Firebase Application - Written Report

# Introduction

The purpose of this assignment was to further understand the uses and functions Firebase has to offer. In this application we focused on the use of Firebase Authentication, Firebase Storage and the Firebase Database known as Cloud Firestore. The application purpose is to allow the users to take a photo of plants/wildlife and for this information to be shared on a network so other users can view what other users had taken.

# Application Run-through

The user opens the application on their android device it will prompt them for permissions due to the location services this application requires. The user will then see a login and register screen, the user has the ability to log into the application and remain logged in using their Google account. If a user wants to register using an email address and password all they have to do is click sign in with email and follow the steps to creating a new account. Once through the authentication the user is then displayed a map, this map will include all the markers from other users. The user can now click on any of these markers to see what’s around them. If the user wishes to upload his/her sightings on to the application, they can do so by clicking the add in the right-hand bottom of the screen. A dialog will appear and after upload has been clicked the marker will appear on the map.

# Permissions

Location Services are very important for this application to run in a sophisticated manner, without them the application will try and use nearest phone masks to pinpoint the user’s device although this is not the best way of finding the users location it does allow the user to interact with the application by not restricting them to having to use GPS location services. The application requires the internet permission to use any functionality on this application.

# Implementation

Before you can implement Firebase into your application you need to add it on to your Gradle Script file with their implementation code, I have 4 Firebase implementation codes in my file each code allowing the application to use more Firebase functionality. The Firebase Core code allows your account to speak to your application. You will need a google-services.json file for this, firebase will ask you to add this into your configuration file. Once Firebase and your application are talking you can add the other functionality by adding their code into the script. So that our images can load in our marker information dialog boxes we need to use a tool called Picasso which allows the application to render image URL’s on the fly. Similar to what we used for our first assignment with API’s the XML files are placeholders and the data is then altered on the fly by Cloud Firestore.

# User Experience

During the application build process, I was constantly thinking of the user, by adding the Google Account Log In function it makes it easier for users who have their Google account on their device. I made the application to the material design guidelines and everything you see is based off cards, this was the correct design for the user group this application was for, the users will be able to work out what the functions on the application do without very much knowledge of the application first-hand. The map is always up to date with Google Maps API and will work in any country providing you have a data connection and location services enabled.

# Conclusion

When the application is working with a good internet connection and accurate location services the user and other users will have a great user experience with minimal problems. However, when location services are disabled the pinpointing area of the application starts to degrade due to the phone’s location being mapped off masks. The application also fails to work in areas where a phone data connection is unavailable. If I was to recreate the application, I would force location services to improve all users experience on the app and I would allow the application to store a snapshot of the user’s image and location to be uploaded when the user next connects to the internet. Therefore, you will not receive as many incorrect mappings.