./

Learning Report – Shadow Project - TrackMyTrip

Course Code: <CODE>



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **To be Approved By** | **Remarks/Revision Details** |
| 1 | 08/12/2020 | Shrawani Bawage (99002688)  R Pavan Raj (99002640)  Arvind Kumar D R (99002493) | Kalpesh Patil  Pagala Prithvi Shekhara |  |

**Document History and Team Details**

Contents

[Shadow Project TrackMyTrip 4](#_Toc58337863)

[Introduction: 4](#_Toc58337864)

[Topics/Subtopics 4](#_Toc58337865)

[Objectives/Key deliverables 4](#_Toc58337866)

[Low level Requirements: 4](#_Toc58337867)

[High level Requirements: 4](#_Toc58337868)

[Basic Flow of Implementation: 5](#_Toc58337869)

[Design 5](#_Toc58337870)

[High level behavioral design 5](#_Toc58337871)

[Low level Behavioral design 6](#_Toc58337872)

[Test Plan 6](#_Toc58337873)

[Unit Testing 6](#_Toc58337874)

[Integration Testing 7](#_Toc58337875)

[Individual Contribution & Highlights 7](#_Toc58337876)

[Summary 8](#_Toc58337877)

[Challenges faced and how were they overcome 8](#_Toc58337878)

[Future Scope (If applicable) 8](#_Toc58337879)

**List of figures**

[Figure 1 High Level Behavioral - Use Case Diagram 5](#_Toc58334460)

[Figure 3 Low Level Behavioral - Activity Diagram 6](#_Toc58334461)

**List of Tables**

[Table 3 Low Level Requirements 4](#_Toc58334468)

[Table 4 High Level Requirements 5](#_Toc58334469)

[Table 5 Unit Testing 7](#_Toc58334470)

[Table 6 Integration testing 7](#_Toc58334471)

# Shadow Project TrackMyTrip

## Introduction:

In this era of smart phones, one can hardly find a phone without tracking application. This application TrackMyTrip, is aimed to record trips of the user. Whenever a user wants to record a trip, he/she will START the trip. The app will record the location (through GPS/4G) at specific interval of time (e.g. 10 seconds, 1-minute etc, a configurable value through menu). Whenever user finishes the trip, he/she will STOP/END the trip. All the trips will be saved locally on the mobile. User should be shown the list of all saved trips. Any trip should be selectable, which will fetch the saved data. There will be integration with GoogleMaps library. All recorded location points will be plotted on GoogleMaps view, to show entire trip (not in GoogleMaps app, but in Track-My-Trip app itself).

## Topics/Subtopics

· Learn Android Studio

· Learn LocationManager APIs

· Learn data storing methods, e.g. SharedPreferences or SQLite DB

· Learn Google Maps APIs

· Learn Android App UI Components

## Objectives/Key deliverables

· Recording of data for 1 trip.

· List of all saved trips

· Loading of data of selected trip

· Plotting of Trip data on Google Maps

· Live plotting of location data on Google Maps view

· Optional: Colouring of location data based on the speed.

## Low level Requirements:

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | Integrating google Maps with marker |
| LL\_02 | Connection with SQLITE database |
| LL\_03 | Showing trip route on Google maps |
| LL\_03 | Implementation of application for background execution |

Table 1 Low Level Requirements

## High level Requirements:

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | Open the Application |
| HL\_02 | Start Button triggers dialog box to take details from input |
| HL\_03 | Dialog box – take trip name and select timer interval |
| HL\_04 | Stop Button to end the trip |
| HL\_05 | Show Trips button to show the previously saved trips |
| HL\_06 | View button to view the trip route on maps |
| HL\_07 | Close the application |

Table 2 High Level Requirements

## Basic Flow of Implementation:

1. To integrate google maps API to view the map and record current locations.
2. Understanding basic mechanism SQLITE Database and connection with same
3. Recording/Storing the trip locations based on details provided in start Button in SQLITE database
4. Ending the trip as soon as stop button is triggered
5. Show trips button to show all the saved trips
6. Retrieving locations from SQLITE Database for each trip
7. Marking the trip route on Maps for each trip
8. Running the application background

## Design

## High level behavioral design

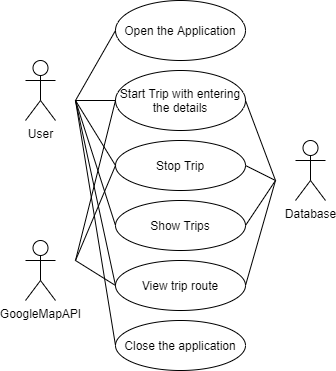


Figure 1 High Level Behavioral - Use Case Diagram

### Low level Behavioral design

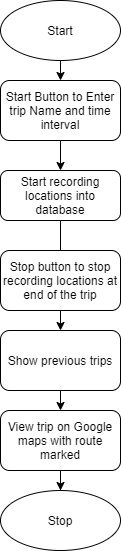


Figure 2 Low Level Behavioral - Activity Diagram

## Test Plan

### Unit Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Test id** | **Description** | **Expected input** | **Expected output** |
| HL\_01 | Open the Application | Click the application icon | The application should open |
| HL\_02 | Start Button triggers dialog box to take details from input | Press the start button | A dialog box should open |
| HL\_03 | Dialog box – take trip name and select timer interval | Enter the details of the trip and select the timer interval | Validate the input and provide message whether the trip has started |
| HL\_04 | Stop Button to end the trip | Press the stop button | Displays message that tri has ended. |
| HL\_05 | Show Trips button to show the previously saved trips | Press the show trips button | Displays all the previously saved trips |
| HL\_06 | View button to view the trip route on maps | Press the view button | Opens a new activity which shows maps with the route marked |
| HL\_07 | Close the application | Click the back button | The application should close |

Table 3 Unit Testing

### Integration Testing

|  |  |  |
| --- | --- | --- |
| **Test id** | **Description** | **Expected output** |
| LL\_01 | Integrating google Maps with marker | Map display with current location marked |
| LL\_02 | Connection with SQLITE database | Viewing database to check the trips |
| LL\_03 | Showing trip route on Google maps | Map display with route marked on it |
| LL\_03 | Implementation of application for background execution | Application runs even when the app Is closed until and unless stop button is pressed |

Table 4 Integration testing

## Individual Contribution & Highlights

Shrawani Bawage

* Week 1-2

1. SQLITE basics  
2. basic commands of create, update and deletions  
3. storing and retrieving data   
4. simple programs to understand linking with database

* Week 3

1. created database  
2. synched database with getting locations continuously based on start button  
3. working on creating dynamic tables  
4. understanding background and foreground services

R Pavan Raj

* Week 1-2

1. Created API key for the Google maps integration on the app

2. A sample app to put the marker on the map on given latitude and longitude

3. Usage of polyline function to draw lines between any two-given latitude and longitude

4. Retrieve data of latitude and longitude using the method getlatitude and getlongitude

* Week 3

Arvind Kumar D R

* Week 1-2

1.How to use background services

2. UI for the application

* Week 3

1. UI part on how to get sync on map activity and database.

2. Used Recycler view to show trips of previous entries

3. Learning more about background services.

4. Dialog box for start button

## Summary

Goals achieved

* Synchronization with google maps
* Implementation of integrating the application of database is achieved.
* Able to store and retrieve locations for every trip
* Able to show details of all the previous trips

Goals yet to achieve

* Plotting of Trip data on Google Maps
* Live plotting of location data on Google Maps view

## Challenges faced and how were they overcome

* Integration of Google Maps API to store updated locations continuously
* Managing SQLITE Database
* Pop up/ Alert dialog box

All the challenges faced were resolved by referring to materials and videos available online

## Future Scope (If applicable)

With further functional improvements, this tracking application can be used for many purposes like to track your run, track car rental services, track delivery services etc.