try to keep their area clean and try to rid of any waste in the respective bins. They currently lack the incentive or the tools to try to clean up the areas around them. Because of the size of their campus it may be tough to try to find an area around campus that needs attention. Along with this, creating a report for an issue on campus is a quite tedious task. Reporting an issue is essentially a hidden feature on the NAU website, which is tough to access on mobile devices and requires knowledge of your surrounding area.

One of our clients, Brock Brothers, manages NAU's Moving and Recycling department where he manages the recycling and trash removal services on our campus. He also manages any maintenance issues including broken fixtures around campus and the Bigbelly trash bins. The main issues he encounters in his daily workflow are:

- Managing the Bigbelly trash bins.
- Managing and receiving reports from students.
- Locating the root of an issue based on reports.

Our other client, Ellen Vaughan, was previously the Sustainability Manager in NAU's Office of Sustainability. This presented her with the responsibility of trying to implement sustainability ideals into universities and colleges, and to manage any initiatives from the Office of Sustainability. As an environmentalist, she is very aware that having toxic waste and non-compostable materials sitting in the ground or getting into bodies of water is never good for our environment or our general health. Her goals include:

- Creating motivation and finding creative ways for students and communities to get involved with their environment.
- Helping students contribute to other sustainability efforts.

Currently reporting cleanliness and maintenance issues is a hassle, which adds delay to the time it takes for the issue to be resolved. The current method of reporting an issue on our campus is logging into NAU Facility Service Work Request Site which is likely an unknown domain to many students. In this process, the user will need to select the campus they are attending then select the zone. This list includes every numbered building on campus, which may be confusing to the users and deter them from

completing their report. The user is finally directed towards a report form where they can describe the issue they discovered. This feature is inconvenient to report an issue on a mobile phone or may be forgotten when the individual finds a computer.

With our clients, we've designed a solution to combat all of these problems NAU currently faces.

3. Solution Vision

Our solution is derived from Mr. Brothers and Ms. Vaughan's vision, and attempts to get students involved with their environment, involved with their surroundings and get inspired to join green projects around them.

We plan to create an application that allows users to take action in cleaning their campus. Our application will direct students towards marked areas of interest and have them report these zones as cleaned or in need of attention. We will include reporting features for these zones that will allow for reports to quickly be created to let maintenance staff know about any dysfunctional or vandalized objects around campus and allow the reports to generate data so high-volume areas can be tracked. This data could also help NAU Green groups to be more organized, with this application being a tool they use.

To construct this application, we plan on using tools such as Google Maps API. This software allows us to create multiple markings on their given map. With all the features given, we will be able to easily display zones, create marks for campus features and various options to allow us to display statistics using heatmaps. We will also be using MySQL for storing any data or images that may be included in reports. This database will be holding any users' account information and map coordinates. Our website will be hosted by firebase, which is the main interface used by administrators to add zones and view reports. The initial application will be designed and built using Android Studio. The application must still be modular enough to be converted into a cross platform application in the future.

We would like to, in an extended solution, include areas on the map that may prove useful for students, such as locations of Bigbelly trash cans or areas that allow the users to recycle. Administrators will have the ability to designate zones on their campus. This will allow the zoning to be more intuitive for users based on information that the administrator knows about how the campus is "used" and viewed, coming from someone who spends a lot of time on, and is intimate with, the campus. These marked zones can be reported as clean or in need of cleaning and will deteriorate over time if the area has not received clean reports in a certain time range.

We are defining users as two types, reporters and administrators. We will be able to collect data on reports created in each zone which will include images that may have been included, information on the report, and user information (such as full name or contact information). Each user of both types must have an account created for the

application and information they used to sign up will be recorded and stored. Any maintenance report created will be displayed on the web portal for administrators to view. We would like our application to create an easier and more accessible view for our maintenance staff, as well as create an easier way for students to create reports.

With this solution in mind, we have outlined our functional, performance, and environmental requirements below.

4. Functional Requirements

Functional requirements outline each specific functionality we will implement, detailing ways that different types of users will be able to interact with the application.

4.1 View Zones

One of the main features of our application is to allow the users to view zones on a map. These zones will act as markers to show the users specific locations that may need cleaning or attention. A timer will function for each zone to degrade the color of the zone over time. If a zone is marked fully clean and receives no reports within a specific time range, the zones color will gradually change from a lighter green to a darker red, to indicate it probably needs attention. Users will know which zones need attention by viewing the color shade the zone indicates from the main interface, rather than needing to tap the zone for more information.

User: Reporters and administrators.
Description: View a section of a map that will contain colored smaller areas that designate areas as clean or in need of attention.
User error: Possibility for user to select a wrong area and send a misleading report.
State of the system before: Map will display area normally, zones will display a color based on the status of their zone.
Flow: User pans in and out of the map area. When they discover their location on the map, they can select a zone.
State of the system after: After tapping, a popup displays the types of reports a user can submit.

4.2 Quick Reports

Once a reporter selects a zone, the application will bring up a small window allowing the reporter to create a quick report or full report. If the user selects Quick Report, they will be presented with default options such as: litter, full trash cans, or the ability to indicate that the zone does not need attention at this time. Depending on the report given, it will either add time to the zone maintaining or improving the zone condition (indicated by color) or decrease the timer on the zone. The quick reports will be the main action that determines a zone's status, making this functionality almost purely based on crowd-sourcing, thus requiring less effort from an administrator or campus worker.

User: Reporters.
Description: The ability to send a predefined report after selecting a zone. The report will only be used to report smaller issues such as litter in an area, an overflowing trash can, or an area is clean.
User error: User may send in a quick report on accident.
State of system before: Zone is displayed in its current status.
Flow: A user sends in a report. This report is sent to our database and is recorded for that zone.
State of system after: Zone color shifts slightly to be redder or greener based on if it was reported as dirty or having been cleaned.

4.3 Full Reports

After selecting a zone, the user also has the option to make a Full Report. This allows users to send in a detailed report about vandalism, a maintenance issue or misplacement of trash. We do not plan on letting the full report modify the status (color) of the zone as these issues are generally out of any reporter's control and will require staff intervention. Zone colors are based solely on crowd-sourced reports of needing to be cleaned or having been cleaned.

User: Reporters.
Description: A custom report that a reporter can submit from a zone. This report will allow for more information on an issue to be received. This report will not modify the status of the zone.
User error: Reporters may accidently report from the wrong zone or create a misleading report.
State of system before: Zone is displayed.
Flow: Reporter taps a zone and selects the full report. They fill out relevant details in a text box and attach a photo if desired.
State of system after: Zone is displayed normally, report is added to the database and displayed on the website to administrators.

4.4 Web Portal

A web portal will be available for administrators that will allow them to review reports, change the location for the map, and add or modify zones. In order for the map to be used on the application and through the web portal, the administrator must enter an API key. This key is responsible for linking any features used to their accounts and applying any charges if they appear. Included on this page will be simple instructions for acquiring information such as finding specific coordinates for zones, or easily entering the information needed to show their campus on the map. They will also be able to create new administrator accounts, and view traffic data.

4.4.1 View Reports

The web portal will display any report made, which administrators will be able to view. Reports can be queried from a date range to simplify the amount of information displayed. They will also be able to view any relevant information about the reports such as the contact information of the sender, the zone the report was sent from, and the content of the report created by the reporter.

User: Administrator.
Description: Full reports can be viewed and sorted based on date or location of the report.
User error: Administrators choose an invalid date or sorting type.
State of system before: User will be presented with a short form allowing them to enter a date range and a location.
Flow: Once the form is completed all reports that match the query will be returned.
State of system after: A list of reports and all related information will be included on the page.

4.4.2 Heatmap

A heatmap will be the minimum way that traffic data is displayed to administrators, specifically displaying the frequency of reports. Information regarding how 'popular' a zone is, meaning how much traffic that zone has seen during a specified time period, will be displayed.

User: Administrator.
Description: Data is displayed in a 'heatmap' format, showing traffic patterns on a campus over a specified time period, regarding reports made.
User error: Administrators choose an invalid date or sorting type.
State of system before: A map displaying the last month (30 days) of traffic data.
Flow: The administrator will enter a date range if desired. After choosing the time period, the map will update to reflect traffic in that time period.
State of system after: A map displays traffic data.

4.4.3 Administrator Registration

A main campus administrator will have the ability to add other administrators through the web portal. Only appropriate parties will be able to have administrator accounts through this method.

User: Administrators.
Description: Administrators will have the ability to add another administrator to the system, giving them certain privileges. Reporters cannot access the website.

User error: A username is entered that is already in use.
State of system before: System is accessible by any administrator with existing credentials.
Flow: Administrator logs into the website and clicks 'Add New Administrator' from a menu. They can then fill out a form to add another administrator account. This new account will have access to the website.
State of system after: Another administrator account can access the website.

4.4.4 List of Current Users

Administrators will have the ability to view a list of existing accounts and their types. This will allow them to remove old unused accounts, or administrator accounts that should no longer have administrator privileges. They will be able to sort by account creation date or type.

User: Administrator.
Description: Administrators can view a list of users.
User error: Administrators choose an invalid date or sorting type.
State of system before: An administrator opens the webpage and clicks to view the list of accounts.
Flow: The information optionally entered in the form is sent to the database to return a list of users, sorted as specified. If no view is selected, accounts are shown in order of creation.
State of system after: A list of accounts is displayed.

4.5 User Registration

Any individual who wishes to use this application will be required to sign up for an account, requiring some basic information. Information such as the user's email address will be needed to ensure that reporters can be contacted if needed, though this is unlikely.

User: Reporters.
Description: A user can register an account and log into the application.
User error: Users enter a password with incorrect case and have trouble logging in.
State of system before: A user is at the login page for the application without the ability to sign in.
Flow: The registration button is selected when the application has finished loading. The reporter can then fill out the form and create an account.
State of system after: Reporter is able to access the application.

These summarize the main functionalities we will implement for the minimal viable product for this project.

5. Performance Requirements

Performance requirements specify expectations for how our project is expected to perform. These requirements are measurable and provable and will work as guidelines for us to create and test our application.

5.1 Simplicity

The application keeps a simple interface so the users of the application can complete tasks in a short amount of time. We have designated three main page interfaces that the average user will be using on the application, being: the main map interface, the quick report menu, and the extended application menu. Submitting reports on the application will be much simpler than NAU's current reporting mechanism, with more functionality.

5.2 Usability

Users of the application should be able to access any feature quickly and easily. An average user can be defined as any student or faculty member on campus. Below are specific measurements we can use to test the usability of the application with many average users.

5.2.1 Quick Report Efficiency

Quick reports will be able to be created in under a minute. These short reports can be created by selecting a zone and choosing quick report. After this option is selected, the reporter will be able to select predefined report options and confirm their submission.

5.2.2 Full Report Efficiency

Full reports will be able to be created in less than two minutes, or three minutes if a picture is included. The user will need to manually enter the details of the report.

5.3 Reliability

Our application will need to handle multiple requests and store a variety of information, whether it be text or image. Our application is expected to handle a minimum 30,000 requests at once, defined by the current NAU student, faculty, and employee populations, plus some room for visitors.

These requirements will ensure our application is both desirable and efficient to use and is an appropriate replacement for current reporting mechanisms.

6. Environmental Requirements

The environmental requirements describe some more technical constraints we may have regarding our project implementation and development environments.

6.1 Gamification

Future expansions of our application have heavy focus on gamification, which will be further elaborated on in section 7. To be mindful of our future additions to our minimal viable product, we are allocating database space for creating user teams and assigning point values to actions, so that different functionalities may be implemented.

6.2 Google Maps API

The Google Maps API outclasses all of its competitors with its built-in features and compatibility with multiple development environments. It should be noted that if we encounter some sort of problem with the Google Maps API, there are hardly any comparable replacements for it. This poses a risk, though it may be insignificant.

6.3 Hosting Service

For our hosting service we need an environment that supports JavaScript and MySQL. This service will be the basis for our application's administrative features, reports view, and statistics.

7. Comfortable Solution

Following the functional requirements we have listed above, we would also like to include other features that will show more information related to reports, add incentive for users to continue reporting or cleaning, and assist users in locating campus features. Geolocation would assist the users in locating their own area while allowing full reports to be created without needing to select a zone. Adding gamification mechanisms will add fun and incentive to keep individuals involved and using the application. We are also looking for a way to allow the administrators to add locations other than zones to pinpoint useful resources around campus.

7.1 Geolocation

Allowing users to submit reports without the need to select a zone would assist with our two-minute average goal for submitting full reports. As an addition to the side menu on our application, users would be able to select 'full report' and submit a report based on their actual location rather than a selected zone. Using this information, we could somewhat confirm the accuracy of reports as we could compare geolocation with the reported zone. The user could also use this information if they are not sure of their current location and need assistance selecting a zone.

7.2 Gamification

Most of the interaction with our application is based off volunteer work, but we would like to implement mechanisms to keep users interested an active with the cleanup effort. A point and high score system would tie a point value to actions on the application, allowing reporters to earn points. Reporters may also be able to form teams using their usernames. This could also allow campus groups and individuals to use our application to incentivize students, i.e., offering rewards for individuals or teams who accrue a certain amount of points on the application. Ideally these functionalities would lead to a larger amount of reports and heightened accuracy in zone status.

7.3 Extensive Map Icons

We would like to see people use this application as a tool to easily locate different features around their campus, making it that much more useful. If a student was looking to find a station for recycling, we would like to make this search easier by including a filter on the map that marks these locations. These locations could be used for multiple different resources, such as: recycling booths for glass or other materials, trash bins, or assistance for students to locate green or related organizations on campus.

These are the main functionalities we would like to implement in a comfortable solution after our minimal viable product is fully implemented and are our next-highest priority.

8. Potential Risks

Risks for our project are what could make our project fail, or just less viable in any capacity. The main risks we have noted are cost, report spamming, and report integrity.

Most of the risks associated with our minimal viable product arise from the price to operate our application. Though we are aware of these costs, most of these can be avoided by using lower quality features within the technologies or is avoided by low traffic. As this project is currently only planned to be used within NAU, costs related to traffic are essentially.

Many of the features from our comfortable solution include extended map features. Some of the more advanced requests, such as geolocation, cost more based on their usage. As more features are requested from the map API, the more this feature may cost. Initial features such as using the map on the application, marking areas, zones and map lookups are extremely cheap and are generally covered by the free credit Google provides all users. With extended usage and more rich features, the usage may go over this given credit and will create a charge for our clients. This can almost be avoided by using cheaper features such as static map loading, however we would like to include the best features we can for the look and feel of our application.

The minimal viable product of our application will not include any validation for either quick reports or any reports for bigger issues. This permits users to mark an area as clean even though no attention was given to the area, or users to inappropriately mark areas as dirty. We do not expect earlier users to send fake reports, as we expect the first wave of users to be students who genuinely want to help out their campus. Even if they do, we will implement logic rules to prevent multiple spam reports within a period of time. However, us and our clients have together decided that currently, the overhead of implementing some sort of verification method beyond preventing a flood of false reports is not appropriate at this time given the expected traffic and user-base of our application.

Though these risks may make our solution less viable, we believe we have a handle on the probability of them occurring and some steps towards mitigation we can take.

9. Project Plan

At this point in time we are working towards completing our technical demonstration. This tech demo is a proof a concept which will showcase the technologies we have selected to use during our development. At the end of the Fall 2018 semester, we will have a simple application showing the key functionalities implemented. We consider a successful demo being a simple application that displays a zone in Google Maps. That zone will be created based on coordinates stored in a database. We will also be able to add coordinates to that database by tapping points on the map in the application. We will also store a picture to the database. We will have a simple website display the same Google Map zone with color changes based on time and display the picture sent.

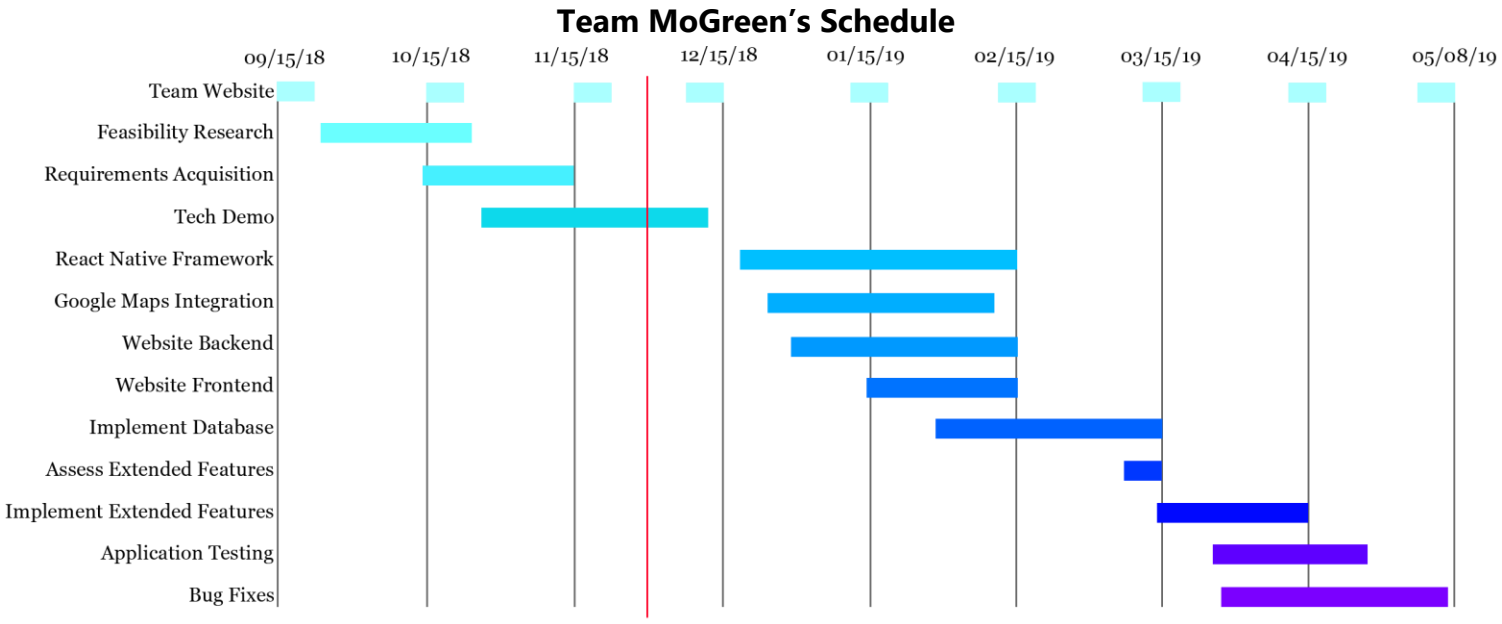
During this December, we plan on building our application using the Android Studio framework. At this time, we will assess the functionality of Android Studio and potentially decide on an alternative environment as needed. This will be done in parallel with integrating the Google Maps API. Creating zones colored with time degradation is the first major milestone we have and is essential for further development of the application. Once we implement that feature within Android Studio, development of the administrative website backend can begin. We will apply the same zone and time degradation scheme to the website. We will be using test values and simulated regions at this time. Development of the website's frontend can begin once the Google Maps API has been integrated with the website backend.

When Spring 2019 semester begins, we should have quick reports implemented and ready to send data to a database. The website frontend should be started with the goal of displaying individual point markers, regions, and photos by mid-January. The backend should be ready to store photos and pull simple data from the database as well. Full reports should be close to finished at this time and ready to send photos to the database. This is when the database is initialized.

By mid-February the application should have quick reports, full reports, zone selection, and other minimal functions fully implemented. The website backend should have

smooth handling of data from the database while the frontend displays all necessary information. The database at this point should be containing actual zones of the NAU campus, references to images, quick report and full report information.

We expect the minimum viable product of the application to be completed by spring break with the database containing all coordinates for the zones of the NAU campus. We will assess where we are in development and decide which additional features we can implement within the remaining time. After spring break, the majority of our development time will be dedicated to stress testing the application and fixing bugs. A bug log will be created and maintained during the bug and test phase. Documentation will be created and updated regularly throughout applications development.



(Figure 1. This Gantt chart shows the tentative schedule as of December 1, 2018. This schedule is subject to change and the most recent chart is located at the Team MoGreen website https://www.cefns.nau.edu/capstone/projects/CS/2019/MoGreen-S19/)

10. Conclusion

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

The requirements specification document will help our clients get an overview of what is our plan for implementation and the specific functionalities we are promising to implement, as well as our highest priority 'extra' features. We have noted the risks we foresee during development, their likelihood of occurring, and what steps we can take to mitigate them. The project plan showcases completed and uncompleted milestones with an estimated time for completion of the application.

This application will act as a tool for getting students and faculty more involved with keeping their campuses clean and taking responsibilities 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 sustainable efforts.