

# Backtrack(Team 1)

---

## Software Implementation Document



**Monil Gandhi (010020231)**

**Chris Ngai (010444733)**

**Dhruv Soni (010012574)**

**Matthew Swerdon (009642269)**

**Kumar Vaibhav (010110295)**

## 1. Preface

### **Purpose:**

The purpose of this document is to elaborate on the teams implementation of the app, Backtrack. It lists architectural and detailed design changes that were made and the reasons as to why they were made.

### **Document Overview:**

- a. **Introduction:** Details the implementation process of the application and back-end
- b. **Architectural Design Changes:** List of architectural design changes and why they were made.
- c. **Detailed Design Changes:** List of detailed design changes and why they were made.
- d. **Requirements Changes:** How these changes affect the requirements of the project

## **2. Introduction**

This document describes the process of the implementation of the Backtrack project. It will show how both the front-end and back-end have been implemented according to the core requirements defined in both the requirements document and design document. It will also describe the necessary changes that needed to be made to complete the implementation.

### **2.1. Front-end**

For our front-end we developed our UI/mobile application in Android Studio. This allows the user to easily download our application through an APK or in the future, the Google Play Store if they have an Android device. It will include a login and register screen that will connect to the back-end database storing user information. The next page will include a switch that allows the user to turn the location sharing for their device on and off. It also has a button to take the user to a map with all the other devices being tracked on the account.

### **2.2 Back-end**

For our back-end we used Firebase. This service allows us to store our user information and our location data from our trackers. It easily integrates into our application and saves us the trouble of setting up our own database. The authentication section holds of the users. It shows their emails and assigns them a UserID. The next section holds the locations of the tracked devices using latitude and longitude.

### **3. Architectural Change**

Any architectural changes that we made during our implementation process as well as the reasoning behind them.

#### **3.1. Firebase Back-end**

- For our back-end, we have decided to use firebase to provide a scalable database that integrates with our android application smoothly. The firebase will keep track of our users information as well as the trackers.
- We decided to use Firebase instead of creating our own mySQL database because it made for a more efficient implementation with what we were trying to build.

#### **3.2. Hardware Switch to Android Smartphones**

- Our team worked hard to integrate an external tracker using an Arduino, but had trouble implementing it with our application as none of us have a strong background experience with it. Rather than forfeiting the project, we have decided to implement the software using android smartphones as the subject being tracked instead.
- The software is constructed in a way where a future external tracker may be used, but for the sake of our deadline, we have opted to track smartphones instead.

## **4. Detailed Design Change**

Any detailed design changes that we made during our implementation process as well as the reasoning behind them.

### **4.1. List of Trackers Now List of Users**

- Rather than going forward with a list of trackers a user has, we implemented a list of users that the users can see. Each user will be part of a group this user is a part of. This will act as our list of trackers.
- Since we are unable to get the actual trackers for the project, we are simulating it through this functionality of seeing other smartphones connected to the group. Theoretically, a user could use a smartphone as the tracker to track their bags.

### **4.2. Allowing Location Sharing**

- Instead of turning on and off trackers we are going to allow the holder of the smartphone to turn location sharing on and off.
- The reason for this change is the context of the tracking device. Since a smartphone is not a device exclusively used for this application, it makes the most sense to allow the location sharing to be turned off for when the application is not being used.

## **5. Requirement Change**

### **5.1. User Requirement Changes**

#### **5.1.1. Functional Requirements**

- The original functional requirements revolve around the user having an external tracker and activating it. The application now needs the users permission to connect a smartphone and allow its location to be tracked.

#### **5.1.2. Non-Functional Requirements**

- The non-functional requirements remain the same other than how the back-end and front-end are implemented. The front-end will have a login screen, a tracker setting screen, and a map screen connected to the Google map API. The back-end will specifically be using Firebase to handle all user data.

### **5.2. Does This Change Our Design?**

The changes made to our requirements slightly changes our design and how we implement our application. The user no longer needs to buy an external tracker, but the target market has shifted. Less users are likely to use a smartphone to constantly track luggage or a backpack. The loss of an external tracker changes up how we gather the locations for our mobile application, but the functionality of tracking multiple devices will remain the same.