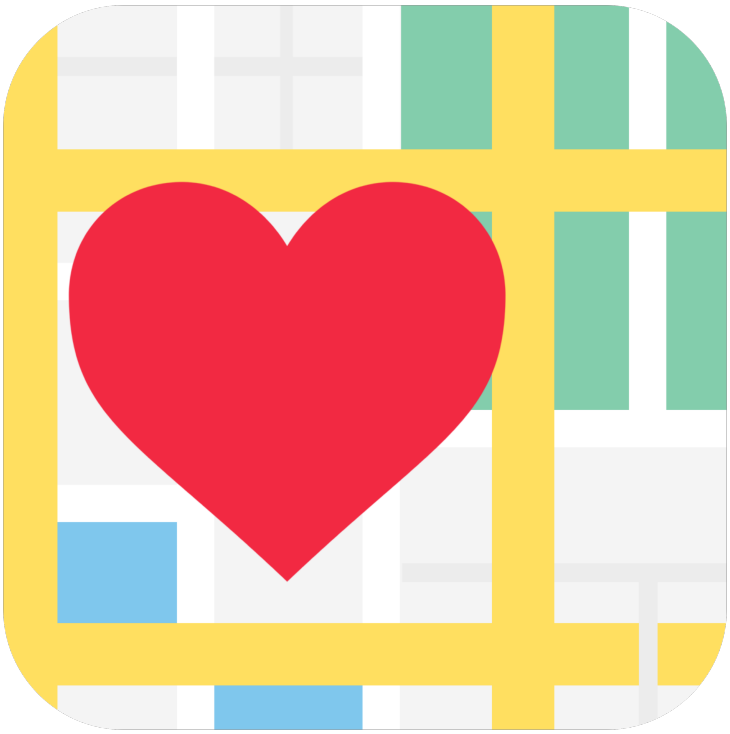

Figure 1: Safety Beacon Logo

DESIGN DOCUMENT

SAFETY BEACON - CMPT 275

Group 7

Nathan Tannar
Jason Tsang
Philip Leblanc
Josh Shercliffe
Youjung Kim

Website

github.com/nathantannar4/Safety-Beacon

Table of Contents

1. Revision History

Version	Status	Publish/Revision Date	Authors
1.0	Created	October 19, 2017	Nathan Tannar Jason Tsang Philip Leblanc Josh Shercliffe Youjung Kim

2. Guidelines

2.1 Technical

The development of Safety Beacon will be accomplished using Apple's Xcode 9 IDE. It features iOS simulators to use for basic testing and advanced diagnostics tools such as instruments for performance analysis. Like many iOS apps, Safety Beacon will strive to follow the iOS Human Interface Guidelines set by Apple. In addition, this project will utilize Git version controlling and store the source code and assets on GitHub.

The development of the MongoDB database and backend system will be done using any standard text editor, as the server will be running a Node.js runtime that links the API to the database. This is an open-source solution known as Parse Server.

2.2 Ethical

Caretakers and patients should be informed that the location data sent by the patient's iPhone is encrypted and sent securely. The location information, while being stored on Safety Beacon's database, is owned by the user that submitted the location; therefore, should they ever wish that their data be purged, a request via email can be submitted and we will comply. Their location information will not be sold or shared with any 3rd party.

2.3 Legal

By using the app, patient's consent to letting Safety Beacon store, transmit and analyze their location information. While our goal is to assist patients with Alzheimer's, the safest option to take when in an emergency is to call local emergency services for immediate assistance.

3. System Diagrams

3.1 Feature Diagram

In this diagram, the main features of our app, as described in the Requirements document are presented in relation to the type of user. The patient's main feature is navigation, while the caretaker has features for tracking and location monitoring to help ensure their patient does not get lost.

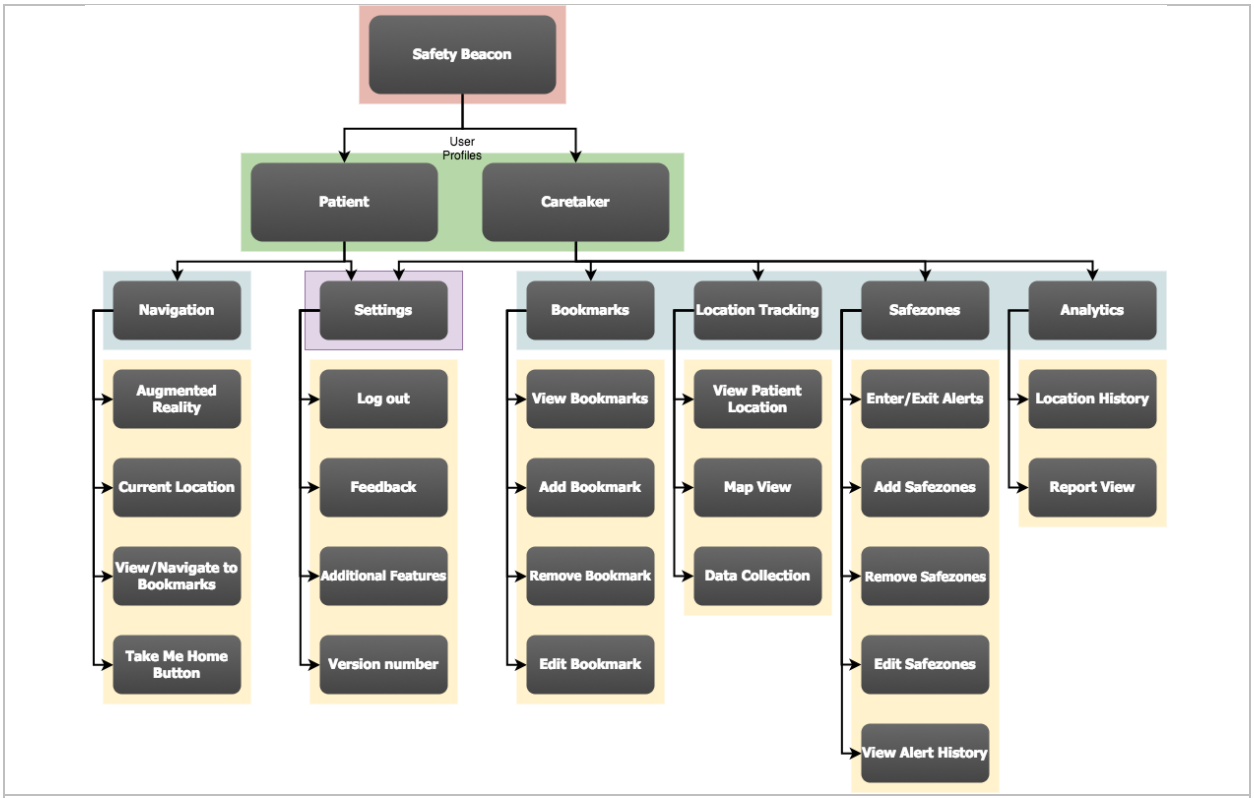

Figure 3.1: Feature diagram showing the list of features split between the two types of users

3.2 View Controller Diagram

This diagram shows the major View Controllers of our system and how they interact with each other.

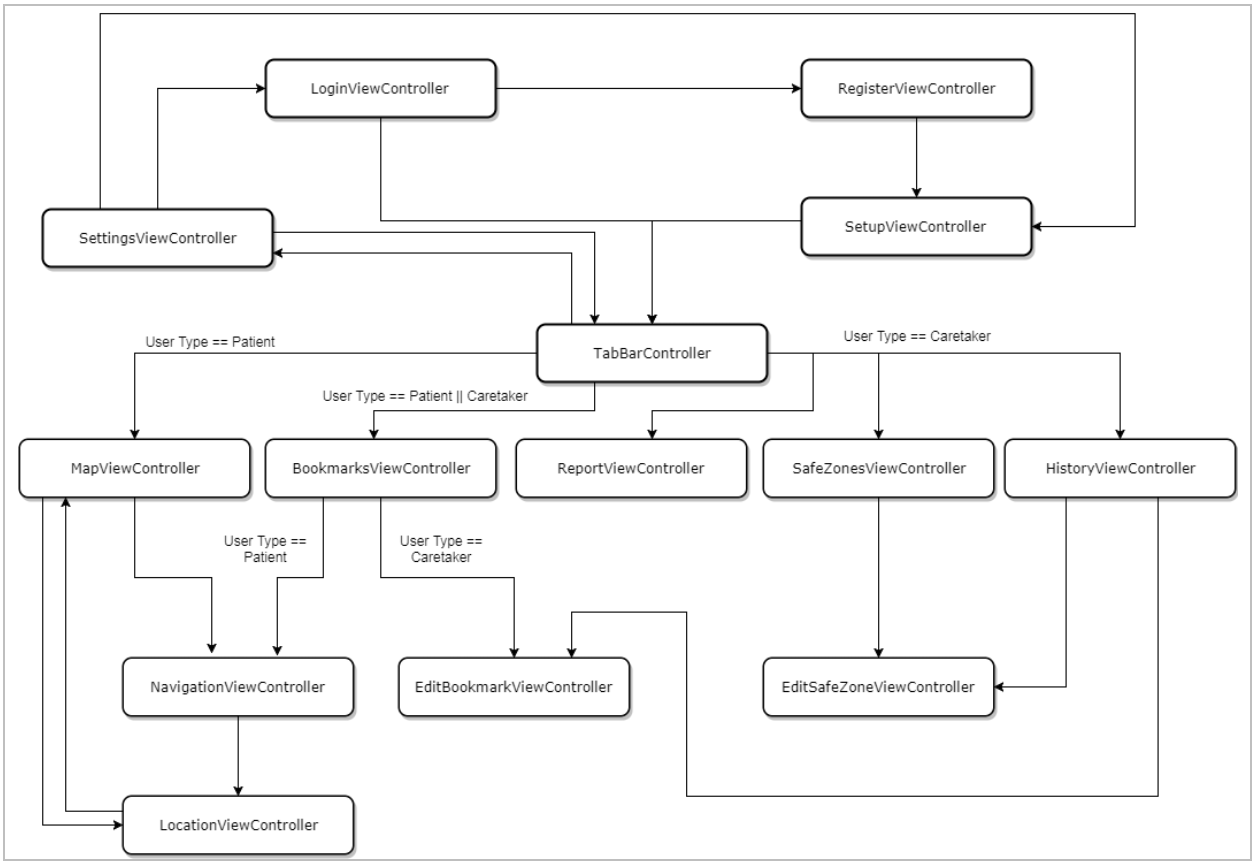

Figure 3.2: View Controller Diagram to show the flow of different controllers

3.3 Class Diagram

Our class diagram shows our major classes. For example, our Location Manager will be a significant component for both types of users as it tracks the patient's location and detects the entering and exiting of Safe Zones.

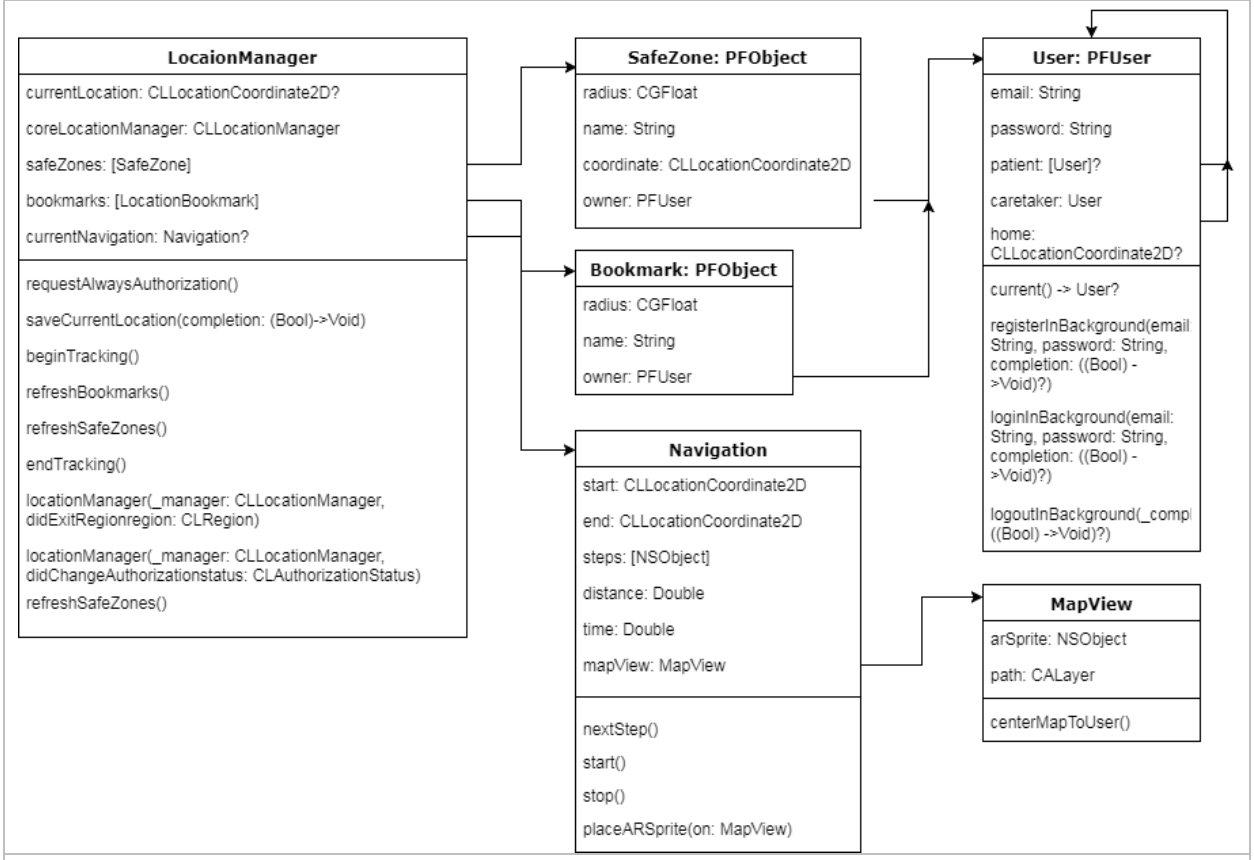

Figure 3.3: Class diagram showing the major classes for our system and how they relate

4. Data Requirements

The primary data inputs will be from the iPhone's touch screen, and the iPhone's GPS chip for location tracking and navigation. Using the touch screen, the caretaker can browse the app to input new bookmarked locations for the patient, view the patient's location history and set Safe Zones. All of these actions require either a query of the database to receive data or the creation of new data records into the database. The communication between the app and the server will be done via REST. Data is encrypted in JSON (JavaScript Object Notation) format when sent between the device and server. The app then parses the JSON data into useful objects such as GPS coordinates, strings, numbers and user records.

Secondary data inputs include the iPhone's camera and gyroscope, when the patient enters the augmented reality mode. This is done using Apple's ARKit framework, which performs all of the calculations required: to superimpose a computer-generated image on a user's view of the real world through the device. The images are superimposed based on the GPS navigation coordinates sent by the server in JSON format.

Data will be outputted onto the iPhone's screen, as well as through the vibrator motor and speaker for notifications.

5. Feature Priority

5.1 Version Release Schedule

Version	Date
1	November 6, 2017
2	November 20, 2017
3	December 4, 2017

Version	Feature	Description
1	Login/Account Setup	▪ Login with email or Facebook ▪ Setup account as either a Caretaker or Patient
1	Patient Menu	▪ A basic map view consisting of the user's current location and required navigation buttons
1	Settings Menu	▪ View basic account/app information ▪ Logout
1	Location Tracking	▪ Sending location information from the device to the server
1	Bookmark Locations	▪ Create, edit, delete location bookmarks for the patient ▪ Displayed on the Map
1	“Take Me Home” button	▪ Immediately guides the user home ▪ Home displayed on the Map
2	Safe Zones	▪ Create, edit and delete Safe Zones ▪ Push notifications to caretaker when patient exits a zone
2	Location History	▪ A map view of the patient’s previous locations with the ability to filter the results to specific days/times
2	Analytics	▪ A simplified report of where the user was during a given day
2	Basic Navigation	▪ A standard navigation path to direct a user to a location on a map
3	Augmented Reality Navigation	▪ Navigation with Augmented Reality to make it easy to guide a user to their destination
3	Turn-by-Turn Navigation	▪ An expansion on Basic Navigation to provide a detailed list of steps in a turn-by-turn fashion ▪ To be used in conjunction with AR Navigation
3	UI Refinements	▪ Focus on perfecting the UI/UX; that is, animations, color and text
3	Video Tutorials	▪ Videos to explain the applications use to a potential patient and/or caretaker