**InPace Mobile Application Elaboration 1 Document**

April 7, 2015

**By:**

Yogesh Pandare Pooja Mahesh

Meghan Jordan

**Pace University**

# Dr. Francis Marchese

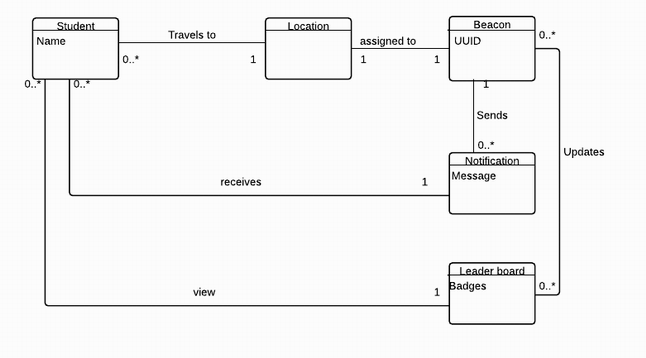
Table of contents:

1. **Domain Model**
2. **Design Model**
3. **Software Architecture**
4. **Data Model**
5. **Use Case Storyboards**
6. **Implementation Model**
7. **UI Prototypes**

## 1 Domain Model

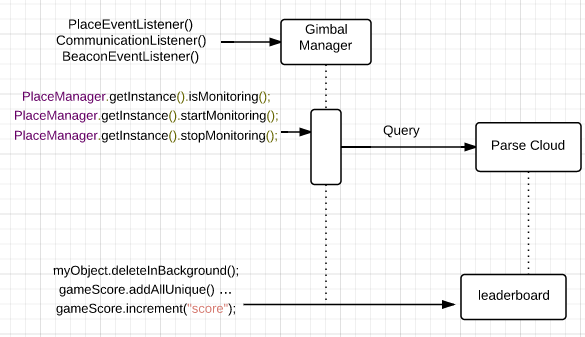
InPace Domain Model

https://www.lucidchart.com/invitations/accept/022e1d77-5081-4e3f-b45e-8e3e269a3a04 - use think link to view and edit diagram



## 2 Design Model

https://www.lucidchart.com/invitations/accept/73f40462-7fdd-4e9e-828a-d26c6e2b4bca use this link to view or edit design model



**3 Software Architecture**

# Architectural Representation

This document will describe the architecture of InPace in a collection of views including use case view, logical view, process view, and deployment view.

# Architectural Goals and Constraints

There are several requirements and constraints on the InPace system.

1. Gimbal Beacons require the use of the Gimbal Manager where Beacons must be registered, activated, maintained, and paired with the mobile application via the package and the Gimbal API key. The Gimbal Manager also allows for notification setup so that Beacons can communicate with the application. Thus the application relies heavily on the Gimbal server.
2. Gimbal Beacons may sometimes require firmware updates and configurations that would result in the need for the Gimbal Manager iOS application.
3. InPace will be implemented as an Android application, published, and easily accessible from the Google Play store.
4. Parse Cloud that handles the leaderboard activity

# Use Case View

The Use Case View describes the scenarios and functionality of the system as they relate to the architectural design considerations.

* Download the application from Google Play.
* Register name and enable Bluetooth.
* Seek Beacons at designated locations.
* Receive/send notifications based on Beacon findings.
* View Leaderboard and then collect a badge.

# Logical View

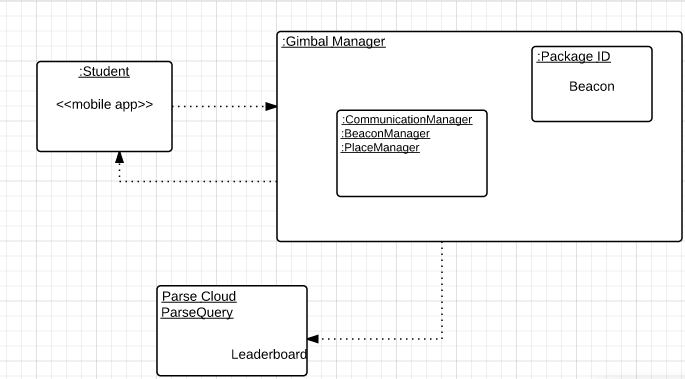
The Logical View of InPace system considers the Google Play service where the application is available for download to users. This is necessary for distribution to Android phones. The Gimbal server also plays a huge role in Beacon management. Each Beacon is activated and registered such that the package ID of InPace: package com.cs639.pace123.inpaceimport; is assigned to the Beacons for the scavenger hunt. The Gimbal Manager provides an app-id and api-key, each of which are associated with the package ID and placed into the Java code. The Java code classes from the Gimbal SDK are engineered to receive communication from the Beacon hardware upon arrival at the destination. A database and list view is also implemented such that users can track their progress with a badge and compare their performance with others via leaderboard.

# Process View

The SDK is responsible for communication with the Beacon. The Parse cloud will be updated frequently and will reflect the leaderboard and UI.

# Deployment View

https://www.lucidchart.com/invitations/accept/75c5ae08-a435-4037-88df-0a722f5043c0 - use this link to view and edit diagram



**4 Data Model**

The following data objects will be presented and managed by the system:

## 1. The Gimbal Manager

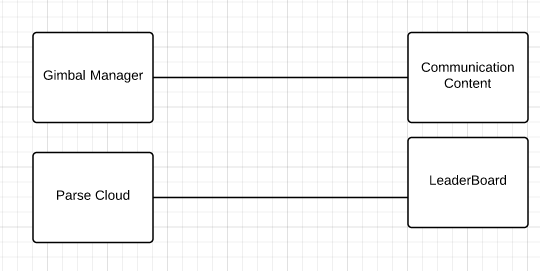
The Gimbal Manager is where the Beacons required for the application to function are activated, registered, monitored, and configured. This is also where the communication is able to be setup and made active for each Beacon sighting

## 2. The Leaderboard

Upon Beacon sightings, score is incremented by 1 in Parse Cloud to acknowledge receipt of Beacon sighting. The leaderboard is updated to new score.

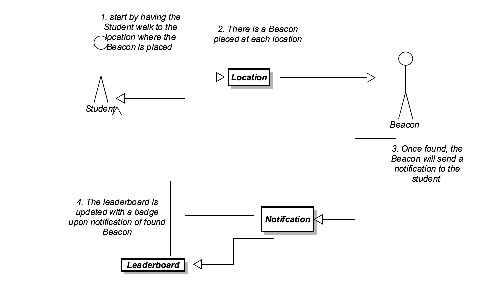
## 3. The Content

The content of the Beacon scavenger hunts must be managed. https://www.lucidchart.com/invitations/accept/80832460-eb8e-4b05-866b-0c6d0ea669ac -use this link to view and edit data model



## 5 Use Case Storyboard

1. The Use Case starts with a student travelling to the location where the Beacon has been placed
2. Once the Beacon has been located it will send a notification to the student that it has been found
3. The system will mark that this location has been found and direct the student to the next location
   1. The student receives a badge in the final location after traversing all the locations.
   2. The leaderboard is updated
4. Steps A-C repeat for 5 locations



## 6 Implementation Model

https://github.com/poojamahesh/AndroidInPace

This code snippet shows the implementation of the beacon listener and communication manager as discussed in the diagrams.

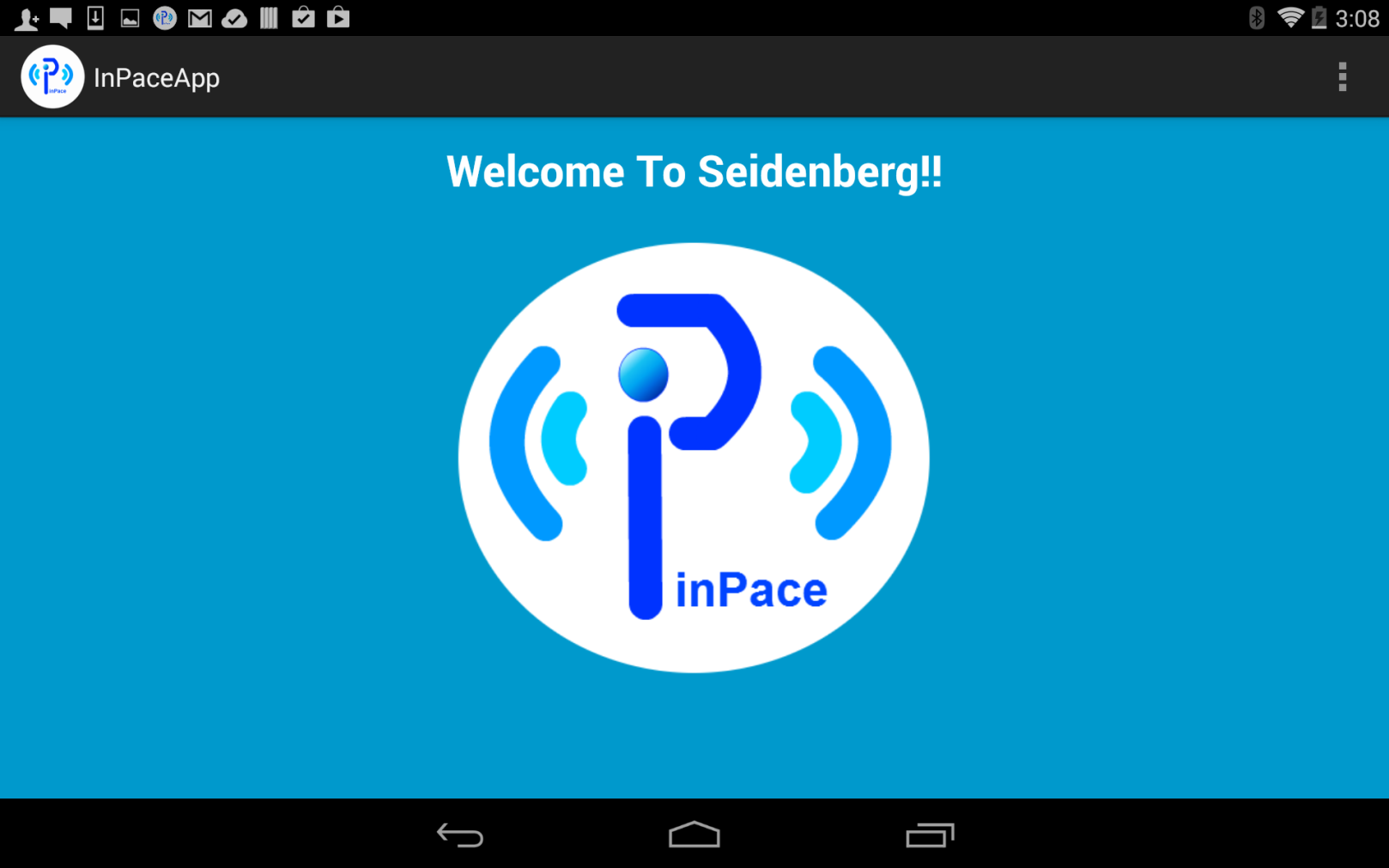




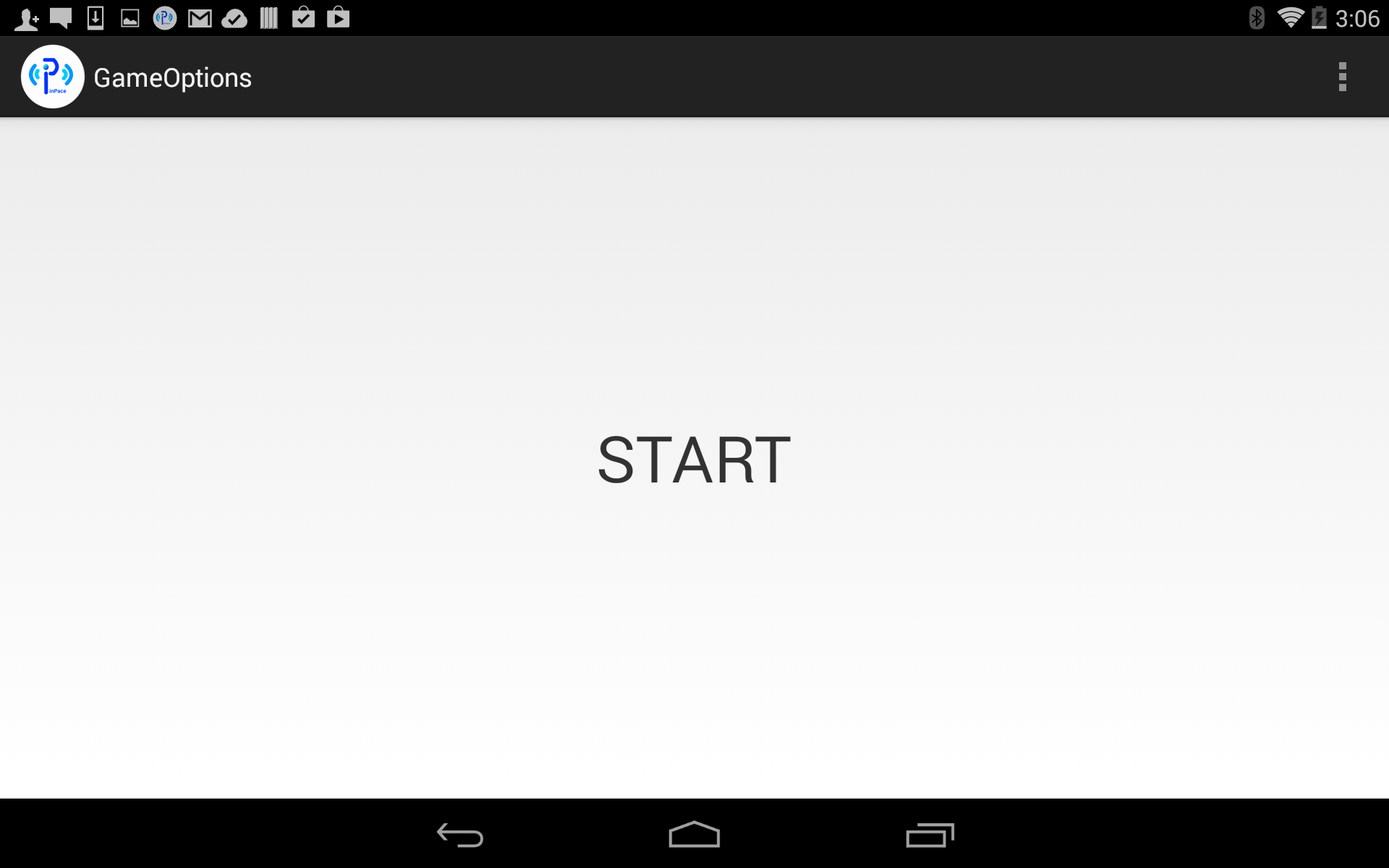


## 7 UI Prototype

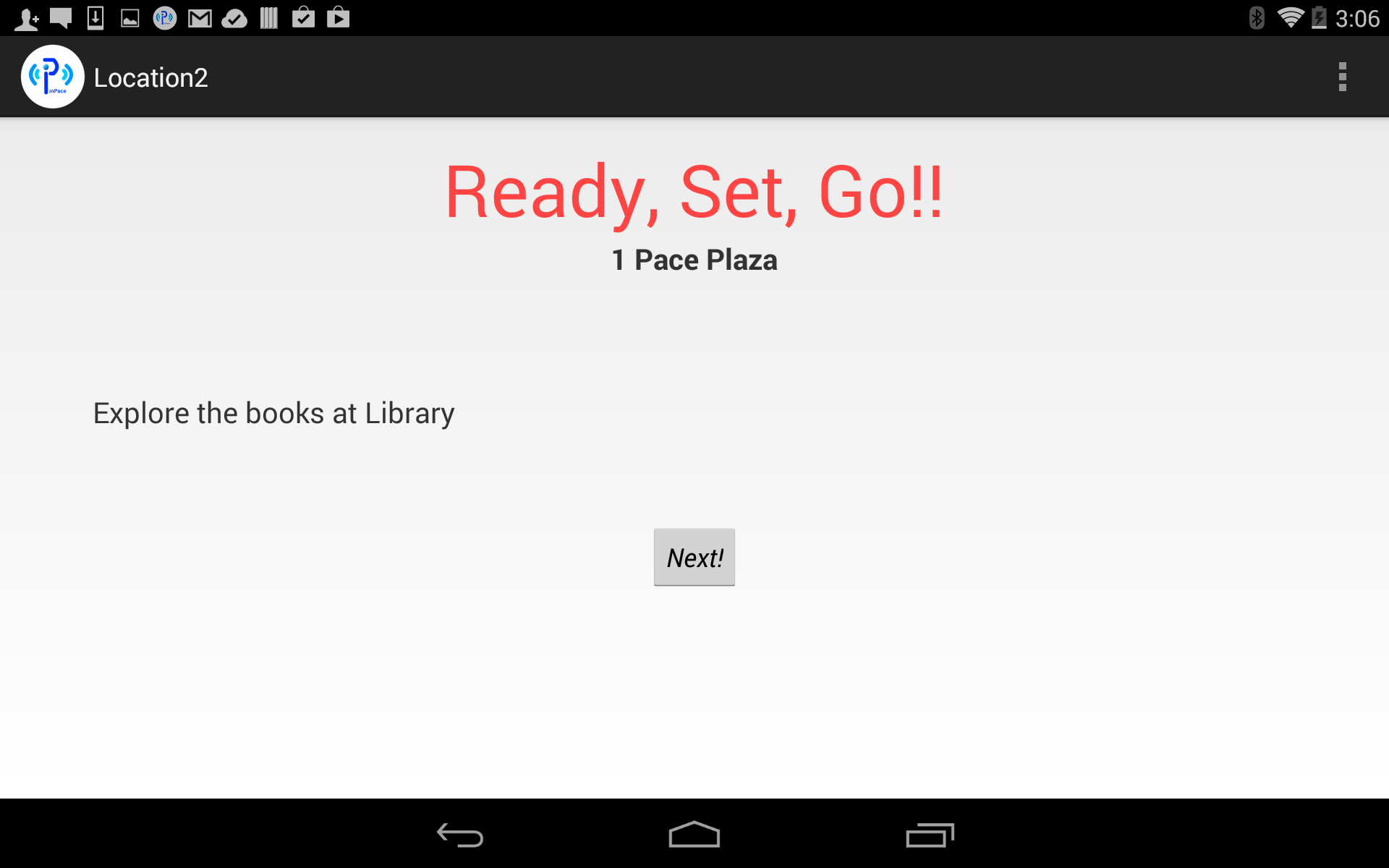
Splash Screen



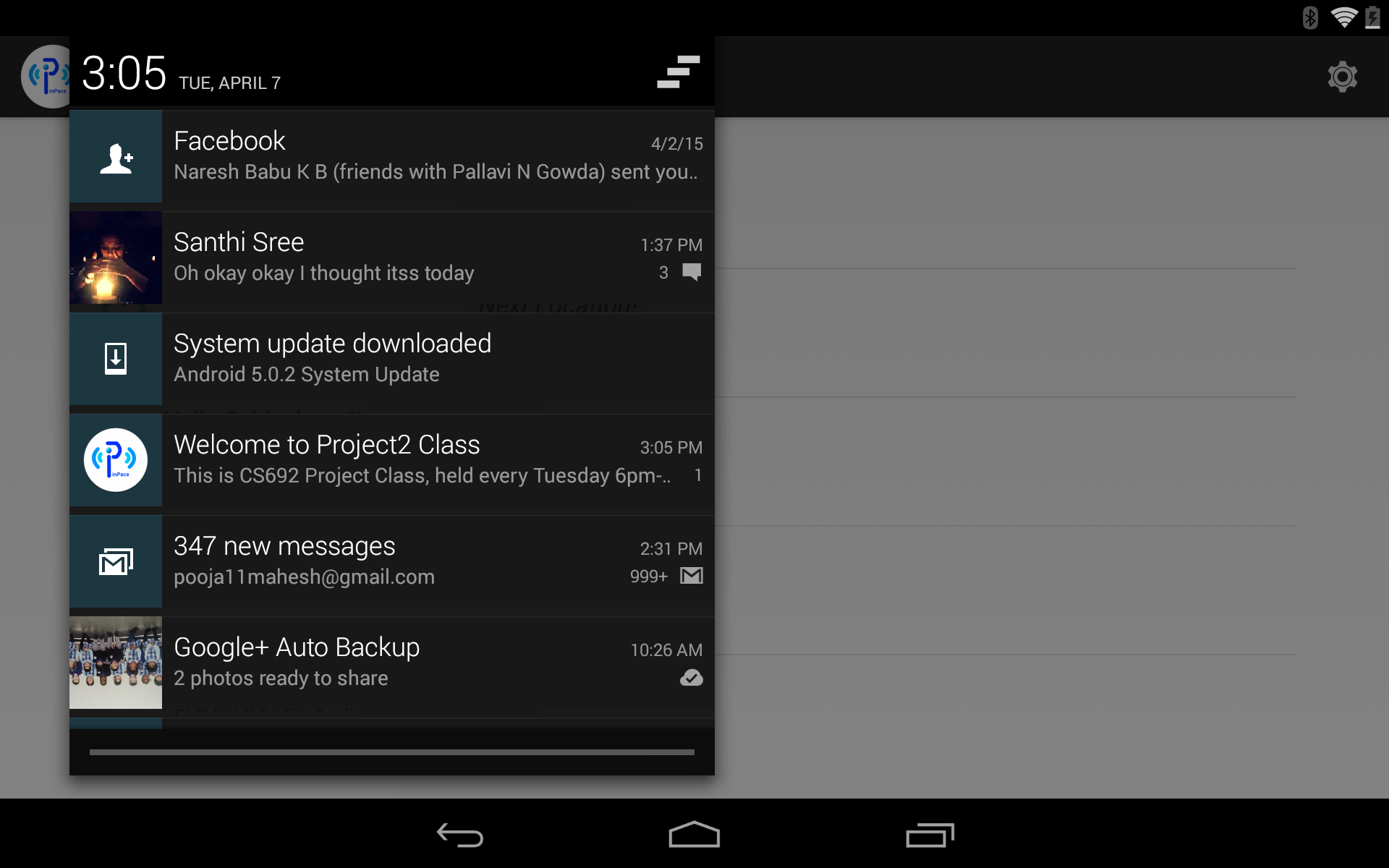
Start the Scavenger Hunt



Directed to Location



Notification description



Beacon Sighted & Notification

