Comp-3004 Deliverable 2

Application Name: Beacon
Members: Julian Clayton, Jason Bromfield, Nolan Hodge, Cameron MacQuarrie
Demo:
For the demo, two devices with **Beacon** installed on them will be needed. The two devices will be denoted as *Client A* and *Client B*. A demo of most of the application's functionality can be carried out as follows:

Part 1: Adding a Friend
Client A will use the user-search functionality to find *Client B. Client A* will then proceed to send a friend-request to *Client B. Client B* will accept the friend request. Both users will have the other user in their friends-list.

Part 2: Sending a Private Beacon
Client A will navigate to their friends-list activity and click on *Client B's* username. *Client A* will then proceed to send *Client B* a Beacon. When the notification pops up on *Client B's* device, *Client B* click on the notification and accept *Client A's* Beacon. *Client B* will be taken to the *Compass* activity and the arrow will point to *Client A's* location. *Client B* will then return to the *Map* activity and *Client A's* location will be displayed on the *Map* activity. *Client B* will click on this Beacon and a dialog will appear prompting *Client B* to track this location. Clicking **TRACK** will take *Client B* back to the *Compass* activity.

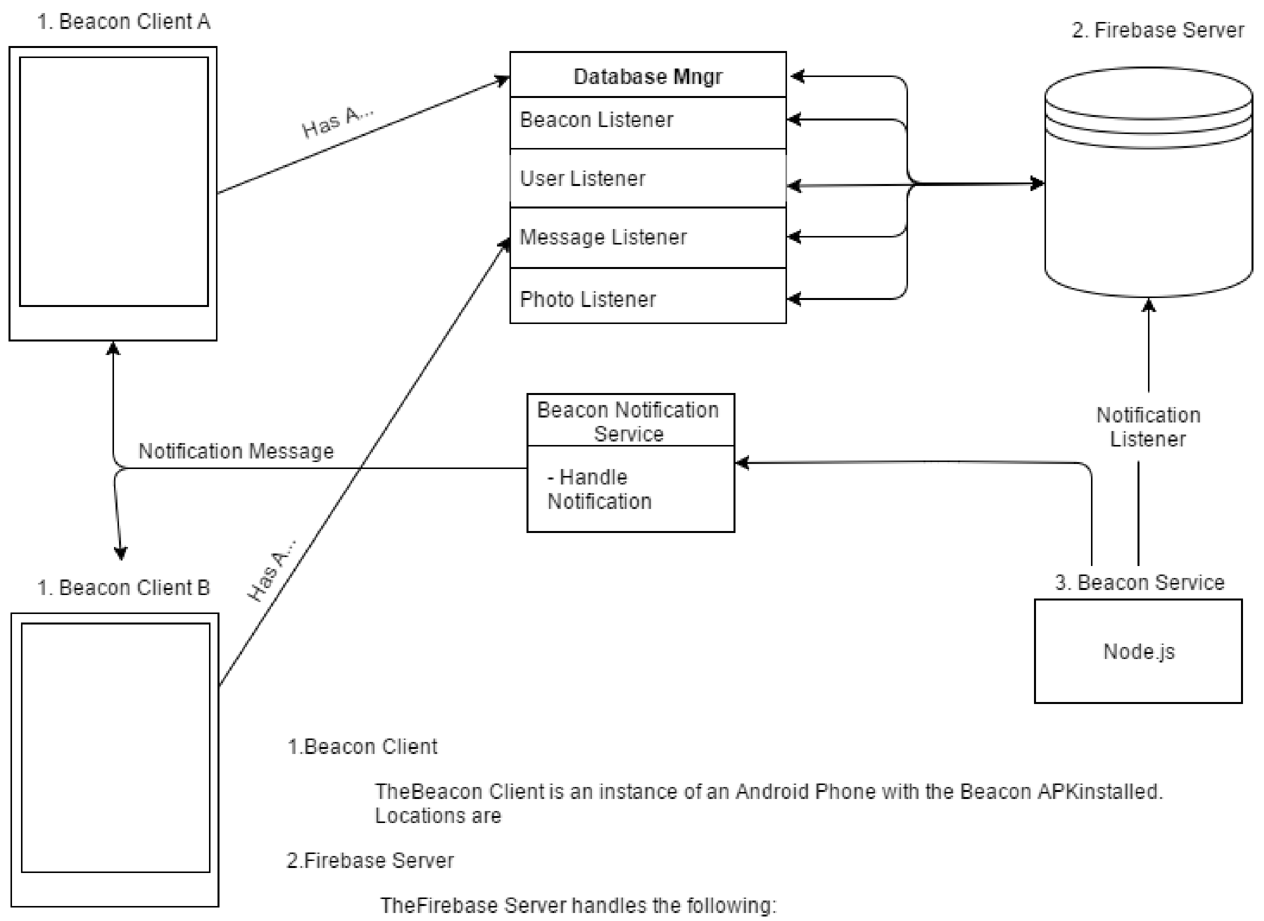
Part 3: Creating a Public Beacon and Attaching a Photo to a Beacon.
Client A will navigate to the *Camera* activity and take a picture of their location. Next *Client A* will click and hold on a location on the map. A dialog will appear asking *Client A* if they would like to create a Public Beacon. *Client A* will accept, this Public Beacon will be broadcast to all Beacon users. *Client B* will navigate to the *Public Beacons* activity and click on *Client A's* username. *Client B* will be asked if they would like to follow *Client A. Client B* will accept then navigate to their *Beacons* activity. Upon clicking on *Client A's* username they will be taken to the *Compass* activity. *Client B* will click on the Photo Button and will be able to see *Client A's* photo that was just taken.

Part 4: Nearby Locations. Request a Beacon
Client A will briefly show that Public Beacons can also be created at Nearby Locations. *Client A* will then navigate back to the *Friends* activity and send *Client B* a Beacon Request. *Client B* will accept the request and *Client A* will be prompted to follow *Client A.*

Part 5: Instant Messaging
Client A and *B* will demonstrate the straight-forward messaging capability by sending messages back and forth.

Status Report
All main functionality that we outlined in our proposal is done for project the next few weeks of development will be solely dedicated to debugging, testing, UI upgrades and increasing the overall robustness of the application.
Current Difficulties:

- Crashes, the application is still a little unstable and crashes when inconsistent data is written to the database.
- Making the application more user-friendly; several changes need to be made to do this. There needs to be ways for a user to delete their own beacons, view there own beacons in an affective way as well as making the GUI easier to navigate and prettier.
- Small bugs that interfere with the overall experience. Such bugs include Beacon Photos loading twice and sometimes not at all, the Message activity reloads when a message is received etc.



1.Beacon Client

TheBeacon Client is an instance of an Android Phone with the Beacon APKinstalled. Locations are

2.Firebase Server

TheFirebase Server handles the following:

- Database
- Authentication
- Photo-Sharing

3.The Beacon Service

The Beacon Service is a small Node.js service that handles all themessaging between phones. When a new notification is present in thedatabase the Beacon Service will send the notification to the correctphone. The notifications are received by theBeaconNotificationService on the Client and the appropriate messagehandling is instigated.