**The University of the West Indies**

**COMP3990 Project**

**2016-2017 Semester 2**

**Project Implementation Document**

**Project Name: BeckonUs**

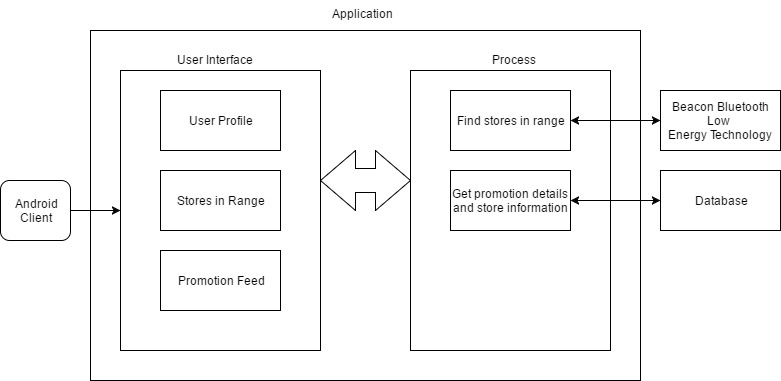
**Project Members: Dineile Guevara**

**Vishal Ramkissoon**

**Christine Russell**

**Architechtural Design**

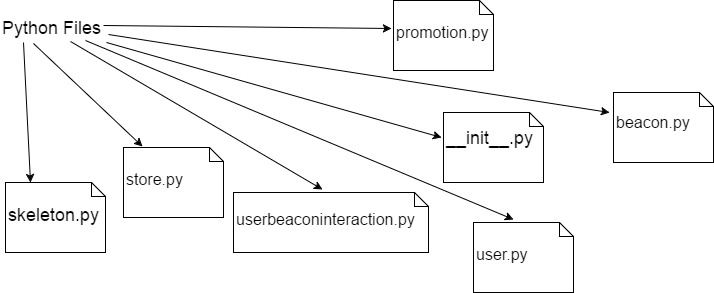
***Component Digrams***



The above diagram shows the architectural design which outlines the key functions of the system.

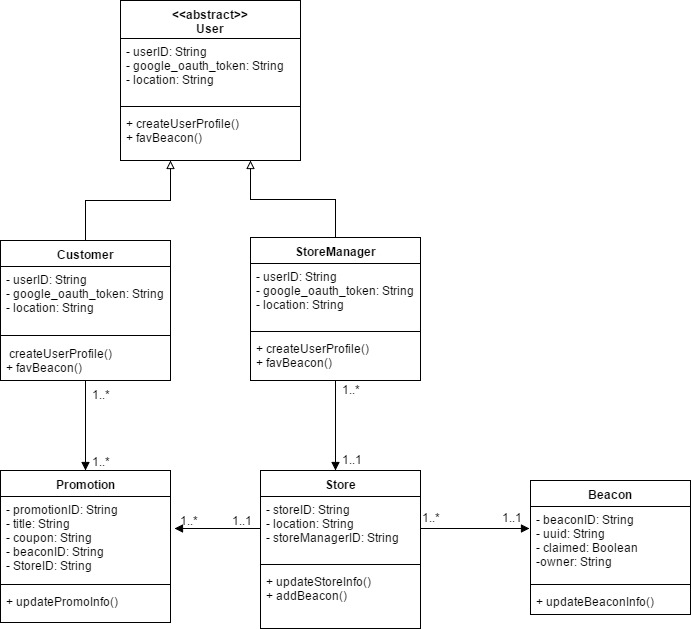
***Description of Components***

The User Interface component displays the major functions of the application. User profile is where the clients’ information is stored, along with the stores that they have favorited. The “Stores in Range” feature allows the client to view all the stores within the range of the Android device and the Promotion Feed gives a listing of all promotions available in nearby stores and favorited stores, including coupons, if any. The application finds all the stores in range through the use of the Bluetooth Low Energy Technology and the promotion details and store information are available from the database.



The above diagram show the Python Files used in the project.

**Class Diagram**



The above diagram outlines the attributes, methods and interactions of the major classes/modules in the system.

**Entity Relationship Model**

****

The above diagram specifies the entities, datatypes and relationships that are important for the project domain.

**Sequence Diagram**

****

The above diagram illustrates object interactions for the major use case scenarios for the project.

**Description of Technology Used**

***Beacon***

A beacon is a tiny, low power computer designed to attract attention to a specific location. Estimote beacons are used in this project. They detect human presence and behaviour using proximity technologies and trigger pre-programmed actions. Each beacon has a Bluetooth Low Energy (BLE) transmitter which broadcasts tiny radio signals over the air containing unique, location-specific data. Modern smartphones constantly scan for these signals. If they enter their range an associated app responds with the desired action. Personalized content is displayed as a notification or directly in the app.

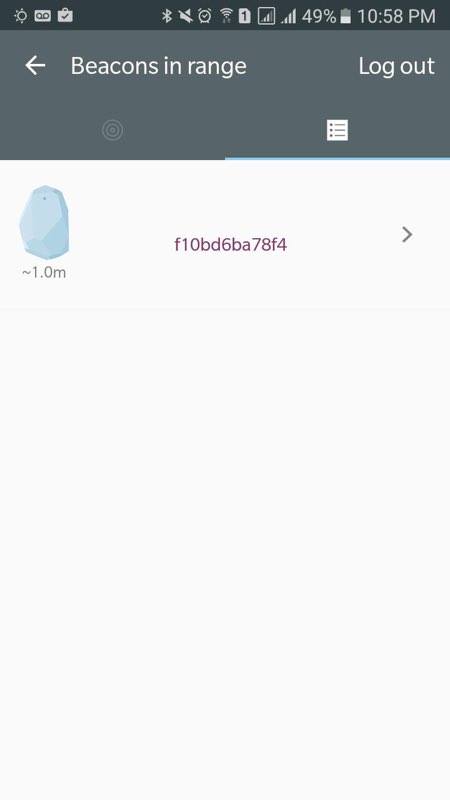
The typical range of Bluetooth low-energy radio modules is anywhere from 1 meter (3 ft) up to 200 meters (650 ft). It depends on the location and circumstances, because radio signals could be absorbed or diffracted. However, the beacons just send the Bluetooth signals, and it is the responsibility of a custom built app to trigger the action on a user's screen, based on proximity to a particular beacon.



*An Estimote Beacon*

***Estimote***

The Estimote App needs to be downloaded from the GooglePlay Store to manage the Estimote beacons on Android devices.



*Screenshot from the Estimote App showing all the beacons in range of the device*

***Android***

Android is an operating system developed by [Google](https://en.wikipedia.org/wiki/Google), based on the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and designed primarily for [touchscreen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer). Android's [user interface](https://en.wikipedia.org/wiki/User_interface) is mainly based on [direct manipulation](https://en.wikipedia.org/wiki/Direct_manipulation_interface), using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a [virtual keyboard](https://en.wikipedia.org/wiki/Virtual_keyboard) for text input.

# ***Android Studio***

Android Studio is the official Integrated Development Environment for Android which provides the fastest tools for building apps on every type of Android device.

***Database Design- RethinkDB***

While considering what features could be implemented to provide a better experience for customers and more incentive for store managers, RethinkDB presented itself as an option for scalable pushing. Whatever updates occur in the database is automatically pushed out to the clients. This would be very useful for the way the feed system for the customer works, as well as allowing Real-time statistics for the store manager. However, this database isn’t well suited for analytical data and processing so additional trend data will require the database to be converted to another database.

The nature of the RethinkDB allows for tables to quickly be made programmatically which allows us to quickly section off our store tables within the same store database. This allows us to keep individual store data clean and concise upon lookup.

***Python***

Python is a widely used [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language) for [general-purpose programming](https://en.wikipedia.org/wiki/General-purpose_programming_language).  Python has a design philosophy which emphasizes code [readability](https://en.wikipedia.org/wiki/Readability) using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block) rather than curly braces or keywords, and a syntax which allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) than possible in languages such as C or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). The language provides constructs intended to enable writing clear programs on both a small and large scale.

***RESTful API***

A RESTful API is an application program interface ([API](http://searchexchange.techtarget.com/definition/application-program-interface)) that uses HTTP requests to GET, PUT, POST and DELETE data. It is based on representational state transfer ([REST](http://searchsoa.techtarget.com/definition/REST)) technology. A RESTful API breaks down a [transaction](http://searchcio.techtarget.com/definition/transaction) to create a series of small modules. Each [module](http://searchenterpriselinux.techtarget.com/definition/module) addresses a particular underlying part of the transaction. This modularity provides developers with a lot of flexibility, but it can be challenging for developers to design from scratch.

***Flask***

Flask is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and based on the Werkzeug toolkit and [Jinja2](https://en.wikipedia.org/wiki/Jinja_(template_engine)) template engine. Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

***Flask Restful***

**Flask-RESTful** is an extension for Flask that adds support for quickly building REST APIs. It is a lightweight abstraction that works with your existing ORM/libraries. Flask-RESTful encourages best practices with minimal setup. If you are familiar with Flask, Flask-RESTful should be easy to pick up.