

BEACON Preliminary Design Document

INTRODUCTION

We are planning to build Beacon as an Android app that allows users to find events near their location in real-time, share events they are hosting with other Beacon users, comment and post photos on the page of an event they went to/are currently attending, and broadcast their events to a wider radius if they are a business account.

This document will cover the high-level architecture and design of our system.

SYSTEM ARCHITECTURE AND RATIONALE

BEACON consists of the following major components:

1. Android application
2. Google Maps API
3. Firebase server

Android Studio (Java)

We are going to be working with Android Studio to write and test all of our code for the app. We have chosen Android Studio because it allows us to easily simulate our application on multiple Android devices. Most of our team has experience using Android Studio, so we can begin our implementation faster.

Google Firebase

We are employing the use of Google's Firebase platform in order to handle multiple aspects and functionalities of our application. These include:

- User Authentication
 - Google
 - Facebook (Later)
- Remote, Real-Time Database, To Store,
 - User Information
 - Event Information
- Storage of Media
 - Photos related to an Event
 - Videos related to an Event
- Targeted Advertisements
 - Provide users with ads which are relevant to their search history

We chose to use Firebase instead of available alternatives because of the platform's simplicity. All of the features listed above are, for the most part, handled by Firebase. This allows us to focus more closely on the user's experience and making the app more useful, as opposed to spending time on the technical server-side aspects of the project.

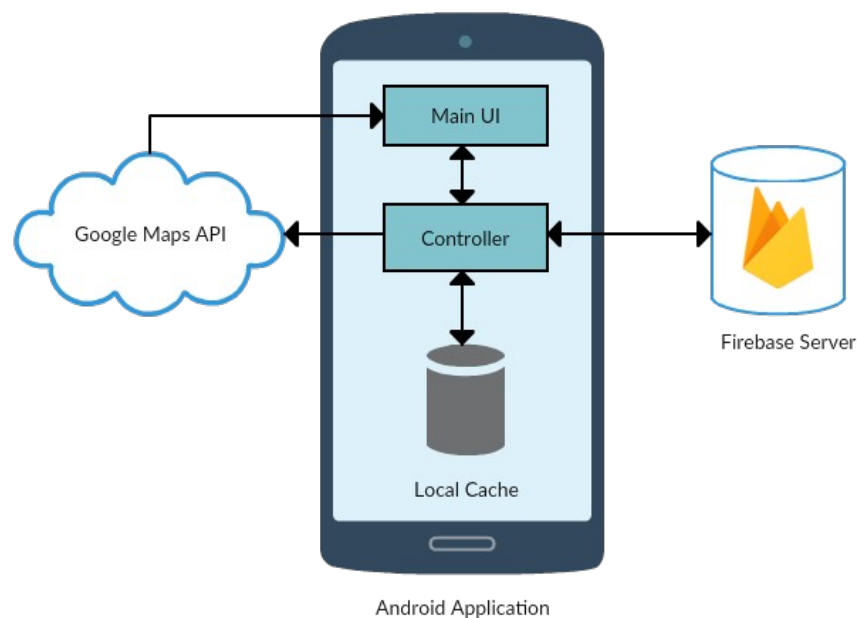
Furthermore, Firebase offers offline services. This means that if a user is using BEACON while they have no internet connection, the full functionality of the app will not be compromised. Any changes made by the user during their offline usage can be seamlessly merged with the

existing, online data. This capability will increase the robustness of BEACON and improve the overall user experience.

Google Maps

We will be using the Google Maps API for the app's location-based features. Google Maps is by far the most used and robust map API. Google Maps will allow users to see locations that have been pinned by people hosting events in an intuitive UI. Since many users have already used Google Maps, it will make our application very intuitive for the average user. It will also provide directions to these locations for users who have Google Maps installed on their device.

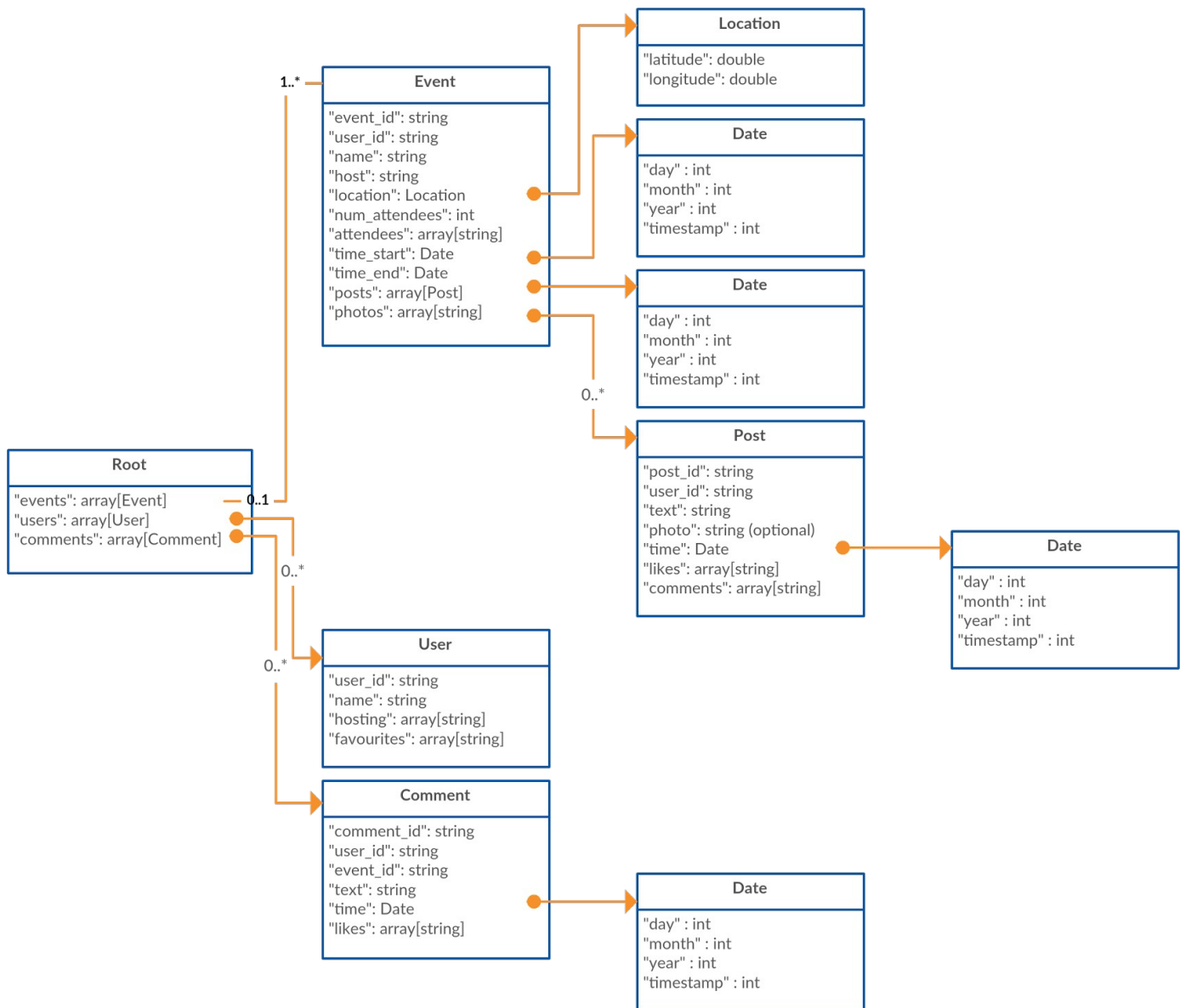
Model-View-Controller Architecture of the App



- The Main UI is the different pages/ views that the user can see. It sends information to the Controller telling it which button has been pressed and the Controller sends information back telling the UI which page to display.
- The Google Maps API interacts with both the Main UI and the Controller. The Controller sends locations for BEACONS to be placed on the map. The map tells the Main UI to display the map with the updated pins.
- The Local Cache stores user information, favourites, and a select number of closest nearby events. The Controller is then able to access this data.
- The Firebase server will contain all of the event and user information in its database. That information within the Firebase database will be called upon and written to by the controller. Furthermore, the controller will contact the Firebase server in order to

DATA

Our application's data model is as follows:



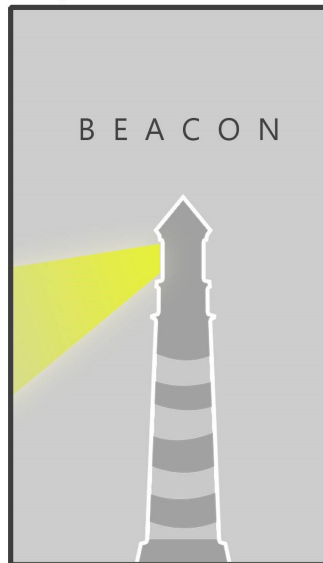
GUI

The UI is based on Google's material design. It will consist of an interactive map, and a list-view where nearby BEACONS will be displayed. Every BEACON will be a link to an event page, where information, comments, and pictures of the events can be viewed.

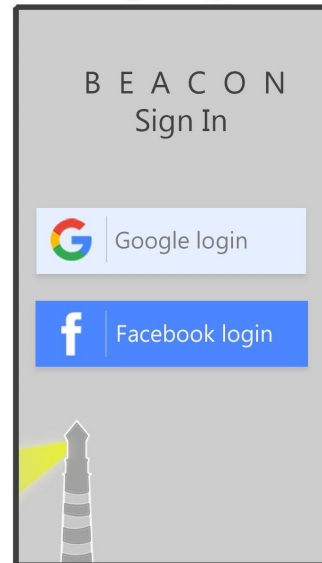
Login View

After clicking the BEACON icon, a splash-screen will display while the app is loading. Upon opening the application the Users will then be presented with a Google sign in page.

Splash Screen View

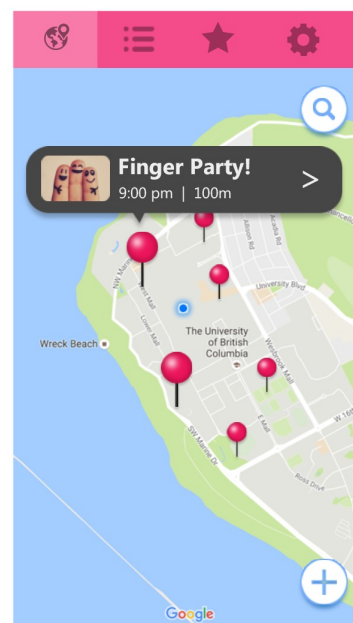
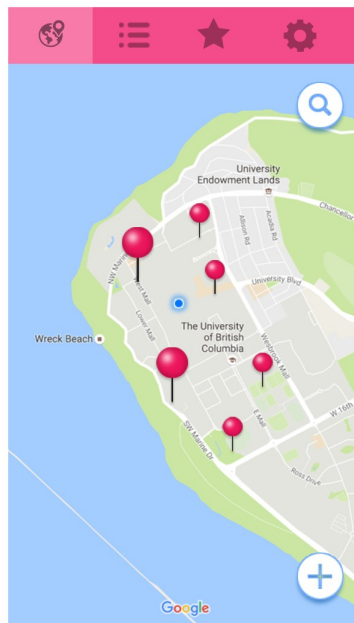


Login Page



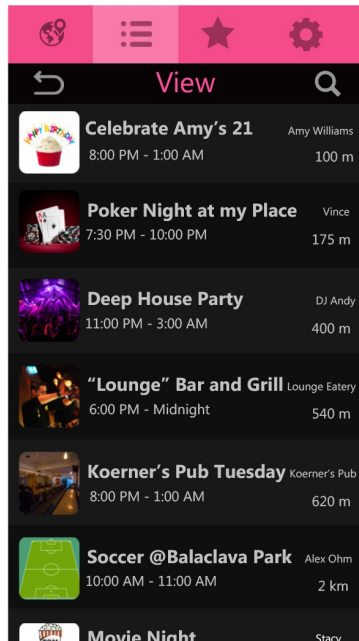
Map View

After users go through the login view, the main page is displayed. This page contains the map view - an interactive map that shows the BEACON pins that user's location.



List View

Users can also choose to see BEACON pins in the list view, which sorts events by name, location (i.e. distance), or time; and shows said basic information.



Event Page

The event page will display all of the information that relates to an event: location, title, short description, comments

feed (includes text and photos)





VALIDATION

We have built a GUI prototype that allows us to see the basic layout and interaction scenarios. We are planning to have design reviews with clients as we further develop our application and GUI.