Smart Train Tracker

A dissertation submitted for the Degree of Master of Information Technology

A. A. T. P. De Silva
University of Colombo School of Computing
2020

Abstract

Sri Lanka Railway (SLR) is considered as one of the most popular transportation modes in Sri Lanka for many decades. Currently, due to impracticable static train timetable, daily train scheduling problems, inability to capture the real-time location of trains and alert passengers on real-time updates of train arrivals, the inevitable issue which is faced by the train commuters is the delaying rails frequently. This dissertation presents a solution, Smart Train Tracker is a Real-time Location based Crowdsourcing Train Tracking Android Application to enhance the effectiveness and efficiency of public train transportation. The proposed application is developed by combining Global Positioning System (GPS), mobile computing and crowdsourcing technologies to gain information from the passengers and provide visual positioning using Google map in real-time. Additionally, it predicts Estimated Time to Arrival (ETA) of a train to any given railway station for better user experience and for better admin management.

User registration by firebase phone authentication with OTP, detection and displaying nearest train stations available within 2km radius of user's current location, searching train schedule by giving start and destination stations, detection of moving train location by passengers' mobile phones, map plotting with real-time train location, calculating Estimated Time to Arrival of a particular train, displaying distance closure notification within 1.5km to destination and user feedback emails to app developer are some services that are provided by the mobile application.

Furthermore, as the specific hardware implementation is not in this application, the zero capital investment cost can be considered as a key benefit of the application. Ultimately, the availability and efficiency of the railway services would be improved by using the Smart Train Tracker.

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: A. A. T. P. De Silva
Registration Number: 2017/MIT/014
Index Number: 17550145

Signature: Date:

This is to certify that this thesis is based on the work of Ms. A. A. T. P. De Silva under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. G. D. S. P. Wimalaratne

Signature: Date:

Acknowledgement

First and foremost, I acknowledge the immense guidance and assistance was given by Dr. G. D. S. P. Wimalaratne, the project supervisor with the deepest sense of guidance through the learning process of this master thesis.

Furthermore, I hereby forward my heartfelt gratefulness towards the MIT Coordinator of UCSC and to the Project Examination Board for letting me to follow this worthy and excellent world recognized master degree.

I acknowledge the greatest support which was given by Information and Communication Technology Agency (ICTA) of Sri Lanka for providing credentials to access Static Train Schedule API from Lanka Gate API Store.

I make this a chance to convey my appreciation and thankfulness towards Mr. Danishka Navin and Mr. Yasitha Chinthaka who extended their greatest assistance and guidance to make my effort a success.

Then my thanks and appreciations go to my colleagues and people who have willingly helped me out with their abilities.

Finally, my sincere gratitude goes to my beloved family for the utmost support they have given throughout this achievement.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API - Application Programming Interface

APK - Android Package Kit

ETA - Estimated Time to Arrival

GIS - Geographical Information System

GPRS - General Packet Radio Service

GPS - Global Positioning System

GSM - Global System for Mobile Communication

GUI - Graphical User Interface

IDE - Integrated Development Environment

OTP - One Time Password

RFID - Radio Frequency Identification

RUP - Rational Unified Process

SLR - Sri Lanka Railway

UI - User Interface

UML - Unified Modelling Language

Chapter 1. Introduction

1.1 Identification of the Problem

Sri Lanka Railway (SLR)/ Ceylon Government Railway was conceived by Sir Henry Ward, the Governor in 1864 in order to deliver coffee from estates by constructing the mainline from Colombo to Ambepussa. Because of the replacement of tea due to the collapse of coffee plantations, the railway has expanded to transport tea to Colombo. Furthermore, manure, machinery and labour were transported to Kandy. Due to cheapness and safety, the British expanded railway into hinterlands in order to transport coconut and rubber productions. The railway system was opened towards the east because of Trincomalee natural harbour. Likewise, the railway system has spread all over the country serving several purposes. The railway network in Sri Lanka consists of 1561 km of rail tracks and maintained nine railway lines [1].

Even though the main purpose of the railway system was on freight transportation in the early days, currently it has changed into passenger transportation. Currently, 369 trains which include 16 intercity and 67 long-distance trains are operated and about 3.72 million passengers are carried by Sri Lanka Railway daily [1]. Trains are used as the mode of transportation by many commuters daily due to the low cost, comfortability and relaxation.

According to the statistics of Sri Lanka Railway 2017, thousands of passengers and plenty of freight are transported daily island-wide and the SLR is earned 83% of the total revenue from passenger transportation [2]. It is confirmed that passenger transportation and passengers can be considered as their main source of revenue. Therefore, it is their responsibility to provide reliable and effective service towards passengers.

Even though both vision and mission of Sri Lanka Railway have emphasized the efficiency and punctuality of railway service, delaying trains can be considered as the main drawback of the rail transport system [1].

In our country it is essential to SLR to provide their service to carry the daily commuters to their workplaces. Unfortunately, the delaying rails has become more frequent nowadays.

Even though delaying of trains is unavoidable, time wastage of passengers has become the most crucial matter which passengers face due to delaying rails. The impracticable train timetable, scheduling problems, inability to capture the real-time location of trains and alert passengers on real-time updates of train arrivals are the main causes for wasting of passengers' precious time. Therefore, it has recognized the importance of introducing a prior awareness system to train commuters. It could save their precious time as well as enhance the efficiency of the rail service.

Though the daily train timetable schedule has existed in the Sri Lanka Railways website, due to lack of real-time updates it does not provide accurate arrival and departure time of the trains. Consequently, it makes the passengers more inconvenient and creates a bad impression on the railway department.

Furthermore, to have a reliable and efficient service it is very important to introduce a new accurate method to inform the commuters about daily prevailing conditions of trains.

In order to provide accurate real-time train schedule information, it is proposed here to give a technological support system. In brief an app which allows the passengers to get fully updated at once, about the daily train schedule.

1.2 Motivation

Since the train delays are unavoidable due to impracticable train timetable, daily train scheduling problems, inability to capture the real-time location of trains and alert passengers on real-time updates of train arrivals, there is dissatisfaction about the prevailing railway service among current commuters. By minimizing the time wastage of the commuters will lead the way to reduce the dissatisfaction of them as well as enhance the quality of life by allowing the community to manage time. Eventually, it will uplift both the income of the railway system and efficiency. Hence, establishing a real-time system would be the ideal solution for this matter.

1.3 Aim and Objectives

The aim of the project is developing a real-time train tracking mobile application to enhance the time management and efficiency for both community and the Sri Lanka Railway.

- Track locations of the trains in real-time
  The proposed system will detect the geo positioning of the train and its speed in real-time by using Global Positioning System (GPS) data of train passengers' mobile phones and provides the locations of the trains through the Android mobile application with a live-updating map to the train passengers.

- Predict Estimated Time to Arrival (ETA) of a train to any given railway station
  The proposed system will predict and notify approximately correct train arrival time to any given railway station by using the location and speed of the train. Therefore, passengers will be able to access this information for reaching the station on time without time and energy wastage and find an alternative train instead of their usual train for reaching their destination without any inconvenience and hesitation.

- Minimize the time wastage of the train passengers
  Minimizing the time wastage of train passengers will be another major objective of the suggested application. Since the train tracking in real-time and estimating train arrival time, train commuters will be able to get train timing on time without wasting their precious time and make their day to day more efficient.

- Enhance the accuracy and efficiency of the railway system
  Because of providing accurate train arrival and departure times for the train passengers, passengers would not waste their precious time and it will help to uplift the impression of the railway services. Ultimately the efficiency and accuracy of the railway system will be enhanced by the system.

1.4 Scope of the Project

- Tracking Location of the Train
  The proposed Smart Train Tracker is a Community Based (Crowdsourced) Train Tracking Application which detects and streams the location of moving trains in real-time by allowing users to share information generated by Global Positioning System (GPS) in their mobile phones. Then the location of the train will be displayed to the train passengers through the mobile application. According to the speed of a particular train and the distance to the destination station, the system will automatically update the train timing and will be displayed and alerted it to the users in real-time. The proposed mobile application will be beneficial to the train passengers who will use the proposed application and the train administrators in Sri Lanka Railway. The system will have consisted of a dashboard application for the train administrators and mobile application for train passengers.

- Estimating Time to Arrival of a Train
  By using Global Positioning System location of the mobile, departure time in the current railway station and the train speed, the system will automatically estimate the Time to Arrival of the train in the next railway station and it will be displayed to the train commuters via the mobile application.

- Alerting Services for Users/ Train Passengers
  For enhancing the service provided by the mobile app it is introduced an alerting feature in this mobile application. Push notifications can be used for notifying the users regarding the train arrival time according to user preferences. Furthermore, a dashboard can be introduced for SLR for alerting.

1.5 Dissertation Overview

The **Introduction** chapter is the first chapter of this dissertation which consists of an overview with motivation, aim and objectives and scope of the project. After that a review of similar systems is included in second chapter **Background**. The **Methodology** chapter consists of three major sections such as Analysis, Design and Implementation. The Analysis section focuses on an overview of system analysis. The functional and non-functional requirements for suggested application are given under the analysis phase. The proposed system design with relevant design diagrams (Activity Diagram, Use case diagram, Sequence diagram and Class diagram), Database structure diagram and User Interface design for the suggested solution are illustrated under the Design section. In the Implementation section, the implementation environment, techniques and tools which are used for the Smart Train Tracker development are included with important code segments. The user evaluation and A critical evaluation of the project are elaborated in the **Evaluation** chapter. A summary of the problems faced, lesson learnt and future enhancements are discussed in the **Conclusion** chapter. The URL references and other essential citations which helped to complete this dissertation are included in this **References** section. The detailed views about the system and other supplementary materials are provided in **Appendices** for those who are interested on development of the Smart Train Tracker.

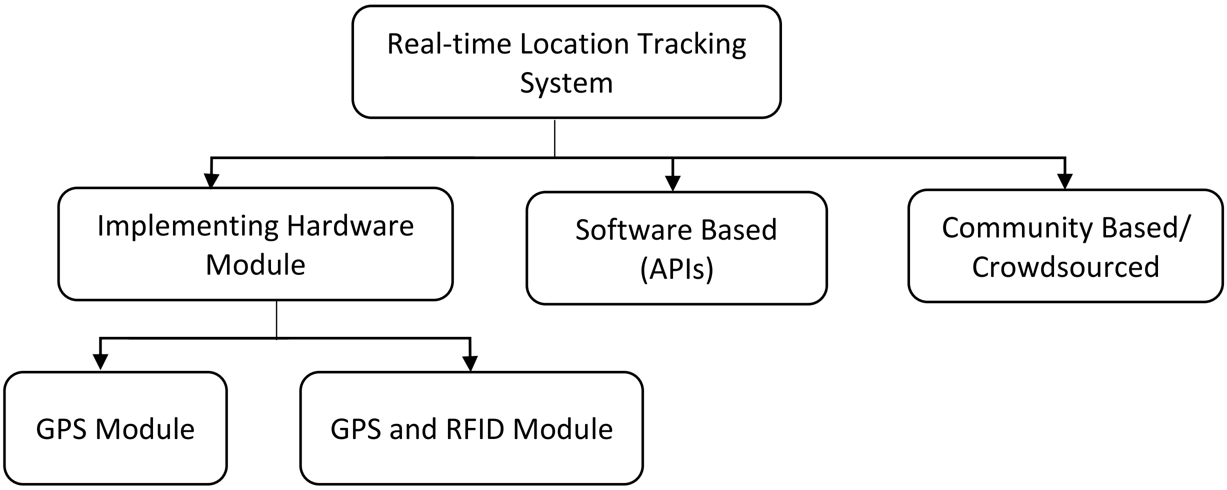
Chapter 2. Background

2.1 Overview

The main focuses of this chapter are the background of the train systems and an overview of various types of technologies which are used to location tracking. A subsection of this chapter discusses about a review of similar systems which are already existing with highlighting their features. Finally, a comparison of alternative design strategies is discussed in this chapter.

2.2 A Review of Similar Systems

By considering existing applications, a vast knowledge could be grabbed on how to implement required functionalities and the system. A few similar systems which are related to real-time location tracking system were reviewed to develop the proposed system.



Real-time Location Tracking System
Implementing Hardware Module
Software Based (APIs)
Community Based/ Crowdsourced
GPS Module
GPS and RFID Module

Figure 2. 1: Real-time Location Tracking System

According to Figure 2. 1, similar systems of the real-time location tracking system can be classified into three major classifications such as the systems with implementing hardware modules, software based (APIs) and the crowdsourcing/ community-based systems. The systems which is implementing hardware modules as the solution can be divided into further such as implementing GPS modules and GPS and RFID modules.

2.2.1 Online Train Schedule

Online Train Schedule Service offered by the Sri Lanka Railways - The static daily train timetable schedule is existed in the Sri Lanka Railways website and this is the online service offered by Department of Railways, with the assistance of Information and Communication

Technology Agency of Sri Lanka (ICTA) [3]. The backend for this service is hosted as a web service, and same service is used in both mobile and web applications.

This service is aimed mainly at displaying the static train schedule for the users - anyone with access to the service.

An enquiry can be placed by providing start and destination stations. schedule of the train, details of the stations and price of tickets can be viewed based on the relevant searching condition.

In this system, only the static schedule data is displayed and no way of confirming for the passengers if the train is available or not in real-time. As well as the system does not offer a method to view train delays and way to locate the trains in real-time. These reasons can be considered as the drawbacks of this system.

2.2.2 GPS/ GSM Based Train Tracking System

In order to make use of mobile networks to assist public transport services, Dileepa Jayakody and his colleagues presented a **GPS/ GSM Based Train Tracking System** by merging technologies such as Global Positioning System (GPS), Geographical Information System (GIS), Global System for Mobile Communication (GSM) and particular software [4].

The location tracking system identified the position of the train with maximum accuracy and transferred Geo Location data to the central server via GSM technology. It has been developed by using intrinsic GPS component for data manipulation and information analysis. By using maps, location data is prepared to provide visual positioning and an overall view on location of the train is granted by using GIS technology. Latitude and longitude position of the train with its speed can be used to provide accurate scheduling which is related to the arrival and departure time of the train on each station. Moreover, the assistance is provided by the system to the staff administration to consider the achievable security concerns and overcome the identified issues effectively by utilizing provided communication techniques.

This tracking solution has been consisted of the following three key components.

- The mobile device which is used to track positioning of the train (GPS/ GSM train locator component)
- Remote server that handling received location data and requests from users those who access the system simultaneously
- Graphical User Interface that facilitates services for users

In their study, the cost factor, size of the module, durability and low power consumption are considered when designing and implementing the train locator unit in the train.

The same paper has been explained that the feasibility of the train locator unit is depended on the power supply unit of the module and for the locator module to function properly, consistent power delivery and a flat voltage must be maintained.

Furthermore, it reported as the GPS receiver of the module is competent of detecting the location and velocity of the specified train by detecting data from the GPS satellites. Then the location data is transferred to the remote server periodically via the GSM transmitter and ability to store data in a buffer when the failure of GSM connection. Finally, when GSM connectivity is reconnecting it can be synchronized with the central server.

In that paper it is further mentioned that in comparison to GSM with GPRS, GSM has high feasibility and accessibility island wide while GPRS has poor coverage and consumption in rural areas. GSM services can be supplied at reasonable price due to the competition among great number of GSM service providers. These are the reasons which paved the way to choose GSM for communication method between the GSM component and the remote server by them [4].

In their proposed system, they mainly focused on train administrative staff to enhance the efficiency and safety of their services rather than the passengers. Considerably high cost of implementation and infrastructure cost can be considered as the main drawback of this system.

2.2.3 Railway Train Dynamics Analysis System/ Train Traffic Optimization System

In 2014, **Railway Train Dynamics Analysis System/ Train Traffic Optimization System** has been carried out by joined effort of University of Colombo School of Computing and the Research and Development Unit of the Department of Railways of Sri Lanka [5]. The primary objective of the study is to figure out appropriate heuristic algorithm to optimize the train traffic.

Five key modules are included in this research.

- Module that real-time data capturing
- Module that engine optimization
- Data input controlling module
- Module that data processing and expose
- Module for scheduling

In addition, they have mentioned that the need of real-time train position data in order to better manage the railway traffic. Hence, a mechanism was applied to gather real-time position data of moving train against the accuracy of the schedules and optimizations were evaluated in this research.

As initial step the southern section of the railway track is monitored. There are fifteen trains operate in the southern railway track. A device was implemented on the driver desk of a locomotive for detecting location data, Then, by using Android smart phone train GPS location and train speed were captured and transmitted to a remote server periodically. Rectangular binding box principle of GPS coordination was used to detect just-in-time train positions.

Additional to general public, the information from the system is available for the internal staff such as railway control staff, staff of the railway stations and drivers, facilitating them to monitor train movement.

Collected manual data from specific stations were compared against the real-time captured data in order to evaluate the accuracy. According to this analysis, the mean accuracy was 93% [5].

In the study, cost of implementing GPS tracking devices in each locomotive and some issues related mobile phones which were facilitated by the government such as using those mobiles for personal matters and stolen were the reasons for failure of this study.

2.2.4 Transport Tracker Solution

The **Transport Tracker Solution** is presented by Google Developers in 2015 as a complete solution for managing and tracking mobile assets such as delivery trucks and buses [6]. The solution is an Android app which is capture the location of vehicles and store it in a Firebase Real-time Database. The real-time visibility of vehicles, routes and schedules are provided using a map built with the MAPS JavaScript API, and Places API. The project is open sourced and can be customized according to user preferences.

The solution consists of the below modules.

- Vehicle locator - An Android application that records the locations to the Firebase Real-time Database using a fused location provider with the Google Play Services Location API.
- Data store - A real-time database that stores the locations vehicle connected with the Road API and Firebase synchronizes data to the backends as well as the map.
- Backend - A backend built via Node.js that handles locations from real-time storage and uses the Directions API to estimate time for travel.
- Map - A web solution which presents a customized map using Maps JavaScript API to show routes and location of buses.
- Administrator’s overview - A web application that provides vehicle movements for administrative staff using Maps APIs.

2.2.5 RailRadar GPS

RailYatri.in has launched a tracking platform that detects the real-time train movements using GPS technology. The train movements are displayed using Google Maps with the current status of the train.

The article further is reported that the prior approaches of installing GPS on trains were failed as the high implementation cost of the project. Therefore, a community-based location tracking was presented. Location data is captured by passenger’s smart phone and implementation expenditure is not required. A sample of over hundred trains is been considered and evaluated with hundred smart phones [7].

Moreover, the CEO of RilYatri.in Manish Rathi [7] has expressed his view regarding crowdsourcing such that mathematical modelling is utilized to verify the location of the train by the crowd sourced model. The technology minimizes human errors, while high availability regardless of any climatic situation or vision impairments. One smart mobile phone RailYatri traveler on each train is all it requires [7].

On the map trains operating on time are shown with a green indicator and delayed trains are marked with a red mark. Further, the number of trains tracked and the percentage of delayed and scheduled trains are indicated.

According to the definition of crowdsourcing by Merriam-Webster, this is the process of gaining idea, content or service by soliciting contributions from numerous of individuals and online communities, rather than traditional employees or suppliers [8].

2.3 A Comparison of Alternative Design Strategies

Though the great number of similar systems were reviewed throughout this chapter, Table 2. 1depicts a comparison of the characteristics of similar systems which are mentioned above.

	[3]	[4]	[5]	[6]	[7]
Specialty	Online service	Bespoke	Bespoke	Open sourced	Crowd sourced
Used Technology	Web services	GSM, GPS, GIS	GPS	GPS	GPS
Platform	Mobile/ Web	Mobile	Mobile (Android)	Mobile (Android)	Mobile
Focus Group	Public	Public and Railway administrative staff	Public, Internal staff (Driver, Station master, Controllers)	Public (Bus,Delivery truck)	Public (Train)
Input	Start and End Stations	Train location and Speed	Train location and Speed	Train location and Speed	Train location and Speed
Data Input	Via interface	GPS receiver component	GPS tracking device	Smart phone sensors	Smart phone sensors
Data Processing/ Storage	Web server	Central server	Central server	Firebase	Firebase
Output	Station details, Train schedule, Ticket price	Real-time arrival and departure time in each station	Real-time location of the train	Routes, Schedule, Real-time location	Real-time location, ontime/ delay status
Status	Static	Real-time	Real-time	Real-time	Real-time
Cost	N/A	High	High	Medium	Zero
Advantages	Address to both mobile and web users	Store data in buffer when failure of GSM		Customization	High availability, Reduce human error
Drawbacks	Static schedule, No real-time updates	High cost of implementation and infrastructure cost, Specific	Cost of implementation, Matters related to provided	Though it is open sourced, need to customize	High battery consumption

	on delay or location	power supply, Size of tracking device	phones (stolen, personal matters)		

Table 2. 1: Similar Systems Matrix

According to the current literature review evidence base, it can be listed out some important points regarding the real time train tracking system.

- When using GPS train locator unit for capturing location data, the feasibility of train locator unit, the power supply for the module and ability to store data in a buffer when the failure of GSM connection are the main issues which should be addressed properly for better real time user experience [4].
- The installation of GPS hardware on trains is affected for increasing project costs and it is caused to failure of some projects [7].
- As the solution of above-mentioned issues, it can be introduced a crowdsourcing real time train tracking system which requires zero capital expenditure for capturing data [7].
- There are many advantages of using crowdsourcing solution when compared using a system with a GPS train locator such as requiring zero capital expenditure, reducing human errors and availability under all weather conditions [7].

Chapter 3. Methodology

3.1 Overview

This chapter discusses about analysis, design and implementation of the proposed Smart Train Tracker. The Analysis section focuses on further expressions of functional and non-functional requirements with the help of fact-finding techniques. The proposed system design with relevant design diagrams (Activity Diagram, Use case diagram, Sequence diagram and Class diagram), Database structure diagram and User Interface designs for the suggested solution are illustrated under the Design section. The development environment, techniques and tools which assisted to the development of the suggested application are included under the Implementation section of this chapter. Finally, User Interfaces and the important segments of code lines which include in the project code are presented in the same section of this chapter.

3.2 Analysis

3.2.1 Fact Gathering of the System

The following fact gathering approaches are used to capture functional and non - functional requirements.

- **Observation / Field Visit**
  The current system was monitored to comprehend the complexity in authentic working manual system. This was very helpful to clarify and understand the problems and find solutions through the real nature of the situation of the problem. If the passenger wants to find a train to reach their destination or delaying of a given train, he or she should reach the nearest railway station and find from the manual low-tech departure and arrival time indicator board situated in the railway station.

  Manual low-tech departure and arrival time indicator board situated at Galle railway station is shown in Figure 3. 1.

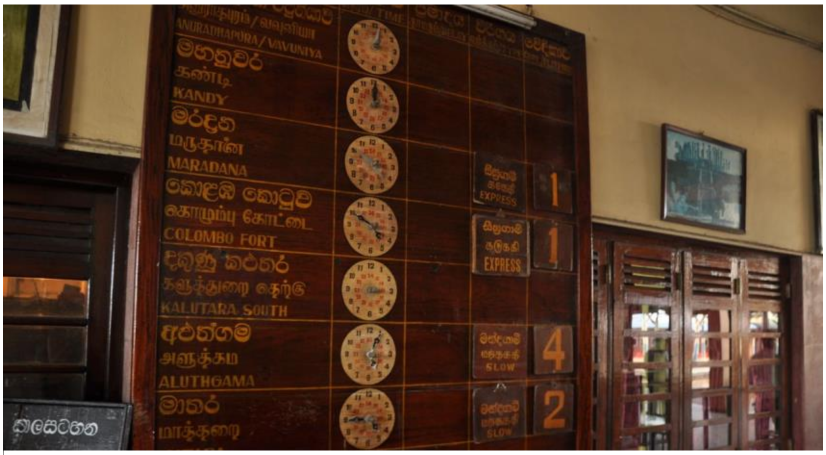
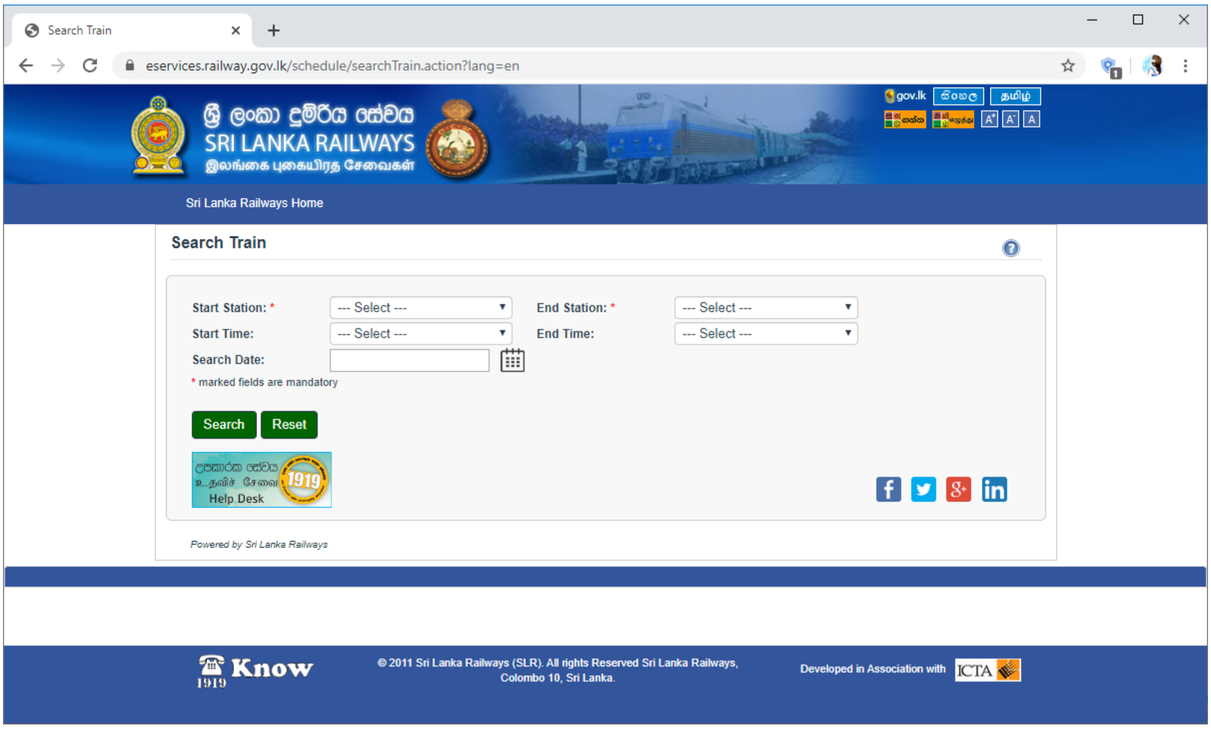


Figure 3. 1: Manual Low-tech Departure and Arrival Time Indicator Board at Galle railway station

- **Existing Website Analysis**

  It was very helpful to identify and understand visiting Sri Lanka Railway official website and collect some evidence on the daily train time table schedule has existed in the Sri Lanka Railways website.

  The interface of the train timetable schedule in Sri Lanka Railway official website is shown by Figure 3. 2 [3].



Search Train

Start Station: * --- Select ---

End Station: * --- Select ---

Start Time: --- Select ---

End Time: --- Select ---

Search Date:

* marked fields are mandatory

Powered by Sri Lanka Railways

© 2011 Sri Lanka Railways (SLR). All rights Reserved Sri Lanka Railways, Colombo 10, Sri Lanka.

Developed in Association with ICTA

Figure 3. 2: The Interface of the Train Timetable Schedule in Sri Lanka Railway Official Website

3.2.2 Requirements of the Proposed System

3.2.2.1 Functional Requirements

The functionalities which encompassed in the proposed Smart Train Tracker application are listed below.

- The mobile app allows cellular data or Wi-Fi to turn on the app.
- The mobile app allows to collect location data when the user using the app to facilitate tracking.
- The mobile app allows to send notifications.
- The mobile app registers user by using phone no, OTP, email, and etc.
- The mobile app restricts unauthorized users from accessing the user's sensitive data such as location and email.
- The mobile app permits users to sign in only if they are registered with the app.
- The mobile app searches a train from nearest station to destination station in given date and time.
- The mobile app lists all the possible trains with details such as date, time, location, available classes, and etc. after searching.
- The mobile app views details of selected train.
- The app displays a train on a map when the train travels in real time.
- The mobile app estimates and displays ETA of a train in all the possible stations within beginning and destination stations when the train travels in real time.
- The mobile app allows users to edit user profile details such as mobile number, name, and email in order to send notifications, build user loyalty, increase retention and performance across multi devices.

3.2.2.2 Non - Functional Requirements

Some non-functional requirements which were identified at this stage for the proposed application are as follows.

- Usability - The graphical user interface of the mobile app must be user-friendly and understandable. This quality attribute can be achieved by using various techniques such as attractiveness and simplicity of user interfaces and easy navigation.
- Performance - After the user enables GPS before starting the app, the mobile app must be able to respond within 15 seconds intervals and able to run smoothly.
- Privacy - Since the sensitive data of the mobile users are captured and collected, the mobile application must be preventing unauthorized access by providing user registration using mobile phone and OTP.
- Reliability - The mobile app must aware the mobile users regarding collecting of users' location details and to be obtained permission, before collecting them. Moreover, the mobile app must be restricted view sensitive data of mobile users such as name, telephone no and etc. to other registered users and must not be indicated on the map.
- Accuracy - All the operations of the app and the information given by the app such as location details, train details and etc. must be accurate for providing quality services to the app users effectively and efficiently.

- Availability - The mobile application must be available 24 hours a day for the mobile application users without any clashes.

3.3 Design

3.3.1 System Overview of the Proposed System

The proposed solution, Smart Train Tracker will be a Community Based (crowd sourced) Train Geo Location Tracking Mobile Application in real time and allow users to share their GPS data via mobile phones only when they are traveling by the train. The captured Geo location and the velocity of the train will be passed to the central server and calculate the distance and Estimated Time to Arrival (ETA) of the train to the following railway station. The location of the tracking train will be displayed on a live updating map with ETA. At the point when the mobile user solicitation to notify the train details with the ETA, the proposed system will be alerted on demand information such as Train ID and ETA to the users. For administrative purposes user profile information and train schedules will be saved on a real time database. The Static Train Schedule API from the Lanka Gate API Store powered by ICTA of Sri Lanka will utilize for the application in order to obtain the train schedule and station list in Sri Lanka as well as Google Maps APIs by Google for searching nearby stations for the passengers and other map related work. User privacy will be one of significant concerns and the security features will be incorporated as much as possible.

The proposed system design for the Smart Train Tracker application is illustrated by Figure 3. 3.

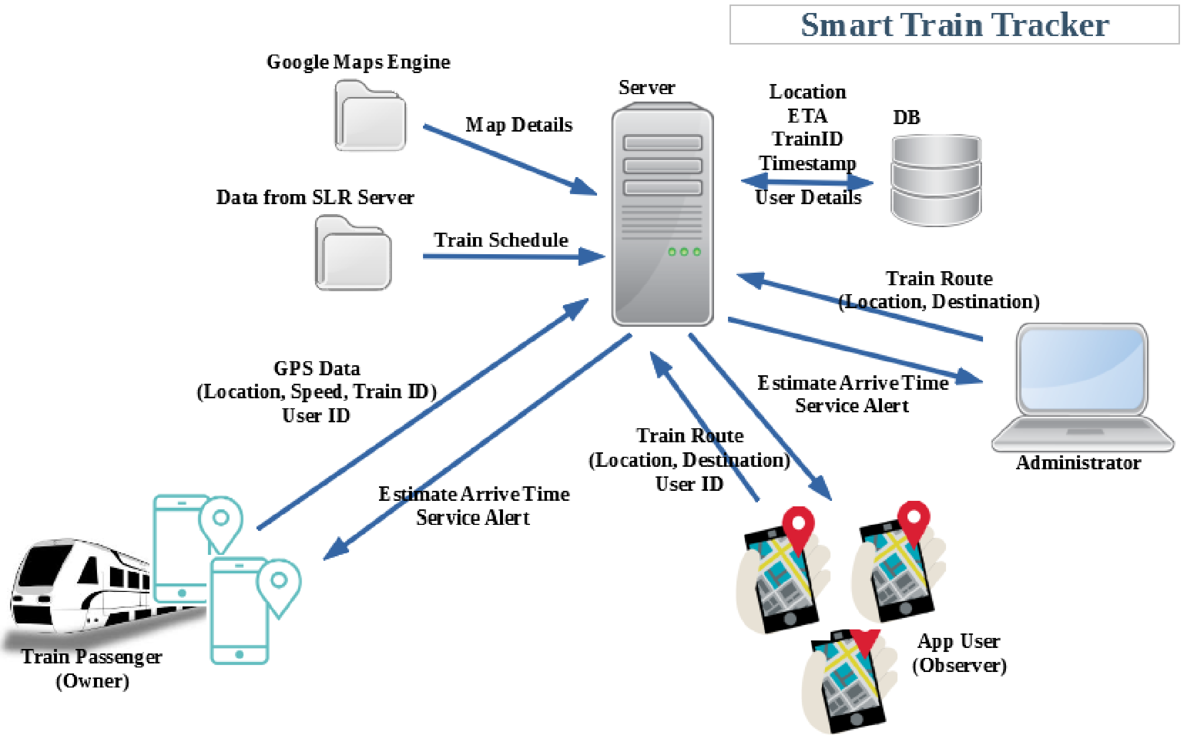


Figure 3. 3: Proposed System Design

3.3.2 Proposed Architecture for the System

As illustrated in Figure 3. 4, the extensible three-tier architecture is introduced for the proposed Community Based (crowd sourced) Real Time Train Geo Location Tracking Mobile Application.

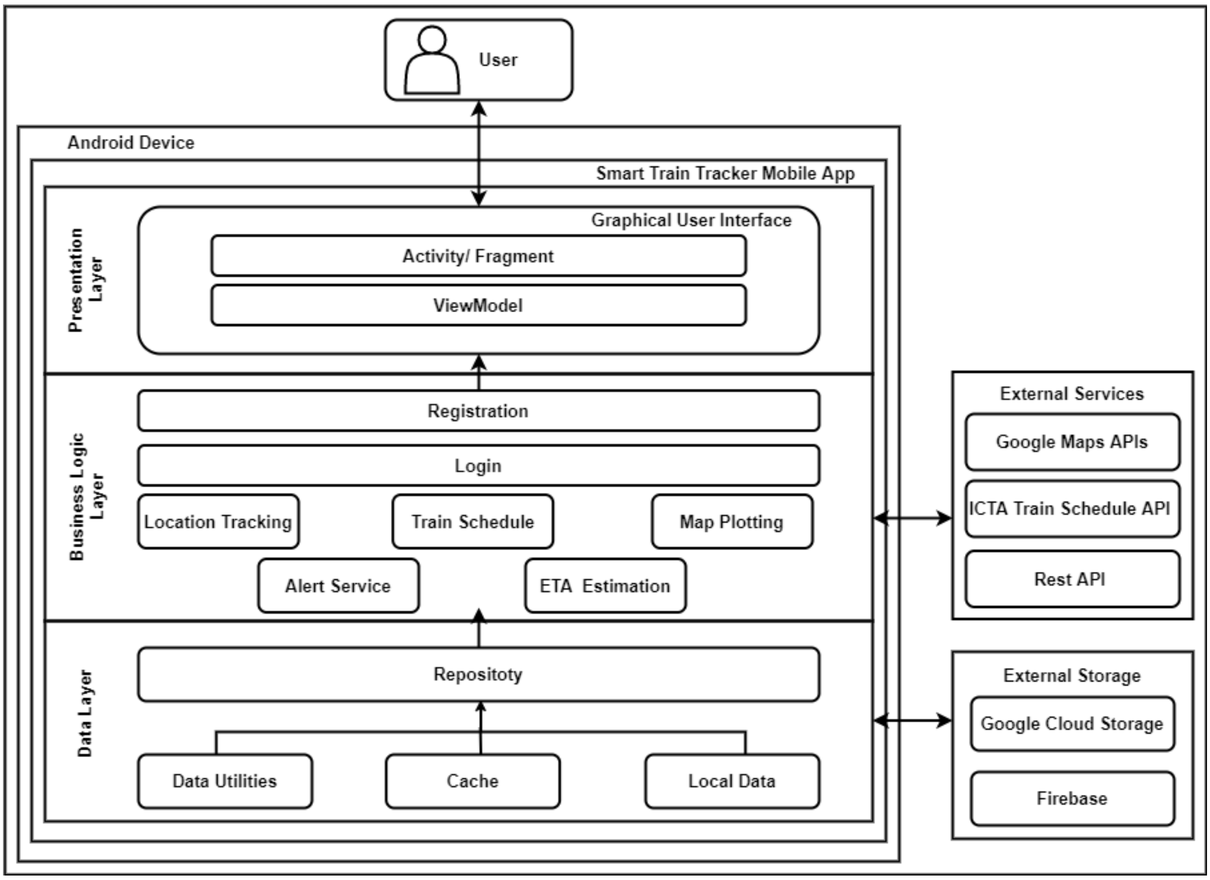


Figure 3. 4: Proposed System Architecture

Presentation Layer - The presentation layer comprises Graphical User Interface which consists of specific UI Components and UI Process Components such as Activities, Fragments and ViewModels. Since the application is based on crowd sourcing, the passengers in the train whose location is tracked and the end users who are willing to use the app interact with the application through this layer.

Business Logic Layer - Registration, Login, Location Tracking, Train Schedule, Map Plotting, ETA Estimation and Alert Service are the key modules of the proposed mobile application. In addition to mentioned key functionalities, external services such as Google Maps APIs, ICTA Train Schedule API and Rest API are supportive services for application's core functionalities under the business logic layer.

Data Layer - The data layer is composed of two distinct categories such as the internal data repository which is inside the mobile app of android device and the external storages which are connected with the respective application. Meanwhile data utilities, cache and local data can be considered as the internal data repository, Google Cloud Storage and Firebase as the external storage.

3.3.3 Methodology for the Proposed System

As the process model for the suggested application Rational Unified Process (RUP) is selected, since it is an iterative software development framework. Further it provides advantages such as managing the changing requirements, support of object-oriented development and performing management easily. RUP comprises four key stages of Inception, Elaboration, Construction and Transition and is shown in Figure 3. 5 [9], [10].

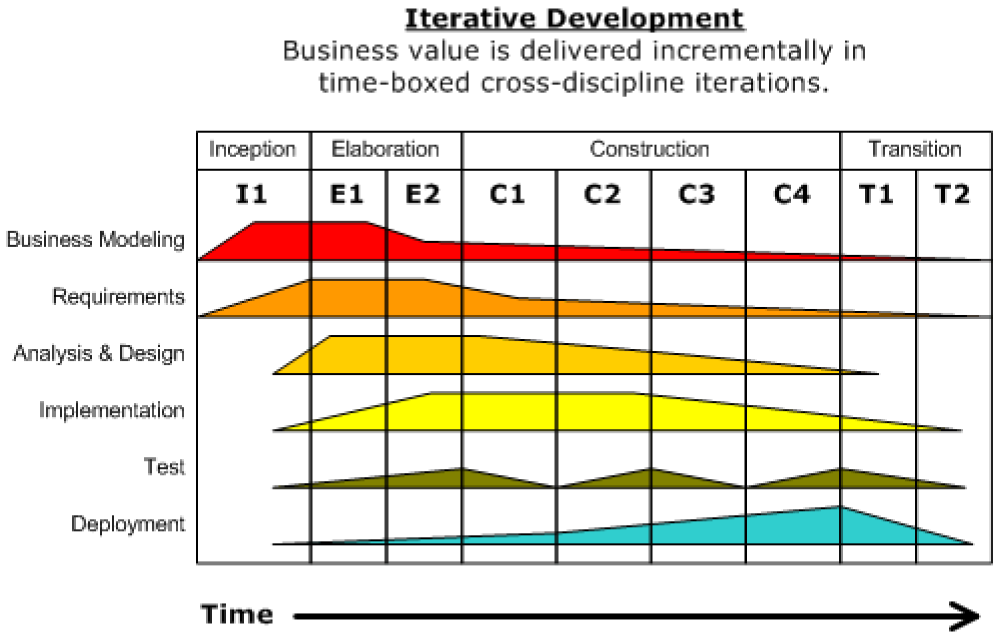


Figure 3. 5: The Major Phases of RUP [9]

3.3.3 Design Assumptions

The following assumptions are made for the suggested solution

- Accuracy of the received GPS location data of the mobile device is considerable
- At least two passengers in the given train use the proposed mobile application
- Users can only get on/ off the train from any station between starting and destination station, but cannot between stations
- Train guard/ driver in the given train uses the mobile application
- There are no more trains traveling parallel to the particular train
- There are no roads in parallel to the rail way lines
- Battery performance is not considered
- Mobile application is allowed to utilize the mobile data connectivity throughout the journey by the train passengers who used the app

3.3.4 Use Case Diagram for the Proposed System

The diagram graphically describes who will be the system users (actor) and in what ways (use cases) the user expects to associate with the proposed solution.

According to the Figure 3. 6, end users and train administrators can be considered as the actors of the proposed system. Furthermore, since the proposed Smart Train Tracker application is based on crowd sourcing end users can be specialized as the Train Passengers (Owner) who help to capture the train Geo position of their phones and the App Users who are eager to use the proposed application.

The proposed system has been categorized into few distinct modules for the better comprehension of development process. The modules are as follows;

- User Registration/ Login Module
- Location Tracking Module
- Train Schedule Module

The Figure 3. 6 illustrates the high-level use case diagram for the proposed Smart Train Tracker application.

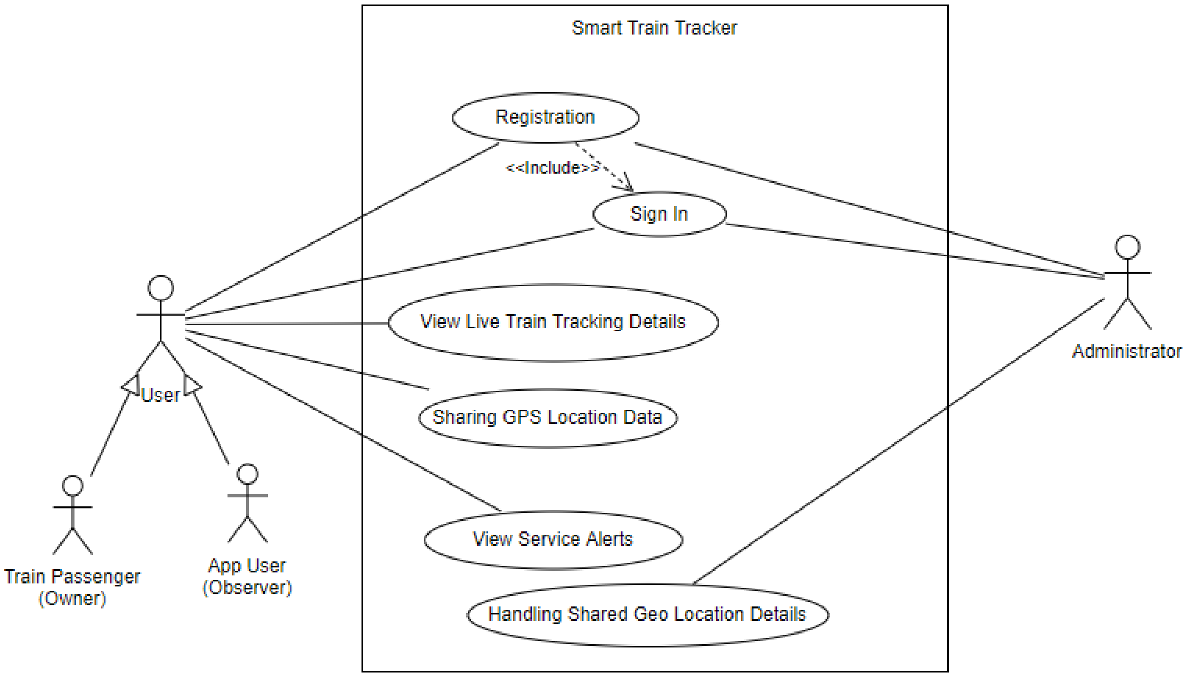


Figure 3. 6: High Level Use Case Diagram for Proposed System

- **Design of User Registration/ Login Module**
  When running this app for the first time on the Android smart phone, the system is checked whether the new user's data is available or not in the database. If the new user's data is not available in the database, the user must register with the phone number, name and e-mail in order to access the system. If the new user is already registered with the system, the system is checked whether the new user is logged in or not as the current user. If the user is already logged in, the main screen of the app will display. Otherwise, user must login

into the system by providing phone number and the OTP which is sent by the Firebase Console to access individually. Firebase Phone Authentication feature is included for the user registration as well as Login and OTP is sent from the Firebase only if the entered phone number is valid.

The logic behind the User Registration/ Login Module is depicted by Figure 3. 7.

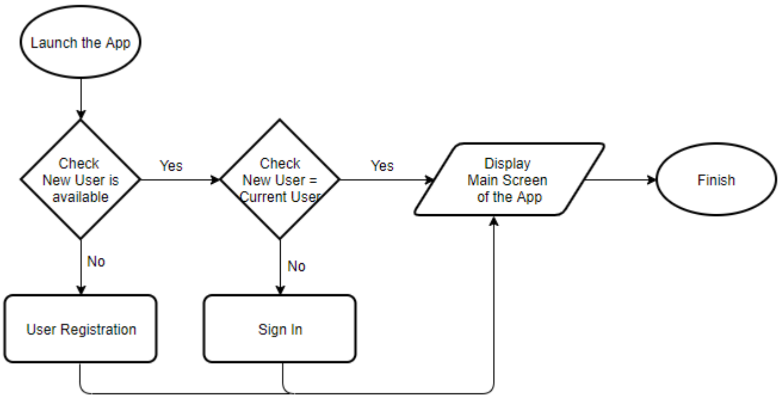

Launch the App
Check New User is available
Yes
Check New User = Current User
Yes
Display Main Screen of the App
Finish
No
No
User Registration
Sign In

Figure 3. 7: Flow Chart for the User Registration/ Login Module

- **Design of the Location Tracking Module**
  It is needed to grant Internet and location permissions from the users in order to capture the geo location data via mobile phones. The location data is sent periodically to the database server of the Smart Train Tracking system in every 30 seconds of time intervals. Detection of the nearest train station, detection of the train that the user is traveling and handling location data from the users those who access the system simultaneously can be considered as the key functionalities under this module.

  The flowchart for the Location Tracking Module is illustrated by Figure 3. 8.

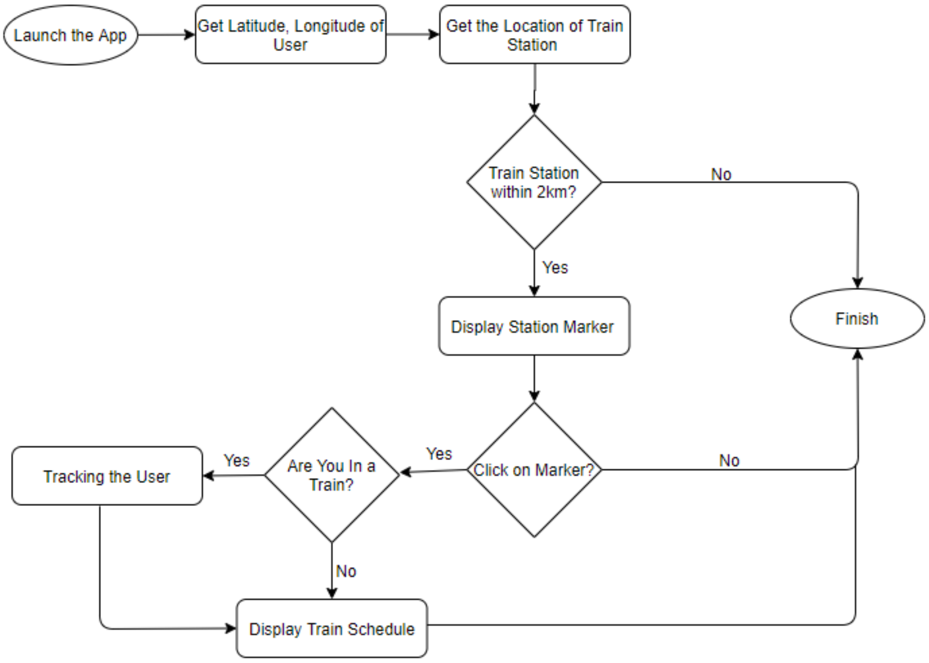


Figure 3. 8: Flowchart for the Location Tracking Module

- **Detection of the Nearest Train Station**
  When the location data is received, the system checks whether a station is available within 2km radius of current location or not. At the point a station is detected within specific area, system is anchored identified nearby stations on the map using markers. Then the user is able to select the nearest station by click on the marker.

  The logic of detecting Nearest Train Station is shown in Figure 3. 9.

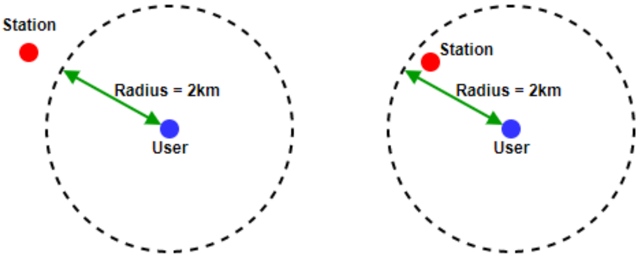


Figure 3. 9: Detecting Nearest Train Station

- **Detection of the train that the user is traveling**
  As soon as the station is selected by the user, the user is asked “Are You In a Train?” as the pop up message by the application. If “In a Train” option is selected by the user, a list of trains is displayed for the user according to the train schedule and the current time. then the train is selected by the user and the passenger’s geo location is tracked as the train location without displaying passenger’s sensitive data.

- **Handling location data from the concurrent users in the same train**
  When detecting the Geo coordination of a specific train from the train passengers, if there are multiple passengers in the same train simultaneously, the latest update of the location is considered as the Geo coordination of the particular train. Further, concurrent user updates can be handled by using Firebase without difficulty.

- **Design of the Train Schedule Module**
  For searching the train schedule in the app, train schedule and train list in Sri Lanka will be directly fetched from the Static Train Schedule API from the Lanka Gate API Store powered by ICTA of Sri Lanka [31]. User must search a train with the StartStationID, endStationID, searchDate, stratTime and endTime and train schedule will be displayed with respected to request data.

- **Design of the Map Plotting**
  Google Maps API is used for map plotting of the application and only the rail route layer is fetched from the entire transit routes. User must be able to view the rail route between selected start station and the destination station. Furthermore, when the user is traveling by the train, train route will be displayed on the map for other app users only if he or she has granted location permission.

3.3.5 Class Diagram of the Proposed System

The class diagram which represents the static view of the suggested system is illustrated by Figure 3. 10. For an instance, a user can only be in one location and one location can have one or more users. Further one train can only exist at one location and there may be one or more trains in one location.

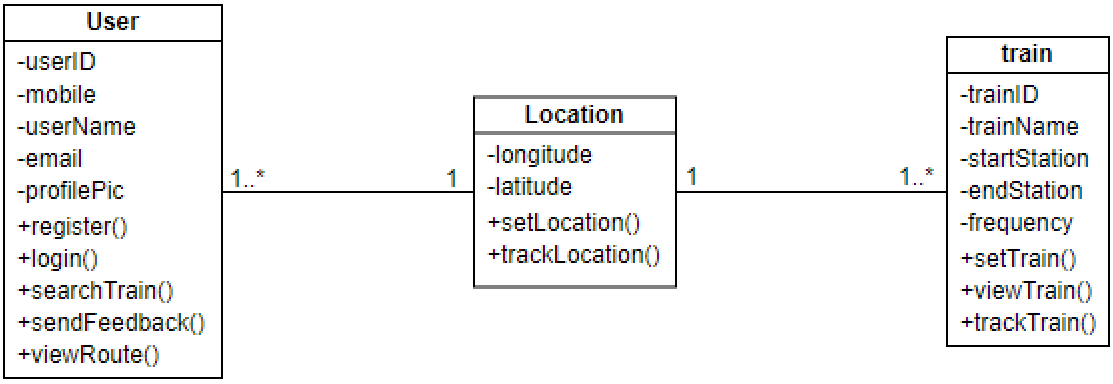


Figure 3. 10: Class Diagram for Proposed System

3.3.6 Database Design of the Proposed System

The Figure 3. 11 is depicted the database structure of the proposed Smart Train Tracker mobile application.

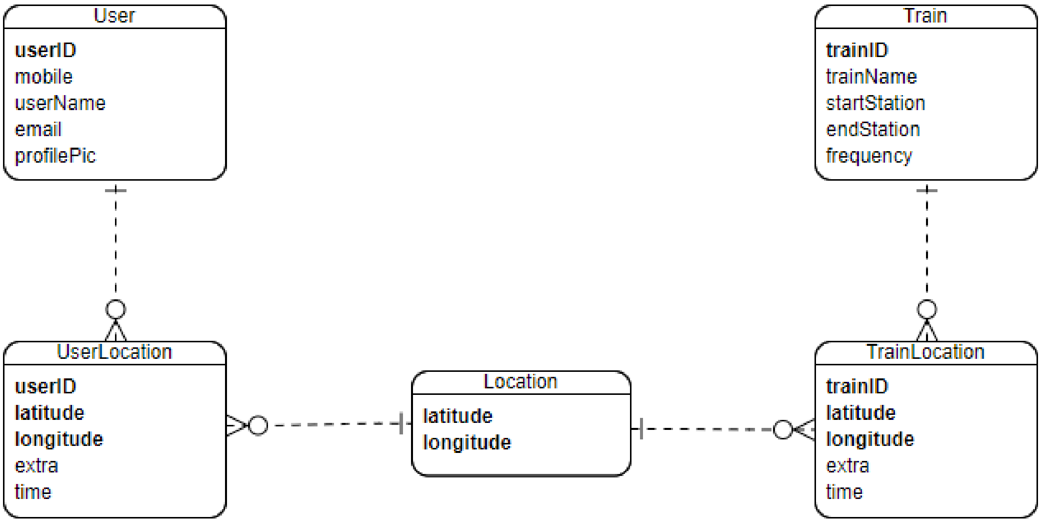


Figure 3. 11: Database Structure for Proposed System

3.3.7 Activity Diagram for Train Schedule of the Proposed System

The activity diagram which is related to the train schedule of the Smart Train Tracker is illustrated in Figure 3. 12.

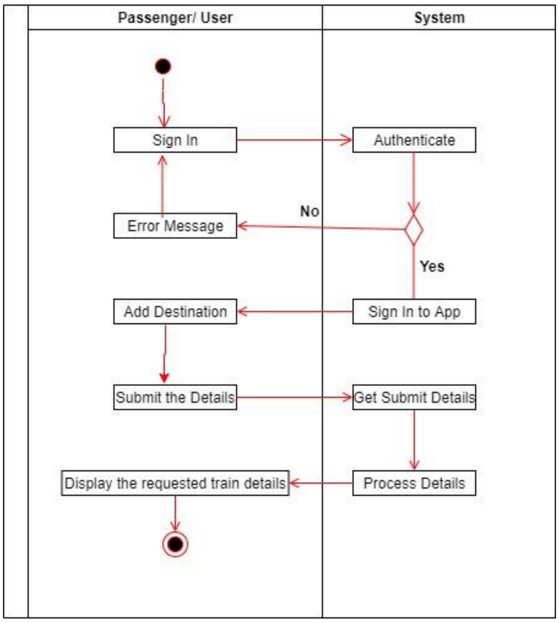


Figure 3. 12: Activity Diagram for Train Schedule

3.3.8 User Interface Design of the Proposed System

Some wireframe designs of the proposed Smart Train Tracker application has given below.

Welcome Screen
After installing the mobile app and when launch the app, it will display the welcome screen first.

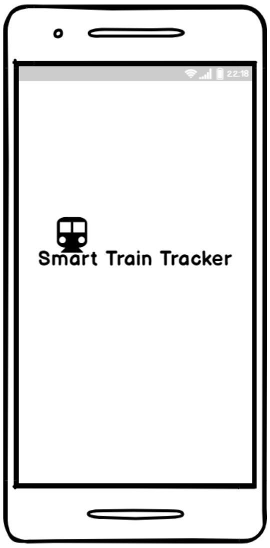
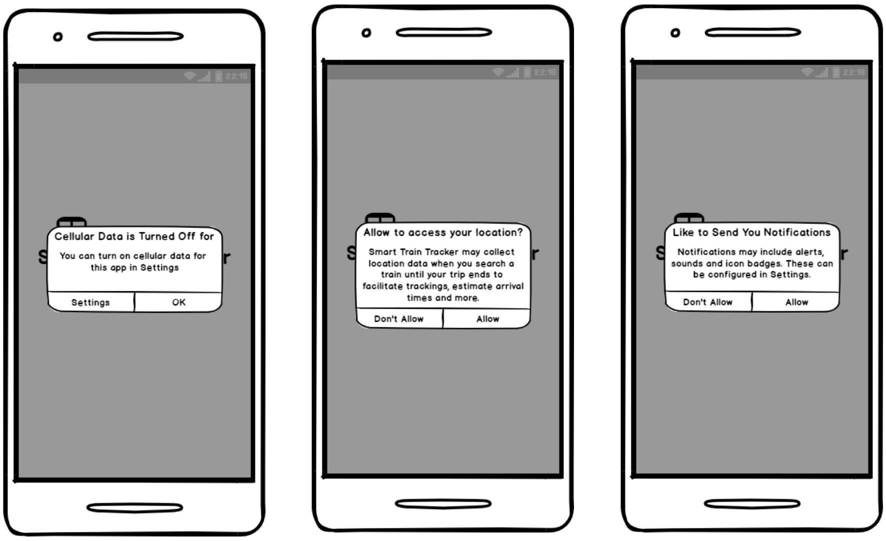


Figure 3. 13:Wireframe for Welcome Splash Screen

Permission Screens

When opening the application, it is needed to grant the permission from the user such as cellular data, location and notification permissions.



Cellular Data is Turned Off for

You can turn on cellular data for this app in Settings

Allow to access your location?

Smart Train Tracker may collect location data when you search a train until your trip ends to facilitate trackings, estimate arrival times and more.

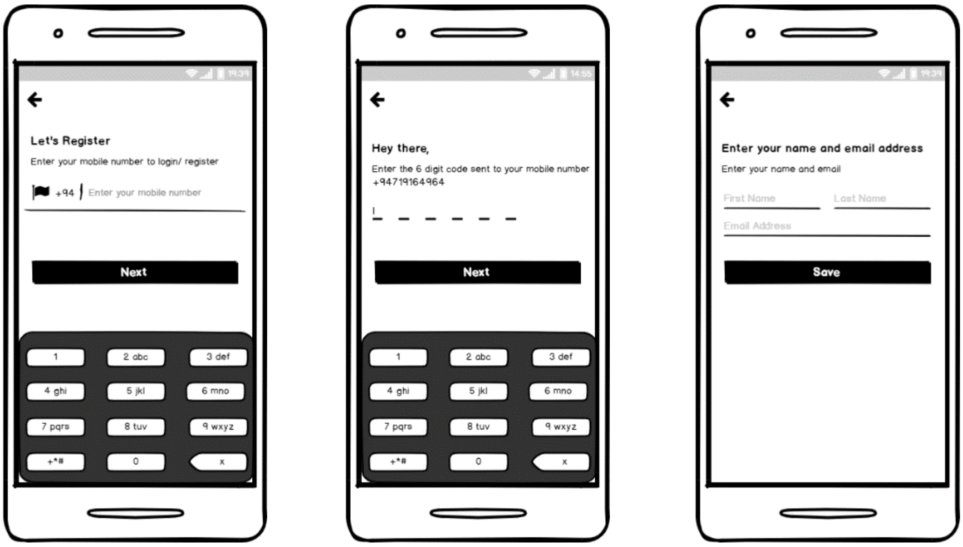
Like to Send You Notifications

Notifications may include alerts, sounds and icon badges. These can be configured in Settings.

Figure 3. 14: Wireframes for Permissions

User Registration/ Login

User is asked to fill required data for user registration or login by using Firebase Phone Authentication.



Let's Register

Enter your mobile number to login/ register

+94 Enter your mobile number

Hey there,

Enter the 6 digit code sent to your mobile number +94719164964

Enter your name and email address

Enter your name and email

First Name

Last Name

Email Address

Figure 3. 15: Wireframes for User Registration/ Login

Location Tracking

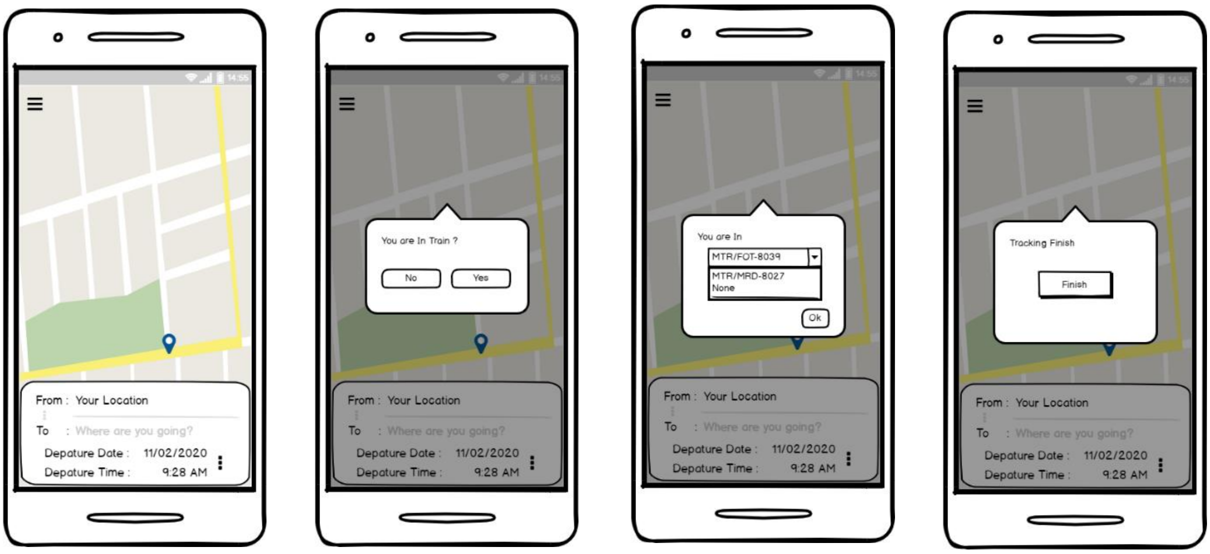


Figure 3. 16:Wireframes for Location Tracking

Train Schedule

At the point when it is needed to search a train; the app gets the nearest station of the passenger’s current location as the starting point and the current date and time as departure date and time. Then it will show the alternative trains. If the train is available in real time it will display in the map with Estimate Train Arrival values.

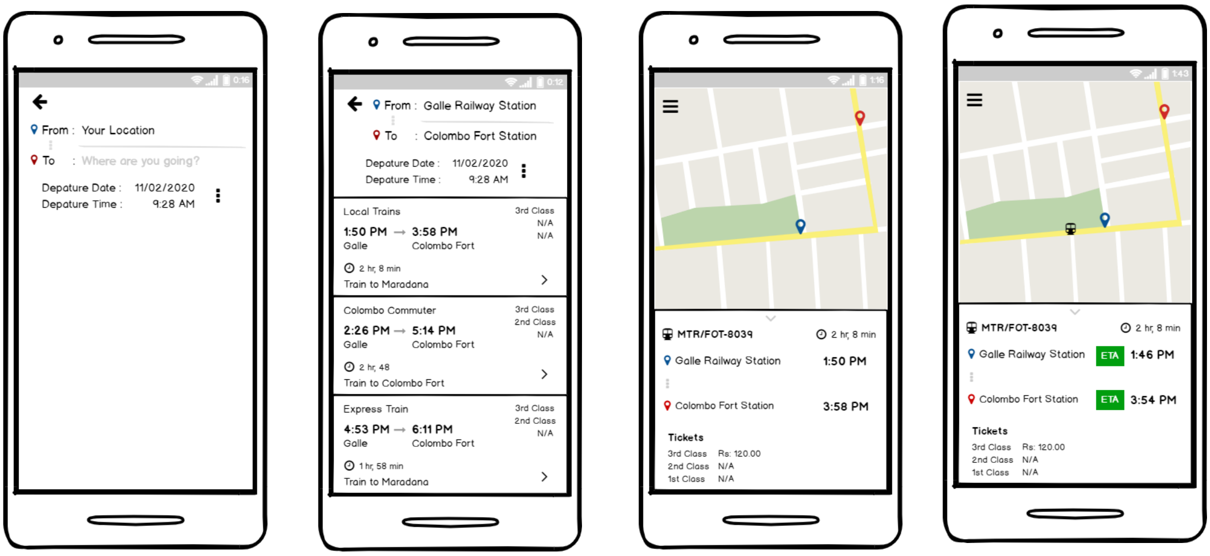


Figure 3. 17: Wireframes for Train Schedule

Other design diagrams and wireframes related to the system design are attached to the Appendix B.

3.4 Implementation

3.4.1 Implementation Environment

The Hardware Environment and Software Environment which were assisted in the development of the app are shown by Table 3. 1.

Hardware Environment	Software Environment
• Intel Core i5 2.20 GHz Processor • 8 GB RAM • 1 TB Hard Disk • An Android Phone • Dongle	• Windows 10 • Android Version 7.0 • Google Maps APIs • Firebase • Android Studio 3.6 • Vysor 3.0.84 • Firebase • Web Browser

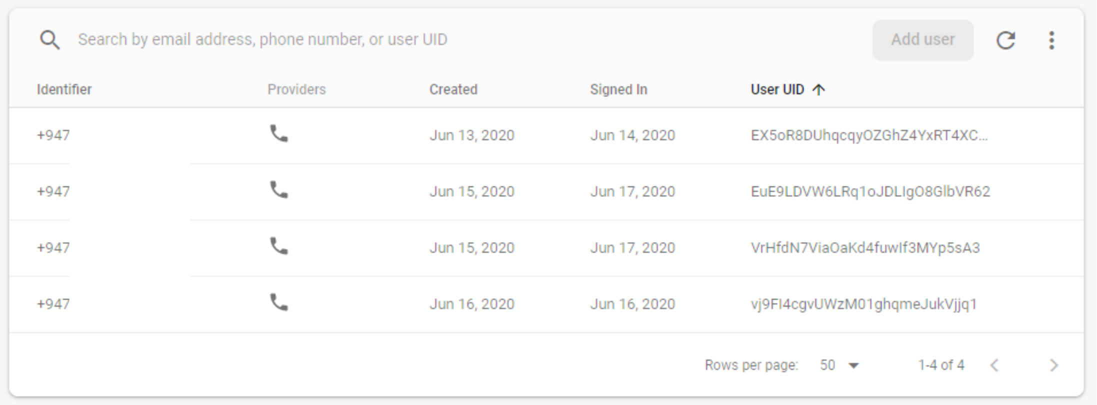
Table 3. 1: Implementation Environment

3.4.2 Development Tools and Techniques

When implementing the mobile application, the following tools and techniques were used.

Firebase

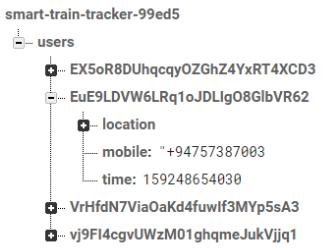
Firebase supports mobile applications as well as web applications as a backend solution with having much services such as Real-time Database and Authentication [11]. Firebase Authentication was used for user registration and login of the application and the phone authentication details in Firebase are depicted in Figure 3. 18.



Identifier	Providers	Created	Signed In	User UID ↑
+947	☎	Jun 13, 2020	Jun 14, 2020	EX5oR8DUhqcqyOZGhZ4YxRT4XC...
+947	☎	Jun 15, 2020	Jun 17, 2020	EuE9LDVW6LRq1oJDLIgO8GlbVR62
+947	☎	Jun 15, 2020	Jun 17, 2020	VrHfdN7ViaOaKd4fuwIf3MYp5sA3
+947	☎	Jun 16, 2020	Jun 16, 2020	vj9FI4cgvUWzM01ghqmeJukVjjq1

Figure 3. 18: Phone Authentication in Firebase

The Firebase Real-time Database is used as a NoSQL database which is hosted in cloud and the data in database is stored as JSON objects. The JSON object which is related to the application is shown by Figure 3. 19.

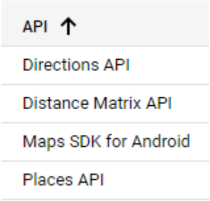


```
smart-train-tracker-99ed5
  users
    EX5oR8DUhqcqyOZGhZ4YxRT4XCD3
    EuE9LDVW6LRq1oJDLIgO8GlbVR62
      location
      mobile: "+94757387003
      time: 159248654030
    VrHfdN7ViaOaKd4fuwlf3MYp5sA3
    vj9Fl4cgvUWzM01ghqmeJukVjjq1
```

Figure 3. 19: JSON Tree Structure in Firebase Realtime Database

Google API Console

Some of the Google Maps APIs such as Places API, Directions API and Maps SDK for Android were enabled by Google API Console and the list of enabled APIs are shown in Figure 3. 20 [12].



API ↑
Directions API
Distance Matrix API
Maps SDK for Android
Places API

Figure 3. 20: List of Enabled APIs

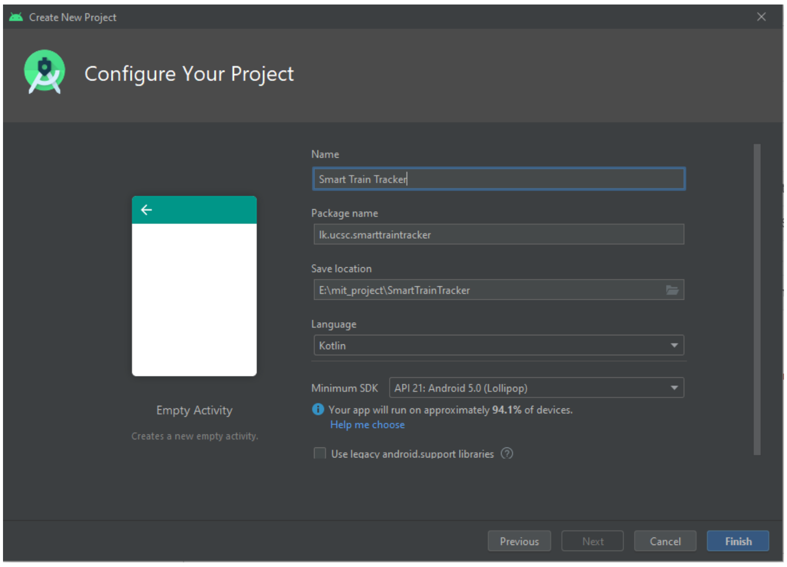
Lanka Gate API

Train details, Train schedule and station list are fetched from the Static Train Schedule API from the Lanka Gate API Store powered by ICTA of Sri Lanka [13].

Android Studio

Android Studio 3.6 was used as the IDE for the system development since it provides better coding experience by having GUI with drag and drop facility and code auto completion. Furthermore, Android studio utilizes the fast developing Gradle build framework [14].

Create New Project Interface with respect to the application is shown in Figure 3. 21.



Create New Project

Configure Your Project

Name: Smart Train Tracker

Package name: lk.ucsc.smarttraintracker

Save location: E:\mit_project\SmartTrainTracker

Language: Kotlin

Minimum SDK: API 21: Android 5.0 (Lollipop)

Your app will run on approximately **94.1%** of devices.

Help me choose

□ Use legacy android.support libraries

Empty Activity

Creates a new empty activity.

Figure 3. 21: Create New Project Interface in Android Studio

Balsamiq Wireframes
Balsamiq wireframes tool was used because it is the online available, fastest, easiest and most focused low-fidelity wireframing tool in the industry [15].

Kotlin
Kotlin is a concise programming language and is used for developing the application. It provides advantages such as reducing lines of codes and complexity of the code. Eventually it helps to improve compiler performance and build speed.

3.4.3 Implementation Architecture

The application was developed according to the MVP architecture. The MVP architecture comprises Model, View and Presenter and it is depicted by Figure 3. 22.

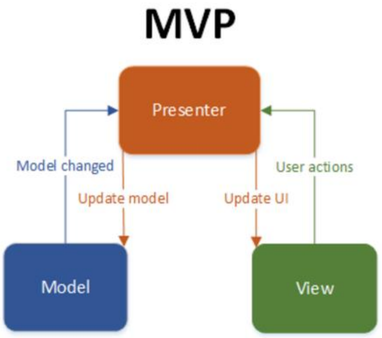


Figure 3. 22: MVP Architecture

Model is the data layer which communicates with databases and handles the business logic. The View uses for displaying the data and notifying user actions. Presenter acts as an intermediator which retrieves the information from the model and applies the UI rationale.

The following Figure 3. 23 illustrates the module structure of the application.

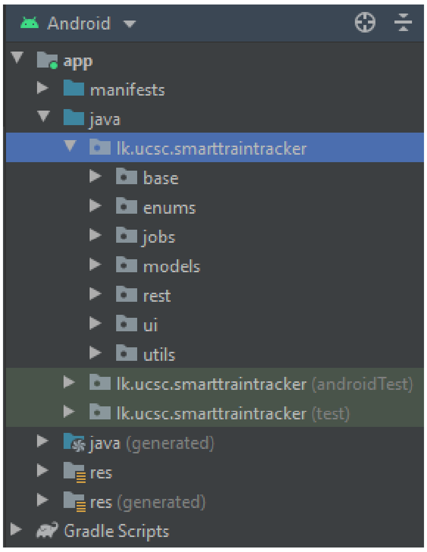


Figure 3. 23: Module Structure of the System

3.4.4 Major Code Segments

AndroidManifest.xml

All the configurations regarding the application such as permissions and handling hardware features are in the Android manifest file which is located in the “app/manifests/” folder and the code segment are displayed in Figure 3. 24.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="lk.ucsc.smarttraintracker">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>

    <uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

    <uses-feature
        android:name="android.hardware.location.gps"
        android:required="true" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:networkSecurityConfig="@xml/network_security_config">

        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyCpxdNek70i51hs7CjpcZljr6ZU1R5TaSw" />

        <activity
android:name="lk.ucsc.smarttraintracker.ui.activities.splash.SplashActivity
"
            android:screenOrientation="fullSensor"
android:theme="@style/App_NOActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <activity
android:name="lk.ucsc.smarttraintracker.ui.activities.login.LoginActivity"
            android:screenOrientation="fullSensor"
android:theme="@style/App_NOActionBar"/>
        <activity
android:name="lk.ucsc.smarttraintracker.ui.activities.main.MainActivity"/>
        <activity android:name=".ui.activities.register.RegisterActivity"
```

```
            android:screenOrientation="fullSensor"
android:theme="@style/App_NOActionBar"/>
        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />

        <service android:name=".jobs.LocationJobService"
            android:label="Train Location Service"
            android:permission="android.permission.BIND_JOB_SERVICE" />
    </application>

</manifest>
```

Figure 3. 24: Code Segment for AndroidManifest.xml

TrainSchedule.tk

The TrainSchedule.tk model can be used to fetch the train schedule from the Static Train Schedule API by ICTA Sri Lanka and the code segment is illustrated in Figure 3. 25.

```
package lk.ucsc.smarttraintracker.models

import io.reactivex.Observable
import io.reactivex.android.schedulers.AndroidSchedulers
import io.reactivex.schedulers.Schedulers
import lk.ucsc.smarttraintracker.rest.APIClient
import lk.ucsc.smarttraintracker.rest.responses.SearchResultsResponse

/**
 * get the train schedule details
 */
class TrainSchedule {
    var trainFrequncy : String? = null
    var finalStationName : String? = null
    var classList : ArrayList<TrainClass>? = null
    var trainID : Int? = null
    var trainNo : Int? = null
    var arrivalTime : String? = null
    var endStationName : String? = null
    var arrivalTimeEndStation : String? = null
    var depatureTime : String? = null
    var trainType : String? = null
    var trainName : String? = null
    var startStationName : String? = null
    var arrivalTimeFinalStation : String? = null

    companion object {
        fun requestScheduleByDateAndTime(startStationID :  Int,
endStationID : Int, searchDate : String, startTime : String, endTime :
String) : Observable<SearchResultsResponse> {
            return APIClient.getTrainScheduleByDateAndTime(startStationID,
endStationID, searchDate, startTime, endTime).subscribeOn(
                Schedulers.io()).observeOn(
                AndroidSchedulers.mainThread())
        }
    }
}
```

Figure 3. 25: Code Segment for TrainSchedule.tk

LocationJobService.kt

The following Figure3. 26 represents the LocationJobService.tk job is used to feed the location of the passenger’s smart phone to the Firebase in background within every thirty seconds interval.

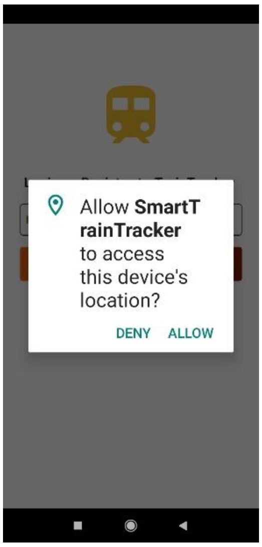
```
package lk.ucsc.smarttraintracker.jobs

import android.app.job.JobParameters
import android.app.job.JobService
import android.content.Intent
import android.location.Location
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.FirebaseDatabase
import io.reactivex.observers.DisposableObserver
import lk.ucsc.smarttraintracker.utils.JobSchedulerUtil
import lk.ucsc.smarttraintracker.utils.LocationUtil
import lk.ucsc.smarttraintracker.utils.LoggerUtil

/**
 * push the location data of the passenger's smart phone to firebase in
background every 30 secs interval
 */
class LocationJobService : JobService() {
    private var mLocationObserver : DisposableObserver<Location>? = null
    private var mAuth : FirebaseAuth = FirebaseAuth.getInstance()
    private var mFireBaseDatabase : FirebaseDatabase =
FirebaseDatabase.getInstance()

    override fun onStopJob(params: JobParameters?): Boolean {
        return true
    }

    override fun onStartJob(params: JobParameters?): Boolean {
        //LoggerUtil.logMessage("Job Scheduler Trigger")

LocationUtil.getCurrentLocation(applicationContext)!!.subscribeWith(getLoca
tionObservable())
        JobSchedulerUtil.scheduleJob(applicationContext!!)
        return true
    }

    private fun getLocationObservable() : DisposableObserver<Location>? {
        mLocationObserver = object : DisposableObserver<Location>() {
            override fun onComplete() {}

            override fun onNext(t: Location) {
                LoggerUtil.logMessage("Location Received")
                LocationUtil.saveCurrentLocation(mAuth, mFireBaseDatabase,
t)
            }

            override fun onError(e: Throwable) {}
        }
        return mLocationObserver
    }
}
```

Figure 3. 26: Code Segment for LocationJobService.tk

3.4.5 Main User Interfaces

Main user interfaces of the working application are shown in this subsection.

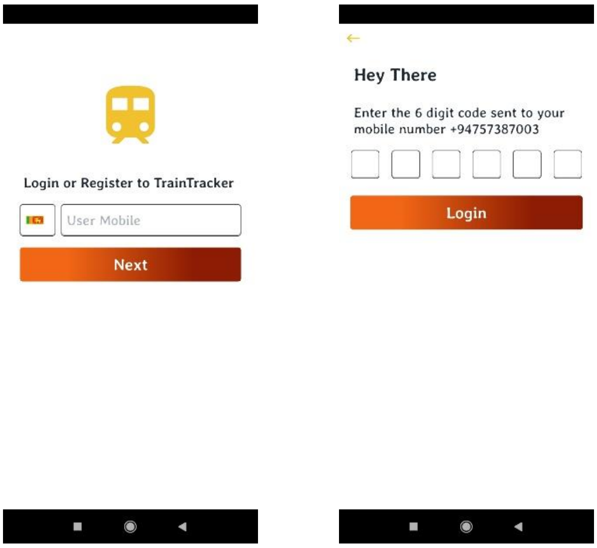
Location Permission
Location permission interface is shown in Figure 3. 27.



Allow SmartTrainTracker to access this device's location?

Figure 3. 27: Location Permission Interface

User Login
Figure 3. 28 displays interfaces of user login.



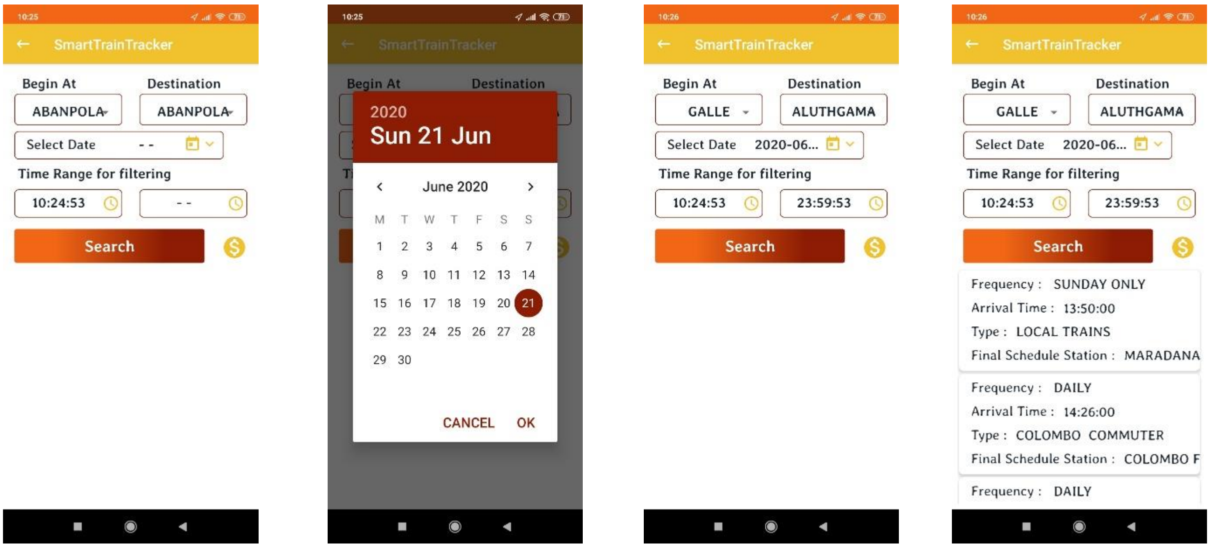
Login or Register to TrainTracker

Enter the 6 digit code sent to your mobile number +94757387003

Figure 3. 28: User Login Interfaces

Train Schedule

Interfaces for searching train schedule are illustrated by Figure 3. 29.



SmartTrainTracker

Begin At: ABANPOLA
Destination: ABANPOLA
Select Date: - -
Time Range for filtering: 10:24:53 – - -
Search

2020
Sun 21 Jun
June 2020
CANCEL OK

Begin At: GALLE
Destination: ALUTHGAMA
Select Date: 2020-06...
Time Range for filtering: 10:24:53 – 23:59:53
Search

Frequency : SUNDAY ONLY
Arrival Time : 13:50:00
Type : LOCAL TRAINS
Final Schedule Station : MARADANA

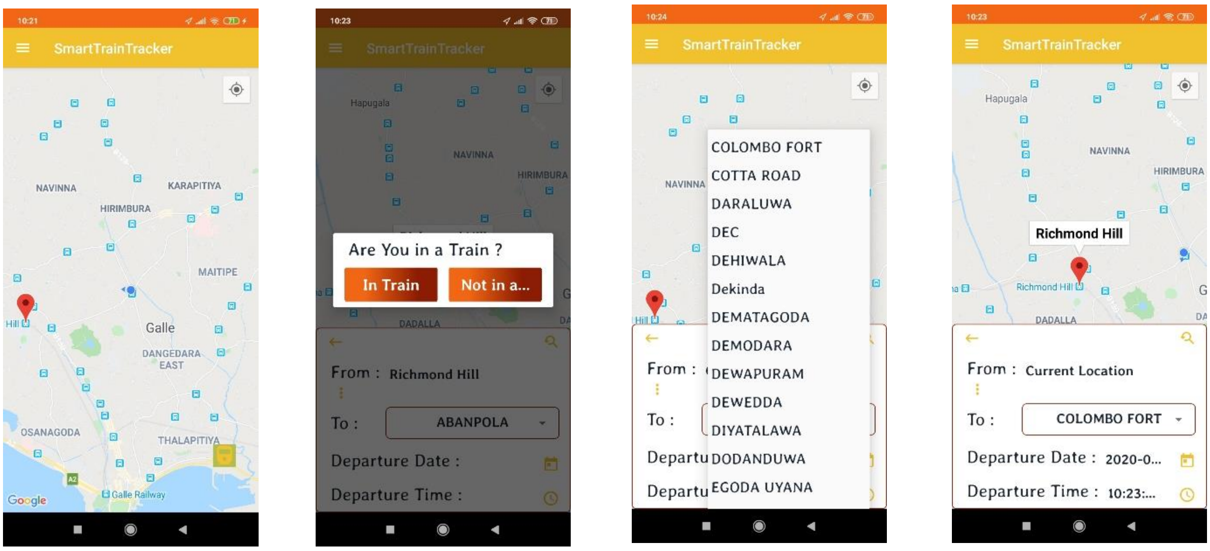
Frequency : DAILY
Arrival Time : 14:26:00
Type : COLOMBO COMMUTER
Final Schedule Station : COLOMBO F

Frequency : DAILY

Figure 3. 29: Train Schedule Interfaces

Location Tacking with Nearby Stations

Figure 3. 30 depicts the interfaces of location tracking with nearby stations option.



SmartTrainTracker

Are You in a Train ?
In Train
Not in a...

From : Richmond Hill
To : ABANPOLA
Departure Date :
Departure Time :

COLOMBO FORT
COTTA ROAD
DARALUWA
DEC
DEHIWALA
Dekinda
DEMATAGODA
DEMODARA
DEWAPURAM
DEWEDDA
DIYATALAWA
DODANDUWA
EGODA UYANA

From : Current Location
To : COLOMBO FORT
Departure Date : 2020-0...
Departure Time : 10:23:...

Figure 3. 30: Location Tracking with Nearby Stations Interface

3.4.6 Implementation Issues

The problems caused by the tools and techniques used in the system development are as follows.

- The Static Train Schedule API by ICTA Sri Lanka only provides fewer information about train schedule and railway stations. For example, Though, list of railway stations is provided, Geo locations of the railway stations are not provided by the respective API. Google Places API was used with the above API to overcome this issue.
- When mapping railway stations in Train Schedule API with the Google Map API in order to obtain location data of railway stations, string mapping is difficult since naming conventions for same station are different in both APIs. First three characters and last three characters are considered as a character mapping mechanism.
- When fetching train schedule data from the API it is necessary to give the destination station. As a solution, a spinner was used for selecting destination by the user.
- The Static Train Schedule API only allows HTTP requests and the functionality of Train Schedule API was limited since Android 9 and high supports only for HTTPS in nature. To overcome this contradiction, the network security configuration settings of the application are customized by adding network_security_config.xml file.

Chapter 4. Evaluation

4.1 Overview

The chapter mainly discusses about the testing of the suggested Smart Train Tracker app with the test cases and authentic test results which obtaining for same inputs compared to the expected results. Further, user evaluation process and the results are elaborated under this chapter.

4.2 Testing Procedure

Since following RUP process model, the suggested application was tested during the development. At that point unit testing and integration testing were done simultaneously followed by the system testing. Finally, the acceptance testing was done with the participation of five users after giving trail version of the mobile application in Galle Railway Station. The system test with user evaluation was carried out using a local train which was from Galle to Aluthgama and the tracking the location of the train from Galle to Hikkaduwa only was considered.

4.3 Test Cases and Test Results

Test cases were created for each major functionality and actual test results were compared with the expected results.

4.3.1 Test Cases

Test Cases for User Login with Phone Authentication
The test cases and expected test results of user login with phone authentication are displayed by Table 4. 1.

Test No	Test Description	Steps of Test	Expected Result
1	Validate User Input Details	Select correct country code Enter correct mobile no Enter correct OTP	Display main screen
2	Validate Phone No	Select correct country code Enter correct mobile no Enter incorrect OTP	Display error message “Registration Failure. Please Try Again”
3	Validate User Input Details	Select correct country code Enter invalid mobile no Enter correct OTP	Display error message “Phone Number Verification Failed”

4	Validate User Input Details	Select correct country code Enter invalid mobile no Enter incorrect OTP	Display error message “Phone Number Verification Failed”
5	Validate User Input Details	Select incorrect country code Enter correct mobile no Enter correct OTP	Display error message “Phone Number Verification Failed”
6	Validate User Input Details	Select incorrect country code Enter correct mobile no Enter incorrect OTP	Display error message “Phone Number Verification Failed”
7	Validate the User Session Data	Select incorrect country code Enter invalid mobile no Enter correct OTP	Display error message “Phone Number Verification Failed”

Table 4. 1:Test Cases for User Login with Phone Authentication

Test Cases for Phone Number Verification

The test cases and expected test results of phone number verification are shown in Table 4. 2.

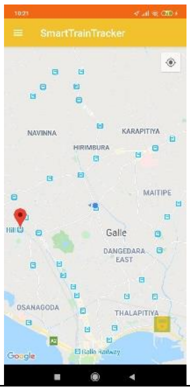
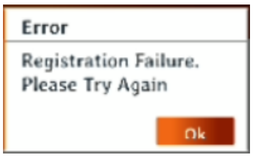
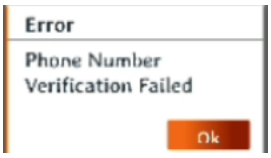
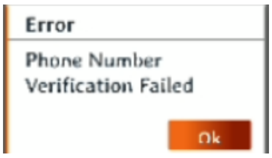
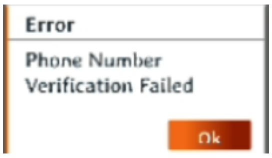
Test No	Test Description	Steps of Test	Expected Result
1	Validate Phone No	Enter correct mobile no	Get the OTP and
2	Validate Phone No	Enter correct mobile no start with 0	Display error message “Registration Failure. Please Try User Mobile Cannot Start with a zero”
3	Validate Phone No	Enter less mobile no	Display error message “Phone Number Verification Failed”
4	Validate Phone No	Enter long mobile no	Display error message “Phone Number Verification Failed”
5	Validate Phone No	Enter invalid mobile no	Display error message “Phone Number Verification Failed”

Table 4. 2: Test Cases for Phone Number Verification

4.3.2 Test Results

Test Results for User Login with Phone Authentication

The test cases and actual test results of user login with phone authentication are displayed in Table 4. 3.

Test No	Steps of Test	Expected Result	Status
1	Select correct country code Enter correct mobile no Enter correct OTP	Display main screen  SmartTrainTracker NAVINNA KARAPITIYA HIRIMBURA MAITIPE Galle DANGEDARA EAST OSANAGODA THALAPITIYA Google	OK
2	Select correct country code Enter correct mobile no Enter incorrect OTP	Display error message “Registration Failure. Please Try Again”  Error Registration Failure. Please Try Again Ok	OK
3	Select correct country code Enter invalid mobile no Enter correct OTP	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed Ok	OK
4	Select correct country code Enter invalid mobile no Enter incorrect OTP	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed Ok	OK
5	Select incorrect country code Enter correct mobile no Enter correct OTP	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed Ok	OK

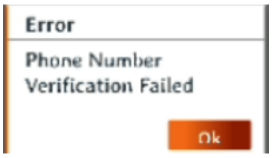
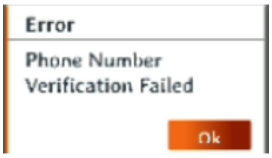
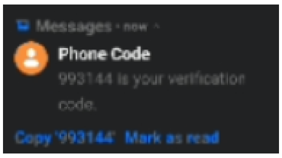
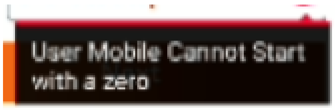
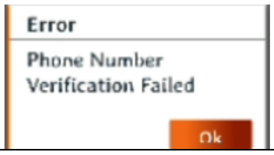
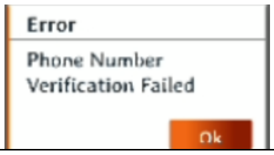
6	Select incorrect country code Enter correct mobile no Enter incorrect OTP	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed 	OK
7	Select incorrect country code Enter invalid mobile no Enter correct OTP	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed 	OK

Table 4. 3: Test Results for Phone User Login with Phone Authentication

Test Results for Phone Number Verification

The test cases and actual test results of phone number verification are shown in Table 4. 4.

Test No	Steps of Test	Expected Result	Status
1	Enter correct mobile no	Get the OTP  Messages · now Phone Code 993144 is your verification code. 	OK
2	Enter correct mobile no start with 0	Display error message “Registration Failure. Please Try User Mobile Cannot Start with a zero”  User Mobile Cannot Start with a zero	OK
3	Enter less mobile no	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed 	OK
4	Enter long mobile no	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed 	OK

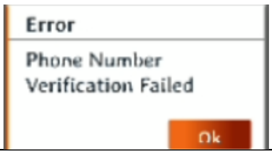
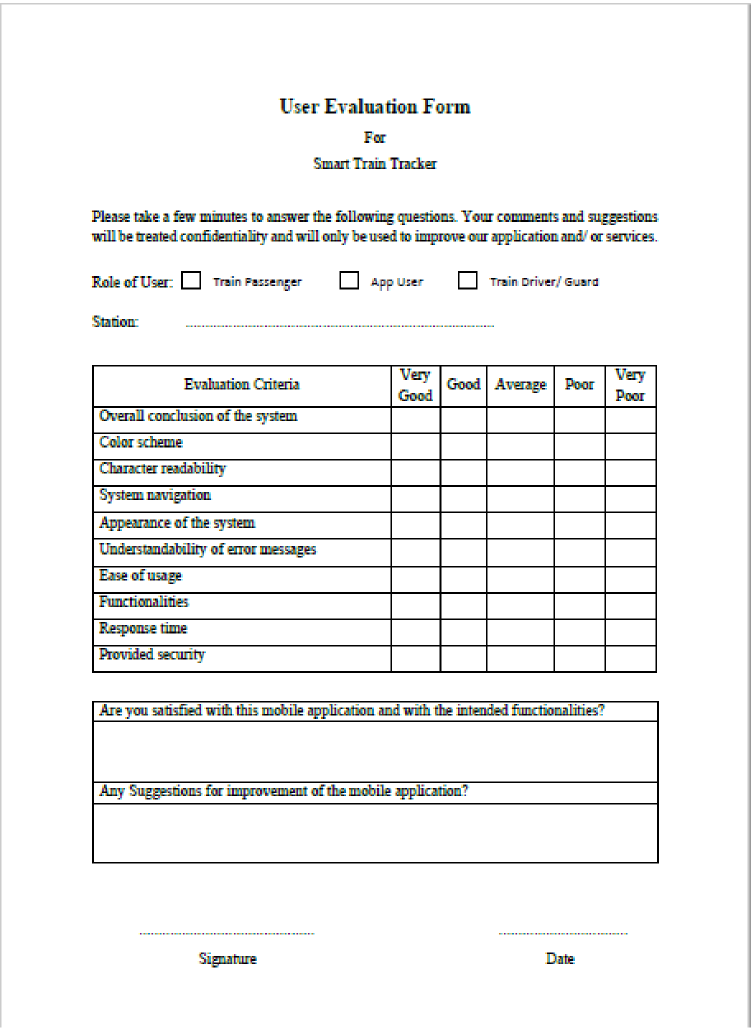
5	Enter invalid mobile no	Display error message “Phone Number Verification Failed”  Error Phone Number Verification Failed Ok	OK

Table 4. 4: Test Results for Phone Number Verification

4.4 User Evaluation

The user evaluation was conducted with the participation of five users in Galle Railway Station. Among them, three users were train passengers and other two were normal app users. Initially the trial version of the mobile application was given for the authentic users and the user feedback was captured by providing a user evaluation questionnaire. The given user evaluation questionnaire was consisted of ten criteria and user’s overall conclusion of the system was shown in first criteria.

The user evaluation form is shown in Figure 4. 1.



User Evaluation Form

For

Smart Train Tracker

Please take a few minutes to answer the following questions. Your comments and suggestions will be treated confidentiality and will only be used to improve our application and/ or services.

Role of User: □ Train Passenger □ App User □ Train Driver/ Guard

Station: ....

Evaluation Criteria	Very Good	Good	Average	Poor	Very Poor
Overall conclusion of the system					
Color scheme					
Character readability					
System navigation					
Appearance of the system					
Understandability of error messages					
Ease of usage					
Functionalities					
Response time					
Provided security					

Are you satisfied with this mobile application and with the intended functionalities?
Any Suggestions for improvement of the mobile application?

....

Signature Date

Figure 4. 1: User Evaluation Form

The collected information from the five users was analyzed and the results are depicted in and Figure 4. 2.

	Evaluation Criteria	Very Good	Good	Average	Poor	Very Poor
1.	Overall conclusion of the system	2	3	-	-	-
2.	Color scheme	3	2	-	-	-
3.	Character readability	1	2	2	-	-
4.	System navigation	1	3	1	-	-
5.	Appearance of the system	3	2	-	-	-
6.	Understandability of error messages	-	2	2	1	-
7.	Ease of usage	1	2	2	-	-
8.	Functionalities	1	3	1	-	-
9.	Response time	1	2	2	-	-
10.	Provided security	1	2	2	-	-

Table 4. 5: Summary of User Evaluation

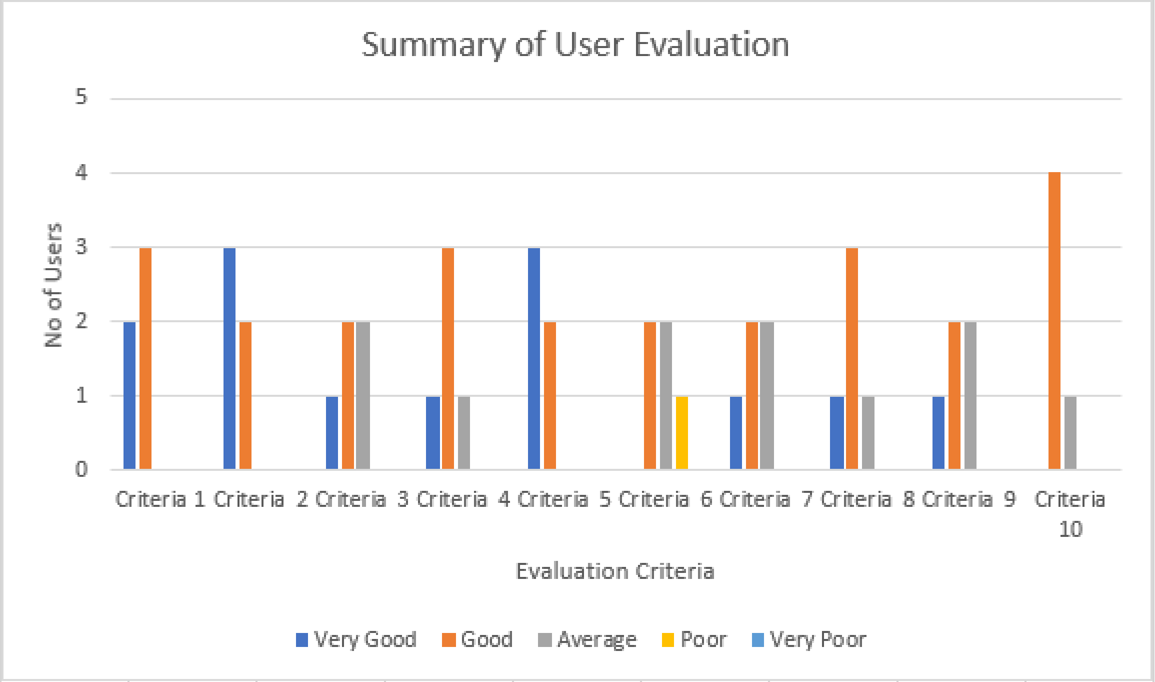


Figure 4. 2: Summary of User Evaluation

According to the summarized data of the user evaluation, the majority of the sample users were satisfied with the suggested mobile application and the intended functionalities. Though the user evaluation results were significantly positive, a few minor modifications of the mobile application were accepted as the future enhancement.

Chapter 5. Conclusion

5.1 Overview

The chapter describes on the critical evaluation of the project, problems faced, lesson learnt and future enhancements of the Smart Train Tracker mobile application.

5.2 Critical Evaluation of the Project

According to the system analysis of the SLR's existing system, it was observed that there was a need for a system to receive real time updates on the location of trains. As a solution, the Smart Train Tracker has been developed as an Android application incorporating the desired functionalities of tracking the location of real-time trains.

A range of services regarding the information of public railway transport service in Sri Lanka is offered for both passengers and the general public as well as members of SLR by this mobile application. The train schedule, location of nearby stations, location of the real-time trains and ETA of the specified trains can be considered as the key functionalities which are provided by the Smart Train Tracker. Since the app was developed as a crowd sourced application, the location of the passenger's smart phone which are in the train is captured in order to track the location of a particular train.

The unit testing as well as the integrated testing was done during the application development and then the system testing was done through the user evaluation. The user evaluation was carried the participation of five number of users by tracking the location of the specific train from Galle to Hikkaduwa only was considered. Then the user evaluation forms were distributed and feedback was obtained.

Because of the services that provided by the application, the train passenger's valuable time would not waste and the services of the SLR can be maintained in a splendid manner. Furthermore, the zero-capital investment cost can be considered as a key benefit of the app, as the specific hardware implementation is not used in this application. Ultimately, the availability and efficiency of the railway services would be improved by using the Smart Train Tracker.

According to the test results, summarized feedback on user evaluation and benefits which are provided by smart train tracker, it can be satisfied on the app with intended functionalities.

5.3 Problems Encountered

Since the lacking of practical knowledge with regard to development tools and techniques of mobile application development, it was more time consuming at the initial stage of development. Therefore, online tutorials and materials were followed for gaining essential knowledge. Another major problem which was encountered was the operating system of the developing machine was crashed and needed to re-install during the mobile application development period. Hence it consumed spare time to back up all the required software for maintaining the development environment as previous. Although it was difficult to test the app in authentic environment since the Covid-19 pandemic situation, the testing was conducted as planned before.

5.4 Lesson Learnt

Developing the Smart Train Tracker paved the way to experiment the knowledge which I gained through the MIT degree program and vast practical knowledge on developing this application including Kotlin, JSON encoding and knowledge of API handing was gained by doing this project. Time management is the most important lesson which was learnt from this project. Further, this valuable lesson will help me to do my forthcoming projects better as well as it will help me to tame success lot of things in my future life. Another valuable lesson which was learnt from this project is having a confidence for doing anything until the completion of it across any obstacle. Finally, developing the project is encouraged me to uplift the technical skills, intellectual skills and interpersonal skills.

5.5 Future Enhancements

Some suggestions for the future enhancements of the system are as follows:

- A leader board design can be added in order to improve and inspire the passengers' involvement for keeping alive this app through the journey
- Location tracking for family members or any person in their authorized circle can be embedded apart from train tracking functionality
- The proposed system can be utilized only by Android users and limits the target audience. Hence the system can be developed as a cross platform application.
- SMS facility can be provided to obtain demanded information which is related to train schedule and ETA from the system
- A fully functional admin dashboard can be developed by providing more facilities for the administrative officers of Sri Lanka Railway

References

[1] "Sri Lanka Railways Official Website," Sri Lanka Railways, 2011. [Online]. Available: http://www.railway.gov.lk/web/. [Accessed 25 06 2019].

[2] Sri Lanka Railways, "Summary of Operating & Financial Statistics Year 2017," 2017. [Online]. Available: http://www.railway.gov.lk/web/images/pdf/statistical_book%20_2017.pdf. [Accessed 25 06 2019].

[3] "Train Schedule," Sri Lanka Railways, 2011. [Online]. Available: https://eservices.railway.gov.lk/schedule/searchTrain.action?lang=en. [Accessed 25 06 2019].

[4] Dileepa Jayakody, Mananu Gunawardana, Nipuna Wicrama Surendra, Dayan Gayasri, Chanaka Upendra and Ranga De Silva, "GPS/ GSM Based Train Tracking System," 2010. [Online]. Available: https://pdfs.semanticscholar.org/349a/76333d57acdd9cf0c431f54353915f84b9d7.pdf. [Accessed 27 10 2019].

[5] A.P. Kasthoori, E.M.W.V. Ekanayake and Kasun De Soyza, "Railway Train Dynamics Analysis System," in *International Conference on Advances in ICT for Emerging*, Colombo, Sri Lanka, 2014.

[6] Google Developers, "The Transport Tracker Solution," 2015. [Online]. Available: https://developers.google.com/maps/solutions/transport-tracker/. [Accessed 09 12 2019].

[7] DQINDIA ONLINE, "Crowdsourced mobile solution enables real time tracking of movement of train on Google Maps," 2015. [Online]. Available: https://www.dqindia.com/crowdsourced-mobile-solution-enables-real-time-tracking-ofmovement-of-trains-on-google-maps/. [Accessed 09 12 2019].

[8] Merriam-Webster, "Crowdsourcing," 2019. [Online]. Available: https://www.merriam-webster.com/dictionary/crowdsourcing. [Accessed 16 12 2019].

[9] WIKIBOOKS, "RUP - IBM Rational Unified Process," 2019. [Online]. Available: https://en.wikibooks.org/wiki/RUP_-_IBM_Rational_Unified_Process/Phases. [Accessed 26 01 2020].

[10] WIKIPEDIA, "Rational Unified Process," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Rational_Unified_Process. [Accessed 09 02 2020].

[11] Firebase, "Firebase," 2020. [Online]. Available: https://firebase.google.com/. [Accessed 08 03 2020].

[12] Google Developers, "Google Developers," 2020. [Online]. Available: https://developers.google.com/. [Accessed 08 03 2020].

[13] Lanka Gate API Store, "Static Train Schedule API," 2020. [Online]. Available: https://api.lankagate.gov.lk:9443/store. [Accessed 21 05 2020].

[14] Android developers, "developers," 2020. [Online]. Available: https://developer.android.com/. [Accessed 07 03 2020].

[15] Balsamiq, "Balsamiq Wireframes," 2020. [Online]. Available: https://balsamiq.com/wireframes/. [Accessed 21 02 2020].

[16] Ministry of Transport and Civil Aviation , "Progress Report 2018," 2018. [Online]. Available: http://www.transport.gov.lk/web/images/pdf/progress_english_2018.pdf. [Accessed 30 06 2019].

[17] Dhruv Pandya, "A Hierarchical Location Tracking System," 2002. [Online]. Available: https://www.imperial.ac.uk/pls/portallive/docs/1/18619761.PDF. [Accessed 13 10 2019].

[18] Infsoft, "Railway Train Tracking System," 2019. [Online]. Available: https://www.infsoft.com/use-cases/railway-train-tracking-system. [Accessed 15 01 2020].

[19] N. K. Das, C. K. Das, Rajesh Mozumder and Jiban Chandra Bhowmik, "Satellite Based Train Monitoring System," *Journal of Electrical Engineering,* Vols. vol. EE 36, No. II, 2009.

[20] R. Immanuel Rajkumar, P. E. Sankaranarayanan and G. Sundari, "GPS and Ethernet based Real Time Train Tracking System," in *International Conference on Advanced Electronic Systems*, 2013 Sep.

[21] VisualParadigmOnline, "VisualParadigmOnline," 2020. [Online]. Available: https://online.visual-paradigm.com/. [Accessed 21 02 2020].

[22] usability.gov, "User Interface Design Basics," 2020. [Online]. Available: https://www.usability.gov/what-and-why/user-interface-design.html. [Accessed 15 02 2020].

[23] Tutorialspoint, "UML Class Diagram," 2020. [Online]. Available: http://www.tutorialspoint.com/uml/uml_class_diagram.htm. [Accessed 09 02 2020].

[24] draw.io, "draw.io," 2020. [Online]. Available: https://app.diagrams.net/. [Accessed 22 02 2020].

[25] web.cse.ohio-state.edu, "Addressing Non-Functional Requirements in Mobile Applications," 2020. [Online]. Available: http://web.cse.ohio-state.edu/~champion.17/5236/Lecture8_NFRTesting.pdf. [Accessed 23 02 2020].

[26] Rewire Security, "What is a GPS Tracker?," 2019. [Online]. Available: https://www.rewiresecurity.co.uk/blog/what-is-a-gps-tracker. [Accessed 15 01 2020].

[27] Android developers, "Android’s Kotlin-first approach," 2020. [Online]. Available: https://developer.android.com/kotlin/first. [Accessed 08 03 2020].

[28] AndroidPub, "Android MVP for Beginners API," 2020. [Online]. Available: https://android.jlelse.eu/android-mvp-for-beginners-25889c500443. [Accessed 21 05 2020].

[29] upday devs, "Android Architecture Patterns Part 2: Model-View-Presenter," 2020. [Online]. Available: https://medium.com/upday-devs/android-architecture-patterns-part-2-model-view-presenter-8a6faaae14a5. [Accessed 21 05 2020].

[30] stackoverflow, "How to allow all network connection types HTTP and HTTPS in Android (9) Pie?," 2020. [Online]. Available: https://stackoverflow.com/questions/51902629/how-toallow-all-network-connection-types-http-and-https-in-android-9-pie. [Accessed 13 06 2020].

[31] stackoverflow, "Androoid Google Map how to check if the gps location is inside the circle," 2020. [Online]. Available: https://stackoverflow.com/questions/30170271/androidgoogle-map-how-to-check-if-the-gps-location-is-inside-the-circle. [Accessed 23 05 2020].

[32] Android developers, "Network Security Configuration," 2020. [Online]. Available: https://developer.android.com/training/articles/security-config. [Accessed 13 06 2020].

Appendix A. System Documentation

The system documentation will be guided for the installation process of the mobile application by using apk file. Further, the phone requirements which needs to be installed are explained.

Step 1 - Phone requirements must be satisfied according to Table A. 1.

Phone Requirements

Phone Requirements	Minimum Requirements
• Android Version	Android Version 5.0 (Lollipop)or high
• Minimum SDK Version	API 21
• GPS	enable
• Internet connectivity	Cellular/ Wi-Fi

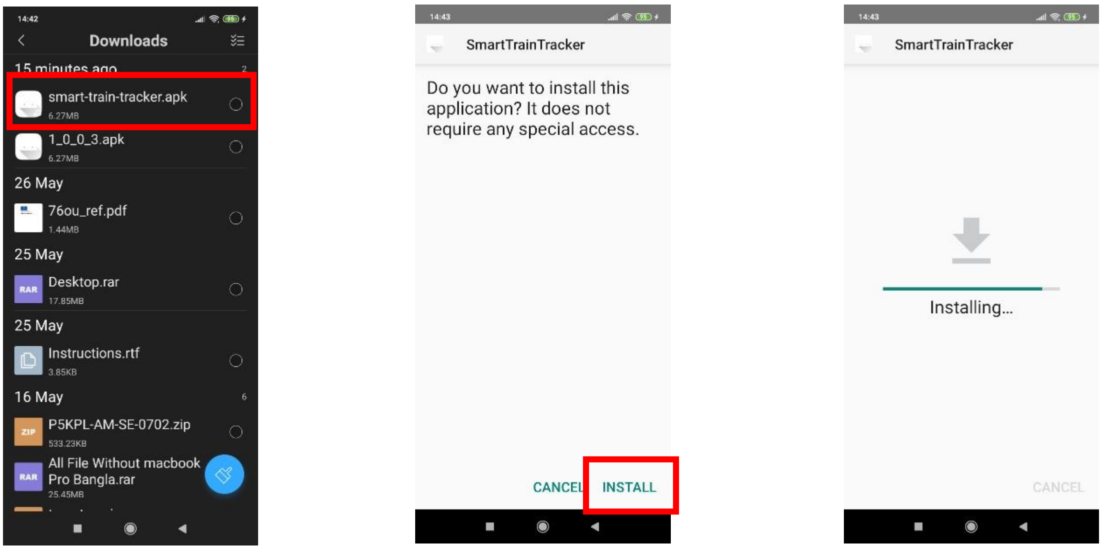
Table A. 1: Phone Requirements for Installation

Step 2 - Installing Smart Train Tracker

Installation Process

1. Copy the provided apk to your phone
2. Go to the location of your phone which the apk file copied
3. Click on the file
4. Click on INSTALL
5. Open the app after installation

Installation of the Smart Train Tracker app is illustrated in Figure A. 1.



Do you want to install this application? It does not require any special access.

Figure A. 1: App Installation

Appendix B. Design Documentation

- **User Registration/ Login Module**

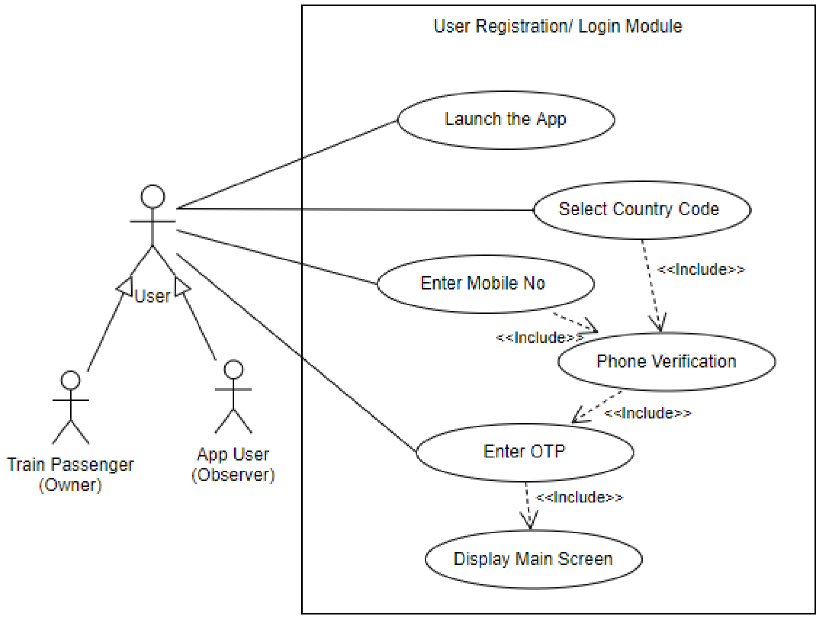


Figure B. 1: User Login Use Case

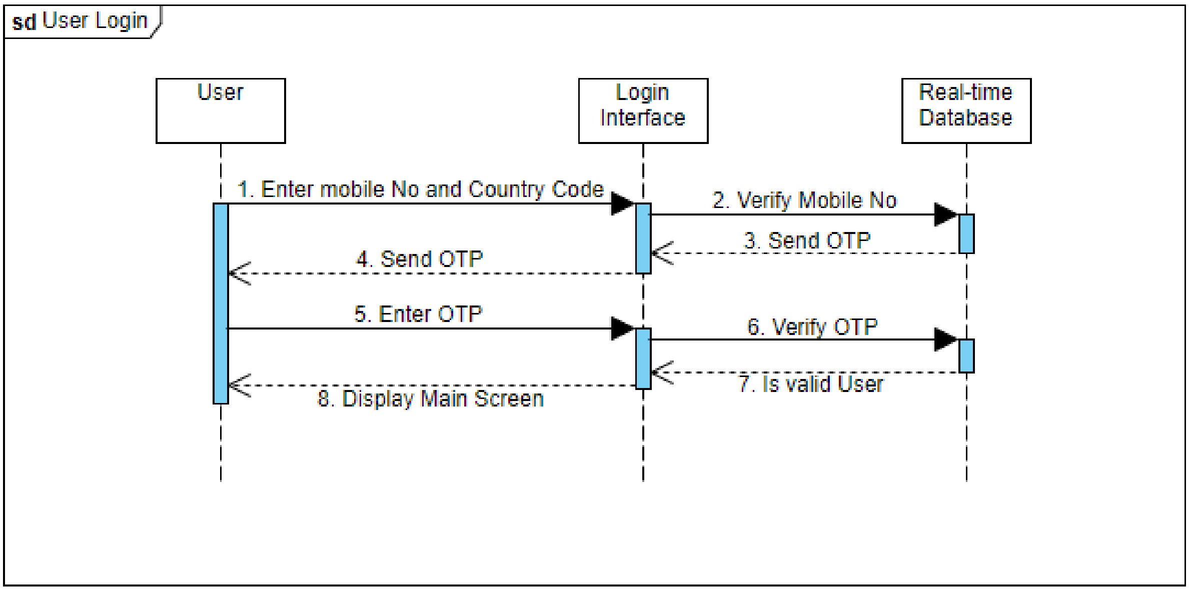


Figure B. 2: User Login Sequence Diagram

- **Train Schedule Module**

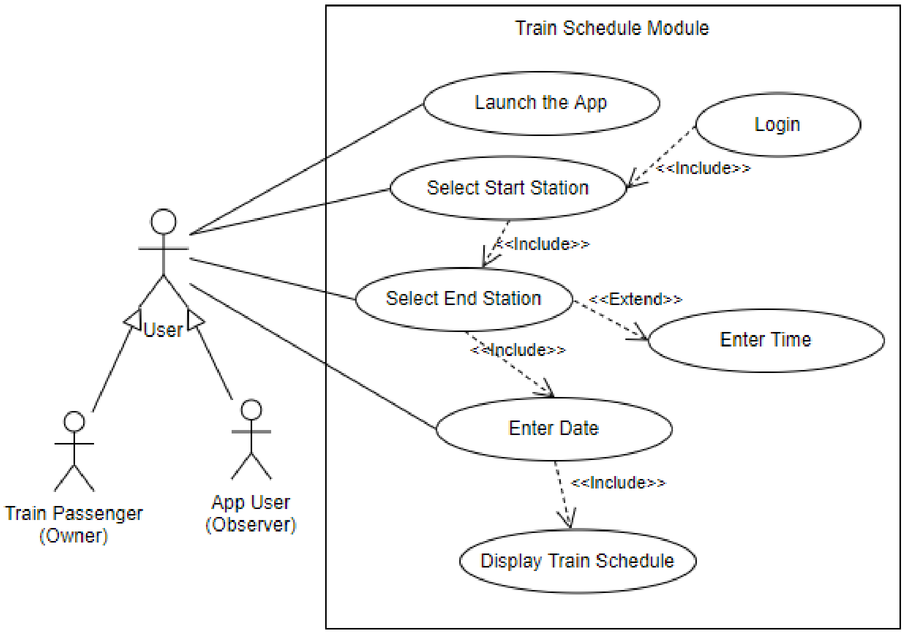


Figure B. 3: Train Schedule Use Case

- **Location Tracking Module**

Use Case Name	Location Tracking
Actors	Train passengers those who are in the train
Overview	The train passenger is tracked by the system
Pre-Conditions	Passenger should be logged in into the app and granted location permission
Normal Course of Events	1. Click on nearest station 2. Select “In a Train” 3. Select the train
Alternative Course of Events	App displays the error message
Post-Conditions	The location data is detected and transferred to the real-time database
Conclusion	Other users can see the train on the map

Table B. 1: Location Tracking Use Case Narrative