MoGreen

User Manual

Version 1

May 2, 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 “Clean My Campus” mobile application and web portal are faster and easier solutions for accumulating facility reports from students and getting students involved in cleaning their campus, and we hope they serve you well. We have created this user manual in the hopes that you can get started using our software easily, as well as be able to maintain it. We hope that we have created a product that will truly enhance your experience taking care of our campus, and we appreciate the opportunity to create the “Clean My Campus” software.

This document will explain how to:

- Install
- Configure
- Use
- Maintain
- Troubleshoot

the components of the “Clean My Campus” software.

2. Web Portal

This section of the manual will explain how to set up, use, and maintain the web portal side of our software. For information on the mobile application, please see page 6.

The administrative website component of our software is where campus administrators can manage zones, place resource markers, track tasks, view a task heatmap, and manage administrative accounts.

The website can add, remove, and edit zones which are seen on the Google Map used by both the application and website. These zones are colorized, starting at green then progressing to yellow, orange, and finally red. The colors represent how long a zone has not been cleaned by a volunteer or by campus maintenance.

The length that each zone remains a certain color can be adjusted by the website administrators for any increment of time in days. Each color except for red can be set to different durations if wanted and have a minimum duration of one day. Red is not changeable since all zones will turn red after the previous colors’ duration have been exceeded.

The website can add and remove resource markers that are show on the Google Map. There are icons for normal recycling, glass recycling, battery recycling, dumpsters, and a temporary marker.

Administrators can view tasks created by the application users for the various issues they have come across around campus. These tasks have descriptions, a task type, timestamp, reporters' email, and an optionally attached photo.

Website administrators can view a Google Map that has a heatmap which shows where tasks have been created and what type of task.

2.1 Installation

The website is located at mogreen-1543284316825.firebaseapp.com.

You can log in using your existing Google account or an alternative email. If you are not assigned as an administrator, you will need to contact the website administrators to assign you as one.

At the time of this writing, the administrative website was developed using Windows 10. Use of the Command Prompt is needed. To install and host the "Clean My Campus" administrative website, you must have Firebase and npm installed. You must also have a Firebase project already started and a Google Maps API key.

1. To start a Firebase project, go to https://console.firebase.google.com
    a. Click "Go to console".
        i. Sign into the account you can associate with developing.
    b. Click "Add project".
        i. Name the project and set the other options as needed.
        ii. Click "Create project".
2. Install npm from https://www.npmjs.com/get-npm, the instructions are located on the site.
3. To get a Google Maps API key, follow the instructions at https://developers.google.com/maps/documentation/embed/get-api-key
4. Create a new folder, this folder will contain the website files eventually and will be your working directory. Example: "C:\Users\User1\Desktop\FirebaseWorkFolder"
5. Open command prompt.
    a. Install Firebase tools by entering "npm install -g firebase-tools".
        i. Enter "firebase login"
        ii. You will be asked to sign in to an account. This account should be the one that created the Firebase project or an account that has been added to the project.
    b. Navigate to the newly created folder.
    c. Enter "firebase init" and follow the instructions.
        i. Are you ready to proceed? Y
        ii. You will be asked to select which features you will be using. Follow the instructions given and select:
            1. Firestore.
            2. Hosting.
            3. Storage.

        iii. Select the Firebase project that was created by following Step 1.
        iv. Use all the basic rules by pressing Enter.
        v. You can name your public directory as anything you would like but anyone else who wishes to add to the project must name their directory exactly the same thing. “public” is default.
        vi. Configuring as a single-page app does not matter as we will be copying our source files over.
        vii. There is one more basic rule to enter then you are completed with initializing firebase for use in that folder.

6. In file explorer, open your “C:\Users\User1\Desktop\FirebaseWorkFolder” and you will see that this folder now contains some files and a folder, by default, named “public”.
        i. Navigate into the “public” file and delete any files inside.
7. Go to https://github.com/LugwellDR/cmcSrcFiles and download as zip, the project.
    a. Unzip the files
8. There are several files that need to be modified for a given Google Maps API key and configuration setup from the Firebase project. Open any editor to edit four HTML files and one JavaScript file.
    a. Your specific Google Maps API key needs to be copied into the script tag that looks like this.
        i. <script async defer src="https://maps.googleapis.com/maps/api/js?key=[GOOGLEAPIKEY]&libraries=drawing,visualization"></script>
    b. The four files that contain this script are:
        i. Home.html
        ii. cmcMapEditor.html
        iii. cmcHeatmap.html
        iv. cmcAbout.html
    c. Next, edit the domainConfig.js file.
        i. Navigate to the Firebase project console and click “Add app”.
        ii. Choose the web option and a code snippet will be shown. This information can be copied and pasted to replace the variables within the domainConfig.js file.
9. Copy all of files and folders into the “public” folder.
10. Return to the command prompt and navigate to the “public” folder. Example: “C:\Users\User1\Desktop\FirebaseWorkFolder\public”.
    a. Enter “firebase deploy”. This will upload the files to the Firebase project and will be readily available to the general public!

2.2 Configuration and Daily Operation

Starting at the index.html page, there isn't anything to configure. You can login using a Google account or by an alternative email. Once logged in, you will be brought to the Home page. The Home page displays a Google Map that represents exactly what application users will see when they open their application. There is a marker display option bar where a campus administrator can show or hide campus resources.

There are no configuration or daily operation needs for the About page.

The Heatmap/Statistics page displays the location and an icon to represent locations of reports made across the campus. You can also generate a heatmap based off the location of the task reports. This page also has an options bar where administrators can hide or show report type icons, customize the heatmap, and apply various filters. Administrators can create a heatmap by using the Generate Button, selecting the 'Heatmap' task type or a task status. When the heatmap is generated, you can modify the radius and intensity of the heatmap points on the Google Map. Selecting a time range is the last option you can choose to filter tasks and the heatmap.

The Task page displays all tasks sent to the database by the application. We can see the task type, date/time, description, reporter, and an optionally included image. By default, the current pending tasks populates the webpage when accessed. Users can view images of the task by clicking the Image button or the Popout button. Users can submit a task as complete by checking the box and then the Submit button. The administrator can apply various filters to help display information. Administrators can display only certain task types, a time range, and sort by timestamp or task type. You can also show completed tasks and apply all the aforementioned filters to the completed task list. You can submit completed task as currently pending by checking the box while looking at the completed task list and submitting. The final filter is Group Duplicates button which only applies to pending tasks. This button will group all tasks based on task type and those that are within ~100 ft of each other as a potential duplicate. This will be denoted by the symbol "|" being inserted into the front of the description. This should help administrators identify duplicate tasks quicker, but it is not 100% accurate.

The Map Editor page is where administrators can manage all things concerning zones, campus resource icons, zone color duration, and update the zone colors for application users. There is nothing to configure as there are default values already included. We have an options bar where administrators can hide or show markers and zones. This helps facilitate zone and marker placement by removing other markers or zones that may not be needed. Markers can be placed by simply choosing the type of marker you wish to place and clicking on the Google Map where that particular resource is located. Administrators can manually reset zone colors by checking the Reset Zone Color box then clicking the desired zones to have their color reset to green. The same process can be done but for deleting zones and markers. Checking the Edit Zone box will highlight each zone vertices. White circles represent current points while greyed circle will create an entirely new vertex. Once edits to zones have been made, you must Submit to save the changes to the database. Administrators can set the duration of how long each zone stays green, yellow, and orange by entering a number above zero and clicking submit for each color duration changed. These zone color changes will not take effect until after the Set button is clicked. Clicking the Apply button identifies all tasks that are contained in zones for later use when incrementing or decrementing time based on the type of task. The remaining text in the options bar shows when the last time an administrator has updated the zone colors for application users and the current duration color cycle. Zones can be created by selected the Draw a Shape button located at the bottom center of the Google Map. When that button is selected, administrators can draw any polygon shape they require. Upon completion of the zone, it will turn green and be immediately viewable by application users.

2.3 Maintenance

There is not much maintenance to do on the website. Tasks are required to be monitored daily and once all zones are created and markers placed, there no need to modify them unless resource locations change, or administrators decide to change zones. The database will eventually need to be cleaned out once in a few years, depending on software use. Storing images and many tasks will eventually increase database usage and storage monthly fees to unwanted amounts. The images in Firebase Storage and task in the Firebase Firestore are the two main ways to decrease database usage and storage usage.

2.4 Troubleshooting

Any issues with map loading may be related to Google Map API keys. The web portal utilizes a key with javascript permissions. Ensure that the Google Map API key is applied as described in Installation: Step 8.

There are still graphical issues on the website mostly located on the Map Editor page. It is recommended to refresh the Map Editor page if any issues appear or zones and markers seem unresponsive.

For multiple developers working on the same Firebase project, currently the source files need to be located externally, such as GitHub. If, during Installation: Step 5.c.iii. you do not see a Firebase project, then you will need to contact an account associated with the Firestore project so they can add your account to project for access.

3. Mobile Application

This section of the manual will explain how to set up, use, and maintain the application side of our software. For information on the web portal, please see page 1.

The mobile application component of our software is where all users (students, faculty, community members) can make reports about litter or maintenance issues, view zones on campus colorized to their status, view markers for campus resources, and mark an area as being cleaned.

Reports can be ‘quick’ or ‘full’. Quick reports include only a report type selected from a dropdown field, and the location of the report. Full reports include the same report type, location of the report, details about the report, and a photo of the event.

Zones on campus are drawn by administrators on the web portal, and displayed on the website. The color of the zone indicates the status of the zone, based on how much time has passed since it was marked clean.

Markers on campus are toggleable, and by default are not visible. When markers are toggled, the symbol for each resource shows on the application at the spot chosen by an administrator on the administrative website.

Users can mark an area clean through the same process as a quick report, then clicking the ‘mark as clean’ button. This will decrement the timer associated with the zone’s color.

3.1 Installation

The application is written using Java in Android Studio. To load the project we must first install all necessary softwares.

1. Download and Install the latest Java SE Development Kit.
2. Download and Install Android Studio including the SDK tools.
3. Open up Android Studio and click Project from Version Control.
    a. Select Git.
    b. Enter https://github.com/cbm97/MoGreenApp in the URL field.
    c. Choose a Directory to install the project in.
4. After this, you have successfully loaded the application! We will now need to get some keys to ensure that the application will operate correctly.
5. Navigate to the Firebase Console and choose the MoGreen project.
    a. Click on the Hamburger Button, then click on the Cog icon, then User Settings.
    b. In the “Your Apps” section, click on the button labeled “google-services.json”
    c. After downloading this file, Move the file to the ‘app’ folder.
6. Go to the Google API Console and select the MoGreen project.
    a. In the Hamburger Menu, choose APIs & Services then Credentials.

   b. Copy the latest Android Key (click on the square button next to or below the key to copy it to your keyboard).
   c. In the loaded project on Android Studio, locate the AndroidManifest.xml document.
      i. App > manifests > AndroidManifest
   d. Locate the meta-data named "com.google.android.geo.API_KEY"
   e. Paste the key in the section below named android:value=
      i. EX. android:value=”<API_KEY>”
7. In Android Studio, go to Build > Make Project.
8. When completed, you can load the application in two ways.
   a. Running the application on a plugged-in device.
      i. Plug in an android device to the computer running Android Studio.
      ii. Go to Run > Run ‘app’.
      iii. Select your phone from the device list.
   b. Building an APK then loading on device.
      i. In Android Studio, go to Build > Build Bundle(s) / APK(s) > Build APK(s).
      ii. A popup will appear near the bottom of the Android Studio window, Click Locate to pull up the APK.
      iii. Add this file to your Android Device.
      iv. From your Android Device, locate the APK from your File Explorer and run the APK to install the application.

3.2 Configuration and Daily Operation

The Google Play should be installed in order to do configuration on the application. Below is how you make a quick and full report on the application.

Quick Report

1. To make a quick report on the application, tap anywhere on the map to report a location
2. Simply click on the “running man” icon or click on the hamburger menu and select quick report
3. From the drop-down menu, you can select a type of report
4. Lastly, click on the “submit” button to submit your report

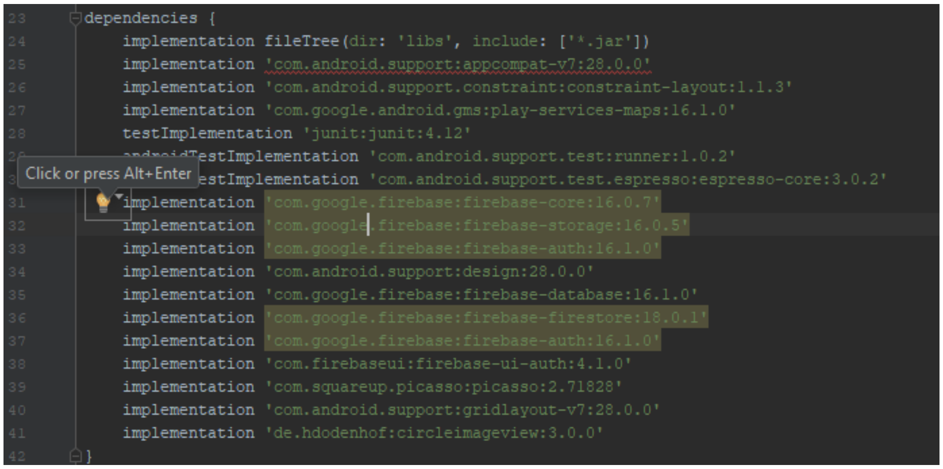
Full Report

1. To make a full report on the application, tap anywhere on the map to report a location
2. Simply click on the “walking man” icon or click on the hamburger menu and select full report
3. From the drop-down menu, you can select a type of report
4. In the “details” box add a description to describe the report

5. Optionally take a picture and attach it to your report
6. Lastly, click on the "submit" button to submit your report

3.3 Maintenance

The application will generally run just fine, however It may be a good idea to update the dependencies. These can be updated in the build.gradle (Module:app) file which can be found in the Gradle Scripts folder in Android Studio. To update a dependency, scroll down to the dependencies tag. If one of the dependencies is highlighted, click on that dependency then click on the lightbulb next to it. After clicking the lightbulb, click on the 'change' option, then go to Build > Make Project.



```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.android.gms:play-services-maps:16.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.google.firebase:firebase-core:16.0.7'
    implementation 'com.google.firebase:firebase-storage:16.0.5'
    implementation 'com.google.firebase:firebase-auth:16.1.0'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.google.firebase:firebase-database:16.1.0'
    implementation 'com.google.firebase:firebase-firestore:18.0.1'
    implementation 'com.google.firebase:firebase-auth:16.1.0'
    implementation 'com.firebaseui:firebase-ui-auth:4.1.0'
    implementation 'com.squareup.picasso:picasso:2.71828'
    implementation 'com.android.support:gridlayout-v7:28.0.0'
    implementation 'de.hdodenhof:circleimageview:3.0.0'
}
```

3.4 Troubleshooting

As the mobile application relies on so many different APIs, sometimes a key may need to be updated. If an error occurs where you are unable to log in while also using the correct username and password, you will need to update the keys for Firebase. To complete this, follow step 5 in the Mobile Application Installation guide.

If you are having issues using the camera or various permissions, make sure the application is allowed all permissions that it requests. You can enable or disable permissions on your mobile phone by going to Settings > Apps > MoGreen > Toggle Permissions.

There is a known bug for some phones that is relatively rare. This bug happens when the camera application has never been opened in the phone before and the camera will be unable to return a picture. This will only happen once if it happens on your device and it can be fixed by pressing the back button, then tapping the take picture button again.

4. Conclusion

With best wishes from your “Clean My Campus” developers: Cassie Graham, Chase Mosteller, Jennie Ryckman, and Justin Shaner. If you would like to contact us our emails are: cassie@nau.edu, cbm97@nau.edu, jtr93@nau.edu, and jms949@nau.edu. While we are all moving on to professional careers, we would be happy to answer short questions in the coming months to help you get the product deployed and operating optimally in your organization.