INSTITUTE of TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

Application Green Quake

Student Name: Peter Lucan
Student Number: C00228946
Supervisor: Chris Meudec

Github Link:
https://github.com/PeterX12/Application-Green-Quake.git
APK Github Link:
https://github.com/PeterX12/Application-Green-Quake/blob/master/ApplicationGreenQuake.apk
APK Google Drive Link:
https://drive.google.com/file/d/1Xe4Ay8YUTcEuXuQa1Zgj9Nlm97CIxCyJ/view?usp=sharing

Abstract

This document is the Design Manual for the Green Quake Application. It's aim is to document the design of the application and allow a new developer to develop this app just by reading this document. To do this all the necessary information has been documented in detail. The tools and technologies to be used are documented, the architecture of the project is documented, the screen designs are documented, the database structures are documented and finally the use cases and a plan on how to approach the project is documented. This allows any developer to develop the Green Quake application just using this document.

Table of Contents

1 Introduction

The purpose of Green Quake is to make it easier and more fun for people to be environmentally friendly. This is achieved by gamification of eco friendly activities and providing the necessary tools for people to be environmentally friendly such as the nearest water refill station locator tools.

The purpose of this document is to explore how the application is to be designed. This document will go through the technologies that are used in this project. Both the Prototype User Interfaces are shown as well as the Final User Interfaces. The schema of the database is laid out and explained and finally all the diagrams necessary to design this application are shown.

The document ends with the final Use Case diagram, all the detailed use cases and then states the plan to be followed throughout this project.

2 Tools and Technologies

2.1 Xamarin Forms

Xamarin Forms was chosen for this application as it is a language that allows the construction of cross platform applications. This means that one can write the majority of the code for the application once and share it between the two operating systems such as Android and iOS. XAML is used for the front end and C# is used for the back end.

Xamarin gives the developer an API for creating User Interfaces across many platforms [1]. This is implemented using XAML and C#. The cross platform elements are converted into native controls on Xamarin.Android, Xamarin.iOS and on UWP at runtime using platform renderers. This gives the native feel and performance to each platform while the developer can reap the benefits of the code being shared across the platforms.

A standard .Net library and individual platform projects make up a typical Xamarin application [1]. The shared library has views and any business logic such as services, models or other code written in XAML or C#.

2.2 Firebase

Firebase is a powerful tool that can be used for mobile applications and will be the database for this project. Firebase is a platform which was originally developed by Firebase in 2011 and was then bought by Google in 2014 and is maintained by them to this day.

Database provides authentication tools for mobile applications instead of having to store the login credentials into the same database as the rest of the data. As well as this the database of Firebase stores data in a JSON format.

2.3 Github

To backup the code of this project and to aid in the following of the agile methodology Github was chosen as the repository of the project. Github allows local git repositories to be pushed to remote repositories on Github. Regular commits will be pushed to this repository with each increment to the application to follow the agile guidelines.
The link to my Github: https://github.com/PeterX12/Application-Green-Quake.git

2.4 Nuget Packages

Below you will find a list of all the Nuget packages that will be used in this application. They all need to be installed for the app to function properly. More can be read on what each of these do as they are being downloaded in Visual Studio 2019 form Tools>Nuget Manage.

- **Acr.UserDialogs (Entire Project)**
- **FirebaseDatabase.net (Entire Project)**
- **FirebaseStorage.net (Entire Project)**
- **Microsoft.AppCenter.Analytics (Entire Project)**
- **Microsoft.App.Center.Analytics (Entire Project)**
- **NETStandard.Library (Entire Project)**
- **Plugin.CurrentActivity (Entire Project)**
- **Rg.Plugins.Popup (Entire Project)**
- **Syncfusion.Xamarin.SfPicker (Entire Project)**
- **Xam.Plugin.Media (Entire Solution)**
- **Xamarin.Android.Support.Annotations (Android)**
- **Xamarin.Android.Support.Compat (Android)**
- **Xamarin.Android.Support.Core.UI (Android)**
- **Xamarin.Android.Support.Core.Utils (Android)**
- **Xamarin.Android.Support.Design (Android)**
- **Xamarin.Android.Support.Fragment (Android)**
- **Xamarin.Android.Support.VersionedParcelable (Android)**
- **Xamarin.Android.Volley (Android)**
- **Xamarin.AndroidX.Browser (Android)**
- **Xamarin.AndroidX.Legacy.Support.V4 (Android)**

- **Xamarin.AndroidX.Lifecycle.LiveData (Android)**
- **Xamarin.Essentials (Entire Solution)**
- **Xamarin.Firebase.Auth (Entire Solution)**
- **Xamarin.Firebase.Common (Entire Solution)**
- **Xamarin.Firebase.Core (Entire Solution)**
- **Xamarin.Firebase.Database (Entire Solution)**
- **Xamarin.Firebase.iOS.Auth (iOS)**
- **Xamarin.Firebase.iOS.Database (iOS)**
- **Xamarin.Forms (Entire Solution)**
- **Xamarin.Forms.GoogleMaps (Entire Solution):**
- **Xamarin.Google.Android.Material (Android)**
- **Xamarin.Google.iOS.Places (Entire Solution)**
- **Xamarin.GooglePlayServices.Maps (Entire Solution)**

2.5 APIs

The API's used in this project were the Firebases API,the Google Maps API and the APP Center API. These allowed us to use Firebase database, firebase authentication and firebase storage, Google Maps and the app centers app monitoring functionality.

3 Best Practices and Architectures

In order to create a successful project the best practices must be followed. These practices are outlined below.

3.1 Best Coding Practices

When it comes to some of the best practices out there, writing comments in code and constantly updating your documentations as you advance with your project is the key to success [3]. Writing Readable and efficient code is also very important. Sometimes it is tempting to write everything in as few lines as possible but this can make it difficult to read. Writing test cases, using helper methods and avoiding hard coding variables are some more important practices to follow but the most important coding practice to follow is to save your work after each incrementation to the project.

In order to construct a successful project all of the above will be followed. Github will be used as a backup for the code. Code will be committed and pushed with every incrementation to the project.

3.2 Xamarin Forms Coding Standards

The best coding standards for Xamarin Forms will be followed to write clean code. These are as follows [4].

- Only assign one page to one View Model.
- Do not pass view modes to methods of other pages.
- All View Models must have an InitializeComponent().
- Do not end xaml file names in view.

3.3 C# Coding Standards

The best coding standards for C# will also be followed to write clean code. These are as follows [5].

- Use Pascal Case for method names and class names.
- Use Camel Case for local variables and method arguments.
- Do not use Hungarian Notation.
- Do not use Capital Case letters only for constant variables.
- Do not use Underscores in identifiers.
- Prefix interfaces with the letter I.
- Name source files according to their main classes.
- Declare all member variables at the top of a class, with static variables at the very top.
- Correctly align brackets.

3.3 Model-View-ViewModel (MMVM) Architecture

This project will follow the MMVM design pattern. The MMVM design pattern consists of the Model, the View and the View Model [6]. The View informs the ViewModel about the user's actions, The ViewModel exposes streams of data relevant to the View and the Model abstracts the data source. The ViewModel works with the DataModel to get and save the data. A visual representation of this model can be viewed below.

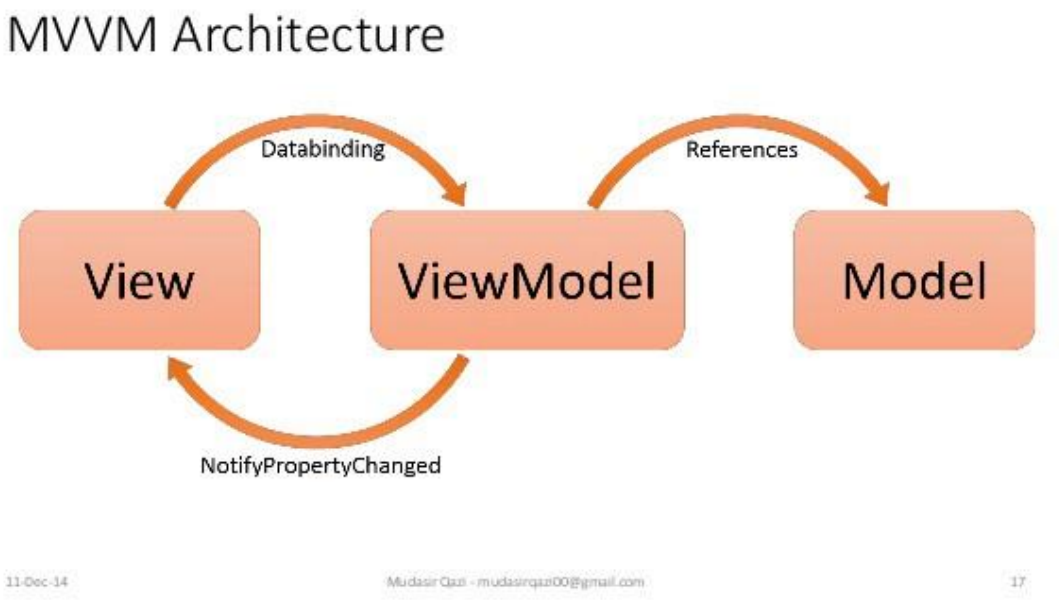


Figure 1: A Visual Representation of the MVVM architecture [7].

4 Prototype UI-UX Design

4.1 Color Palette

The color palette used for the prototype screens can be seen below.

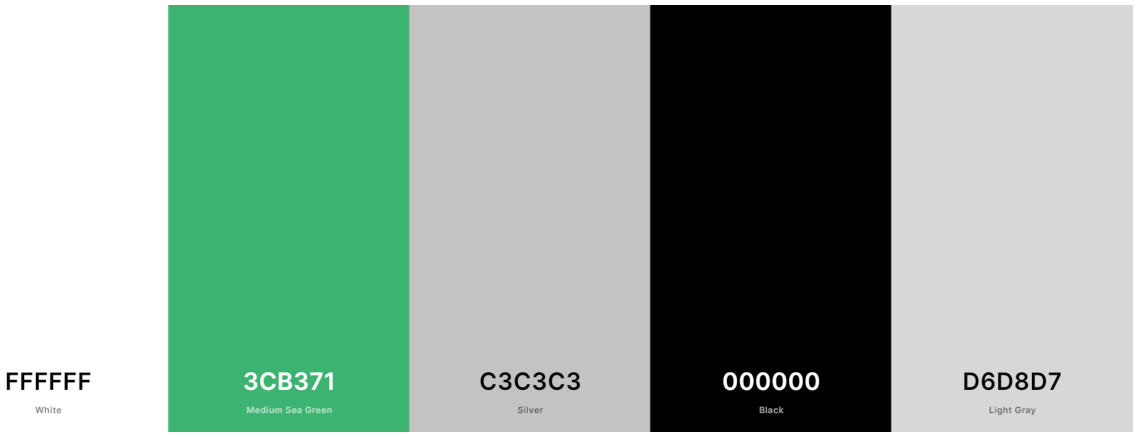


Figure 2: The colours used for the Prototypes.

4.2 Prototype Screens

Prototype screens that will be used for the first few iterations of the project are shown below. These prototypes were designed just to serve as screens that implement the back end as the back end is being developed. The final deliverable will change massively as the front end UI design has not been done at this present moment. A few of the Prototype screens are shown below.

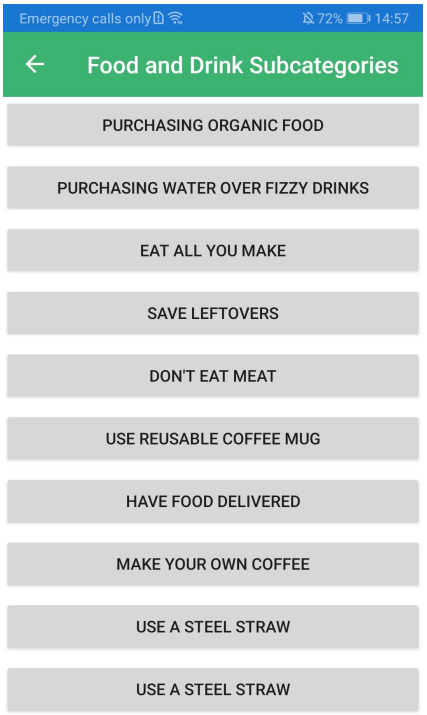


Figure 3: Food and Drink Subcategories Screen.

The image above shows the screen that presents the options of actions to choose from for the sub category page of Food and Drink. As you can see the design is very basic.

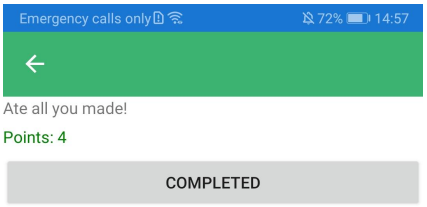


Figure 4: Eat All You Make Action Screen.

As you can see in the image above. The Eat All You Make option was selected from the previous image and the image above is the screen that appears. On this screen the user can log their action.

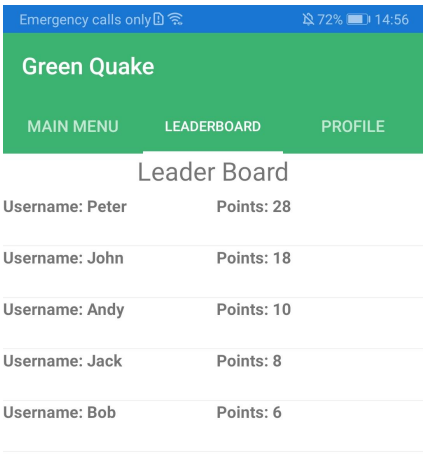


Figure 5: Leaderboard Screen.

The image above is the prototype of the leaderboard which shows the username and the points of the users. A user is also able to navigate to the Main menu and profile screen from here.

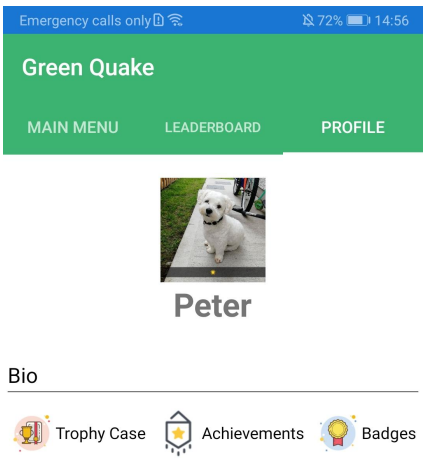


Figure 6: Profile Screen.

The image above is the prototype of the profile screen. It has the profile picture and the bio and three buttons to navigate to Trophies, Achievements and Badges. The user can change the image by tapping on it and change the bio by tapping on it.

5 Final UI-UX Design

The following design was constructed with the aid of research on UI, UX and Eco Friendly Activities and the best colours to use to suit the theme of our project. This research is mentioned in the Research Report.

5.1 Color Palette

After much research the perfect colours for the application were found that correspond nicely to the applications theme and these colours can be seen below. These colours were decided upon as research was conducted on the best designs and colours to use for eco-friendly products.

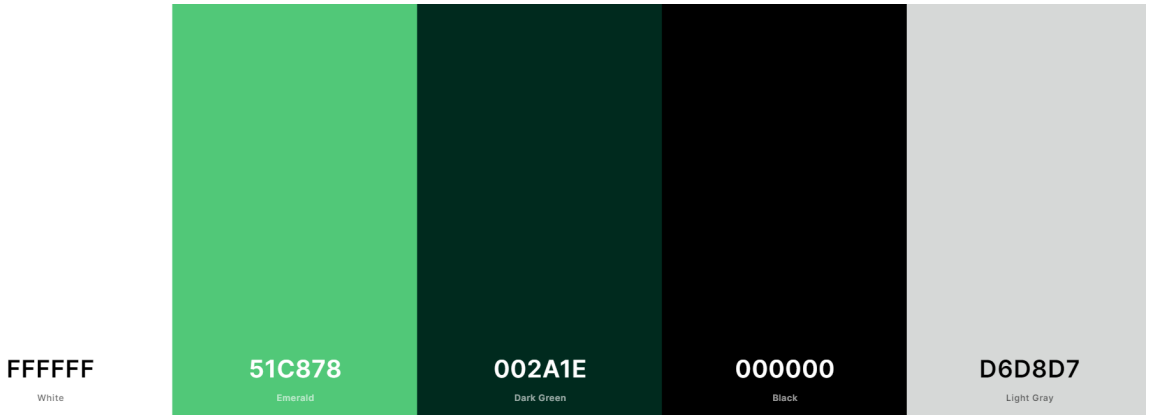
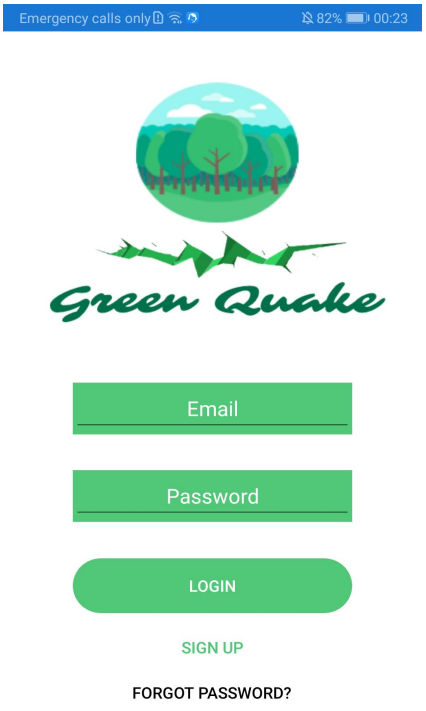


Figure 7: The colours used for the Final Design.

5.2 Final Screens

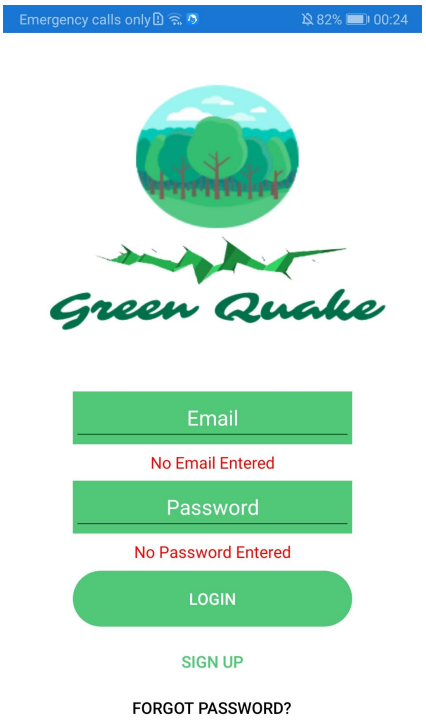
This section shows the final design for the screens to be used in the Green Quake mobile application and some details about them. It also shows the workflow of the application. These final designs can be viewed below.



As you can see this is the **login** screen**.** This will be the opening screen for the application when the user is not signed in. A user can enter their email and password and click the login button to attempt to sign in or they can tap the Sign up option to navigate to the sign up page or they can tap on the Forgot Password? Option to navigate to the Forgot Password screen.

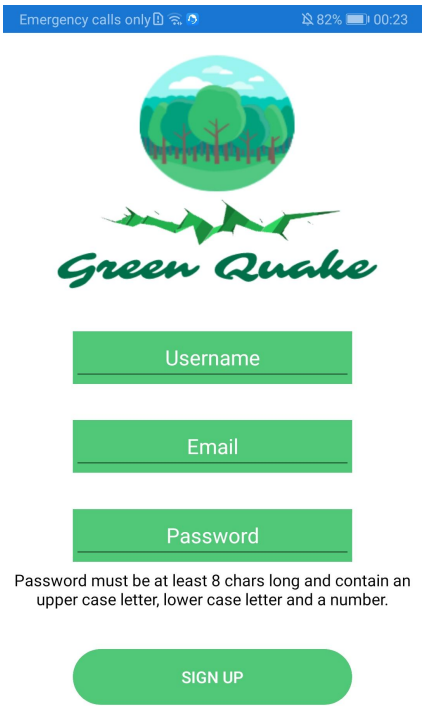
Upon successful login a nice splash screen appears which can not be documented as it is an animated event. After this splash screen the user is brought to the main menu.

Figure 8: Login Screen.



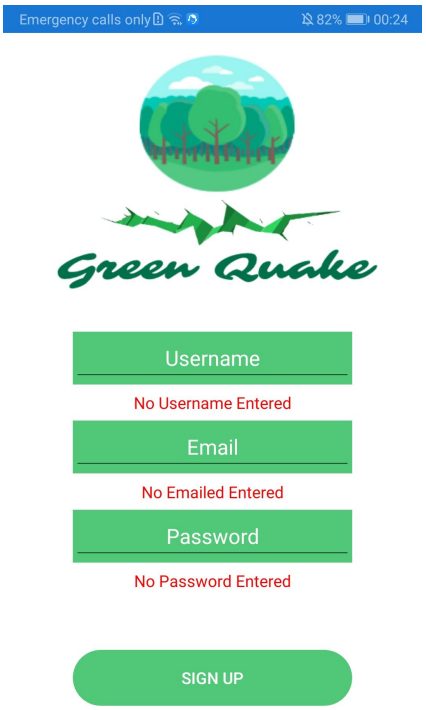
As you can see this is the **login** screen once more but now displays error messages for the invalid input fields. These error messages vary depending on the situation. If the user is not connected to the internet an alert pops up asking them to connect to the internet.

Figure 9: Login Screen With Error Messages.



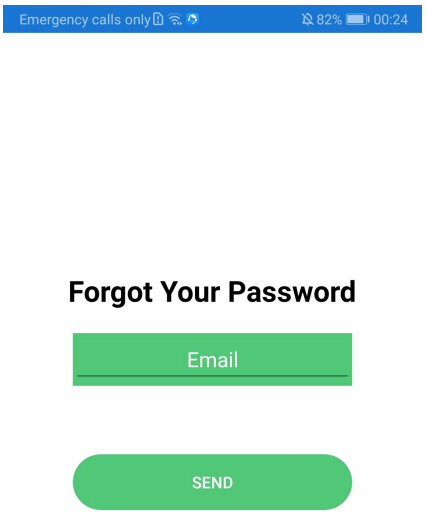
As you can see this is the **Sign Up** screen. A user can enter any username they wish and only a valid non duplicate email and a password which complies with the rules listed under the password fields. Upon successful signup an alert pops us saying your account has been created and redirects the user to the login page.

Figure 10: Sign Up Screen.



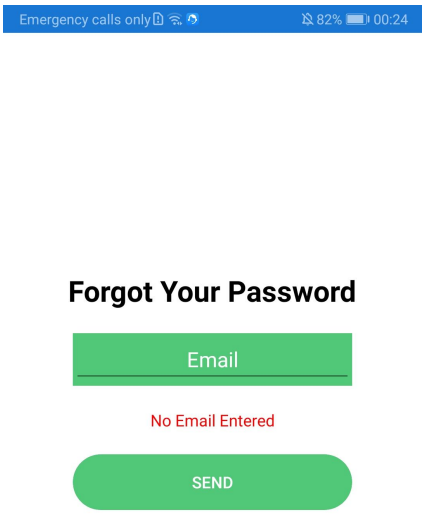
As you can see this is the **Sign Up** screen once more but now displays error messages for the invalid input fields. These error messages vary depending on the situation. If the user is not connected to the internet an alert pops up asking them to connect to the internet. If the email already exists an alert pops up saying the email already exists and if there is an error a generic error alert pops up.

Figure 11: Sign Up Screen With Error Messages.



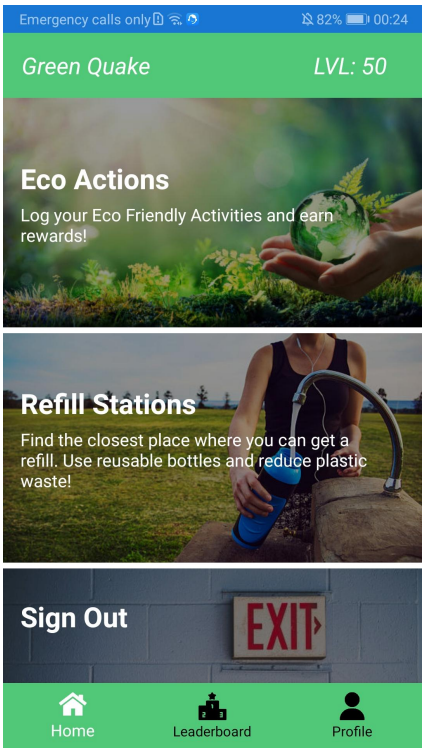
As you can see this is the **Forgot Password** screen. A user can enter their email and if the email exists a password reset email will be sent to that email address. An alert saying “If the email exists a password reset email has been sent to your email address”. Then the user is redirected to the login page.

Figure 12: Forgot Password Screen.



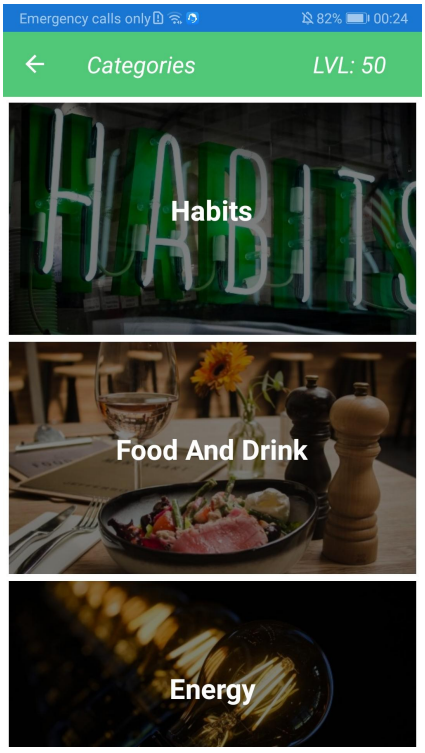
As you can see this is the **Forgot Password** screen once more but now displays an error message for the invalid input field. If the user is not connected to the internet or the email is invalid the corresponding alert box pops up.

Figure 13: Forgot Password Screen With Error Messages.



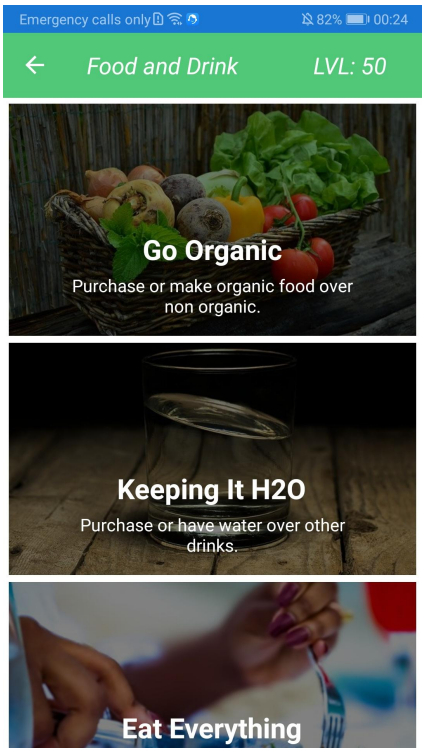
This is the **main menu** screen that appears after successful login. The user can navigate to the Eco Actions section by tapping on it and slo to the Refill Station section by tapping on it. In addition the user can navigate to the Leaderboard or Profile section using the navigation bar at the bottom of the page. The level of the user is displayed in the top right of the screen.

Figure 14: Main Menu Screen.



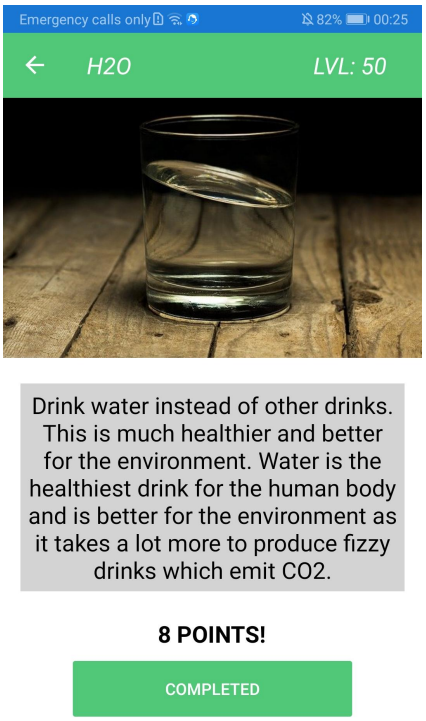
This is the **Categories** screen that appears after a user selects Eco Actions from the previously mentioned main menu. From here the user can select one of 12 categories to log and eco action for. These categories are: Habits, Food And Drink, Energy, Travel, Shopping, Water, Home, Outdoors, Community, Waste, Work and Advanced categories.

Figure 15: Categories Screen.



This is the **Food and Drink** subcategories screen that appears after a user selects Food and Drink from the previously mentioned categories screen.

Figure 16: Food and Drink Subcategories Screen.



This is the **H20** action screen. This screen provides some details and interesting information about this particular action. The amount of points that it is worth is displayed below the text. The user can press the Completed button to log the action. Once this is pressed either the points get posted into the database and an alert box with the successful message is displayed to the user before redirecting them to the main menu or one of two error message alert boxes pop up.

The application does not allow more than 15 posts per day and an alert box saying this will be shown to the user and the post will not be submitted if the user tries to submit more than 15 actions per day. The app also only allows one submission per minute to prevent spamming and an alert box saying this also pops up for the user and does not submit the post.

Figure 17: H20 Action Screen.

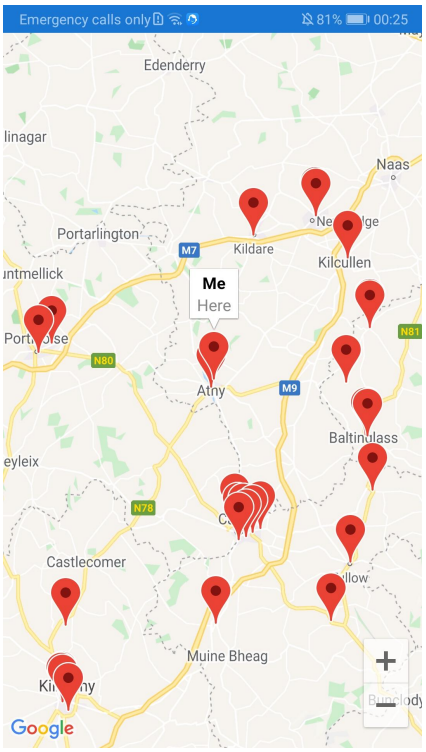


Figure 18: Refill Stations Screen.

This is the **Refill Stations** screen that appears after a user selects Refill Station from the previously mentioned main menu. This screen loads a map at your current location and displays all the Refill Station on the map. The user can clearly see where the refill stations are located in retrospect to their location and can tap on each pin to view more details about it.

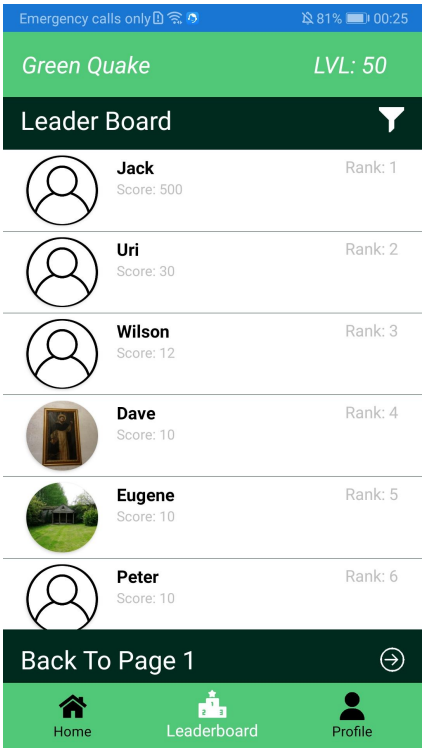
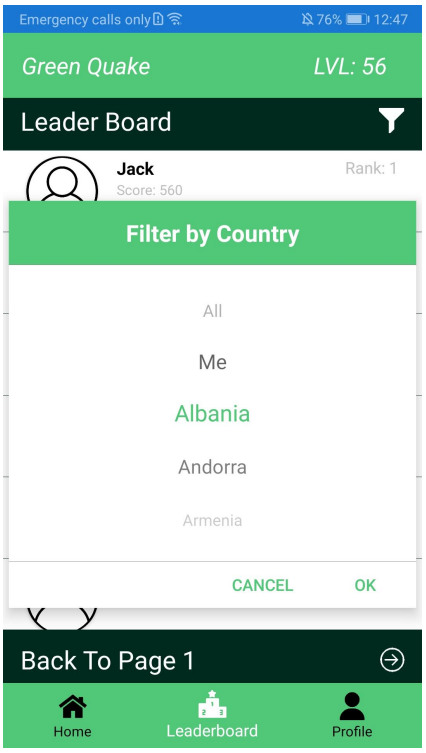


Figure 19: Leaderboard Screen.

This is the **Leaderboard** screen that appears after a user selects the Leaderbaord from the navigation menu at the bottom of the screen on the previously mentioned main menu. On this screen the leaderboard displays 10 rows per page. The user can tap on a profile to view more information about the user. The user can also tap the icon in the top right to filter the leaderboard. The user can also tap the right arrow at the bottom right of the page to see the next 10 entries and tap the Back to Page on text to return to the first page. If there are no more pages to load then an alert box with the appropriate message gets displayed.

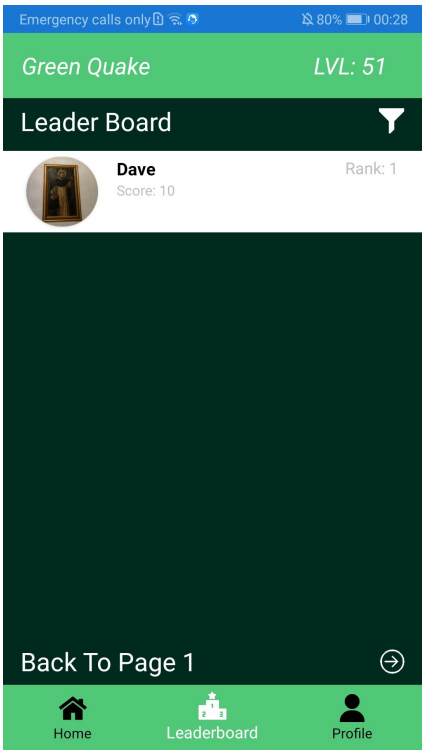


This is the **Leaderbaord** screen when the **filter icon** is tapped. A picker with the options All, Me and the list of nations appears. The user is allowed to filter to display all the users on the leaderboard, filter by nation or view their own position on the leaderboard global.

Figure 20: Leaderboard Screen When The Filter Button Is Tapped.

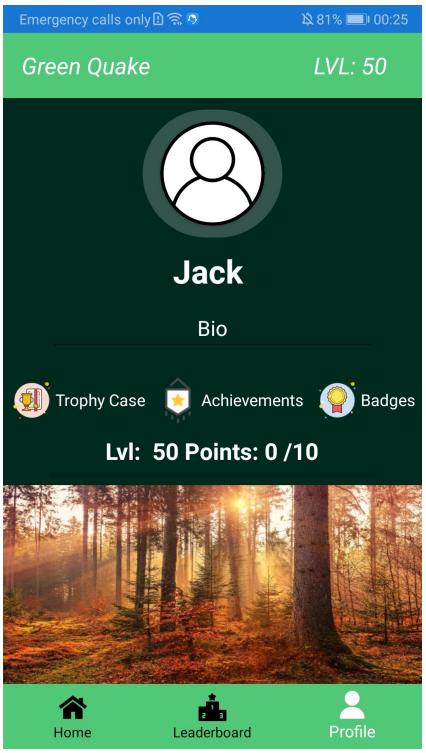
This **popup** screen appears when a profile is tapped on from the leaderboard. It displays more information about the tapped on user. The information displayed is the username, the profile picture, the rank, the bio and the points.

Figure 21: Leaderboard Pop Up Screen When A Profile Is Tapped On The Leaderboard.



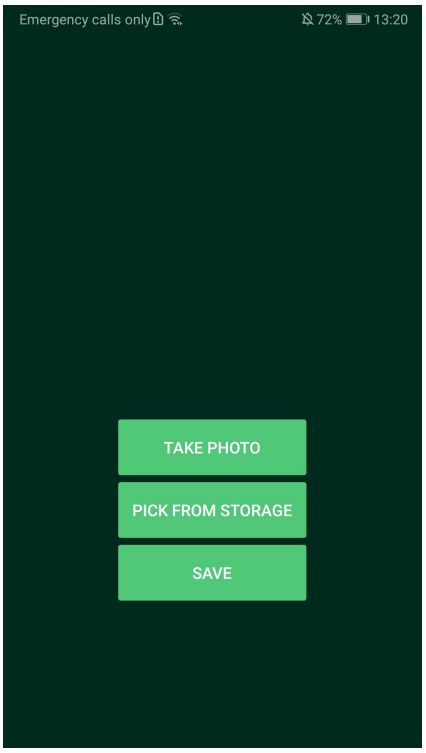
This is the **Leaderboard** after it has been **Filtered by Nation**. Only the profiles with the correct nation are displayed and they are ranked against other accounts of the same nation.

Figure 22: Filtered Leaderboard Screen.



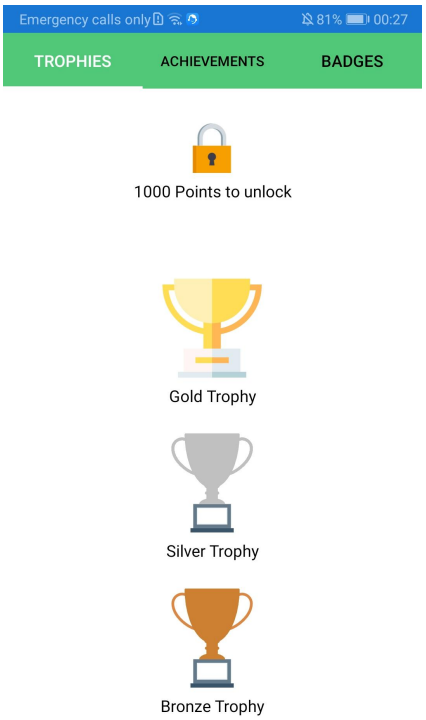
This is the **Profile** screen that appears after a user selects the Profile from the navigation menu at the bottom of the screen on the previously mentioned main menu. The Profile screen has the profile picture and bio that the user can tap to change and their username. Below this are the Trophies, Achievements and Badges. By tapping on each of these the corresponding screen is displayed showing the Trophies, Achievements and Badges. Below this, the level of the user is displayed with a progress bar to the next level indicating how many points out of 10 the user needs to level up. Finally below this is the live avatar that is a mosaic that gets unlocked piece by piece on each level up. The mosaic gets unlocked after each level so each mosaic has 5 stages and there are 20 mosaics to unlock which makes them end at lvl 100.

Figure 23: Profile Screen.



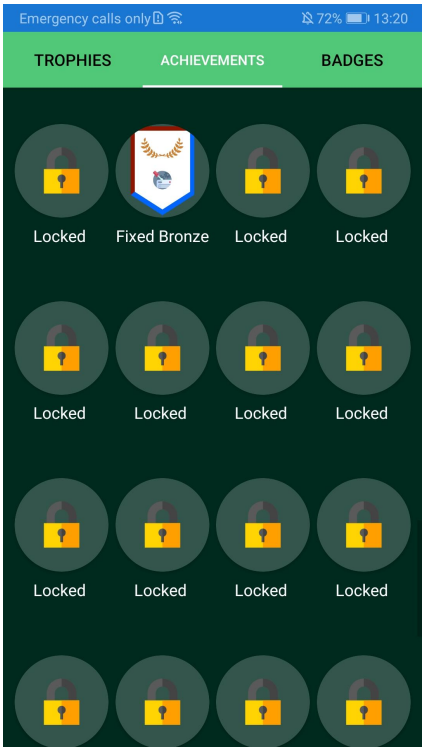
This **popup** screen appears when the user taps on the profile image on the profile screen. This popup allows the user to take a photo and save it as their profile picture or pick an image from the phone's storage and save it as their profile picture. Then the user is redirected to the profile screen.

Figure 24: Popup Screen After The Profile Image Is Tapped.



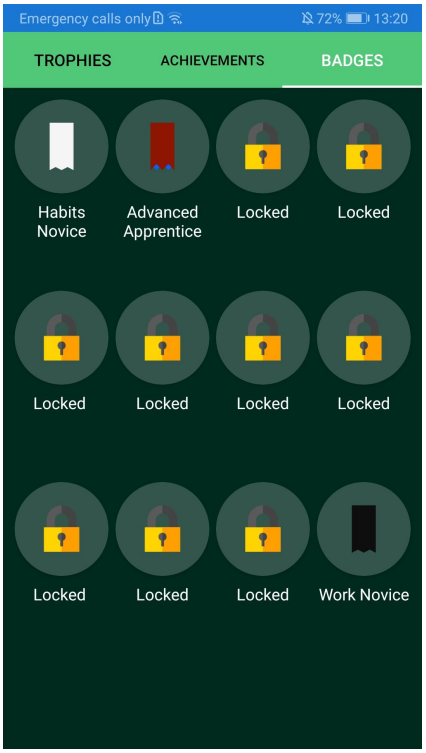
This is the **Trophies** Screen that can be navigated to from the profile screen by tapping on the trophy case icon and text. The trophies are initially locked and get replaced with trophies as the requirements get met. Bronze trophy for 100 points, Silver for 250, Gold for 500 and Diamond for 1000.

Figure 25: The Trophies Screen.



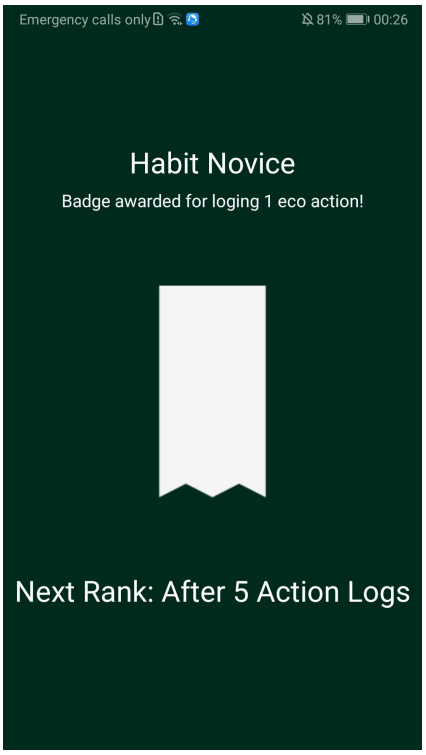
This is the **Achievements** Screen that can be navigated to from the profile screen by tapping on the achievements icon and text. There are achievements for every action that can be logged which makes it over 80 achievements that the user can earn. In addition to this these are live achievements which can be bronze, silver and gold so there are over 260 achievements to earn in total. The achievements are initially locked and get unlocked by meeting the qualifying requirements. A bronze achievement is unlocked when a certain action gets logged 5 or more times, a silver achievement is unlocked when a certain action gets logged 15 or more times and finally a gold achievement is unlocked when a certain action gets logged 25 or more times.

Figure 26: The Achievements Screen.



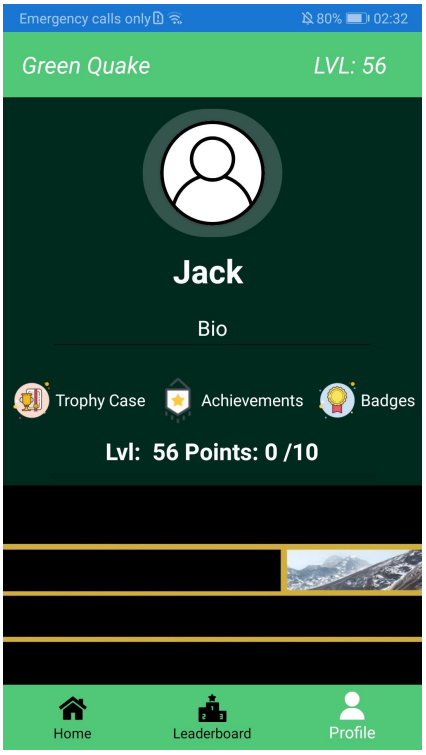
This is the **Badges** Screen that can be navigated to from the profile screen by tapping on the badges icon and text. There are badges for every category that a user makes a login. There are 12 categories and there are 12 badges that can be unlocked and displayed. These are live badges and there are 6 levels in each category making it 72 badges that the user can earn. Each badge has a different design. The badges are initially locked and get unlocked when certain conditions are met. The Novice badge for a certain category gets unlocked when the user makes a single log or more in that category. Apprentice badge for 5 or more, Adept badge for 10 or more, Expert badge for 25 or more, Master badge for 50 or more and Legend Badge for 100 or more. All these badges can also be tapped on to get a close up and view more information.

Figure 27: The Badges Screen.



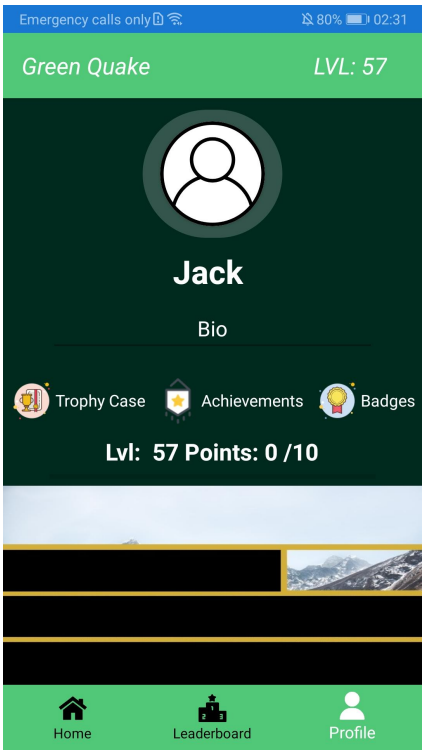
This **Popup** screen appears when a user taps on the badges on the badges page. This image displays a close up of the badge and a description for the badge and what it was awarded for. It also tells the user the requirements to get the next badge.

Figure 28: The Badges Screen Popup When A Badge Is Tapped.



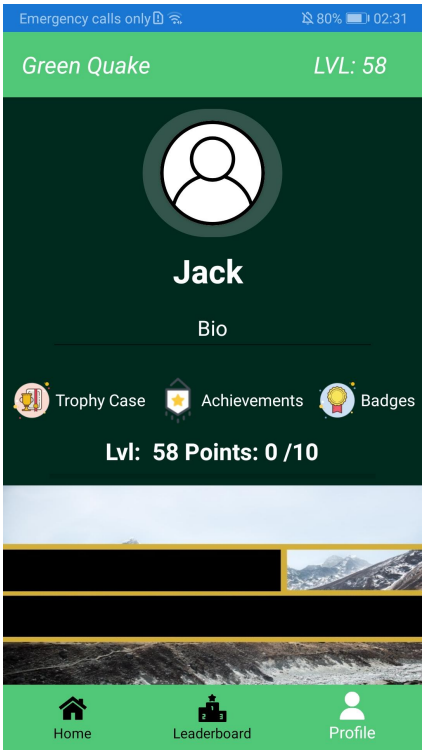
As you can see the mosaic only has one piece unlocked on level 56.

Figure 29: The Live Avatar Mosaics on Lvl 56.



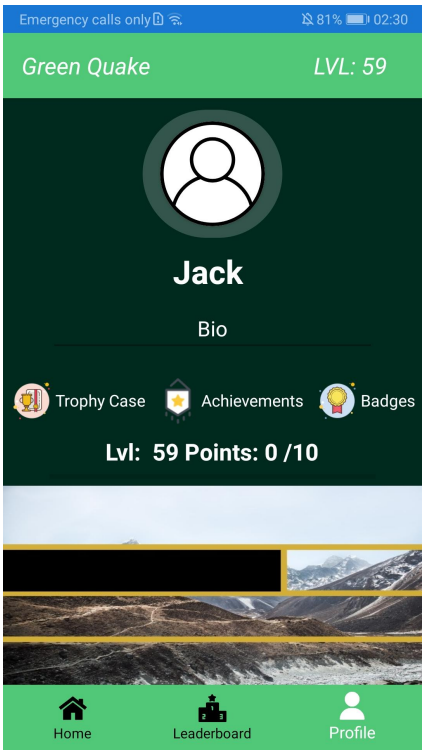
As you can see the mosaic only has two pieces unlocked on level 57.

Figure 30: The Live Avatar Mosaics on Lvl 57.



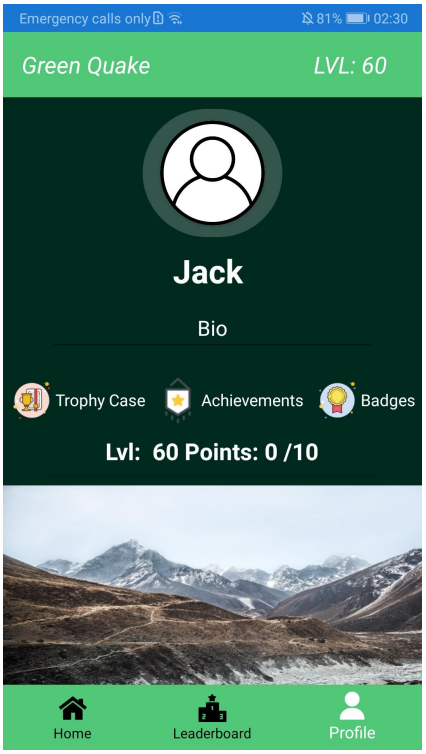
As you can see the mosaic has three pieces unlocked on level 58.

Figure 31: The Live Avatar Mosaics on Lvl 58.



As you can see the mosaic has four pieces unlocked on level 59.

Figure 32: The Live Avatar Mosaics on Lvl 59.



As you can see the mosaic has all pieces unlocked on level 60. The process repeats itself every 5 levels with a new image each time all the way to level 100.

Figure 33: The Live Avatar Mosaics on Lvl 60.

5.3 Trophy Designs

There will be four trophies in this application. These trophies have been taken from Flaticon and are free to use [8]. They were then modified for this application. These trophies will be earned for reaching 100 points, 250 points, 500 points and 100 points. These Trophies can be seen below.

Figure 32: Bronze Trophy For Reaching 100 Points.

Figure 33: Silver Trophy For Reaching 250Points.

Figure 34: Gold Trophy For Reaching 500 Points.

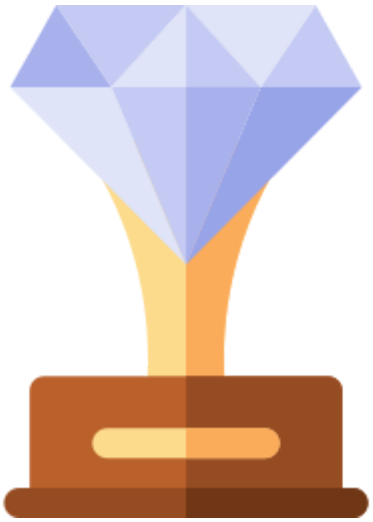


Figure 35: Diamond Trophy For Reaching 1000 Points.

5.4 Badge Designs

There will be 12 live badges for each category of eco friendly actions that a user can log. These Badges will be themed by color to the category that they represent. Each badge will have 6 stages so there will be 72 badges in total. These badges have all been manually designed and made for this project. The badges are initially locked and get unlocked when certain conditions are met. The Novice badge for a certain category gets unlocked when the user makes a single log or more in that category. Apprentice badge for 5 or more, Adept badge for 10 or more, Expert badge for 25 or more, Master badge for 50 or more and Legend Badge for 100 or more. These 6 Badges for the Community Category can be seen below. Each badge transforms to the next one when the user earns enough points.

Figure 36: Novice Badge For Logging First Action In The Community Category.

Figure 37: Changes To Apprentice Badge When Reaching 5 Actions In The Community Category.

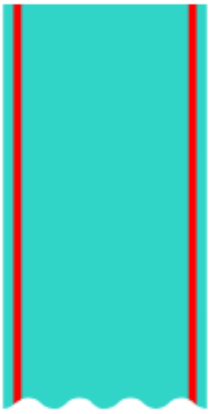


Figure 38: Transforms To Adept Badge When Reaching 10 Actions In The Habits Community

.

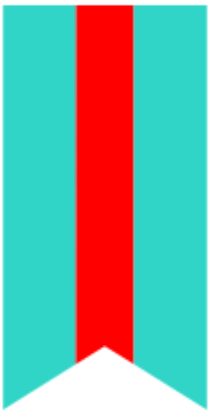


Figure 39: Transforms To The Expert Badge When Reaching 25 Action In The Community Category.

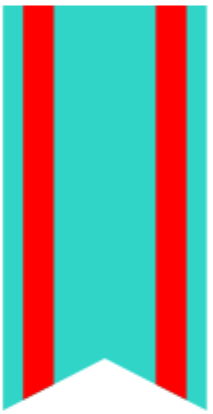


Figure 40: Transforms To The Master Badge When Reaching 50 Action In The Community Category.

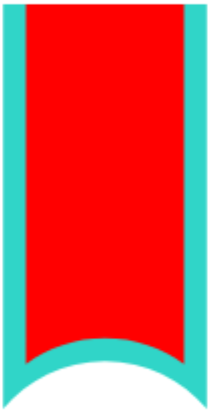


Figure 41: Changes To The Master Badge When Reaching 100 Actions In The Community Category.

As mentioned before, these badges exist for the 12 different categories and are all different colours to the ones above as the ones above are only the badges for the Community Category.

5.5 Achievements Designs

The user will have the chance to earn an achievement for logging every activity in the application. There are 80+ activities to log and there are gold, silver and bronze versions of the achievements making it possible to earn over 260 badges. An Example of these achievements can be seen below. The badges below are the Gold, Silver and Bronze achievements for hang drying clothes.

Figure 42: Gold Badge For Hang Drying Clothes.

Figure 43: Silver Badge For Hang Drying Clothes.

Figure 44: Bronze Badge For Hang Drying Clothes.

5.6 Live Avatar Design

The Profile Page will have a live avatar at the bottom on the screen. This avatar will be a mosaic that will start off as a black screen and will unlock a piece after each level up in the application up to 5 levels each time and then starting over with a new image to unlock. The level is calculated by dividing the users points by 10 and ignoring the decimal numbers. The decimal numbers are then always taken and used to display the progress bar above the image indicating the progress and saying how many more points are needed to unlock the next level.

6 Database Structure

6.1 A Good Example

In order to create a successful application that works well with user data, it needs a database. This database must store and retrieve data efficiently and safely. In Firebase this is achieved by keeping the data shallow so only the data that is needed is queried [2]. In contrast, if the data was nested it would be necessary to loop through all the nested data preceding the element that is being searched for. The Green Quake Application will implement and maintain a shallow structure. Below an example of a nested database structure and a shallow database structure is displayed.

Figure #: Example of Nested Data [2].

In the image above, the data is nested. If likes were to be retrieved the query would have to loop through everything that comes before it and this is highly inefficient.

Figure #: Example of Shallow Data [2].

In this instance, the image above represents data stored in a shallow manner. This allows the query of child elements without the need of looping through all the data. If likes are to be

queried only likes and/or comments have to be queried. This is why the shallow approach will be taken in the design of the database for this application.

6.2 Green Quakes Database Structure

As previously mentioned the database structure has to be shallow in order to optimize query speed so this is exactly what has been implemented in this projects database. The layout of the database is displayed below. The actual database is of the JSON format.

application-green-quake-default-rtdb

-|AdvancedPoints

-|UID

-|fixCount

-|numberOfLogs

-|points

-|username

-|EnergyPoints

-|UID

-|draftSealCount

-|ductSealCount

-|efficientThermostatCount

-|fridgeCount

-|fullDryerCount

-|fullMachineCount

-|hangDryCount

-|insulateWaterCount

-|isolateHomeCount

-|ledLightBulbCount

-|microwaveCount

-|numberOfLogs

-|offSocketCount

-|points

-|reBatteriesCount

-|solarPanelCount

-|username

-|FoodAndDrinkPoints

-|UID

-|eatAllCount

-|foodDeliverCount

-|noMeatCount

-|numberOfLogs

-|organicCount

-|ownCoffeeCount

-|points

-|reCoffeeMugCount

-|saveLeftOversCount

-|steelStrawCount

-|username

-|waterOverFizzyCount

-|HabitsPoints

-|UID

-|brushingCount

-|fullWasherCount

-|matchesCount

-|numberOfLogs

-|offLigtsCount

-|points

-|showerCount

-|timedShowerCount

-|username

-|Points

    -|UID

        -|points

        -|username

-|SecuritySchecks

    -|UID

        -|counter

        -|date

        -|time

-|Station

    -|ID

        -|description

        -|label

        -|latitude

        -|longitude

-|Travel

    -|UID

        -|carpoolCount

        -|cycleCount

        -|ecoCarCount

        -|numberOfLogs

        -|points

        -|transportCount

        -|username

        -|walkCount

-|WastePoints

-|UID

-|billsCount

-|bioBinBagsCount

-|compostCount

-|numberOfLogs

-|points

-|recyclingBinCount

-|setUpRecyclingBinCount

-|username

-|WorkPoints

-|UID

-|numberOfLogs

-|offElectronicsCount

-|paperCount

-|points

-|remoteWorkCount

-|username

-|usernames

-|username

-|Uid

-|Users

-|UID

-|bio

-|nation

-|username

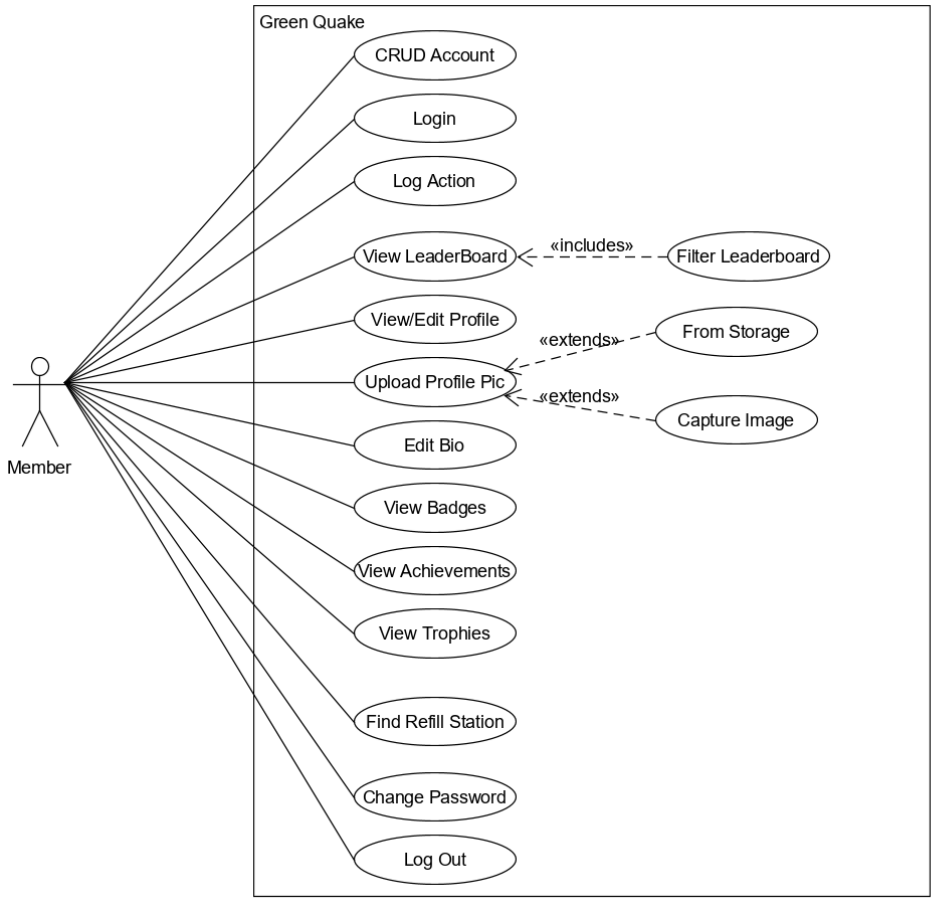
Firebase Storage is used to store images for users and it's structure can be seen below:

-|gs://application-green-quake.appspot.com

-|UID

-|filename

7 Use Case Diagram



8 Detailed Use Cases

8.1 CRUD Account

Use Case Name: CRUD Account

Actor(s): Member

Main Success Scenario: The use case begins when the member wishes to create, read or update their account.

1. A member that is not logged in launches the application.
2. The application displays an option to log in or create an account.
3. The member chooses the create an account option and the application loads the create and account page with a form for the member to fill in.
4. The member enters their username, email and password into the form and submits it.
5. The application validates the fields and creates the members account and redirects them to the main menu page.

6. The member clicks their profile icon where they can perform the read or update option.
7. The members can input their bio and upload an image.
8. The member can choose to delete their account by choosing the “Delete Account Option” and confirming their decision.

Alternative(s):

5a. Validation fails and the account is not created

1. An error message is displayed to the member .
2. The form is reset and the member can try again.

8.2 Login

Use Case Name: Login

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to log into the application.

1. The actor launches the application.
2. The application displays an option to log in or create an account.
3. The actor enters their email and password and submits the form.
4. The application verifies the user.
5. The actor is then logged into the application and is redirected to the main menu.

Alternative(s):

5a. The verification fails and the actor does not get logged in

1. An error message is displayed to the member
2. The form is reset and the member can try again.

8.3 Log Action

Use Case Name: Log Action

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to log eco-friendly actions they have performed such as planting a tree or using a reusable bag instead of a plastic bag.

1. The actor chooses the Eco Actions option from the main menu.
2. The application loads a page of categories for the actor to choose.
3. The actor chooses a category in which they want to log an activity.
4. The application loads a page with a list of eco friendly activities.
5. The actor chooses an activity they wish to log.
6. The application displays the activity and it’s details.
7. The actor clicks log activity.

8. The application redirects to the main menu page.
9. The application saves and calculates the actors points, achievements, badges and trophies.

Alternative(s):

8a. The log activity action is stopped because the user tries to log more than 15 actions in a day.

1. An error message is displayed to the member.

9a. The log activity action is stopped because the user tries to log more than 1 action in less than a minute.

1. An error message is displayed to the member.

8.4 View Leaderboard

Use Case Name: View Leaderboard

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to view the leaderboard and more details about individual entries in it.

1. The actor navigates to the leaderboard section using the navigation bar..
2. The leaderboard is displayed.
3. The leaderboard taps on an entry.
4. The tapped users details are displayed.

Alternative(s): No Alternatives.

8.5 Filter Leaderboard

Use Case Name: Filter Leaderboard

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to filter the leaderboard.

1. The actor navigates to the leaderboard section using the navigation bar.
2. The leaderboard is displayed.
3. The actor chooses the filter on the top right of the pages.
4. The actor selects a nation from the dropdown.
5. The leaderboard with entries from only that nation are displayed.
6. The actor selects “Me” from the dropdown.
7. The leaderboard with only the user's details is displayed indicating their rank and details.

Alternative(s): No Alternatives.

8.6 View/Edit Profile

Use Case Name: View/Edit Profile

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to view or edit their profile.

1. The actor navigates to the profile section using the navigation bar.
2. The profile is displayed.
3. The user can tap on the image to change it.
4. The user can tap on the bio to change it.

Alternative(s): No Alternatives.

8.7 Upload Profile Pic

Use Case Name: Upload Profile Pic

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to

1. The actor taps on the profile picture on the profile page
2. The application displays an option to Take a photo or choose on from storage.

Alternative(s): No Alternatives.

8.8 From Storage

Use Case Name: From Storage

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to upload a profile picture from their gallery.

1. The actor taps on the profile picture on the profile page.
2. The application displays an option to Take a photo or choose on from storage.
3. The actor chooses the From Storage option.
4. The application redirects to the phones gallery.
5. The user selects an image and presses Save.
6. The app saves the image to the database and redirects to the profile page where the new image is now displayed.

Alternative(s): No Alternatives.

8.9 Capturing Image

Use Case Name: Capturing Image

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to log

1. The actor taps on the profile picture on the profile page.
2. The application displays an option to Take a photo or choose on from storage.
3. The actor chooses the Take Photo option.
4. The application opens the users camera.
5. The actor takes an image and presses save.
6. The app saves the image to the Firebase database and redirects to the profile page where the new image is now displayed.

Alternative(s): No Alternatives.

8.10 Edit Bio

Use Case Name: Edit Bio

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to edit their bio.

1. The actor navigates to the profile section using the navigation bar.
2. The profile is displayed.
3. The user taps on the bio.
4. The user enters their bio and presses the back button on their keyboard
5. The system saves the bio and saves it to Firebase.

Alternative(s): No Alternatives.

8.11 View Badges

Use Case Name: View Badges

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to view their badges.

1. The actor navigates to the profile section using the navigation bar.
2. The profile is displayed.
3. The user taps on the badges icon or text.
4. The system displays the badges page.

Alternative(s): No Alternatives.

8.12 View Achievements

Use Case Name: View Achievements

Actor(s): Member, Employee, Manager

Main Success Scenario: This use case begins when an actor wishes to view their achievements.

1. The actor navigates to the profile section using the navigation bar.
2. The profile is displayed.
3. The user taps on the achievements icon or text.
4. The system displays the achievements page.

Alternative(s): No Alternatives.

8.13 View Trophies

Use Case Name: View Trophies

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to view their trophies.

1. The actor navigates to the profile section using the navigation bar.
2. The profile is displayed.
3. The user taps on the trophies icon or text.
4. The system displays the trophies page.

Alternative(s): No Alternatives.

8.14 Find Refill Station

Use Case Name: Find Refill Station

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to find the nearest refill station to them.

1. The actor navigates to the reifl page.
2. The application finds the nearest water refill stations and displays them on the map.

Alternative(s):

8.15 Change Password

Use Case Name: Change Password

Actor(s): Member

Main Success Scenario: This use case begins when an actor wishes to change their password.

1. The actor navigates to the forgot password screen for the login page.
2. The application displays a page with a form.
3. The actor fills in the form and clicks the send button.

4. The system validates the input fields and sends a password reset email to the users email address and displays the corresponding message.
5. The actor closes the message and goes back to the main menu.

Alternative(s):

4a. The system validation fails and an error message displays to the actor.

1. The actor closes the error message.
2. The actor can try to fill out the form again.

8.16 Log Out

Use Case Name: Log Out

Actor(s): Member

Main Success Scenario: This Use case begins when an actor wishes to Log out.

1. The actor taps the Logout button in the main menu.
2. The web application ends the session.
3. The web application loads the Login page.

Alternative(s): No alternatives.

9 The Plan

The plan for this project is to fully follow the agile methodology and work in 2 week long sprints or 3 or 4 week long sprints depending on the situation and task at hand but no longer than 4 week long sprints as this is the longest amount of time allowed for a sprint that is recommended by the Agile manifesto. At the end of every sprint also known as iteration a working deliverable will be shown to my supervisor to ask for feedback. This feedback will be documented and pushed into the backlog. The backlog will be made up of all the tasks needed to be completed in this sprint sorted by priority from High to low priority. The Highest task priority tasks will be taken into sprints first and the tasks derived from the feedback will be placed into the backlog accordingly.

The plan for this project will be to build a basic UI first and implement all the back end functionality and then go ahead and design the screens and implement the front end. A milestone will be reached whenever a use case is fully implemented and working fully and the end of an interaction or a number of iterations.

10 Crash Reporting And Analytics

Microsoft App Center was chosen to monitor and provide reports on app crashes and other bugs. Microsoft App Center allows one to build, test, release, and monitor apps for every platform [9]. This will be implemented into the application by using their API and downloading their SDK and making the relevant code changes.

11 Conclusion

This section concludes the design of the Green Quake Application. This design has been thoughtfully documented throughout this document and will be adhered to closely during the development of the application.

12 References

[1] Docs.microsoft.com. 2021. What Is Xamarin.Forms? - Xamarin. [online] Available at: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin-forms> [Accessed 5 January 2021].

[2] YouTube. 2017. How To Connect Firebase Users To Their Data - 3 Methods. [online] Available at: <https://www.youtube.com/watch?v=2ciHixbc4HE&ab_channel=Fireship> [Accessed 24 January 2021].

[3] Kharlamov, D., 2015. Coding Standards and Best Practices - Aversan. [online] Aversan. Available at: <https://www.aversan.com/coding-standards-and-best-practices-2/> [Accessed 26 April 2021].

[4] Liberty, J., 2020. Xamarin Coding Standards. [online] Jesse Liberty. Available at: <https://jesseliberty.com/2020/02/18/xamarin-coding-standards/> [Accessed 26 April 2021].

[5] Liberty, J., 2020. C# Coding Standards. [online] Jesse Liberty. Available at: <https://jesseliberty.com/2020/01/29/c-coding-standards/> [Accessed 26 April 2021].

[6] Muntenescu, F., 2016. Android Architecture Patterns Part 3:Model-View-ViewModel. [online] Medium. Available at: <https://medium.com/upday-devs/android-architecture-patterns-part-3-model-view-viewmodel-e7eeee76b73b> [Accessed 26 April 2021].

[7] Burford, D., 2018. Using the MVVM pattern with a Xamarin Forms mobile app. [online] Medium. Available at: <https://domburf.medium.com/using-the-mvvm-pattern-with-a-xamarin-forms-mobile-app-b1930976bc3d> [Accessed 26 April 2021].

[8] Flaticon. 2021. Flaticon. [online] Available at: <https://www.flaticon.com/search?word=filter&license=selection> [Accessed 27 April 2021].

[9] Appcenter.ms. 2021. Visual Studio App Center | iOS, Android, Xamarin & React Native. [online] Available at: <https://appcenter.ms/> [Accessed 28 April 2021].