GPS-Android based Location Tracking System for Public Transportation (TrackIT)

R.R.Vithanage
149233B

Faculty of Information Technology
University of Moratuwa
May 2017

GPS-Android based Location Tracking System for Public Transportation (TrackIT)

R.R.Vithanage
149233B

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of the requirements of the Master of Science in Information Technology.

May 2017

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

------------------------------ ------------------------------
Name of the Student Signature of the Student

Date:

Supervised by

----------------------------- -----------------------------
Name of the Supervisor Signature of the Supervisor

Date:

Dedication

This thesis is dedicated to my parents, Mr. S.Vithanage and Mrs. G.Thambawita for their endless love, encouragement and support.

Acknowledgements

First and foremost I would like to offer my sincere gratitude to my research supervisor, senior lecturer Mr. B. H. Sudantha, for his guidance, supervision, encouragement and support throughout this study.

I would not know what research is and how to do research if the lecture series of thesis writing and research methodologies is not offered. So I would like to offer my sincere gratitude to the Prof. Asoka S. Karunananda for feeding the knowledge and guidance for doing researches in proper way.

I would also like to thank my lecturer S. Premaratne and all the lecturers of Faculty of Information Technology – University of Moratuwa, for their guidance and encouragement to get maximum use of knowledge and capabilities. Also I would like to mention the encouragement given by Mr. J.I. Pitiyage during the research.

I am grateful to the management and colleagues at Four Corners Lanka (Pvt) Ltd for their kind support and understanding during this work.

Finally I would like to extend my deepest gratitude to my parents and family, for their continuous support given in every possible way to make this project success.

Abstract

Public transportation has been considered as the main mode of transportation in Sri Lanka. General public can view static time table of public transportation but properly aligning the transportation with the correct time table is not happening due to various reasons such as traffic conditions, bad road condition and some technical or human resource unavailability. From the passenger's perspective, they do not have a method to find the actual location or availability of the bus or train causing to waste of time waiting on bus shelters and railway stations. Therefore some method for location tracking for transportation system is important to enable passengers to reduce their time wastage.

Many researches have been conducted on location tracking and monitoring in the world. With the spreading of information technology, over the past few years there have been some attempts to enhance the quality of public transportation with the application of information and communication technology. However still there is no proper mechanism to find real time location of a bus or train although there are some commercial tracking systems to track private vehicles by installing some costly hardware devices.

This research focused on utilizing commonly used technologies and infrastructure to implement the location tracking system for public transportation sector. The system consisting of a web application and a mobile application to enable location tracking. The web application is the core of the system which provides the interface for the general public to view the schedules and actual location of the bus or train. In addition to the location information, it predicts time of arrival/destination through analyzing previous data and current live data. The administration panel of the main web application enable management of all master data of the system.

With this system, it is expected to facilitate passengers to make better travelling decisions by providing required information for them. Also relevant authorities also can use system data to support decisions to enhance railway service and bus services. During the evaluation, it was shown that system was successful more than 70%.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Prolegomena

This chapter presents the background and motivation of the research, hypothesis, objectives, problem statement, our location tracking system (TrackIT) approach and the structure of the rest of the thesis. Here we describe some of key problematic areas in the current public transportation system in Sri Lanka. Especially in passengers/general public perspective problems will be identified and proposed solution to address the problem is briefly described.

1.2 Background and Motivation

Public transportation has been considered as the main mode of transportation in Sri Lanka. Therefore it is important to further develop and enhance public transportation as the main method of transportation. With the spreading of information technology, over the past few years there have been some attempts to enhance the quality of public transportation with the application of information and communication technology. However, unfortunately some major concerns of passengers still remain unsolved. From the passenger’s perspective, they do not have a method to find the actual location of the bus or train[1]. The only available information is static time table. Due to the road conditions and various other factors such as unavailability of drivers, conductors or technical fault of bus or train, the scheduled time tables may not reflect actual time or availability of the bus or train in a particular route. Then the people have to wait unexpectedly to get a train or bus. Some cases people who travel long distances to work, become unable to arrive the work place due to delays or unavailability of public transportation. Most people may have experienced such situations when using public transport system. For some rural areas, only one bus gives the service. People wait in passenger shelters for the bus. If the bus is not available, people get to know it after passing the time. Then they have not enough time to find an alternate method to reach their destination. If we can address this issue in a simple and cost effective way, it would be a great benefit for those people. It would leads to save many man hours and thus it helps to economic development of the country.

If the people can get to know the actual location or availability of the bus or train, they can decide time to which they should go to the bus-stop or railway station. In the absence of bus or train they can decide alternative transportation methods without waiting for last moment. Therefore implementing some location tracking system of public transportation which can be accessed by general public is very important. In addition to the location of a particular bus or train, if a system can give a predicted arrival/destination time based on various practical scenarios is more beneficial to the passengers.

Location tracking has been used in many areas such as defense, national security, vehicle tracking, and people tracking and so on. Recently there is a huge trend towards applications of Internet of Things (IoT)[2] and location based tracking systems. Fleet management/Tracking, commercial vehicle tracking, and tracking of persons are some of interesting areas of location based tracking systems. In this research, we present our work on location tracking solution for public transportation sector using Global Positioning System (GPS), Mobile network, Android and open source web technologies.

1.3 Problem statement

During our study, we found that, currently there is no proper system to locate public transportation such as buses and trains. Existing solutions provide only static schedule information. Many researchers have work on location tracking systems and most of them given solutions those involve different hardware modules those have issues in installation and costs. Some solutions are difficult to implement practically since they require to equip bus-stops with some microcontroller devices. For Sri Lanka, it is necessary to implement an affordable solution which should be beneficial for the general public who use the public transportation. Therefore utilizing commonly used equipment's, technologies and infrastructure for a location tracking system is a research gap which should be addressed. Accordingly utilizing mobile phone, mobile network and open source web technology to implement a solution for real time location tracking would address the problem. As an addition feature, extending the solution for tracking private vehicle, field officers of a particular business network would be possible.

1.4 Hypothesis

We hypothesis that simple and cost effective location tracking system which would address problems in passengers` perspective, can be developed using Android smartphone, open source web technologies and utilizing mobile networks.

1.5 Objectives

(i) To study the current issues related to location monitoring in public transportation

(ii) To Critically review the available methods for location tracking

(iii) To do an in depth study of location tracking technologies

(iv) To develop a new system for location tracking

(v) To provide a solution to predict time for arrival/destination

(vi) To allow third parties to consume location tracking data

(vii) To evaluate the performance of the system

1.6 Location Tracking - GPS Android based approach (TrackIT)

The proposed system would be consist of a web application and a mobile application. The web application is the core of the system. It is the central administrating panel and the central processing unit. Also it provide the interface for the general public. The web application would be developed in PHP with a use of a CMS (Content Management System). The android mobile application is used to get geo location updates. This application will be given to the authorized users for security reasons.

Main input to the system is location information posted periodically by the android application. Also the master data and configurations can be fed into to the system via the admin panel of the main web application.

The location update data sent by the mobile application, will be validated and saved to the database. Form each device, particular route, vehicle, speed can be identified and those details will be used for next steps. Then the location data will be retrieved to plot on a graph to show real time location information. This graph with location information could be freely accessed by the public through web. The saved location data will be used to analyze patterns and those will be used to form algorithms to predict arrival/destination time. Since the web application is fully responsive, it can be easily accessed by any mobile device without a separate mobile application.

1.7 Structure of the Thesis

The rest of the thesis is organized as follows. Next chapter (Chapter 2) critically reviews the literature on location tracking solutions and identify the research problems. Chapter 3 is about the technology adapted in this new system. The overall architecture of the system, and the reasons for selection of technologies would be described in this chapter. In chapter 4 our new approach for location tracking system for public transportation would be described. Inputs, processes, output and desired user groups also would be described here. In chapter 5, system design is described for the new location tracking system. The overall design, use case diagrams, sequence diagrams and database structure diagrams are included and described in this chapter.

Also the modules available in the system and the interaction among them is also described in detail here.

Chapter 6 would describe the implementation details each module which was described in chapter 5. Here the details about software, hardware used for the system is mentioned, and details of critical algorithms used in system are described. Chapter 7 would describe the evaluation methods used for this system in detail. The design of the questionnaire, selection of users and analyzing of final results would be described in this chapter. Chapter 8 is about the overall achievements of our new system TrackIT, problems that had to be addressed while developing the system, and limitations of TrackIT and how it could be enhanced further in the future.

1.8 Summary

This chapter gave an overall picture of the entire project presented in this thesis. As such we described the background/motivation, problem definition, hypothesis, objectives, and a brief overview of the solution. Next chapter presents a critical review of literature on location tracking solutions.

Chapter 2

Developments and Challenges in Location Tracking Systems

2.1 Introduction

Chapter 1 gave a comprehensive description of the overall project described in this thesis. This chapter provides a critical review of the literature in relation to developments and challenges in location tracking for public transportation.

For this purpose, the review of the past researches have been presented under three major sections, namely, early developments, recent developments and currently available systems.

At the end, this chapter defines the research problem as the unavailability of a location tracking system for public transport system, and identifies that GPS/GSM , Android and open source web technology can be integrated to addressed the problem.

2.2 Early Developments

Radio frequency identification (RFID) technology is also used for location tracking although it was originally invented for military uses. From 1980s, commercial RFID products started to be available and they were mainly applied in areas of supply chains, transport, manufacturing, personnel access, animal tagging, etc. This technology mostly used for local area or indoor location tracking purposes[3]. A research RFID adoption for vehicle tracking in a container terminal has been done by S.L.Ting[4]. This solution presents the utilization of RFID for getting statistical data in addition to location information. However this solution require more experiments on different RFID equipment is necessary in order to sort out the best.

Another research named Accurate Vehicle Location System Using RFID has been done by Y.Ning and his coworkers[5]. It uses both GPS (Global Positioning System) and RFID to accurately locate the vehicles. Since two methods are used it is more accurate. However it requires a complex set up since it needs RFID transmitters and receivers stationed in various places.

Global Positioning System (GPS) networks have become a part of everyday life. When we consider location tracking or monitoring, Global Positioning System (GPS) comes to play a main role. The first known Global Positioning System commenced in 1963. It was TRANSIT and provided Location data to the US Navy's Polaris submarine forces. Due to competing interest between the US Navy and US Air force in the use of this technology led to the implementation of the NAVSTAR satellite network. As a result of the cold war the USSR created their own satellite navigation network known as GLONASS. This network become available for civilian use in 1995[6]. In the passage of time, some satellite networks such as GLOVE-A, GLOVE-B, GLONASS and GALILEO has been launched. A research conducted by Amol, according to his survey GLONASS & GALILEO satellite networks provides more accuracy information. Also he suggest that the GPS is the best technology considering its availability and receiver cost[7].

Many researches on location tracking systems have been conducted in many countries. The application on location tracking systems or real time monitoring systems for public transportation system is an interesting research area where many researchers are working on.

Vehicle Monitor System for Public Transport Management is done by Guilin Li and coworkers in China[8]. This system is developed based on embedded technology. The research is mainly focused on the hardware units which consists of embedded Linux operating system on a chip and integration of GPS and GPRS (General Packet Radio Service) technologies. This system has developed to have capabilities with location tracking, communication scheduling, video monitoring and intelligent passenger counting. System supports dynamic scheduling of public transportation system in

China. Since this system is developed for public transportation management, its direct access to the general public is not available to view the actual location of the vehicle. In China like countries, public transportation system may work in timely manner, So implementing of a location monitoring application for general public may not necessary. The hardware unit is complex and the system consists of heavy programming and resources are some drawbacks. The cost of the entire system also considerably high.

A system for School Bus Environment Monitoring has been done by H.K.Bhangale and his colleague[9]. This system use GSM technology and DGPS (Differential GPS). Differential GPS is a recent technology that has two antennas placed in module. Both antennas gives differential value of the longitude and latitude. Depending upon the differential value correct location can be traced. In addition to location tracking, this system enables temperature sensing and humidity sensing. The hardware unit of this system is integration of Raspberry PI, Carbon Monoxide sensor, temperature sensor, GSM modem and a DGPS modem. This proposed system allows organizations to track their vehicles and to get exact location of vehicle with its temperature and humidity data. Also the system allows those companies to monitor the travelled routes through a web client. Since the system consists of integration of many hardware units, it is a costly solution while it is not tuned to cater public transportation properly.

A research on development on GPS GSM based tracking system has been done by Pankaj Verma and his coworkers[10] . This system is consists of a tracking hardware unit with a microcontroller. Also a GSM module is integrated to the tacking unit to send location data to a separate server. Application consists of a web application to view the vehicle location on a map. On the web application, google map plot the location of the vehicle. This system is mainly designed to monitoring theft of a vehicle in India. The system is suitable to monitor an individual vehicle. However it is not designed or optimized for public transportation systems. In order to monitor a vehicle, a user has to log on to a computer, it is a drawback of this system. It would be more useful if it is optimized for mobile devices. Another main fit fall is a mobile number should be added to the system in software programming in order to proceed with monitoring.

2.3 Recent Developments

With the rise of smartphone usage and the incorporation of GPS technology into these devices, capturing the location data is feasible without an additional cost for separate GPS receiver devices. This lead to give a cost effective method to retrieve location based data. Also smartphones provides GSM and GPRS technologies to facilitate communication of location related data. Few researchers have tried to utilize mobile technology for location tracking systems. For the public transportation systems it would be greatly helpful if the available mobile technology could be utilized.

Very limited number of researches have been done for monitoring, tracking of public transportation in Sri Lanka. One of a most recent research is The real time bus tracking an scheduling is done by Haleem and coworkers[11]. This research is done for Sri Lanka`s public bus system. Wireless sensor technology and GPS, GPRS technologies have been used for this solution. To identify buses, RFID (Radio Frequency Identification) tags are implemented for each bus. The bus-stops and passenger shelters will be equipped with microcontroller based electronic devices which was designed using the Arduino and RFID receiver. Each bus has particular identification number (RFID) placed in bus. Detected RFID information are sent through the Arduino and GSM / GPRS shields which are placed at the bus-stops into central scheduler. Although this system utilize latest technologies, there are practical issues to install the equipment in bus-stops and passenger shelters. Power supply, security and the cost are the main concerns on this system.

Although the research conducted by Medagama and colleagues focused on fleet tracking, it has advantages as it uses open source map server, and it provide options to mark geo-fences in addition to viewing locations. The GIS (Geographic Information System), GPS, GPRS and web technologies are integrated to form the framework for fleet tracking[12]. In this research also they use separate device to get geo location data and it is designed for fleet tracking.

2.4 Currently Available Systems for General Public

There are few currently available methods of information retrieval for train passengers but not for passengers who use buses. Certain drawbacks could be seen on those available methods.

To give the train time table, electronically, Sri Lanka Railway with the Information and Communication Technology Agency of Sri Lanka (ICTA), have implemented eService[13]. This is the main electronic service available to the general public. The service is aimed mainly at displaying the static train schedule for anyone who has access to the web. Although it shows time table it does not represent the dynamic changes such as train late, live location, predicted time etc.

There are few Android Mobile Applications available in the Google Play marketplace. All the applications are focused on static time table for trains. Real time information is not available in any mobile application currently exists.

In 2014, a new train tracking system named Railway Traffic Optimization System (RTOS) was launched for the coastline enabling the general public to view train movements by the Railway Department Sri Lanka with the association of University of Colombo's School of Computing (UCSC)[14]. This solution had implemented only for coastal line trains. However currently that service also not available to the public.

Research on Community Based Train Locating System (CBTLS) has been done by Senevirathna[15]. This research tried to utilize passengers input to the system to track the locations of trains. Mobile application is developed for passengers to update and view train locations. Also it allow passengers to give suggestions, feedback regarding trains. The main advantages are, this system does not use additional hardware units for tracking purpose and any involvement from Railway Department is not necessary. Since location data is updated by general public, reliability and accuracy and proper implementation of this system is not sufficient.

2.5 Problem Definition

The literature review has identified various unsolved problems such as complex set up, cost, reliability, security, practical implementation, unavailability of recent or real time information of applications developed for location monitoring for public transportation systems. Table 2.1 summarize the achievements and the limitations of the key research projects discussed in this chapter.

Research	**Country**	**Technology/Algorithm used**	**Key benefits**	**Limitations**	**Domain**
Vehicle Monitor System for Public Transport Management[8]	China	Embedded technology, GPS and GPRS	Location monitoring, Passenger count and video monitoring	Complex set up. High cost	Public Transport (Buses)
School Bus Environment Monitoring[9]	India	GSM and DGPS	Location, Temperature and humidity monitoring	Many hardware units to integrate. High cost	School Buses
Design and Development of GPS-GSM based Tracking System[10]	India	GPS,GSM, Google maps	Monitor vehicle theft	Separate hardware unit, Not optimized for mobile devices	Private Vehicles
Real Time Bus Tracking and Scheduling System[11]	Sri Lanka	Wireless Sensor RFID, Mobile technology, Microcontrollers	Monitor the actual locations of buses	Hardware units for passenger shelters and bus-stops. High cost, Implementation barriers	Buses
GIS/GPS/GPRS and Web–based Framework for Fleet Tracking[12]	Sri Lanka	GIS,GPS,GPRS,Microcontroller and web technology	Location tracking with geo fences feature	Not optimized for mobile devices. Separate hardware unit is used	Fleet Tracking
eService by The Department of	Sri Lanka	Web technology, Android App	Train time table on web	Static time table, No real time	Trains

Railways			and mobile	monitoring available	
Railway Traffic Optimization System 2014	Sri Lanka	GPS, GPRS and Web	Real Time train location can be viewed	Only for coastal line. Currently unavailable	Trains
Community Based Train Locating System (CBTLS)	Sri Lanka	GPS, Android, Web	No additional devices, Real time location data	Reliability and accuracy issues	Trains

Table 2.1 – Summary of key Researches

According to the above summary it is evident that many reaches have been done with different technologies. As shown in Table 2.1, location tracking systems for transportation has been a hot topic and many researchers have identified the need of further research in this regard.

According to Table 2.1, despite many solutions are available for location tracking, they are rather expensive, utilize separate hardware units. Therefore, we cannot go for a customized solution of the existing approaches, instead should go for a low-cost simple solution that cater the location monitoring facility.

We have further noticed that limited research in location tracking with Android has been recorded in the literature. Since there is a rise in smart phone usage, Android apps has been the most popular light weight, cost effective technology anyone can use. Instead of complex hardware devices, we can utilize a mobile phone to track locations. Since general public mostly using smart phones, they can access responsive web application to find actual location, arrival time of their bus/train. Details of the technology behind the solution will be discussed in Chapter 3.

2.6 Summary

This chapter presented a comprehensive literature review on the location tracking/monitoring research and identified the research problem as the inadequate in simple, cost effective and practical solution for location tracking for public transportation. We also identified that the android smartphone, GPS, GSM and open source web technology can be utilized to address the above problem. Next chapter will discuss the technology to be used for our solution.

Chapter 3

Technology Adopted - TrackIT

3.1 Introduction

In the previous chapter, various researches to address the same issues were critically reviewed. Advantages, disadvantages and features of existing systems and proposed systems were analyzed and listed.

In this chapter, technologies regarding location tracking system will be described. Also technologies and methods and tools used in development, testing and implementation will be discussed.

3.2 Technologies Available

3.2.1 Location Tracking

Location tracking can be categorized to two sections namely local area or indoor tracking and wide area tracking. For local area tracking RFID (Radio Frequency Identification) can be used[16]. In such a case, each asset would be tagged with an RFID tag, and readers would be placed in strategic locations to be able to accurately read those tags within a matter of inches. In our case wireless RFID technology is not suitable since we need to track wide area.

To track a large geographical area, most accurate technology is GPS. The GPS receiver should be installed in the assets/vehicle to be tracked. On the request the GPS satellites give accurate position of the location. The drawback is GPS is limited in indoor areas.

Android smartphones comes with inbuilt GPS feature. So we can use this technology for our tracking system. The user location would be acquired through GPS and Android's Network Location Provider. The GPS is more accurate, it only works properly outdoors, and it consumes more battery power. The Android Network

Location Provider determines user location using cell towers and Wi-Fi signals, and provide location information in both indoor and outdoor. Therefore we use both methods in our application to get the best location update.

3.2.2 Main web application

There are many technologies such as ASP.NET, Java and PHP to develop web applications. PHP is open source scripting language and there are many open source frameworks and tools available to develop applications. PHP supports object oriented concepts and the hosting cost is also comparatively low. Drupal 7 is a CMS (Content Management System) available with many features to develop comprehensive application on top of it. Many comprehensive corporate applications have been developed on this open source technology stack. Therefore we can develop our application on the top of Drupal core by utilizing its advanced features such as User API, Theming layer and security features. We can develop separate modules for our requirements by utilizing features available.

For the development and testing environment we use WAMP server installed on local computer. WAMP server is a web development environment for Windows. The WAMP server consists of Apache, MYSQL and PHP to web development environment. For different operating systems, there are LAMP, MAMP separately for LINUX and Mac OS. XAMPP is also a web development environment similar to above. Since we are using Windows, we use WAMP server for our local development and testing purposes.

As the database we use MySQL since it is free and open source software with community support. Also there are open source tools such as Phpmyadmin, MySQL workbench are available to view and manage database. There are enough convenient integration, scalability and flexibility in MySQL with PHP. Therefore we use MySQL as our database.

For the implementation and demonstration purposes, Linux web hosting with Apache web server and MySQL database is suitable since it is comparatively low cost and it can be scalable. For this purposes we can use shared hosting plan with CPanel access form a hosting provider. For the production mode of the application, cloud hosting solutions could be used due to its high availability and scalability.

The frontend of the web application should be accessible form mobile devices also. It should be supported for all major web browsers. For the frontend theme, HTML, CSS, and JQuery is used. The frontend template is fully responsive, then it can be properly accessed through any mobile device.

For security concerns we use coding practices and necessary modules to avoid XSS (Cross-Site Scripting), (CSRF) Cross-Site Request Forgery attacks. In addition to application level security, web server side security enhancements like WSF (Web Server Firewall) make the solution more secure and stable.

3.2.3 Mobile application

In Sri Lanka, usage of smartphones are rapidly growing. Among smartphones, Android is the majority used type. Also cost is low. Therefore using Android for our solution is more beneficial to the majority. Since smartphone comes with inbuilt GPS feature, we do not want to install separate GPS receiver modules. Development using android is relatively low cost compared to platforms like IPhone, which requires their specific Operating System and tools for development. Also it provides strong built in functionalities and as well as has larger community support. The development tools for android is also freely available to download.

3.3 Technology Stack

Our Technology stack is Apache 2.2 web server, MySQL 5.6 database and PHP 5.6. The overall technology stack can be illustrated as Figure 3.1.

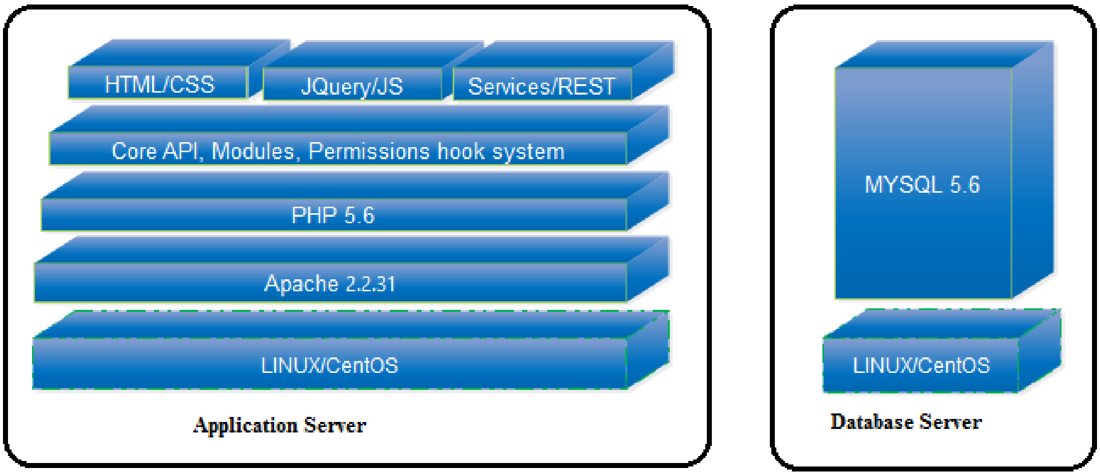


Figure 3.1 – Technology stack

3.4 Summary

In this chapter the technologies used for our location tracking system –TrackIT were described. Reasons for selecting relevant technologies also explained. Some advantages and disadvantages of technologies briefly described.

In next chapter, the approach to implement TrackIT will be described.

Chapter 4

An approach to TrackIT

4.1 Introduction

Having define the problem in Chapter 2 we presented technology required for the proposed solution in chapter 3. This chapter presents our novel approach to use android and open source web technology to address our research problem. The approach is described under the headings of hypothesis, users, input to the system, output of the system, process to convert input to the output and overall features of the system.

4.2 Hypothesis

We hypothesis that simple and cost effective location tracking system which would address problems in passengers` perspective, can be developed using Android smartphone, open source web technologies and utilizing mobile networks.

4.3 Users

Number of users can be benefited by the TrackIT system in multiple ways. More importantly passengers and vehicle owners can directly benefited by the system. Also management and decision makers of SLR (Sri Lanka Railway), SLTB (Sri Lanka Transport Board) and RDA (Road Development Authority) can benefitted by system using data collected to identify patterns and to support decisions. University students who interested in studying location tracking systems and data analyzing can use the system for learning purposes. Three type of users directly interact with the system can be mentioned as,

- Passengers
- Locomotive Engineers/Drivers

- Administrators

Locomotive engineers and drivers are the location updaters who are the authorized person to use location updater application on smart phone for enabling the tracking process. Administrators are the people who manage all master data in the system. They generate necessary reports for management upon request.

4.4 Input to the system

The system can accept location information from multiple smart phones which has location updater android application. The input could be as,

- Username, password (Authentication details)
- Train/Vehicle details (shuttle api key, turn number, tracking period)
- Location data (longitude, latitude and speed)
- Master data (Train time schedule, Railway stations, Bus schedule, Routes)

Any user accessing the system master data and location updates should go through an authentication process. Passengers can browse the system as anonymous users.

4.5 Output of the system

The output of the system would be real time location details of a particular train or bus plotted on a map. The output could be as,

- Real time location with speed
- Predicted time for arrival, destination
- Web service with real time location data
- Time schedules
- Results in reports and graphs

4.6 Process

We have divided the entire process in to six major parts which are the, administration, monitoring, receiving geo location data, sending location data, geo location data update, Frontend displaying process and web services.

4.6.1 Administration process

Administration process consists of all the administration tasks of the system including user management and master data management. First step of administration process is user authentication. Any user should go through authentication process to access the administration panel. In our novel approach we utilize Drupal core for authentication and master data management. Separate interfaces would be created for adding master data for trains and buses. Train lines, stations, bus routes would be categorized as different content entities. User management module is used to generate different user roles with relevant access permissions.

Using graphs and visualization, location and speed information of trains and buses would be shown in graphs and reports. For this, PHP, Views API and google graphs would be used for reports and visualizations separately.

4.6.2 Receiving geo location data

Capturing real time GPS coordinates together with speed details and sending them to the system is the main input source of the system. Although GPS coordinates can be retrieved using hardware devices, our novel approach is to retrieve those details through an android application. Since cost of android smart phone is low and it is equipped with GPS receiver, we utilize it. Android is suited with necessary classes and libraries to handle geo coordinates and speed information. Using android location manager we can receive geo coordinates. Using android services class, request for receive GPS data can be run periodically.

4.6.3 Sending geo location data

Received location data should be sent to the main web application periodically. This is done using the httpClient library. HTTP GET method is used to send data to the main application. Geo coordinates, speed is sent to the web application together with authentication data one android supplication started to run. Time interval to capture and send data allowed to be altered by selecting predefined values. Android services library would be used to run the application in the background.

4.6.4 Geo location update process

This process is handled in the main web application. Different endpoints would be defined separately for updates from trains, buses and private vehicles. Each location data received from mobile applications would be authenticated. In the endpoint received data is validated with different criteria such as availability, correct format, range of valid distance. Valid coordinates and speed details would be saved into the database together with particular train, bus details and time received.

4.6.5 Frontend display process

For each train, vehicle separate dynamic page is generated once added to the system. Schedule details would be shown in tabular format using PHP and HTML. A google map is embedded in this page. Geo location data saved in the database is retrieved in JSON format and real time locations will be plotted on the map together with speed and time.

The search functionality would be implemented for both trains and buses. In the search results predicted arrival time and destination time would be displayed. For the prediction process historical data would be analyzed and average speeds are calculated for particular turns and days of week. Both historical data and real time data would be used to make predictions. Our formula to calculate prediction time is shown below.

Average Speed = (Current Speed + Average Speed for the Turn + Average speed for the day)/3

Predicted Time= Distance/ Average Speed

4.6.6 Web services

External parties are allowed to use real time data from our system through services. Separate REST (Representational State Transfer) web service endpoints would be defined for trains services and bus services. Only data retrieval permission would be made available through the public web service. Schedule information together with real time geo coordinates and speeds are exposed to the REST service through our services module.

4.7 Features

In connection with the input, output, users and process, the over features of the system include following characteristics.

- Online web solution
- Mobile Application solution
- Low implementation cost
- Multiple location trackers
- High level of accuracy
- User friendly
- Easy accessibility
- Big data

4.8 Summary

In this chapter we have described overall solution of our research. We have mentioned problem definition and assumption of the solution to that problem. We have described clearly inputs to the system, outputs of the system and how convert input to output, who has benefits of this system, and features of the solution. Next chapter we will describe in detail, extended design of our process and what our system does.

Chapter 5

Design of TrackIT

5.1 Introduction

The previous chapter gave full picture of the entire solution. This chapter describes the design of solution for the process presented in the approach. We design the solution as a client-server system with a backend database and administration interface. Separate android application is designed to capture geo coordinates as inputs to the system. Here we describe the top-level architecture of the design by elaborating on the role of each component of the architecture.

5.2 Top Level Architecture

The top-level architecture of TrackIT consist of five main modules, namely, Webserver, Database, Central web application and Location updater Android application Figure 5.1 illustrates the Top Level architecture.

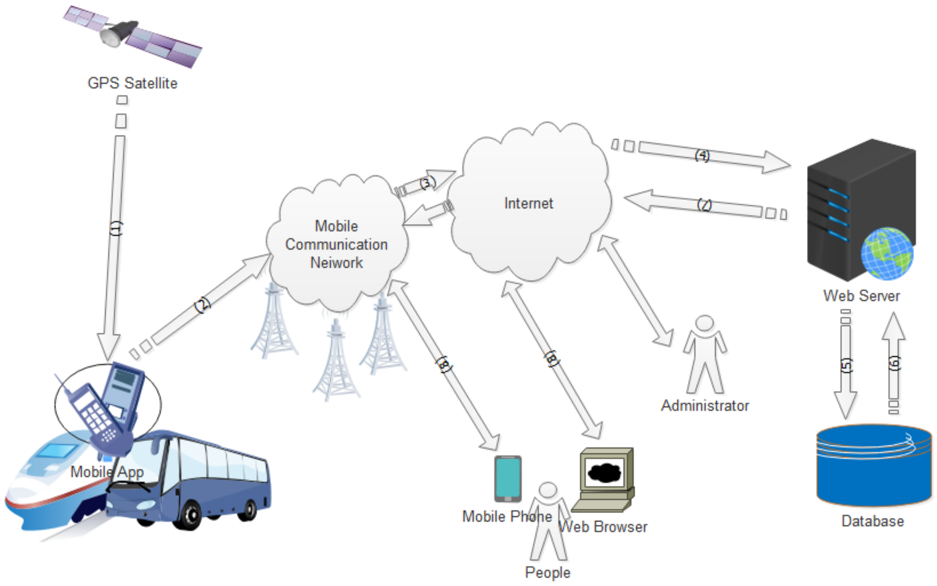


Figure 5.1 – Top level architecture

According to above Figure 5.1 overall process can be summarized. The location updater android app placed in a bus or train. It sends geo location updates over mobile network and the Internet to the central web application. The web application validates the geo-location updates and save to the database with timestamp, route and speed details. The web application has a public interface to view location information. When an anonymous user browse the web page, particular location data will be retrieved and it will plot on a map. The map is updated automatically so once a location update received by the system, it will be show on the map. Accordingly users can monitor location of bus or train near real time. Also passengers can search for bus or train for a particular destination, then the system show available buses, trains with predicted time in addition to static time table. If a particular bus or train is already running on the particular route, it indicates the availability and shows live data on the map.

5.3 Main Web Application

The core of the solution is the main web application hosted on a public web server. This system consist of a user management module, user authentication module, geo location data handling module, public user interface (map display) and a web service. Figure 5.2 show the overall structure of the web application.

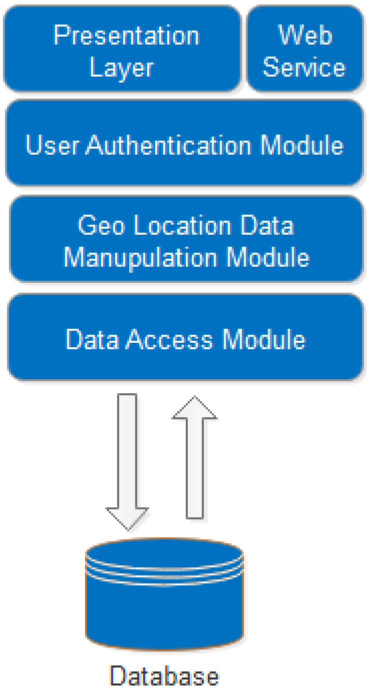

Presentation Layer
Web Service
User Authentication Module
Geo Location Data Manupulation Module
Data Access Module
Database

Figure 5.2 – Web Application Structure

The presentation layer consist on the public user interface. It allows users to select routes and view location updates on a map. Also it allow to search buses/trains and view static time tables and predicted information together with live data if available. This is the main interface use by the general public.

The user management module allow to manage administrative users. Also it provides user authentication features. It provide unique identification for each location updater app. This is implemented to avoid invalid inputs to the system. Also it ensures that data inputs are from trusted and correct sources.

The geo-location data module is the main module directly deal with the location update data. This module validate the received data categorize them according to the route. Location information for the particular route is send to data access layer to save to the MYSQL database. Also this geo location module retrieve data to plot data on the map.

The data access layer supports perform CRUD (Create, Read, Update and Delete) operations with the database.

The web service is available for third parties to retrieve real time location updates for their applications or systems. Once someone request for particular route, the service sends location updates as REST service.

The web application is hosted on Apache web server and use MYSQL as the database.

Different modules namely trackit track, trackit train, trackit map, trackit front, trackit api and trackit coordinates are developed to handle buses, train and vehicles inputs, process and outputs. For geo coordinate validations and processing a separate module is used while prediction and presentations use separate modules. Details of above modules will be discussed in next chapter in details.

5.4 Location Updater – Android Application

This is the native android application which would communicate with the main web application as the input source. In this application, the user location would be acquired through GPS and Android's Network Location Provider and is sent to the web application as a series of geo coordinates. Using Android Services Class, it is scheduled to send geo coordinates together with speed and identification data periodically. Instead of using separate hardware units to receive and send GPS coordinates, we use smartphone with our location updater android application. So it is highly available and cost effective.

5.5 Public web interface

This interface is the public frontend of the main web application. It acts as a main output of the system. It retrieve location data and plot in a map to display real time location information. Also it enable to view static time table information and predicted information. This interface is developed using PHP, HTML and JQuery. For maps google map API is used. This interface is responsive, so it can be accessed from any mobile device.

5.6 Public web service

This is an optional feature to enable third party web sites, users can use to get real time geo location data to display in their applications. This service is developed as REST (Representational State Transfer) service which is lightweight and easy to build and use.

5.7 Use case Diagrams of the system

Appendix A illustrates use case diagrams of the system.

5.8 Sequence Diagrams of TrackIT

Appendix B consists of sequence diagrams of the system.

5.9 Database Diagram of TrackIT

Please refer appendix C for database diagram of the core system and for specific modules for TrackIT.

5.10 Summary

This chapter mainly described the overall architecture and the design of each components with relevant technologies and their interconnections. Reason to use particular component and its functionality also described. Further use case diagrams, sequence diagrams and database diagrams also listed in this section.

The following chapter is mainly discussed the implementation details of the TrackIT system. It will present some important code segments and related implementation details.

Chapter 6

Implementation of TrackIT

6.1 Introduction

Chapter 5 presented the design details of TrackIT. This chapter describe detailed implementation of the system. The overall solution has been implemented as an Open source application running on apache web server. An android application is implemented as the main input source for the system. The algorithms, hardware, software, pseudo codes and relevant code segments of the implementation are presented in this chapter.

6.2 Overall solution

Overall solution has been implemented as a web application that can be accessed by any client running on any OS including Windows or Linux. This is primarily client-server architecture. The application is primarily PHP based solution with additional use of Android application as the main input source.

6.3 Implementation plan for demonstration purpose

For the demonstration purpose of this research, only a single rural bus route (Panadura – Galpatha) and a single train between selected stations (Panadura - Kollupitiya) were considered. Further it was considered office transport times where availability is high and many passengers used to go to work and come back. Master data for both train and bus was entered through the administrative interface. The schedule data is taken from official site of Sri Lanka Railway and bus data was directly collected form the bus staff. Initial version of the system is hosted on a public shared server to collect data.

6.4 Software and Hardware used in TrackIT implementation

At the time of implementation a set of tools were used for the development purposes as described in this section. For the web application, integrated development environment (IDE), Netbeans 8.0.1 was used. Netbeans IDE selected for development as it is a comprehensive development tools with a large number of plugins available and a widely used IDE and it supports PHP and support for html markup language also. As the core of the web application, Drupal 7 which gives capability of extending its features through its hook system is utilized. It provide API to make customized developments for each module such as user permissions, business logic and theming layers. Comprehensive custom developments can be implemented on top of it through modular approach in which each component can be developed as separate modules. Developing different components as modules enable maintainability, scalability and portability. Also the modules can be contributed to the open source community to use them.

WAMP server which consist of apache, MySQL and PHP was used for local development purposes and testing purposes of the application. Phpmyadmin interface was used for designing and creation of database. For the hosting for data collection and demonstration purposes, web application is hosted on Linux shared hosting plan. The hosting environment consist of Apache 2.2 web server, MySQL 5.6 database and PHP 5.6.

For the development of location updater android application, Android Studio 2.2.2 IDE was used. Android Studio is also a widely used IDE which has many development supporting features, debugging and simulation features. For the testing and data collection through android application, Samsung J5 smart phone was used.

6.5 Main Web Application

The main web application is developed on PHP which is a scripting languages with many features. Its efficiency is high since it consumes few of system resources. It is open source and has object oriented features. PHP is cross-platform since it is running on LINUX, UNIX and WINDOWS. Cost effective hosting solutions are available to host PHP applications. Popular and most stable and extendable framework Drupal has been used as the core of the application. This was used considering the stability, security and further scalability. Also it enable the use of standard hooks instead of re-inventing the wheel. Highly customizable, complex applications, modules can be developed on the top of the core.

Major components of our application is developed as different modules which enable easy troubleshooting, maintainability, scalability, portability and reusability. Different permission levels can be granted to different modules and it enables to restrict access of unauthorized calls. It provide the feature to enable, disable when needed. The web application project set up in the Netbeans IDE is shown in Figure 6.1

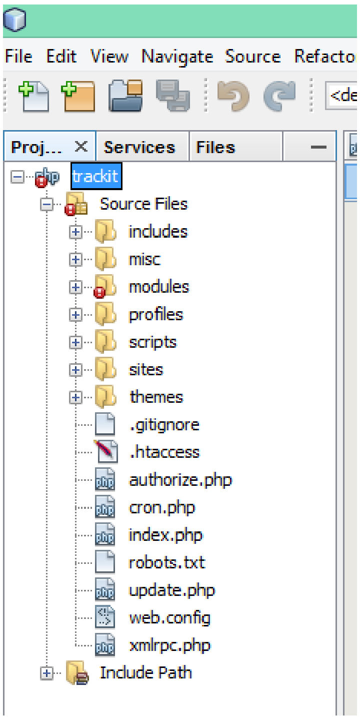
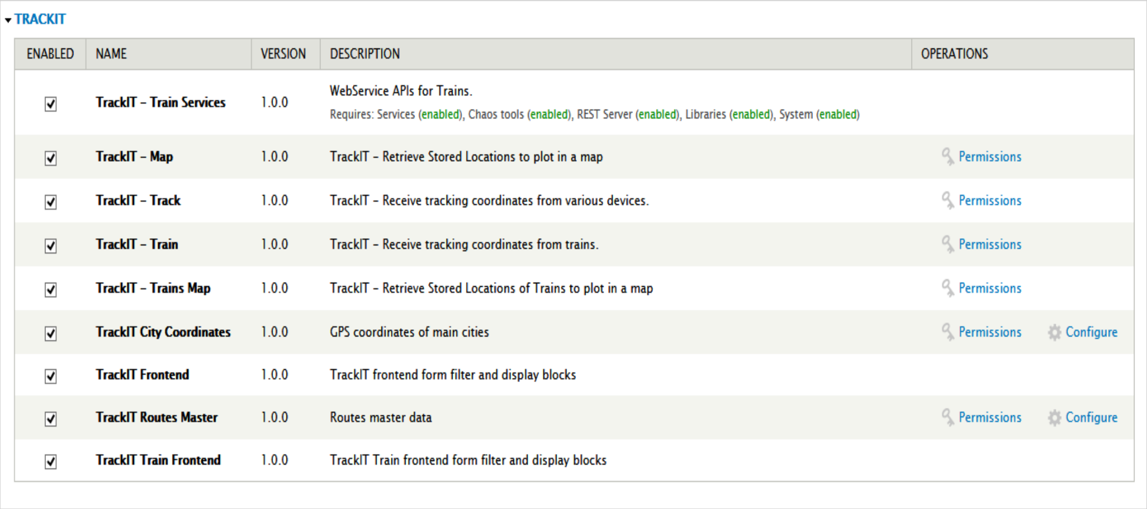


Figure 6.1 – Web application structure in Netbeans

The main web application consist of separate module to manage users, permissions, manipulate geo coordinates from different sources, prediction and calculations and frontend display. Also there are sub modules to support main features. Detail of each module is discussed below. Figure 6.2 shows set of developed modules listed on administration panel.



TRACKIT

ENABLED	NAME	VERSION	DESCRIPTION	OPERATIONS
☑	**TrackIT – Train Services**	1.0.0	WebService APIs for Trains. Requires: Services (enabled), Chaos tools (enabled), REST Server (enabled), Libraries (enabled), System (enabled)	
☑	**TrackIT – Map**	1.0.0	TrackIT – Retrieve Stored Locations to plot in a map	Permissions
☑	**TrackIT – Track**	1.0.0	TrackIT – Receive tracking coordinates from various devices.	Permissions
☑	**TrackIT – Train**	1.0.0	TrackIT – Receive tracking coordinates from trains.	Permissions
☑	**TrackIT – Trains Map**	1.0.0	TrackIT – Retrieve Stored Locations of Trains to plot in a map	Permissions
☑	**TrackIT City Coordinates**	1.0.0	GPS coordinates of main cities	Permissions Configure
☑	**TrackIT Frontend**	1.0.0	TrackIT frontend form filter and display blocks	
☑	**TrackIT Routes Master**	1.0.0	Routes master data	Permissions Configure
☑	**TrackIT Train Frontend**	1.0.0	TrackIT Train frontend form filter and display blocks	

Figure 6.2 – List of developed modules in administration panel

6.5.1 User Module

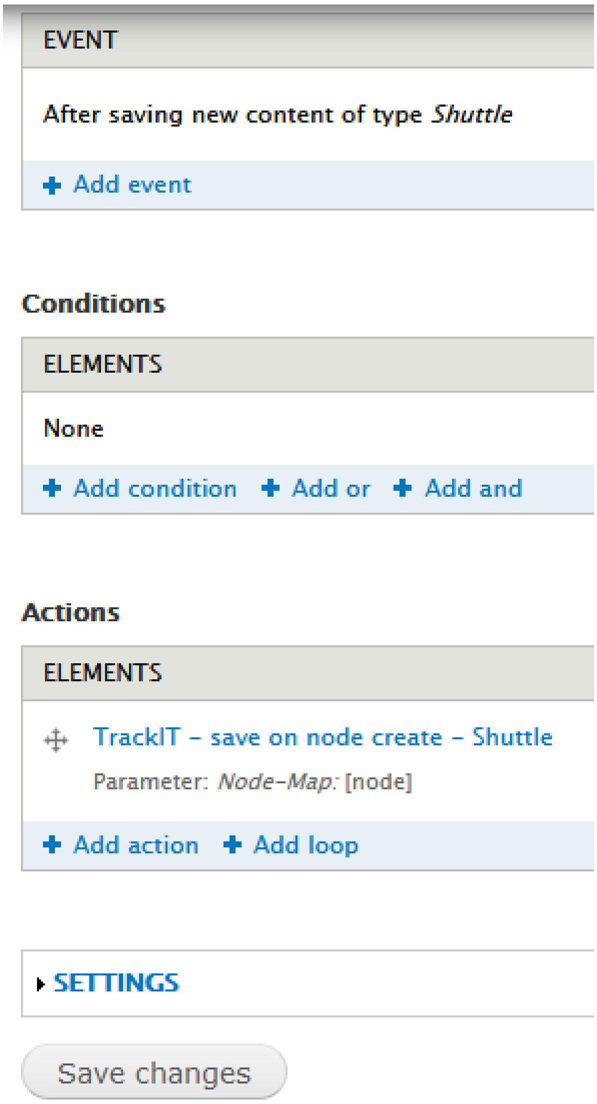
User module is used to add user roles, grant different permissions and manage system users. Also it supports user authentication process. Since different user permissions should be given to different users, separate user roles are created for administrator, anonymous user and location updater. Permission levels were handles for different user roles using the hook permission.

Please refer Appendix D for the important code segments of user permissions and interfaces of user module.

6.5.2 TrackIT Track Module

This module was developed to manage master data related to buses and routes. Also it receive geo coordinates from location updater android app and validate and store in the database. In the core system, it allows to define rules to trigger different actions based on particular events. For buses, content type named Shuttle has been created with necessary fields to add schedules and stops. The rules are configured for each CRUD operations of shuttles. Then this custom module captures the values and save to the relevant tables of the database. Those data handling actions have been coded in this module.

Definition of a rule for a shuttle (bus) adding event is shown in Figure 6.3.



EVENT

After saving new content of type *Shuttle*

Conditions

ELEMENTS

None

Actions

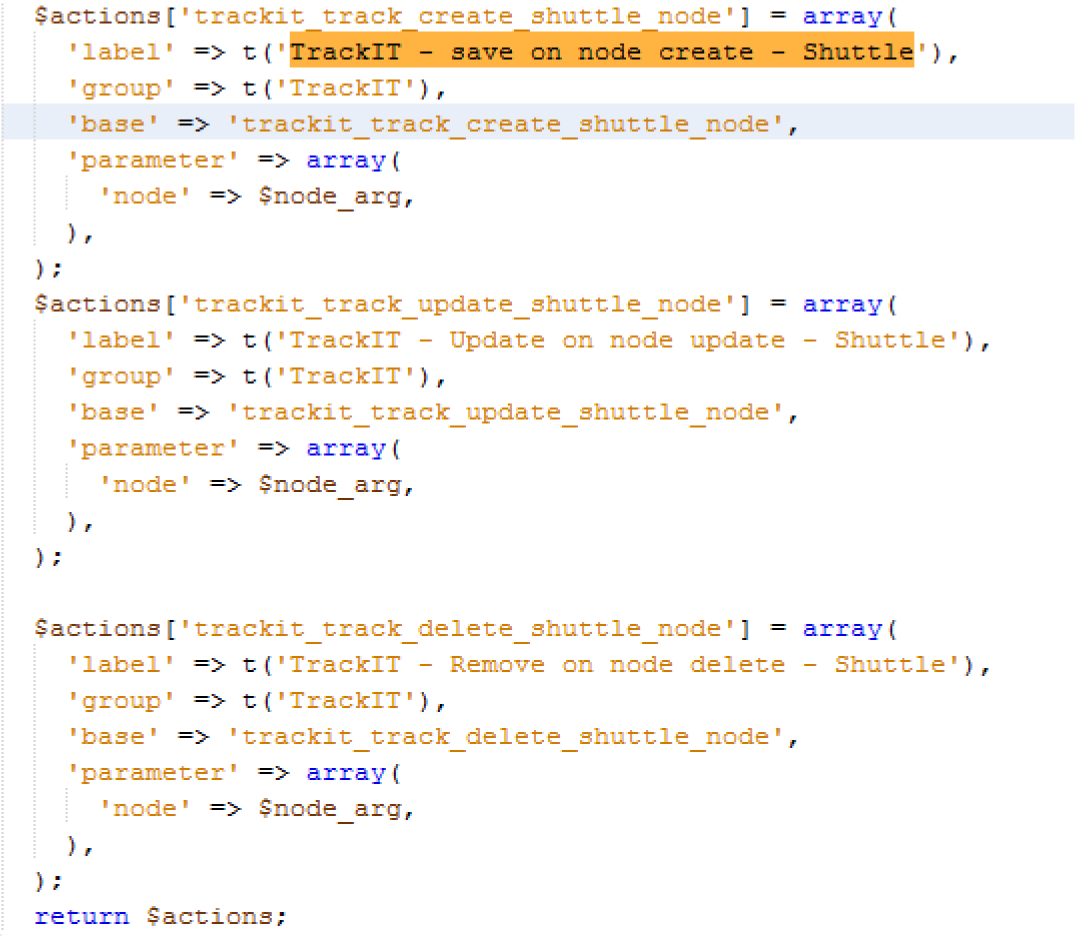
ELEMENTS

TrackIT – save on node create – Shuttle

Parameter: *Node-Map:* [node]

Figure 6.3 – Shuttle node create actions

Code segment to define the above action is shown in Figure 6.4.

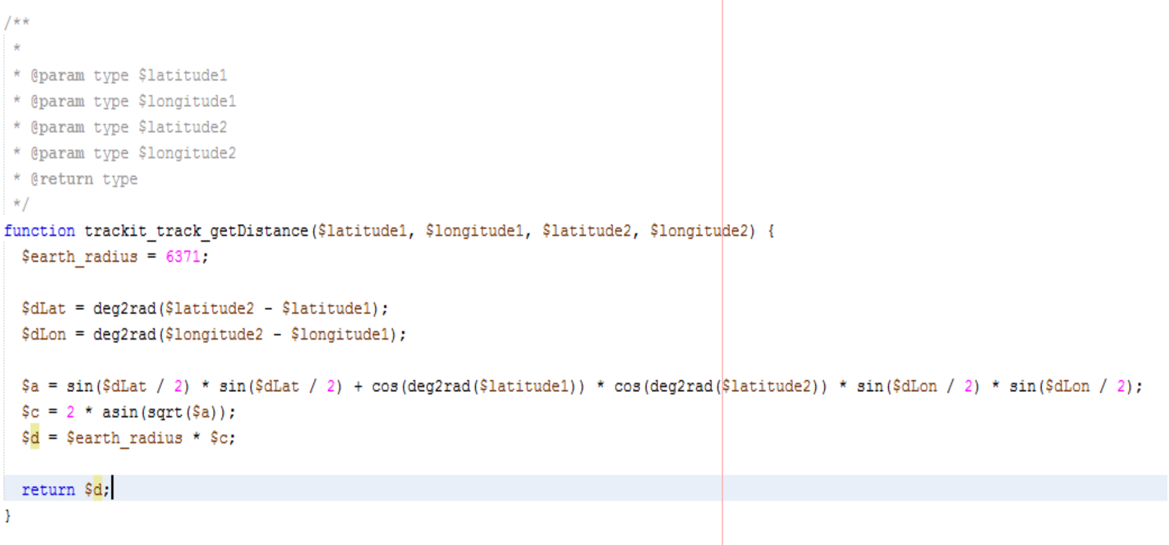


```
$actions['trackit_track_create_shuttle_node'] = array(
  'label' => t('TrackIT - save on node create - Shuttle'),
  'group' => t('TrackIT'),
  'base' => 'trackit_track_create_shuttle_node',
  'parameter' => array(
    'node' => $node_arg,
  ),
);
$actions['trackit_track_update_shuttle_node'] = array(
  'label' => t('TrackIT - Update on node update - Shuttle'),
  'group' => t('TrackIT'),
  'base' => 'trackit_track_update_shuttle_node',
  'parameter' => array(
    'node' => $node_arg,
  ),
);

$actions['trackit_track_delete_shuttle_node'] = array(
  'label' => t('TrackIT - Remove on node delete - Shuttle'),
  'group' => t('TrackIT'),
  'base' => 'trackit_track_delete_shuttle_node',
  'parameter' => array(
    'node' => $node_arg,
  ),
);
return $actions;
```

Figure 6.4 – Defining actions for shuttle nodes

Separate functions were defined for CRUD operation of master data. Screens of managing master data related to shuttles/buses is illustrated in appendix E.

Another main functionality of this module is validate the geo coordinate data and store them in the database. Permission level to accept geo coordinates was defined. So it allowed to accept geo location data only form allowed levels. This module enable a url path (endpoint) to send data from the mobile application. The mobile application send geo location data together with session id, shuttle key information to identify the particular shuttle/bus. In the web application module end point, it filter the received values to avoid vulnerable inputs. Then check for the required parameters such as longitude, latitude, shuttle key, etc. If those data is not available the request will be rejected. If the inputs are valid it check for the validity of the key and the turn number. If it also validated, then received coordinates and relevant speed data is stored in the database if the received coordinated within a defined range in the master data. To get the distance between two coordinates Haversine formula was used. Implementation of Haversine formula in our track module is shown in Figure 6.5



```
/**
 *
 * @param type $latitude1
 * @param type $longitude1
 * @param type $latitude2
 * @param type $longitude2
 * @return type
 */
function trackit_track_getDistance($latitude1, $longitude1, $latitude2, $longitude2) {
  $earth_radius = 6371;

  $dLat = deg2rad($latitude2 - $latitude1);
  $dLon = deg2rad($longitude2 - $longitude1);

  $a = sin($dLat / 2) * sin($dLat / 2) + cos(deg2rad($latitude1)) * cos(deg2rad($latitude2)) * sin($dLon / 2) * sin($dLon / 2);
  $c = 2 * asin(sqrt($a));
  $d = $earth_radius * $c;

  return $d;
}
```

Figure 6.5 – Implementation of Haversine formula

6.5.3 TrackIT Map Module (trackit_map)

This module was developed to retrieve location data and route information to plot the map. For this Google maps API is used which is a way in which a Google map is embedded in the web page through JavaScript. The API provides a large number of practical feature such as marking, distance calculation and information windows for dealing with the map.

The map display is handled in the theming layer of the web application while this module handles only the processing part and the data retrieval part. The request endpoints are defined in this modules to cater location requests from the theming map layer. When a request is made, this module retrieve relevant data from the database and process. The processed data is send back to the theming layer in JSON format.

6.5.4 TrackIT Front Module (trackit_front)

This module gives the functionality of searching buses for particular source and destination. Form API was utilized to develop frontend search form. It shows the time table and live availability of the bus. Also this module calculates predicted time based on both live data and historical data. Appendix G illustrates all screens of these modules.

6.5.5 TrackIT Train Module (trackit_train)

This module is similar to above explained TrackIT Track Module. The difference is this module is designed to manage master data related to trains and receive geo location data sent from the Android application.

In the core system, when it allows to define rules to trigger different actions based on particular events. For trains, content type named Train was created with necessary fields to add stations and schedules. The rules are configured for each CRUD operations of train nodes. Then this custom module captures the values and save to the relevant tables designed for trains in the database. Those data handling actions were defined in this module. Implementation screens of managing master data related to trains is illustrated in appendix F.

6.5.6 TrackIT Train Frontend Modules

The modules trackit train map and trackit train front, separately used to display locations plotted on a map and enable search functionality to find trains. Additionally trackit train front module predict time based on live data and historical data analysis.

6.5.7 TrackIT Train API Module

This module enable third parties to access train, bus details through services. Access is limited to information retrieval. The response of the service is given in both JSON and XML formats to choose. Initially the endpoint is defined and anonymous requests were allowed to use the service. Detailed screens of web services module is explained in appendix H.

6.5.8 TrackIT Vehicle Module

This module is similar to above explained TrackIT train Module. The difference is this module is designed to track vehicles added by the registered users. This module also separate endpoint to receive geo location data from the mobile application. Implementation details are illustrated in appendix G.

6.5.9 Prediction Method

To predict the arrival time for particular destination, few methods can be used. The simplest method is using the distance and speed. It calculates the distance to the destination form the current location. Then by taking the current speed, predicted time can be calculated.

Time= Distance/Speed

Result of this calculation is not accurate since the speed is not uniform in practical scenario. Therefore this prediction is much deviated from the actual result. To make it smooth, instead of taking instant speed, calculate the average speed of a particular turn of the train or bus. This average speed leads to better prediction.

Approximate Time = Distance/Average speed for a particular turn

Since system collect large amount of data, we can analyze those data to identify patterns. Through the analysis of collected data, speed patterns could be identified. Accordingly average speed for a particular train, bus for a relevant turns/trips were recorded. Also average speeds for particular days of the week also identified. While increasing the amount of collected data, those calculated information could become more accurate.

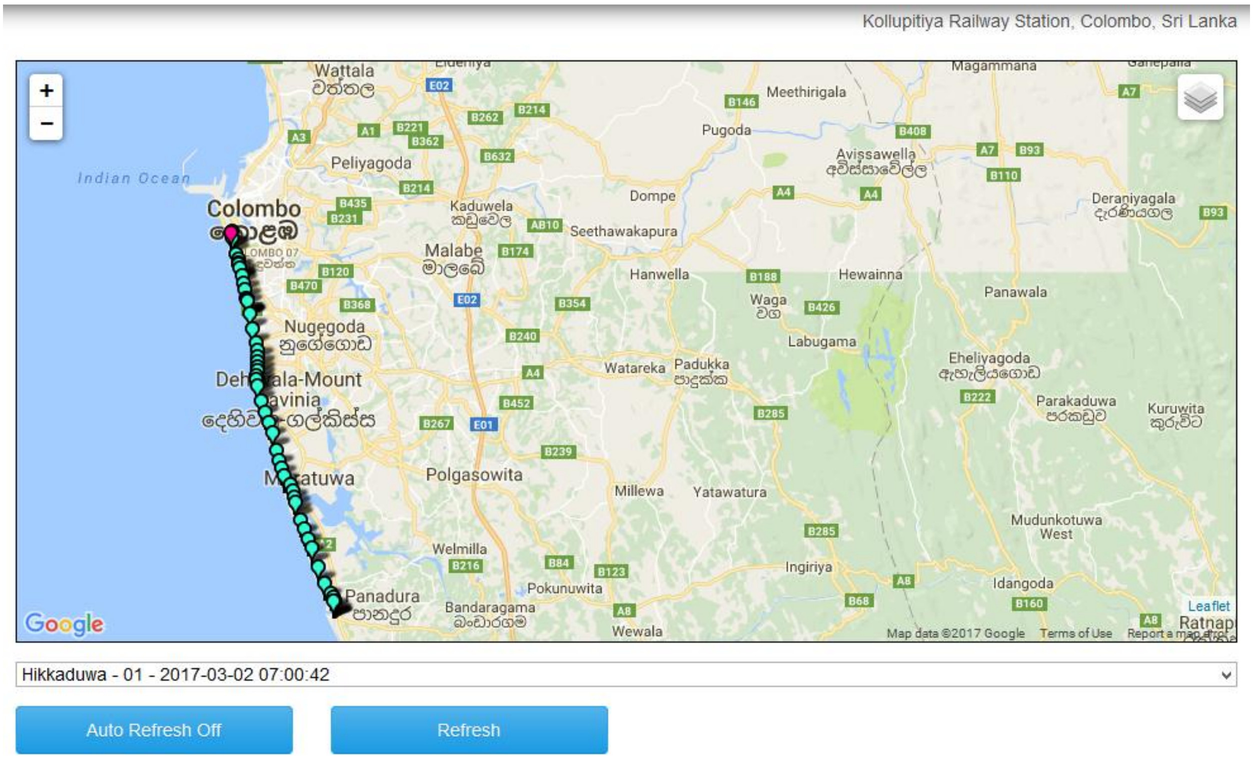
By considering above factors algorithm to get predicted time is taking the mean of current speed, average speed and the average speed for particular day. The formula is shown below.

Average Speed = (Current Speed + Average Speed for the Turn + Average speed for the day)/3
Predicted Time= Distance/ Average Speed

Data analysis section of this chapter consists of more details regarding speed distributions.

6.5.10 Public Web Interface

This is the frontend of the main web application visible to general public. The location update data is retrieved from the database through trackit map module and front modules as described above. These formatted data is sent to the theme to plot on the google map. To implement this PHP, HTML and AJAX was used. Figure 6.5 shows the frontend map for a train named Hikkaduwa.



Kollupitiya Railway Station, Colombo, Sri Lanka

Hikkaduwa - 01 - 2017-03-02 07:00:42

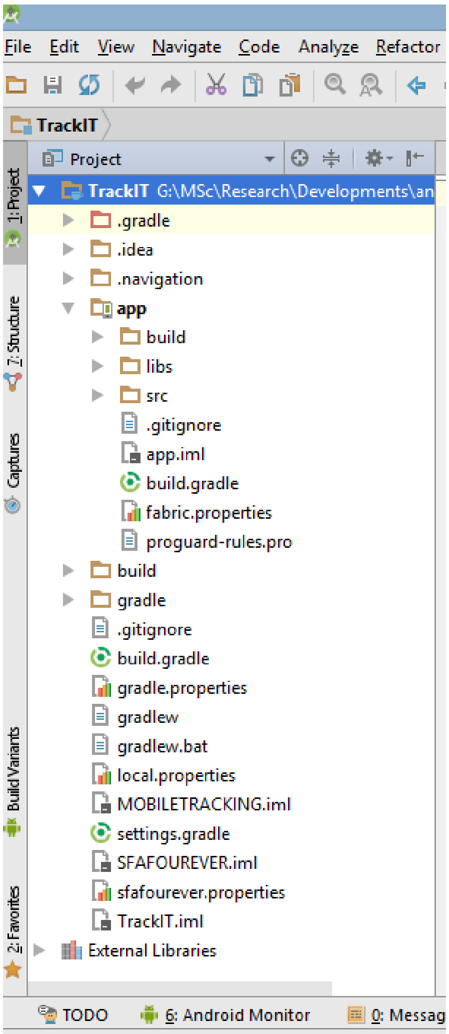
Figure 6.6 – Frontend map for Hikkaduwa train

All frontend interfaces are illustrated in appendix G.

6.6 Location Updater Mobile Application

This android application is the main input source for the system. Instead of using separate hardware units, smartphone is used to get GPS location updates and send them to the central application for processing.

For the development of android application, Android Studio 2.2.2 IDE was used. The project structure in Android Studio is shown in Figure 6.32.



```
▼ TrackIT G:\MSc\Research\Developments\an
  ▶ .gradle
  ▶ .idea
  ▶ .navigation
  ▼ app
    ▶ build
    ▶ libs
    ▶ src
      .gitignore
      app.iml
      build.gradle
      fabric.properties
      proguard-rules.pro
  ▶ build
  ▶ gradle
    .gitignore
    build.gradle
    gradle.properties
    gradlew
    gradlew.bat
    local.properties
    MOBILETRACKING.iml
    settings.gradle
    SFAFOUREVER.iml
    sfafourever.properties
    TrackIT.iml
▶ External Libraries
```

Figure 6.32 – Location Updater App project structure

The user location would be acquired through GPS and Android's Network Location Provider. The Location Manager class is used to get location updates. The GPS is more accurate, it only works properly outdoors, and it consumes more battery power. The Android Network Location Provider determines user location using cell towers and Wi-Fi signals, and provide location information in both indoor and outdoor. Therefore we use both methods in our application to get the best location update. Since the application should periodically send location data to the central application, getting GPS coordinates and posting them to the endpoint url is should be happen in background. To achieve this android service component was used. Android Service is a component that runs in the background to perform long-running operations without needing to interact with the user.

For the testing and demonstrate purpose, android application is developed by allowing to change location update periods as 15 seconds, 30 seconds 1 minute, 5 minutes, 15 minutes etc. This enabled to select the best period for location updates. Also in demo application endpoint url is also made available to entered manually since the url may change during the data collection period. Also location data for trains and buses send to different endpoints.
Please refer appendix I for more implementation details.

6.7 Deployment View

The main application is hosted on Apache web server. The web server is connected to a MYSQL database server. For testing and demonstration purposes, a shared Linux hosting plan is used. The Firewall and related network infrastructure handled by the hosting provider. Following diagram illustrates the deployment environment of TrackIT.

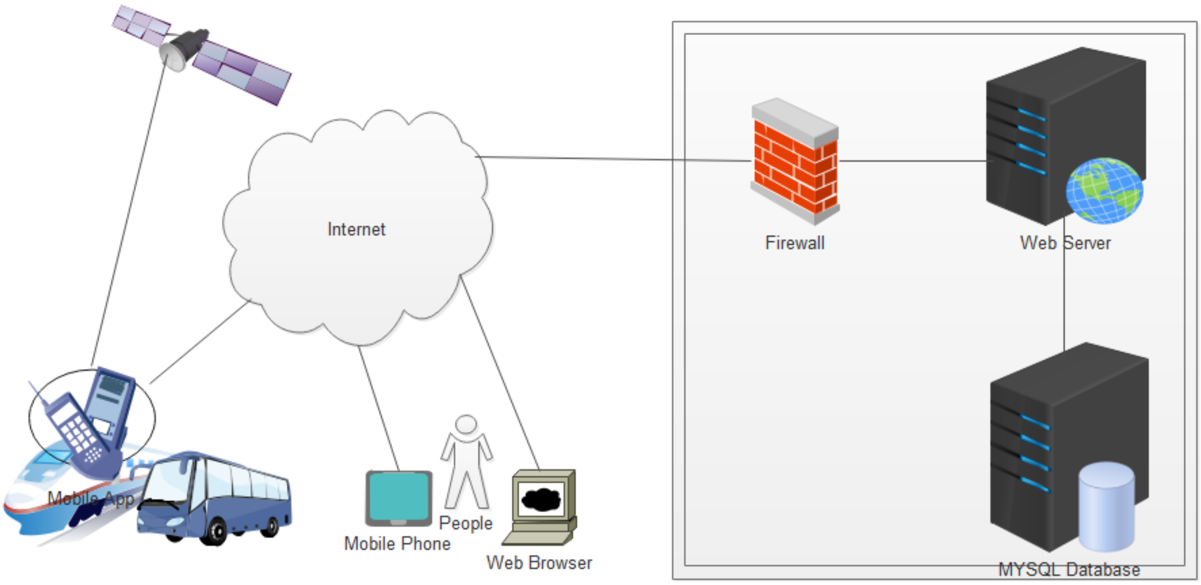


Figure 6.36 – Deployment view of TrackIT

6.8 Data Analysis

Through the TrackIT system, data related to buses and train was collected. These set of data collection grows daily. By analyzing collected data, some patterns would be identified and could be used to generate new knowledge. For the demonstration purposes, data collected separately for selected bus and train for 4 weeks for both.

6.8.1 Train data analysis

TrackIT system is developed to track the live location of a particular train or bus. In addition to live data, prediction mechanism which enable passengers to know expected arrival time, destination time. For this purpose simply time can be calculated using current speed and the distance. Current speed of the particular train, bus is fed into the system through our TrackIT Location updater android application. Distances between stations are in master data in the system. Also it can be calculated using geo coordinates. This simple method may not accurate in every situations since speeds may vary due to various conditions such as traffic, track repairs, number of rail gates etc. Since the system collect geo location data with speed and time, those data can be used to analyze and get some knowledge form those previous data.

During the study of the domain, and through the collected data analysis few hidden factors were identified those could be useful for improving the service. Although large data set should need for a better analysis for pattern recognition, using available data some useful facts were recognized.

Distribution of various speeds of the train is shown in figure 6.37. Since system has speed values, time and geo coordinates, those used to analyze patterns of speeds with the date and time.

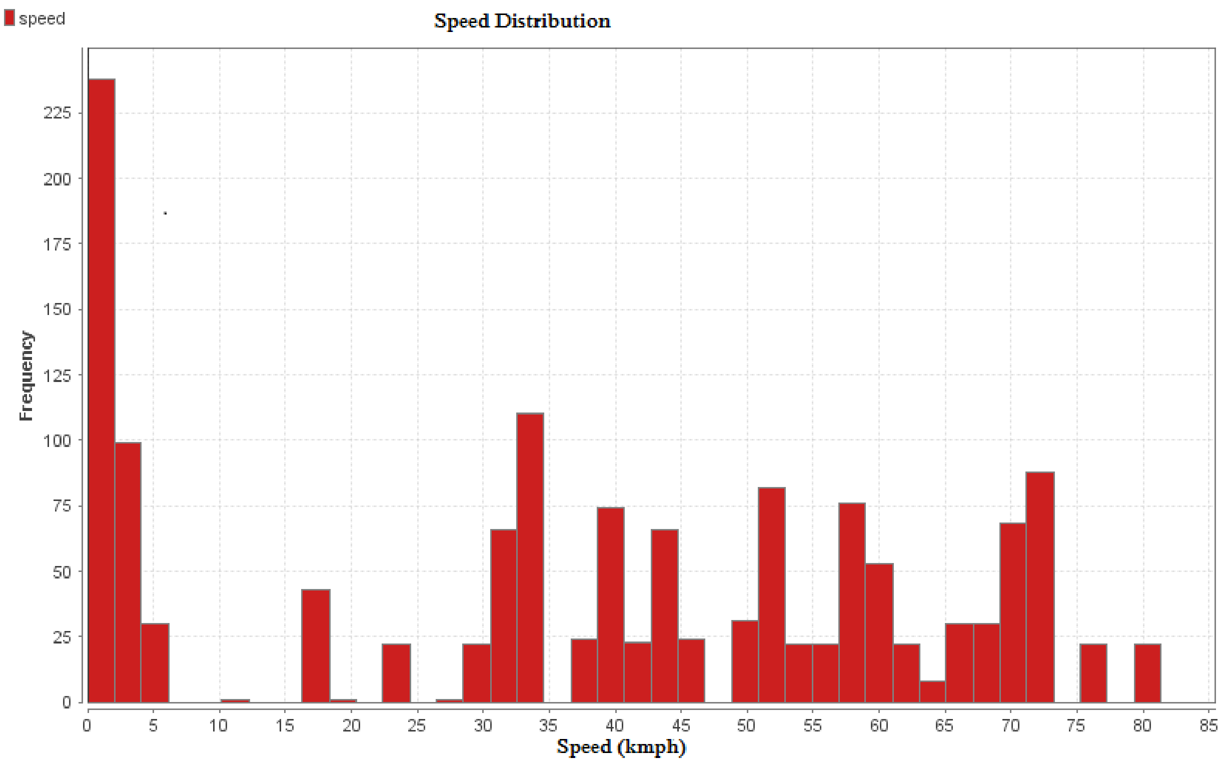


Figure 6.37 - Speed Distribution graph for train

For the considered train, two trips (turns) were considered. Those trips shown different characteristics. Initially average speed calculated by taking the average of instant speeds sent by the android application. Then it could be identified that there was a deviation of predicted time and the actual time taken for the trip. Then using the available data, average speed calculated by travel distance divided by time spent for the trip. Average of above values caused to more accurate value.

Figure 6.38 illustrates averages speeds for a trip.

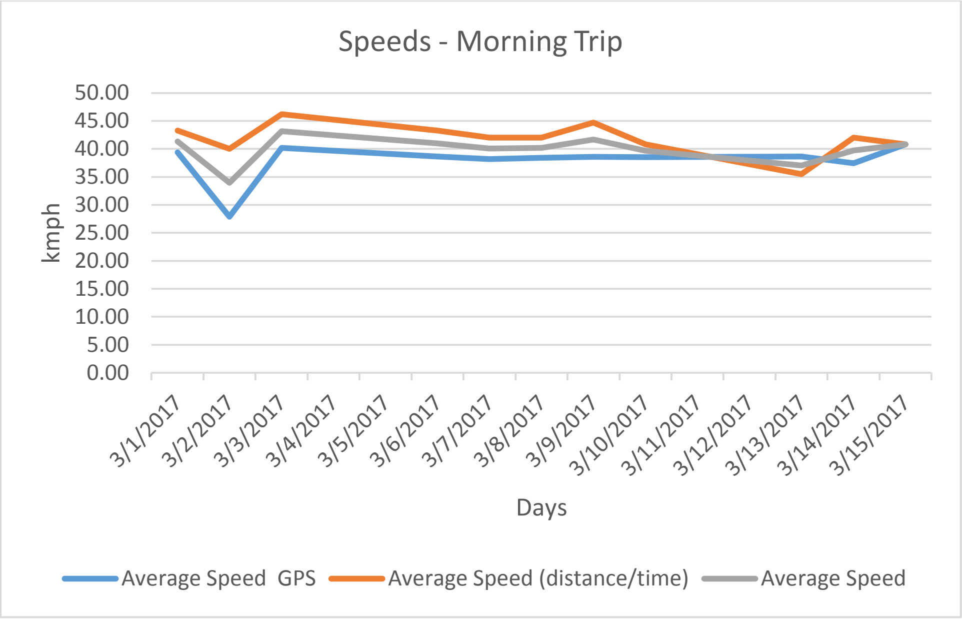


Figure 6.38 – Average speeds – Train Morning Trip

Average speeds were identified for days of the week. Figure 6.39 shows speeds with days of the week.

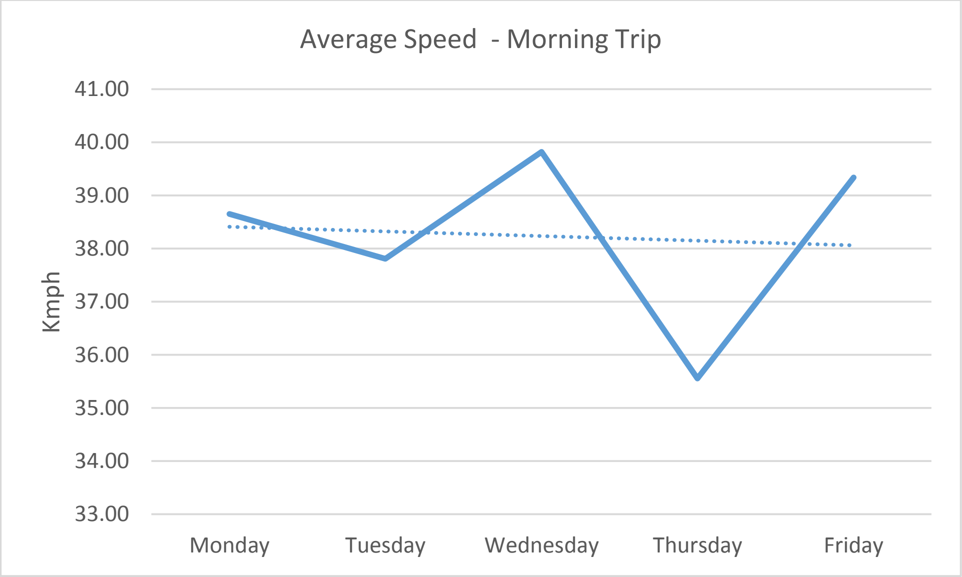


Figure 6.39 – Speed distribution for train for days

Similarly other trips (Evening) also considered and average speeds were identified using collected data. Figure 6.40 shows speeds of evening trip.

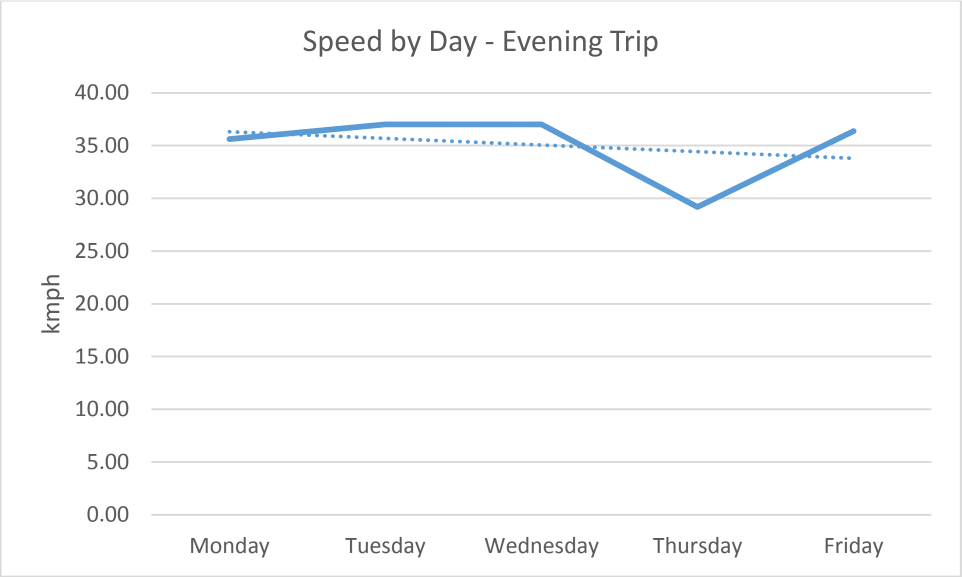


Figure 6.40 – Speeds with days - Evening trip

Those identified information was used for calculate predicted time for a particular query.

Further analysis could be done with big data. Accordingly speeds and delays between stations, relationship of number of railway gates and train delay could be identified. If a train is delay between two stations, it may be due to some damaged/old bridge or some fault on the track. So management decisions also could be taken from the generated knowledge form data taken from the system.

6.8.2 Bus data analysis

Average speeds are calculated by analyzing selected route bus data. Speed distribution is shown in figure 6.41.

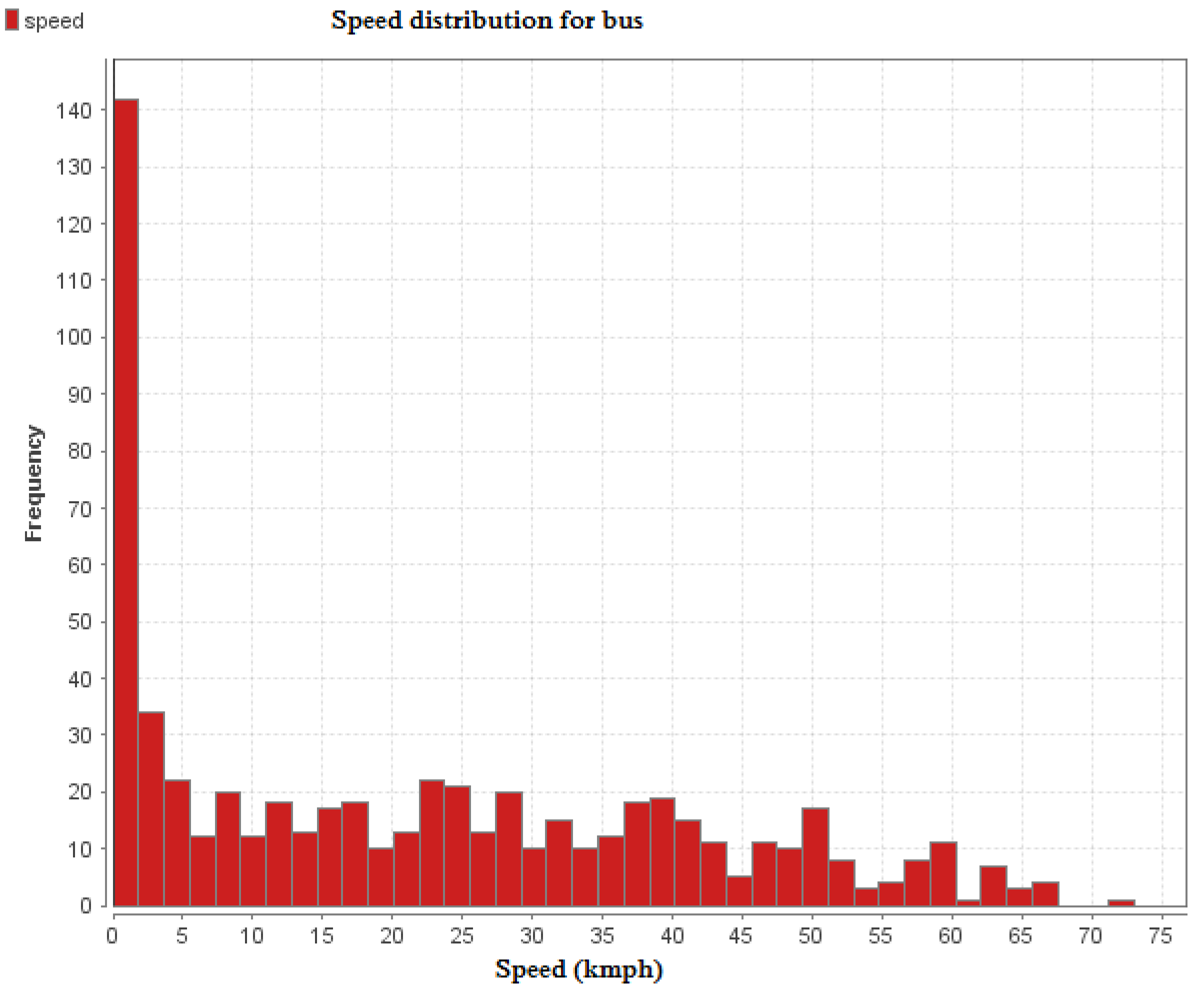


Figure 6.41 – Speed distribution of bus

For the prediction of bus arrival, destination time, previous data could be used in addition to the live information. Through past data analysis, speed variation for days could be identified. Figure 6.42 illustrates speed details for some days and figure 6.43 illustrates particular speeds identified for the days of week.

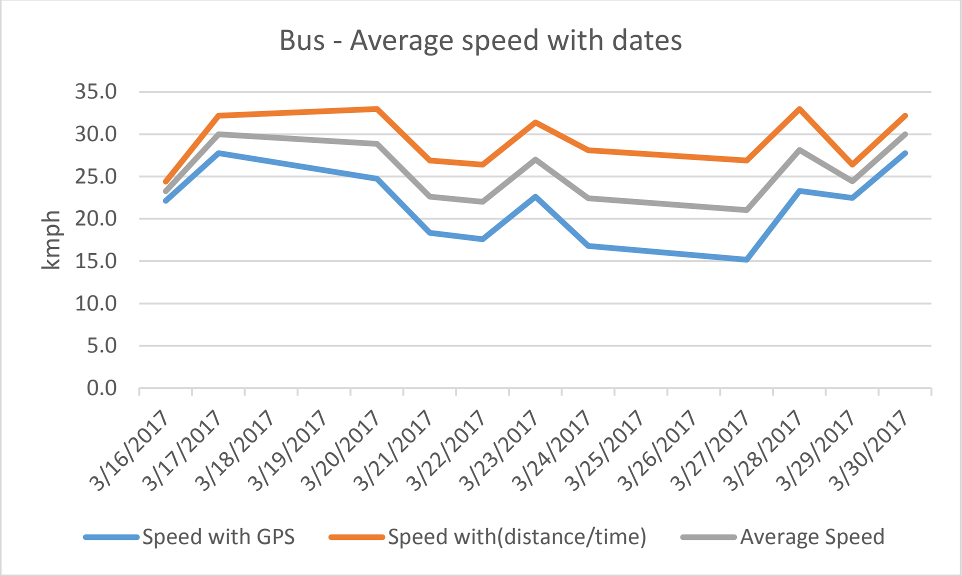


Figure 6.42 – Average speeds of bus

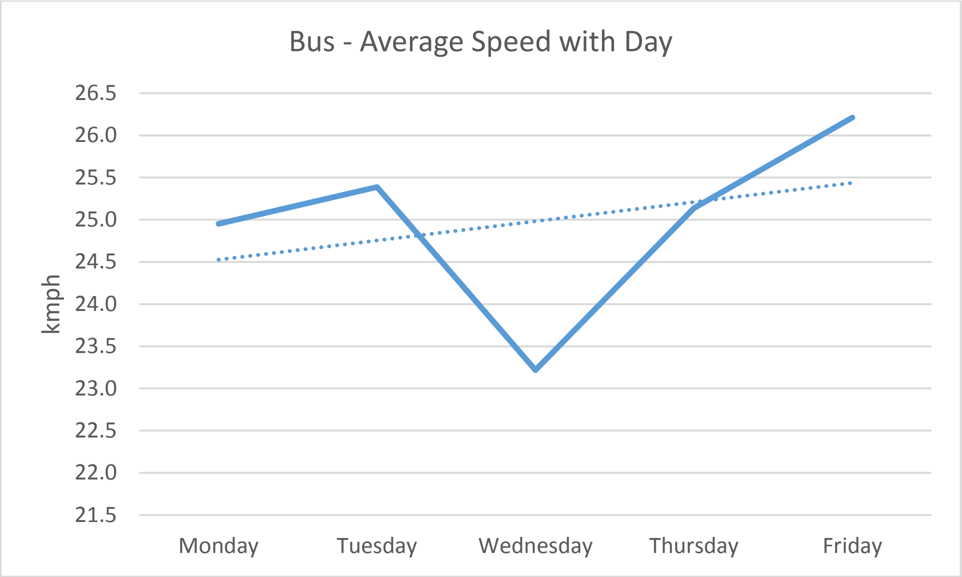


Figure 6.43 – Average speeds of bus for days

In addition to the prediction of arrival, destination time, collected data could be used to identify causes for delays. For example, if particular bus`s speed regularly decrease on a particular area, it may due to bad road condition on that area or dues to many pedestrian crossings in that area. Also speed patterns could be identified with particular driver details.

As a summary, data gained through TrackIT system, could be used to support management decisions in addition to supporting passengers to manage their journey.

6.9 Summary

In this chapter, it was explained in detail how the implementation done for TrackIT, based on the design which was describe in Chapter 5. Data collected through the system, was analyzed and shown that knowledge generated could be used to support management decisions in addition to frontend solution for passengers. In next chapter it will be discussed how this implemented system was evaluated by a selected set of users.

Chapter 7

Evaluation of TrackIT

7.1 Introduction

In the previous chapter the implementation details of the TrackIT system was discussed in detail. The evaluation of the implemented system will be discussed in this chapter.

7.2 Evaluation Methodology

For proper evaluation of the system functionality, system should be deployed in an actual production environment. Therefore the main web application was hosted temporarily on a public domain. Since selected train and a bus chosen for the demonstration purposes, selected set of passengers were requested to give feedback on system using a predefined set of questions in a questionnaire.

For the evaluation of the Location updater android application, it was shown to set of possible users. Then they were allowed to give feedback to a separate predefined set of questions.

The composition of questions in the questionnaire were categorized as bellow.

Main Web Application Evaluation:

In terms of functionality, usability and overall impression five questions for each category were prepared. Since the web application is used for both train and buses, two sets of users were selected separately for trains and busses. Same set of questions were given both set of users to give feedback.

Mobile Application Evaluation:

The location updater mobile application is not for public usage. It is designed to get location updates from authorized users. The users would be Locomotive engineer, Railway guard or bus driver. For the evaluation purpose, the mobile application was shown to set of users and their feedback was taken for a predefined set of questions those cover the usability, functionality and the overall impression about the application.

For each question on system evaluation features, following evaluation factors were assigned with weighted marks assigned to each factor.

1 - Very poor (1), 2 – Poor (3), 3 – Average (6), 4 - Good (8), 5 - Excellent (10)

A decision should be made through a statistical method by analyzing the feedback received. For this purpose, marks calculated for each question and average is taken for the considered features separately. The calculated averages were checked against defined critical line.

Number of users who has given the feedback - N
Number of questions on an evaluation feature – Q
Total marks per question = M
Number of questions marked as Very poor – VP
Number of questions marked as Poor – P
Number of questions marked as Average - A
Number of questions marked as Good - G
Number of questions marked as Excellent - E
Marks per evaluation item= (E*10 + G*8 + A * 6 + P * 3 + VP *1)/N

Average marks per evaluation feature
$\sum_{i=0}^{Q} (\mathrm{E} * 10 + \mathrm{G} * 8 + \mathrm{A} * 6 + \mathrm{P} * 3 + \mathrm{VP} * 1) /$ Q*M

The critical line = 40%

7.3 Evaluation Forms

Please refer Appendix J for evaluation forms.

7.4 Final Evaluation Results

Main Web Application:

Number of users involved in giving feedback: 10

Evaluation results for main web application for train passengers shown in table 7.13

Feature	**Percentage**
Usability	69.8
Functionality	75.8
Overall Impression	79.2
Average	74.9

Table 7.13 – Main web application evaluation results - Trains

Evaluation results for main web application for bus passengers shown in table 7.14

Feature	**Percentage**
Usability	67.8
Functionality	72.6
Overall Impression	77.6
Average	72.7

Table 7.14 – Main web application evaluation results - Buses

Mobile Application:

Number of users involved in giving feedback: 10

Evaluation results for mobile application shown in table 7.15

Feature	Percentage
Usability	60
Functionality	62.4
Overall Impression	60.6

Average	61.0

Table 7.15 – Mobile application evacuation results

Overall evaluation results are illustrated in figure 7.1.

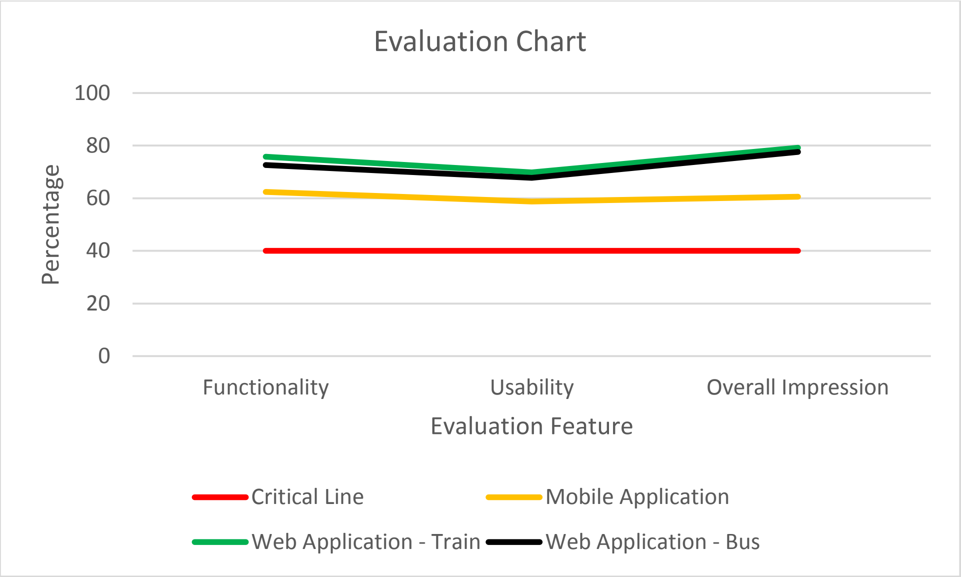


Figure 7.1 – Evaluation Chart

According to the evaluation chart given here, in Figure 7.1, all the evaluation lines are above the defined critical line and therefore system has approached its target evaluation level. The web application is gained high values in evaluation while the mobile application gained relatively slight drop. Reason may be for the highly acceptable, its functionalities and high usability for the passengers since such a system was not available previously. Reason for slight drop of mobile application may be due to location updaters may not willing to use the application since they may pay their attention to start and stop it. Also they may think that it may harm for them by knowing the actual location by other competitive buses.

According to the overall evaluation result, the system is successful and it has been approved by the users with high level of acceptability.

7.5 Summary

According to the evaluation done in this chapter, the system has maintained its evaluation above the critical line in all evaluation features for both main web application and the mobile application. In next chapter, the conclusion and the further possible enhancements for TrackIT system will be discussed.

Chapter 8

Conclusion and further work

8.1 Introduction

This chapter concludes the research presented in this thesis. We recall our hypothesis as simple and cost effective location tracking system can be developed using GPS, Android and open source web technologies. It is evident from the evaluation stated in chapter 7, that the acceptance level of the new system is above 70%. This chapter present our overall conclusion, objective-wise achievements and suggested further work.

8.2 Overall conclusion

The evaluation evident that the simple and cost effective location tracking system developed to address problems in passengers` perspective, using GPS, Android and open source web technologies, is in above 70 % acceptance level. Selected route, limited historical data, nature of participants could have affected to the overall result below 100%. Accordingly our hypothesis is proven to be true.

8.3 Objective-wise conclusions

We recall the objectives:

(i) To study the current issues related to location monitoring in public transportation

(ii) To Critically review the available methods for location tracking

(iii) To do an in depth study of location tracking technologies

(iv) To develop a new system for location tracking

(v) To provide a solution to predict time for arrival/destination

(vi) To allow third parties to consume location tracking data

(vii) To evaluate the performance of the system

Objective (i) has been achieved by conducting a comprehensive literature survey comprising more than 20 research papers related to location tracking systems. Here we have discovered many key problems and defined the research problem of this thesis as simple and cost effective location tracking system can be developed using GPS, android and open source web technologies.

The achievement of objective (ii) has also been supported by the literature review chapter. In addition, chapter 3 presented a description on each technology selected for developing the proposed solution which fulfill the achievement of objective (iii).

Achievements in objectives (iv), (v) and (vi) are evident form the details in chapters for approach, design and implementation.

The objective related to the evaluation of the hypothesis is presented in chapter 7. It was a systematic testing process with selected set of users who use public transportation daily. The overall success of the solution has been 70%.

8.4 Discussion

The party whom are mainly benefited by this TrackIT system is the passengers. With currently available systems, they are only capable to see static schedules. Also currently there is no proper single system to get both train details and bus details. This system facilitates to view both train and bus schedule details.

This new system will provide following information to passengers.

- Availability of particular train or bus
- Current position of selected train or bus

- Facility to search train or bus
- Predicted time for arrival/destination based on both historical data and current live data

Features provided by the new system, allow passengers to reduce the waiting time at the stations or bus stands. Also they can select an alternative transportation method in time if desired train or bus is not available for desired time. This leads to save productive man hours for the development of the country.

Since the system provide a web service with live location information, it would be used to display train, bus schedule and live details on large screens in railway stations and bus stands. Then it facilitate everyone to get the benefit from the system.

Although system is designed to mainly address the problems in the passenger's perspective, it gives benefits to third party developers, private users and Sri Lanka Railway (SLR) and Sri Lanka Transport Board management (SLTB).

Third party service providers/developers can use web service provided by this new system to get live location update data. It enable them to develop own applications by consuming the data feeds provided by this system. Also system can be used by people to track their own vehicles by registering in the system. This enable them to generate reports on past data of their vehicle movements.

Authorized users would be able to analyze large set of daily collected large set of data and identify patterns of transportation such as daily speed drops, delay time etc. Such identified information would be used to make management decisions which cause to provide better service for passengers. For example, daily speed drop in a particular area may be due to a fault on the track or bridge. Since system has all details, management can take a decision to repair the fault in the particular place. Especially this data would be utilized by the Sri Lanka Railway to identify unidentified causes for train late. Similarly pattern of bus transportation would be used by Sri Lanka Transport Board and Road Development Authority to make decisions on time schedules and road maintenance.

The location updater mobile application would require GPS or Android`s Location Provider activated. Also the application would require an active internet connection to send geo location data to the central server. These could be considered as limitations of the system, since some users would not be agreed with these terms. Thus the main risk for the system would be the availability and the inaccuracy of the data received from the location updater through the mobile application. The inaccuracy could be reduced through validation, but the availability of input data would be depend on the location updaters , Locomotive engineer, Main railway guard or bus driver. They have to enable the mobile application to tracking to be initialized. To avoid the reluctance to use the mobile application, some mechanism to encourage them to use it would be officially introduced through the SLR and SLTB.

The core of the system is the main web application which require high performance server to handle expected large amount of requests from general public. At the same time periodic location updates from number of trains and buses should be handled and saved. For this purpose necessary security enhancements such as WAF (Web Application Firewall) would be implemented.

8.5 Limitations

Poor mobile network coverage, delays, mistakes in enabling the mobile application have given negative impact on real time location updates.

8.6 Further Work

Since the TrackIT system collect data daily, a separate data analyzing module would be developed or integrated to the system to enabling authorized users from Sri Lanka Railway or Transport Board to generate reports based on the past set of data. System could be further enhanced by facilitating to enter locomotive engineer, driver information, and number of railway gates, and speed limits for specific locations to be entered to the system. This enable many dimensions to analyze train delays with available data.

The location updater mobile application also would be further enhanced by automating its functionality of capturing the turn number, starting and stopping the tracking process. Also it would be integrated to a weather information system to collect weather information together with real time location updates which facilitate the study of delays with respect to the weather conditions.

Since the system already facilitate a web service, it can be used in railway stations or bus stands to display real time locations and schedule on large screens. This would be more beneficial to all the passengers.

8.7 Summery

In this chapter the overview of TrackIT project was discussed along with the evaluation results and identified limitations. Further possible enhancements were also discussed together with possible practical applications.

References

[1] D. A. Aaker, *Building strong brands*. Simon and Schuster, 2012.

[2] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *J. Comput. Commun.*, vol. 03, no. 05, pp. 164–173, 2015.

[3] Y. B. Bai, S. Wu, H. R. Wu, and K. Zhang, “Overview of RFID-Based Indoor Positioning Technology.,” in *GSR*, 2012.

[4] S. L. Ting, L. X. Wang, and W. H. Ip, “A study on RFID adoption for vehicle tracking in container terminal,” *J. Ind. Eng. Manag.*, vol. 5, no. 1, Jun. 2012.

[5] Y. Ning, W. Zhong-qin, R. Malekian, W. Ru-chuan, and A. H. Abdullah, “Design of Accurate Vehicle Location System Using RFID,” *Electron. Electr. Eng.*, vol. 19, no. 8, Oct. 2013.

[6] P. Hannay, “Satellite navigation forensics techniques,” 2009.

[7] A. Dhumal, A. Naikoji, Y. Patwa, M. Shilimkar, and M. K. Nighot, “Survey Paper on Vehicle Tracking System using GPS and Android.”

[8] G. Li, D. Zhang, J. Zeng, and S. Chen, “Vehicle Monitor System for Public Transport Management Based on Embedded Technology,” *Phys. Procedia*, vol. 24, pp. 953–960, 2012.

[9] A. R. Gutte and H. K. Bhangale, “School Bus Environment Monitoring by Advance GSM and DGPS Technology,” 2016.

[10] P. Verma and J . Bhatia, “Design and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring,” *Int. J. Comput. Sci. Eng. Appl.*, vol. 3, no. 3, pp. 33–40, Jun. 2013.

[11] A. Haleem, S. Lebbe, and S. S. Nawaz, “Real time bus tracking and scheduling system using wireless sensor and mobile technology,” 2016.

[12] M. Medagama, D. Gamage, L. Wijesinghe, N. Leelaratna, I. Karunaratne, and D. Dias, “GIS/GPS/GPRS and Web–based Framework for Fleet Tracking,” in *National Conference on Geoinformatics Applications Sri Lanka*, 2008.

[13] I. SLR, “eService- Sri Lanka Railway,” *eService- Sri Lanka Railway*. [Online]. Available: www.eservices.railway.gov.lk/schedule.

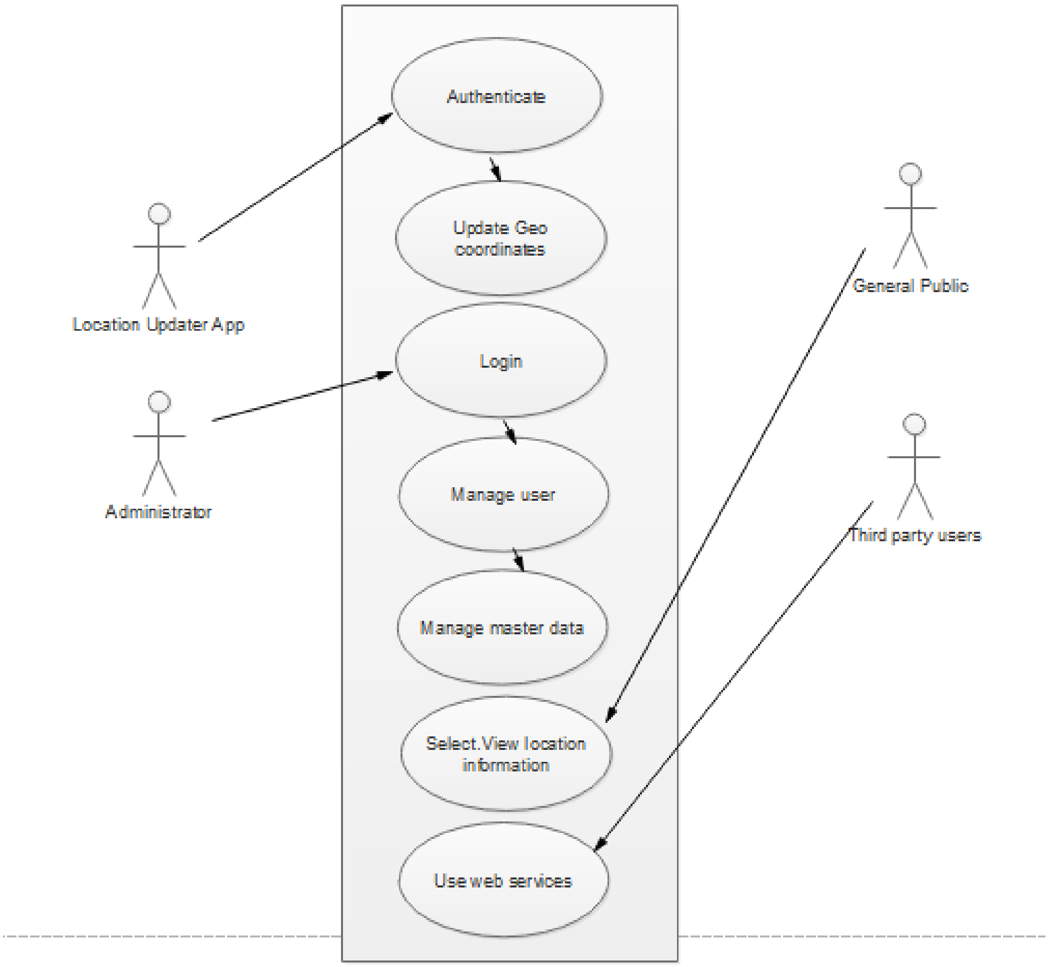
[14] sundaytimes, “Railway Traffic Optimisation System (RTOS),” 2014.

[15] D. N. H. Senevirathna, “Community Based Train Locating System (CBTLS),” in *Community Based Train Locating System (CBTLS)*, 2015.

[16] C. Jechlitschek, “A survey paper on Radio Frequency Identification (RFID) trends,” *FileFwwwcse574-06ftprfidindex Htm*, 2006.

Appendixes

Appendix A – Use Case Diagrams



Use Case Diagrams of the TrackIT System

Figure 5.3 - Use case diagram for TrackIT system

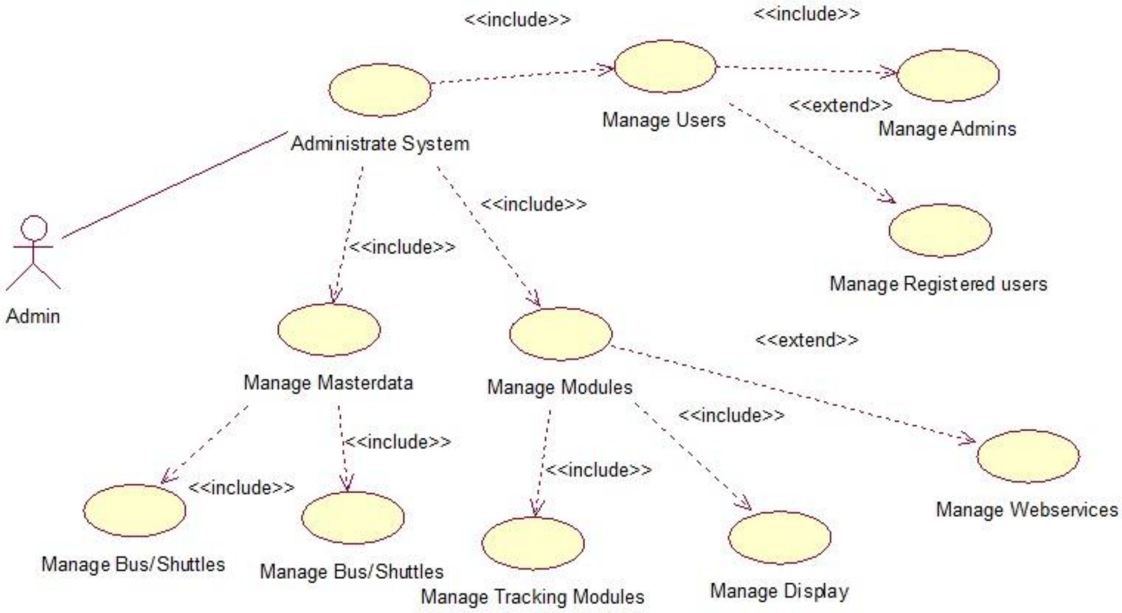


Figure 5.4 – Admin use case

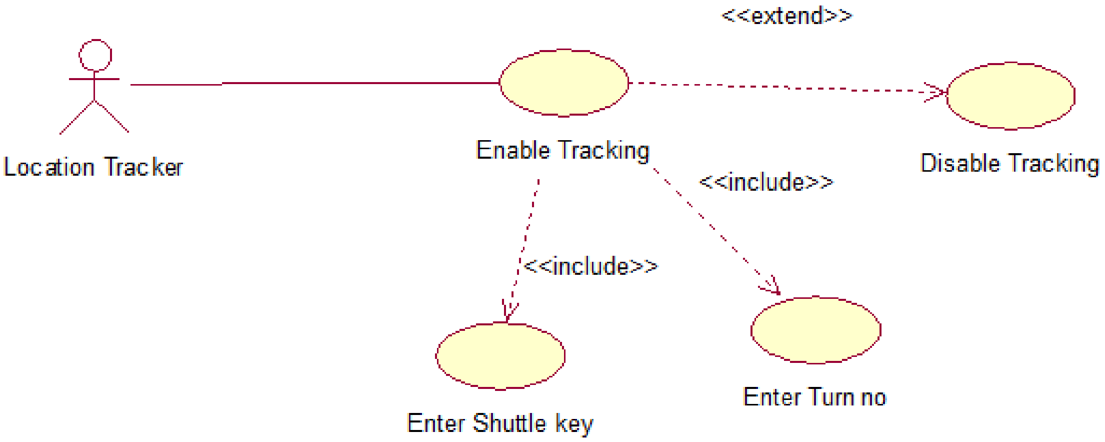


Figure 5.5 – Location Tracker

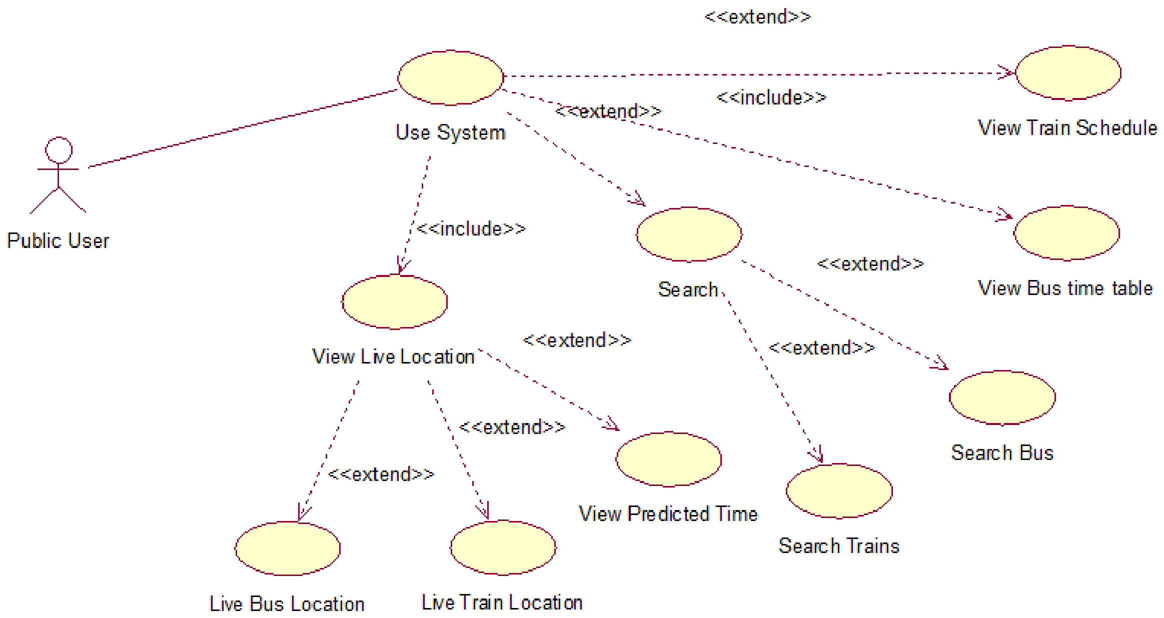


Figure 5.6 – Public user/passenger

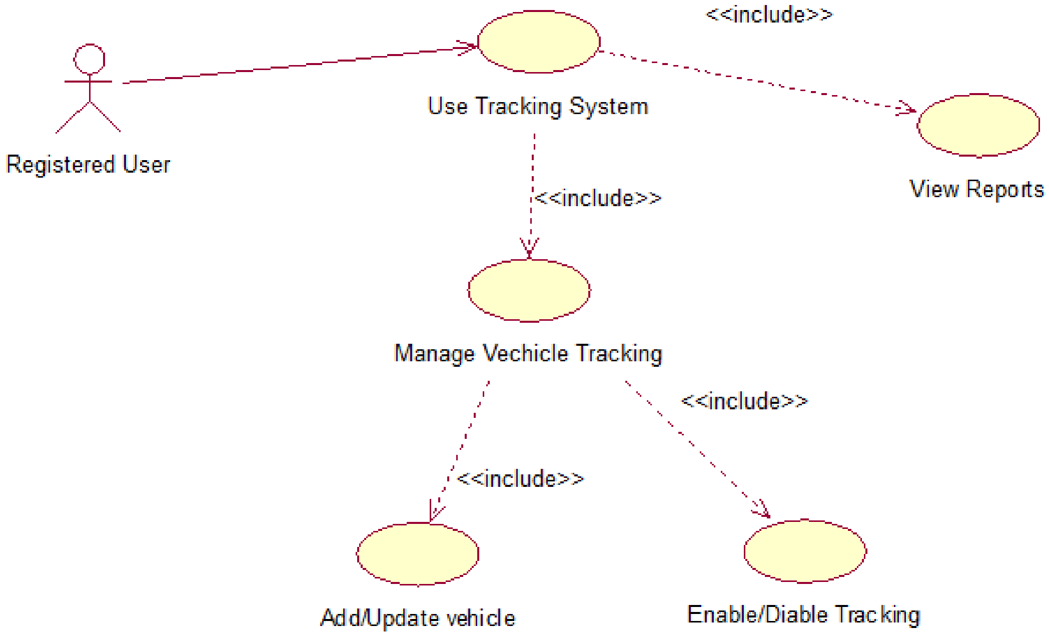


Figure 5.7 – Registered user/Vehicle owner

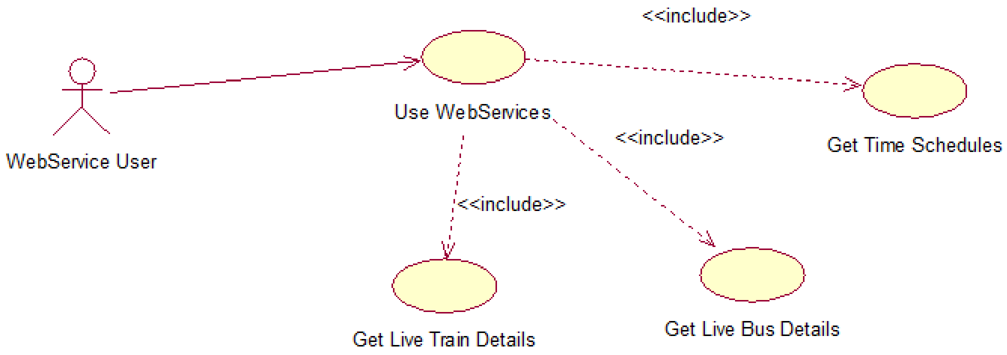


Figure 5.8 – Web service

Appendix B – Sequence Diagrams

Sequence Diagrams of the system

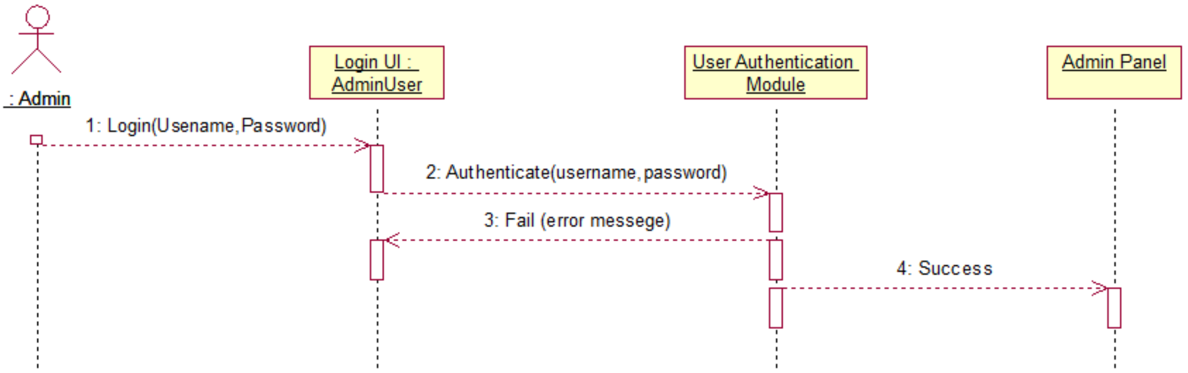

: Admin
Login UI : AdminUser
User Authentication Module
Admin Panel
1: Login(Usename,Password)
2: Authenticate(username, password)
3: Fail (error messege)
4: Success

Figure 5.9 – Sequence diagram for Admin Login use case

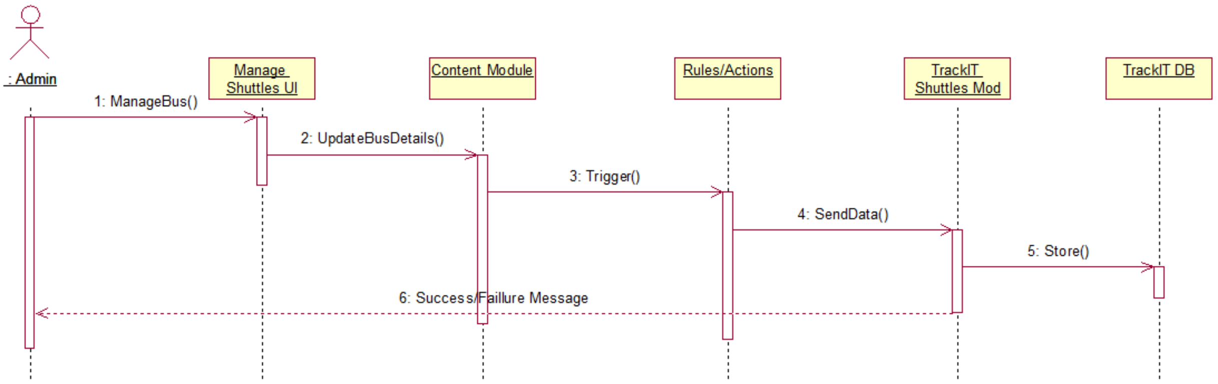


Figure 5.10 – Sequence diagram for manage buses use case

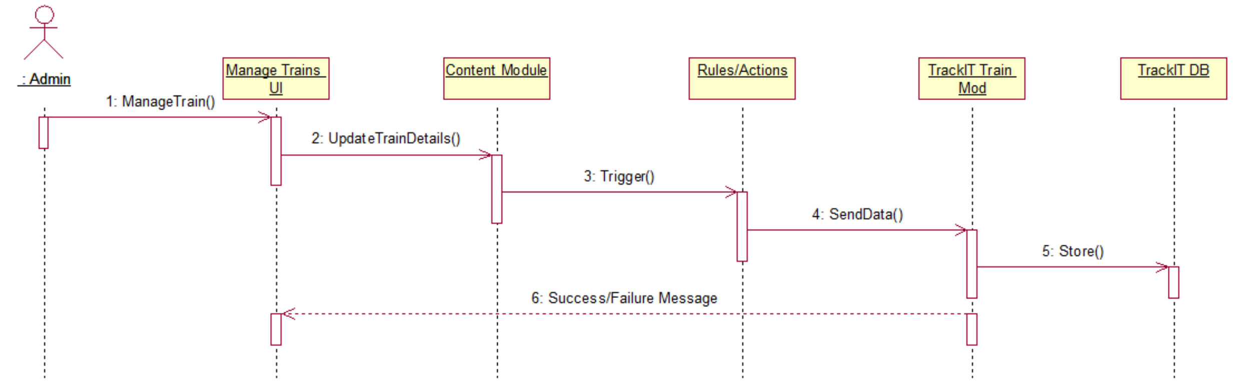


Figure 5.11 – Sequence diagram for manage trains use case

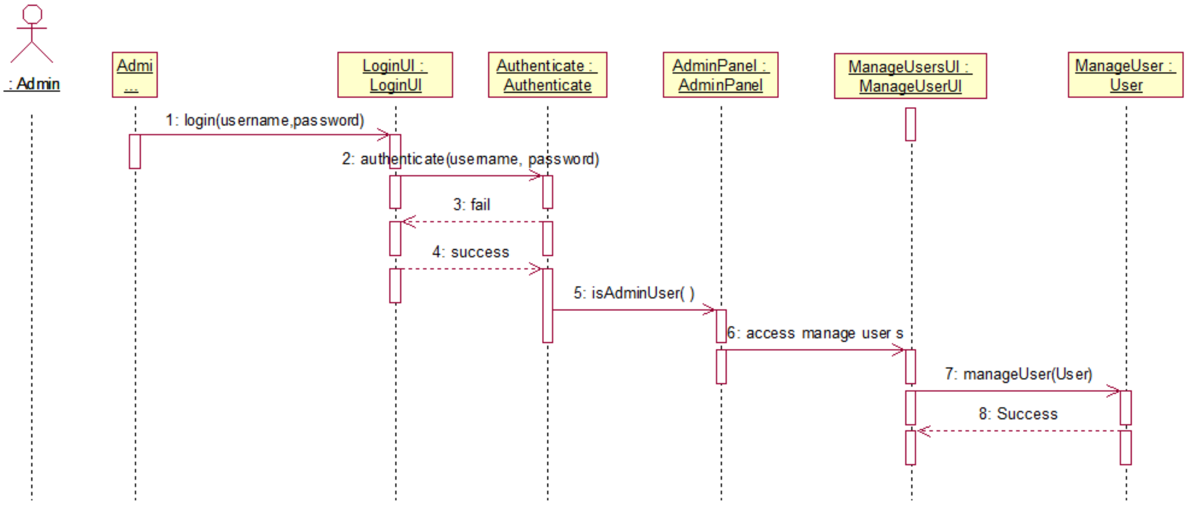


Figure 5.12 – Sequence diagram for manage users use case

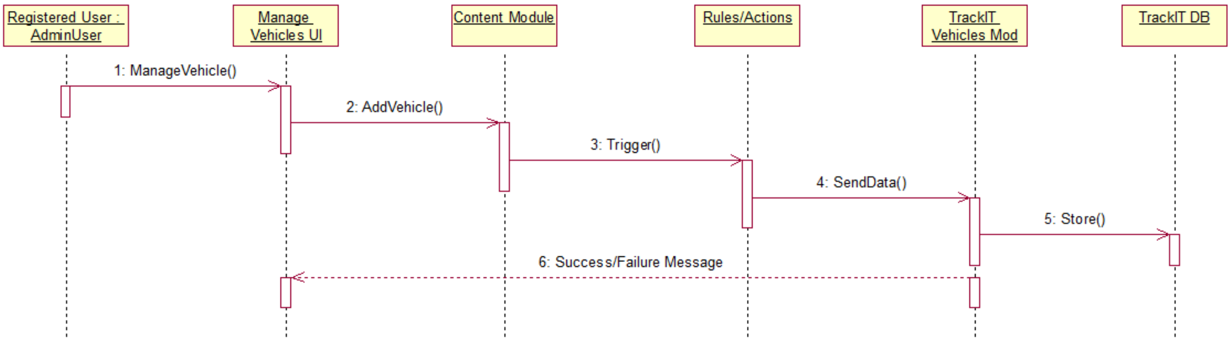


Figure 5.13 – Sequence diagram for manage vehicles use case

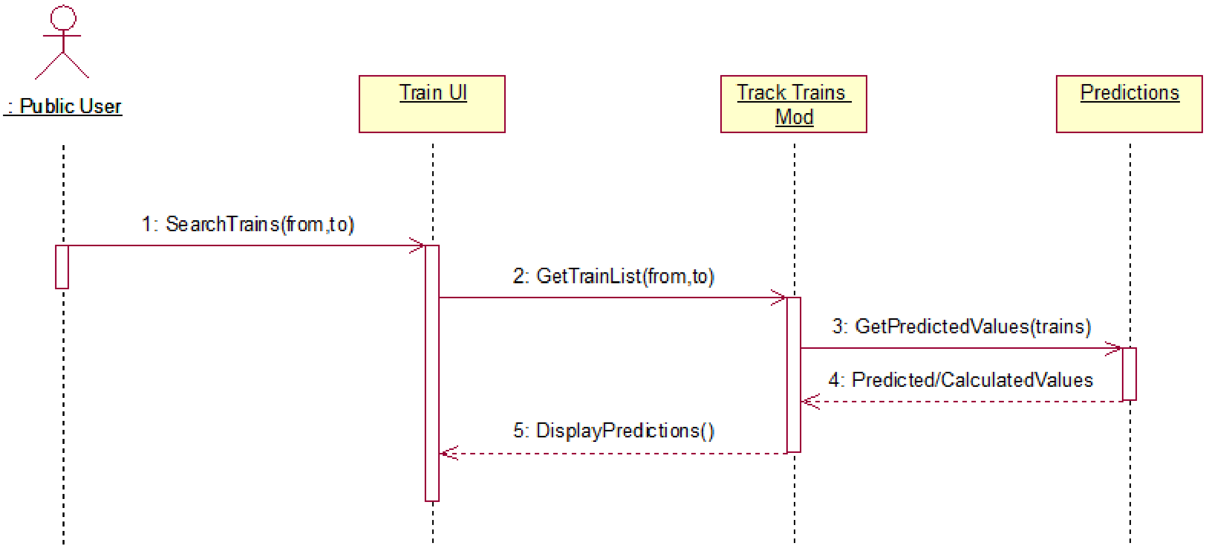


Figure 5.14 – Sequence diagram for search train use case

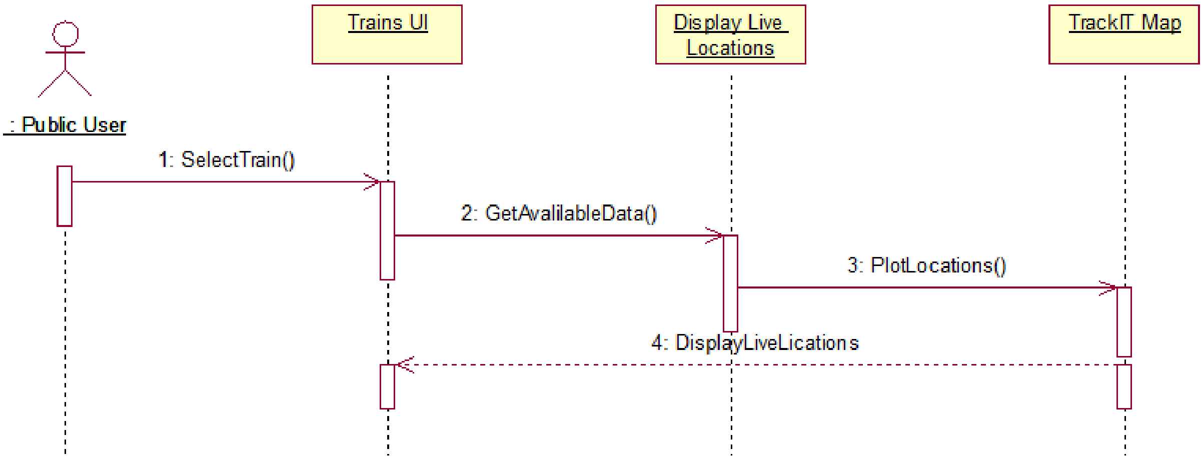


Figure 5.15 – Sequence diagram for view live train location use case

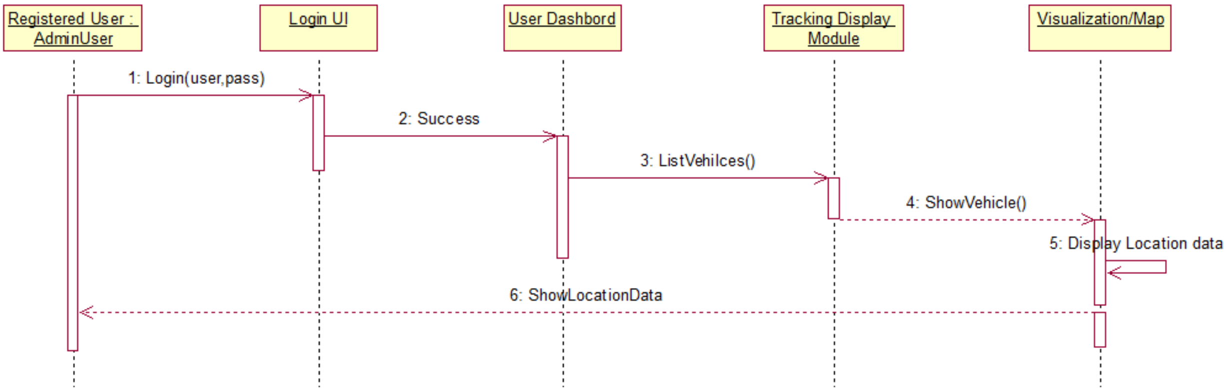


Figure 5.16 – Sequence diagram for view vehicle tracking use case

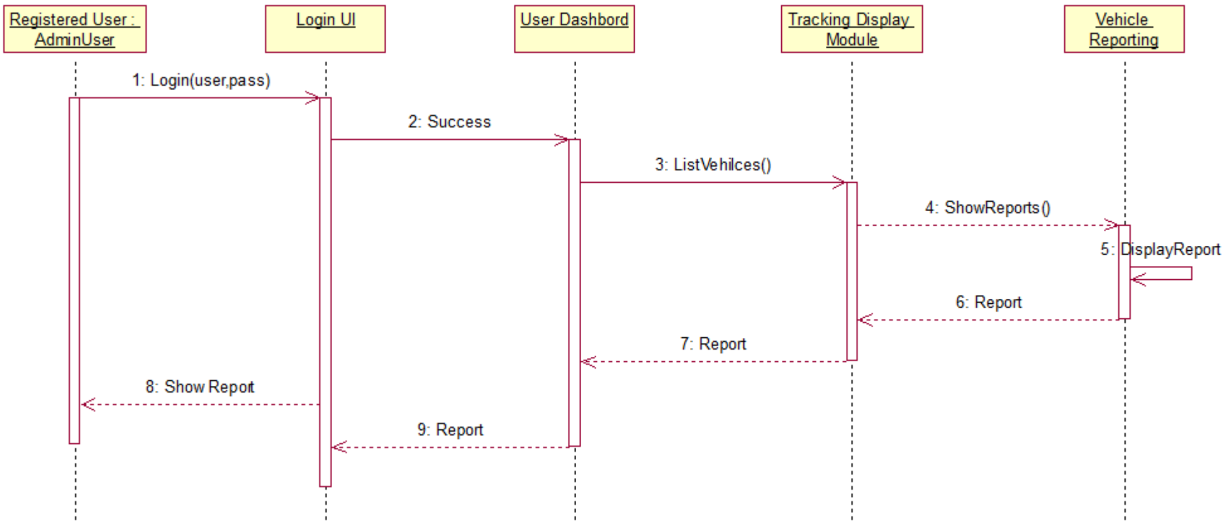


Figure 5.17 – Sequence diagram for view vehicle reports use case

Appendix C – Database Diagrams

Database Diagrams of the system

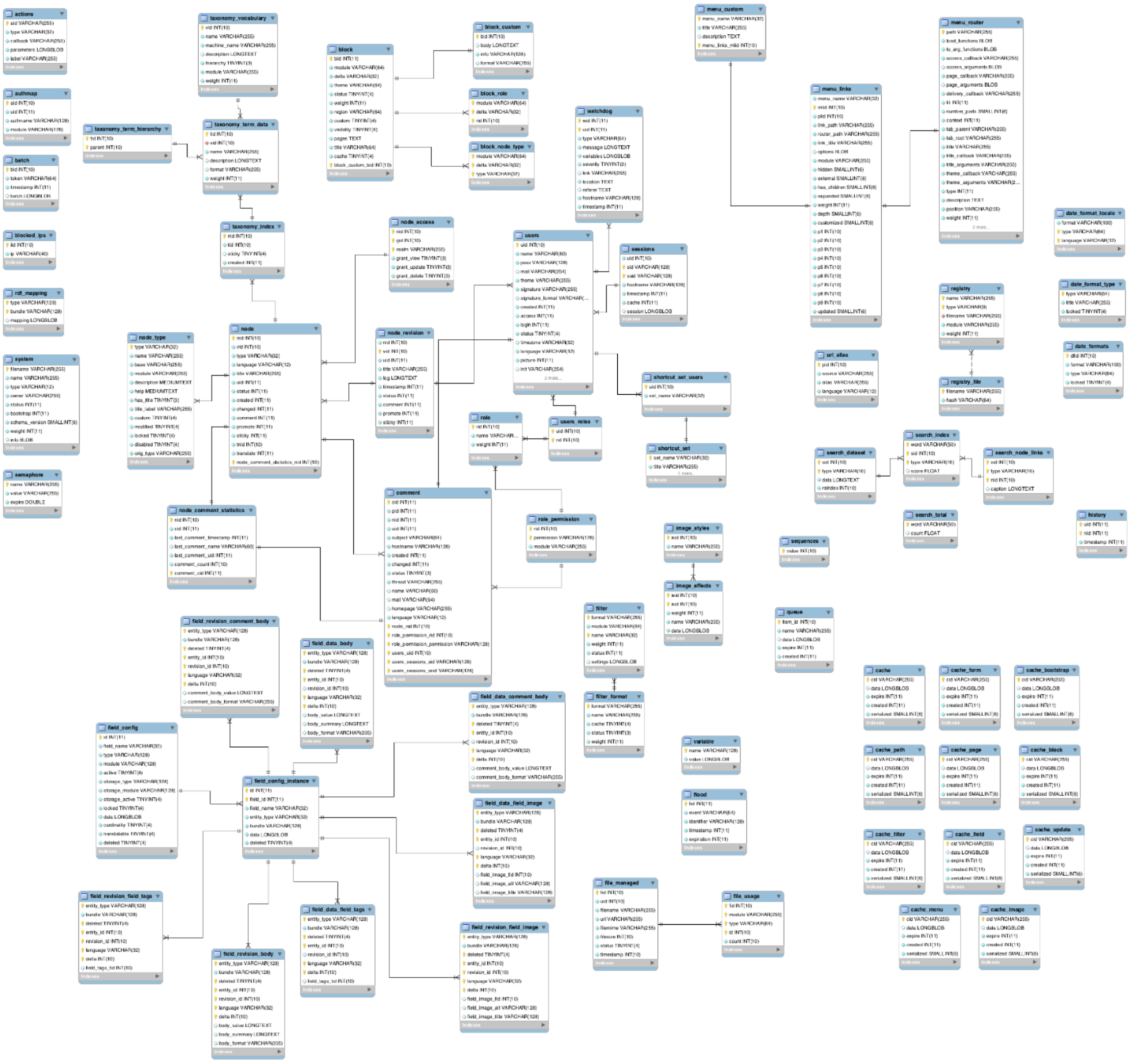
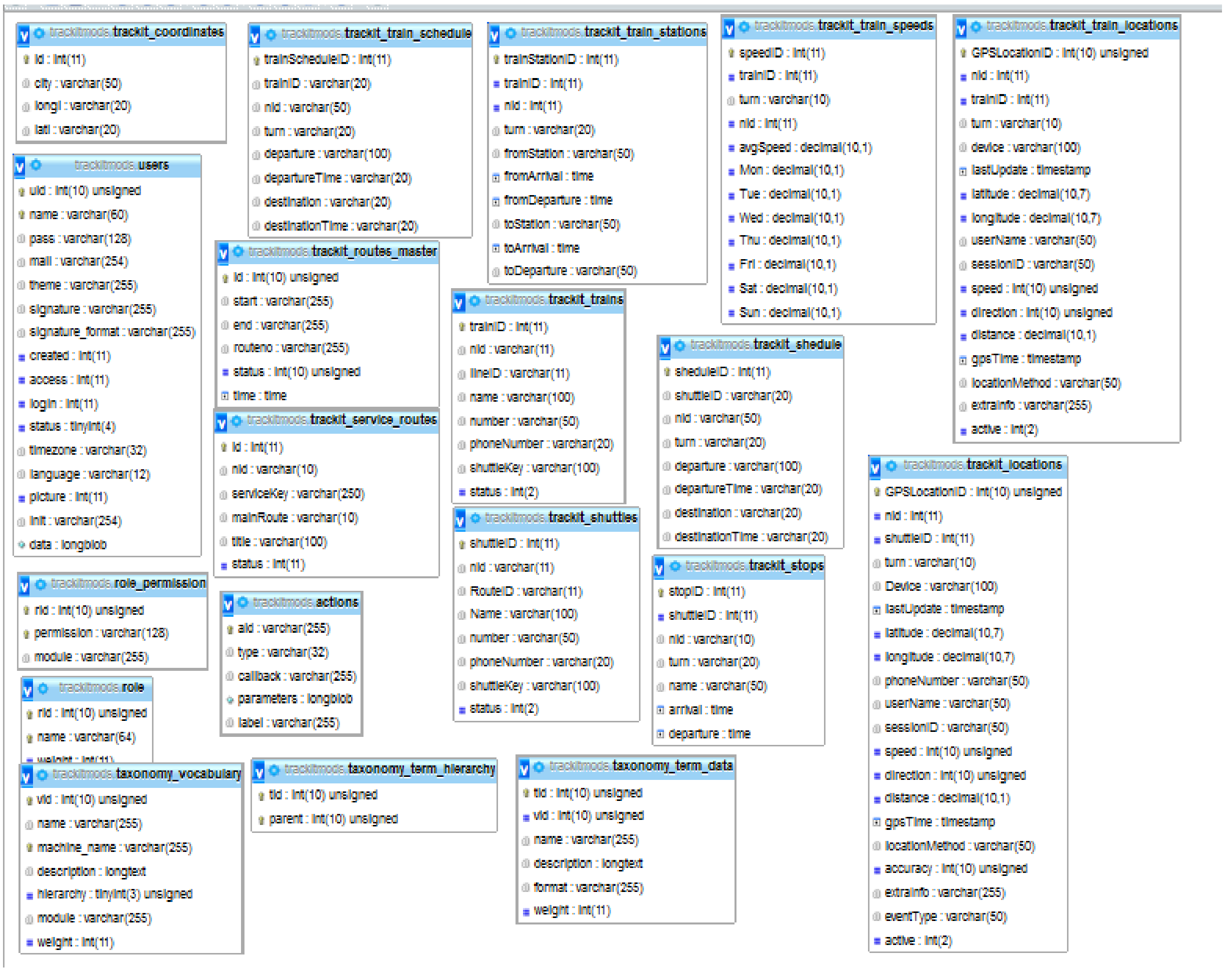
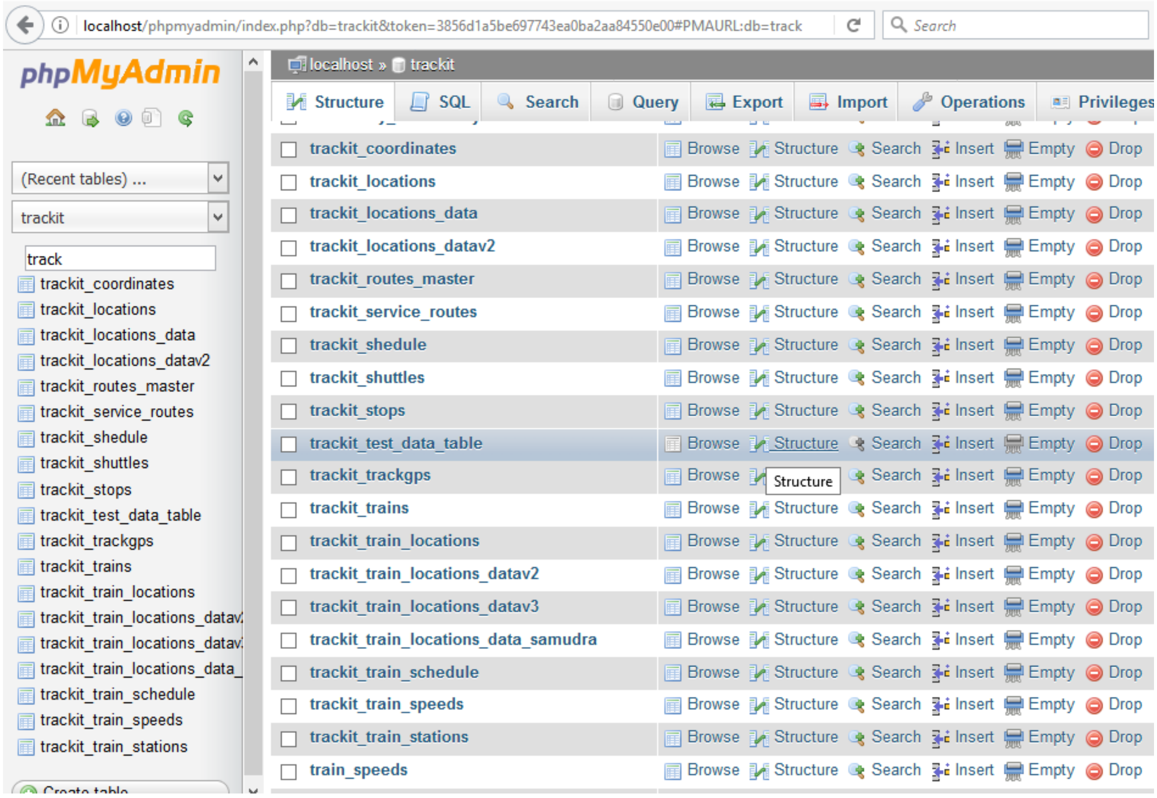


Figure 5.18 – Database diagram of the core system



```
trackitmods.trackit_coordinates
id : int(11)
city : varchar(50)
longi : varchar(20)
lati : varchar(20)

trackitmods.users
uid : int(10) unsigned
name : varchar(60)
pass : varchar(128)
mail : varchar(254)
theme : varchar(255)
signature : varchar(255)
signature_format : varchar(255)
created : int(11)
access : int(11)
login : int(11)
status : tinyint(4)
timezone : varchar(32)
language : varchar(12)
picture : int(11)
init : varchar(254)
data : longblob

trackitmods.role_permission
rid : int(10) unsigned
permission : varchar(128)
module : varchar(255)

trackitmods.role
rid : int(10) unsigned
name : varchar(64)
weight : int(11)

trackitmods.taxonomy_vocabulary
vid : int(10) unsigned
name : varchar(255)
machine_name : varchar(255)
description : longtext
hierarchy : tinyint(3) unsigned
module : varchar(255)
weight : int(11)

trackitmods.trackit_train_schedule
trainScheduleID : int(11)
trainID : varchar(20)
nid : varchar(50)
turn : varchar(20)
departure : varchar(100)
departureTime : varchar(20)
destination : varchar(20)
destinationTime : varchar(20)

trackitmods.trackit_routes_master
id : int(10) unsigned
start : varchar(255)
end : varchar(255)
routeno : varchar(255)
status : int(10) unsigned
time : time

trackitmods.trackit_service_routes
id : int(11)
nid : varchar(10)
serviceKey : varchar(250)
mainRoute : varchar(10)
title : varchar(100)
status : int(11)

trackitmods.actions
aid : varchar(255)
type : varchar(32)
callback : varchar(255)
parameters : longblob
label : varchar(255)

trackitmods.taxonomy_term_hierarchy
tid : int(10) unsigned
parent : int(10) unsigned

trackitmods.trackit_train_stations
trainStationID : int(11)
trainID : int(11)
nid : int(11)
turn : varchar(20)
fromStation : varchar(50)
fromArrival : time
fromDeparture : time
toStation : varchar(50)
toArrival : time
toDeparture : varchar(50)

trackitmods.trackit_trains
trainID : int(11)
nid : varchar(11)
lineID : varchar(11)
name : varchar(100)
number : varchar(50)
phoneNumber : varchar(20)
shuttleKey : varchar(100)
status : int(2)

trackitmods.trackit_shuttles
shuttleID : int(11)
nid : varchar(11)
RouteID : varchar(11)
Name : varchar(100)
number : varchar(50)
phoneNumber : varchar(20)
shuttleKey : varchar(100)
status : int(2)

trackitmods.taxonomy_term_data
tid : int(10) unsigned
vid : int(10) unsigned
name : varchar(255)
description : longtext
format : varchar(255)
weight : int(11)

trackitmods.trackit_train_speeds
speedID : int(11)
trainID : int(11)
turn : varchar(10)
nid : int(11)
avgSpeed : decimal(10,1)
Mon : decimal(10,1)
Tue : decimal(10,1)
Wed : decimal(10,1)
Thu : decimal(10,1)
Fri : decimal(10,1)
Sat : decimal(10,1)
Sun : decimal(10,1)

trackitmods.trackit_shedule
sheduleID : int(11)
shuttleID : varchar(20)
nid : varchar(50)
turn : varchar(20)
departure : varchar(100)
departureTime : varchar(20)
destination : varchar(20)
destinationTime : varchar(20)

trackitmods.trackit_stops
stopID : int(11)
shuttleID : int(11)
nid : varchar(10)
turn : varchar(20)
name : varchar(50)
arrival : time
departure : time

trackitmods.trackit_train_locations
GPSLocationID : int(10) unsigned
nid : int(11)
trainID : int(11)
turn : varchar(10)
device : varchar(100)
lastUpdate : timestamp
latitude : decimal(10,7)
longitude : decimal(10,7)
userName : varchar(50)
sessionID : varchar(50)
speed : int(10) unsigned
direction : int(10) unsigned
distance : decimal(10,1)
gpsTime : timestamp
locationMethod : varchar(50)
extraInfo : varchar(255)
active : int(2)

trackitmods.trackit_locations
GPSLocationID : int(10) unsigned
nid : int(11)
shuttleID : int(11)
turn : varchar(10)
Device : varchar(100)
lastUpdate : timestamp
latitude : decimal(10,7)
longitude : decimal(10,7)
phoneNumber : varchar(50)
userName : varchar(50)
sessionID : varchar(50)
speed : int(10) unsigned
direction : int(10) unsigned
distance : decimal(10,1)
gpsTime : timestamp
locationMethod : varchar(50)
accuracy : int(10) unsigned
extraInfo : varchar(255)
eventType : varchar(50)
active : int(2)
```

Figure 5.19 – Table structures of custom modules



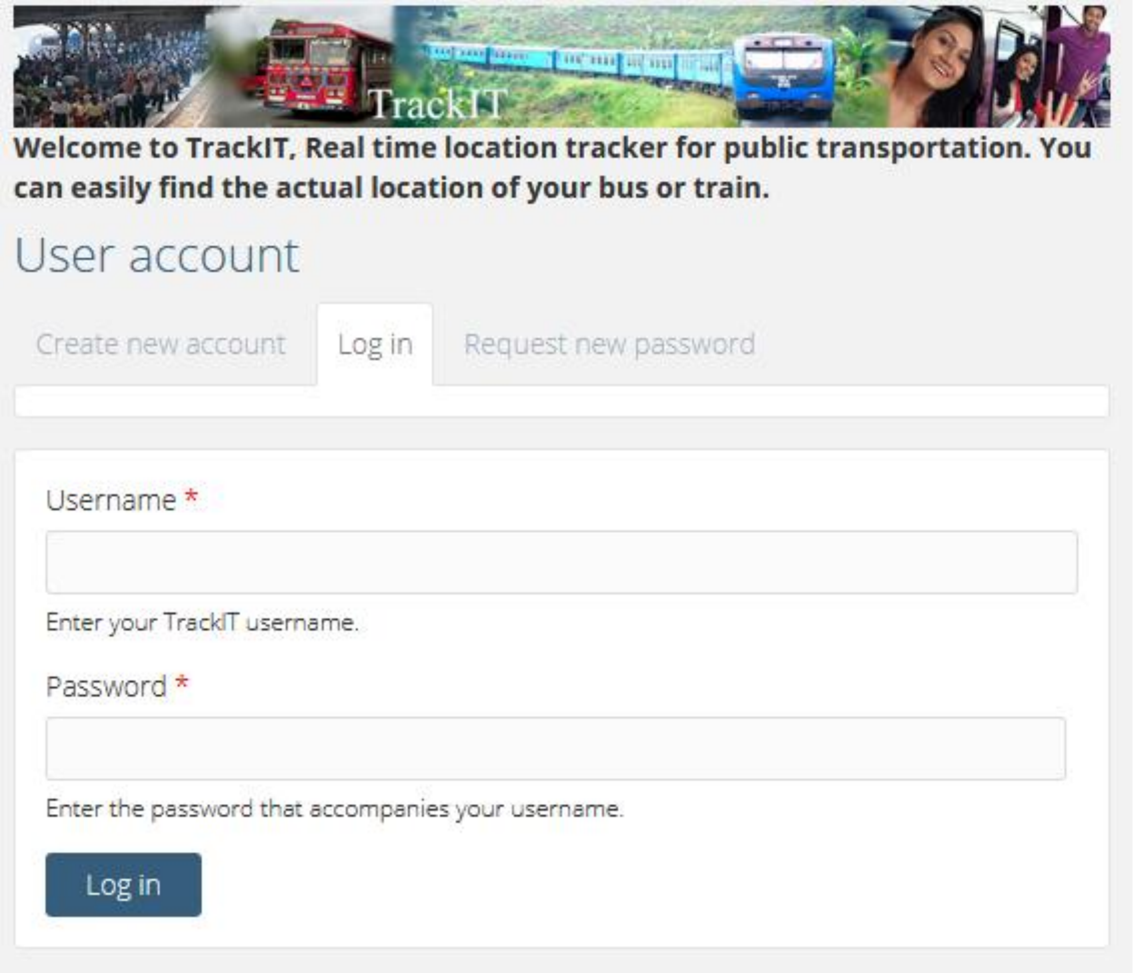
localhost/phpmyadmin/index.php?db=trackit&token=3856d1a5be697743ea0ba2aa84550e00#PMAURL:db=track

□	trackit_coordinates	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_locations	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_locations_data	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_locations_datav2	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_routes_master	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_service_routes	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_shedule	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_shuttles	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_stops	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_test_data_table	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_trackgps	Browse		Search	Insert	Empty	Drop
□	trackit_trains	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_train_locations	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_train_locations_datav2	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_train_locations_datav3	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_train_locations_data_samudra	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_train_schedule	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_train_speeds	Browse	Structure	Search	Insert	Empty	Drop
□	trackit_train_stations	Browse	Structure	Search	Insert	Empty	Drop
□	train_speeds	Browse	Structure	Search	Insert	Empty	Drop

Figure 5.20 – Database administration interface - PhpMyAdmin

Appendix D – User Module

User Module Interfaces



Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

User account

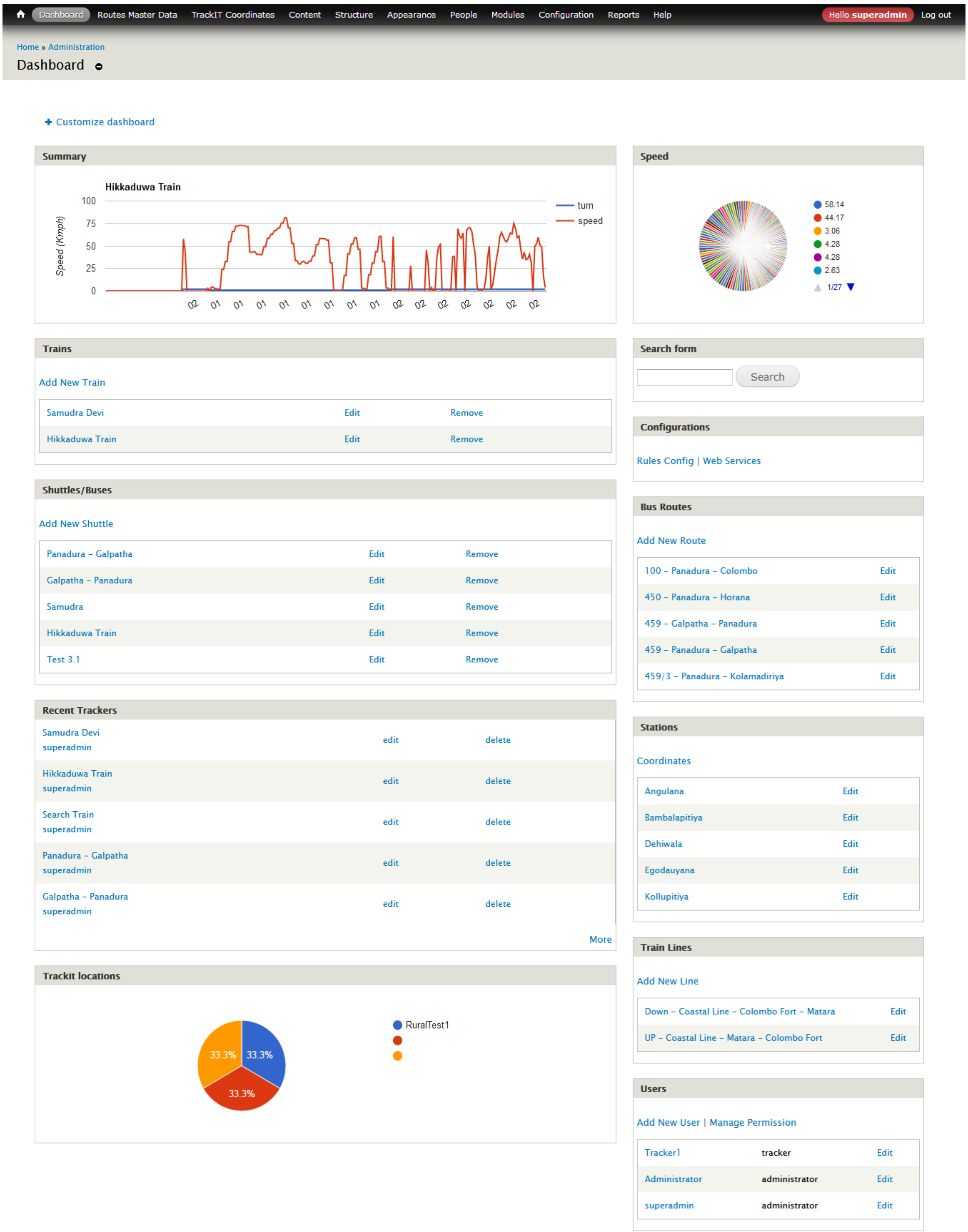
Username *

Enter your TrackIT username.

Password *

Enter the password that accompanies your username.

Figure 6.7 – Log in page



Dashboard

Summary

Speed

Trains

Samudra Devi	Edit	Remove
Hikkaduwa Train	Edit	Remove

Search form

Configurations

Shuttles/Buses

Panadura – Galpatha	Edit	Remove
Galpatha – Panadura	Edit	Remove
Samudra	Edit	Remove
Hikkaduwa Train	Edit	Remove
Test 3.1	Edit	Remove

Bus Routes

100 – Panadura – Colombo	Edit
450 – Panadura – Horana	Edit
459 – Galpatha – Panadura	Edit
459 – Panadura – Galpatha	Edit
459/3 – Panadura – Kolamadiriya	Edit

Recent Trackers

Samudra Devi superadmin	edit	delete
Hikkaduwa Train superadmin	edit	delete
Search Train superadmin	edit	delete
Panadura – Galpatha superadmin	edit	delete
Galpatha – Panadura superadmin	edit	delete

Stations

Angulana	Edit
Bambalapitiya	Edit
Dehiwala	Edit
Egodauyana	Edit
Kollupitiya	Edit

Trackit locations

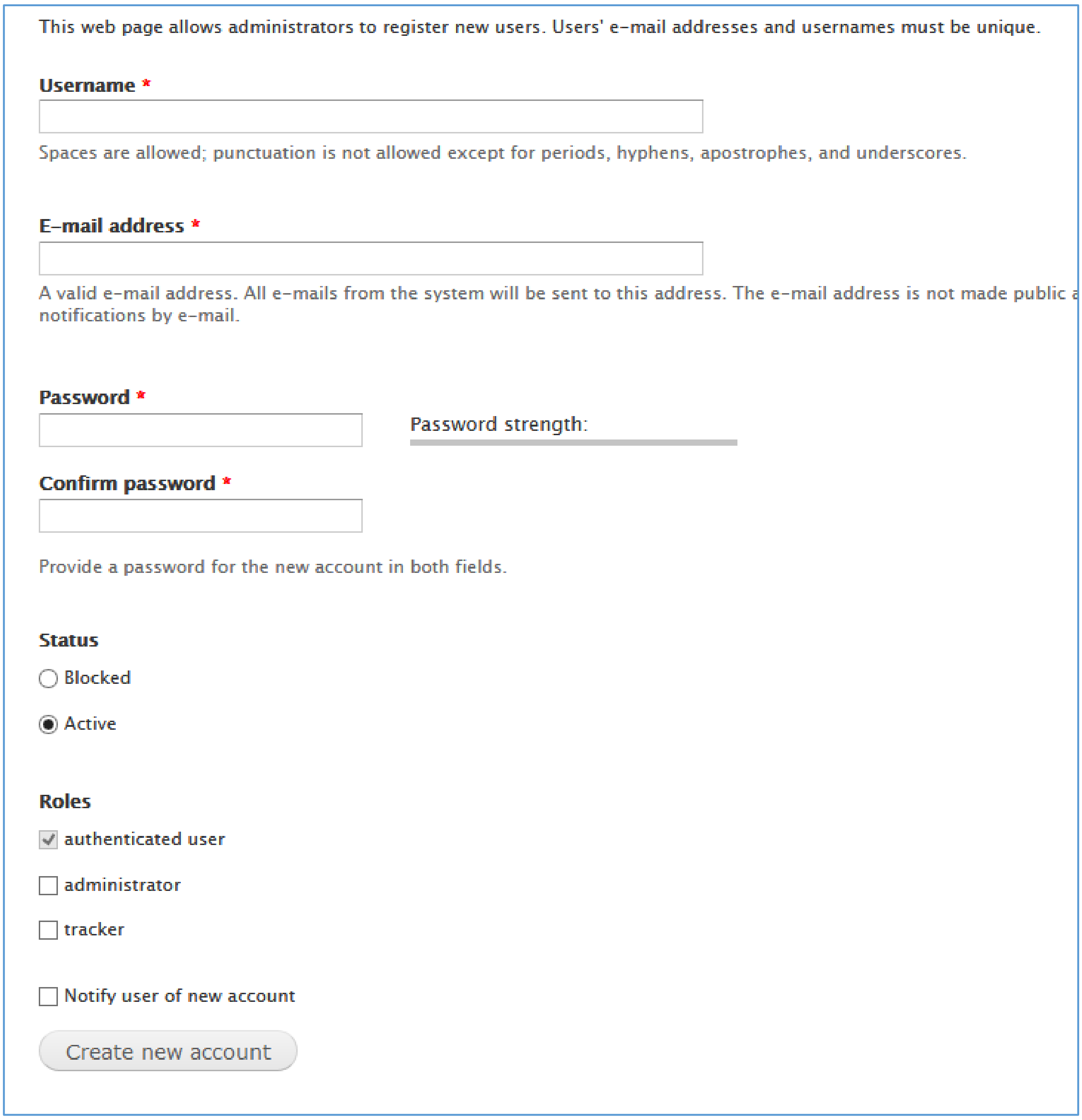
Train Lines

Down – Coastal Line – Colombo Fort – Matara	Edit
UP – Coastal Line – Matara – Colombo Fort	Edit

Users

Tracker1	tracker	Edit
Administrator	administrator	Edit
superadmin	administrator	Edit

Figure 6.8 – Administration Dashboard



This web page allows administrators to register new users. Users' e-mail addresses and usernames must be unique.

Username *

Spaces are allowed; punctuation is not allowed except for periods, hyphens, apostrophes, and underscores.

E-mail address *

A valid e-mail address. All e-mails from the system will be sent to this address. The e-mail address is not made public a notifications by e-mail.

Password *

Password strength:

Confirm password *

Provide a password for the new account in both fields.

Status

○ Blocked

● Active

Roles

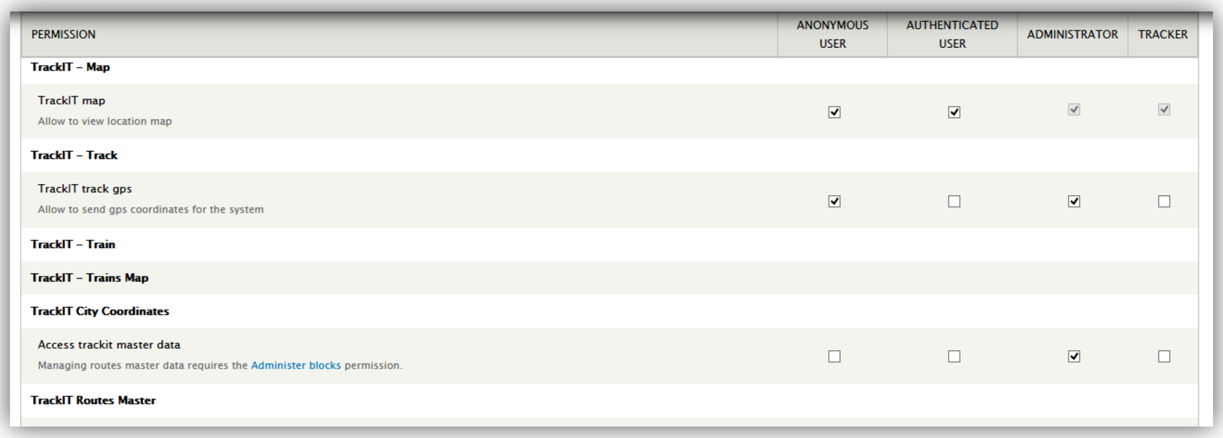
☑ authenticated user

□ administrator

□ tracker

□ Notify user of new account

Figure 6.9 – Add new administrator users

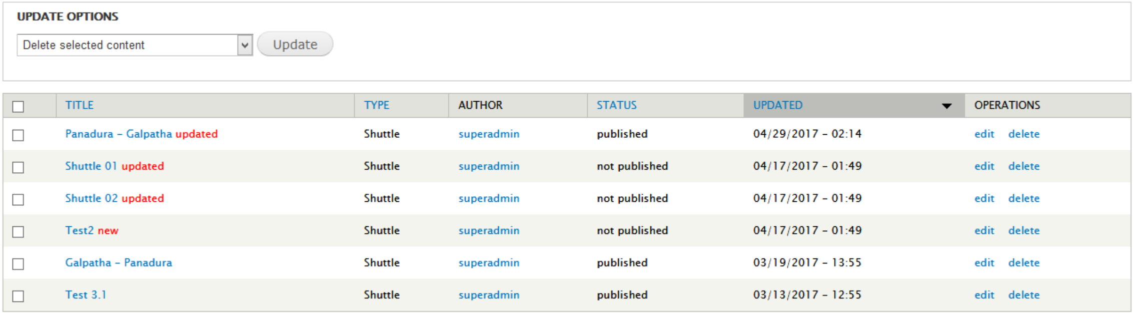


PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR	TRACKER
TrackIT – Map				
TrackIT map Allow to view location map	☑	☑	☑	☑
TrackIT – Track				
TrackIT track gps Allow to send gps coordinates for the system	☑	□	☑	□
TrackIT – Train				
TrackIT – Trains Map				
TrackIT City Coordinates				
Access trackit master data Managing routes master data requires the Administer blocks permission.	□	□	☑	□
TrackIT Routes Master				

Figure 6.10 - Granting permission levels for different user roles

Appendix E – Track Module

Bus Tracking Interfaces

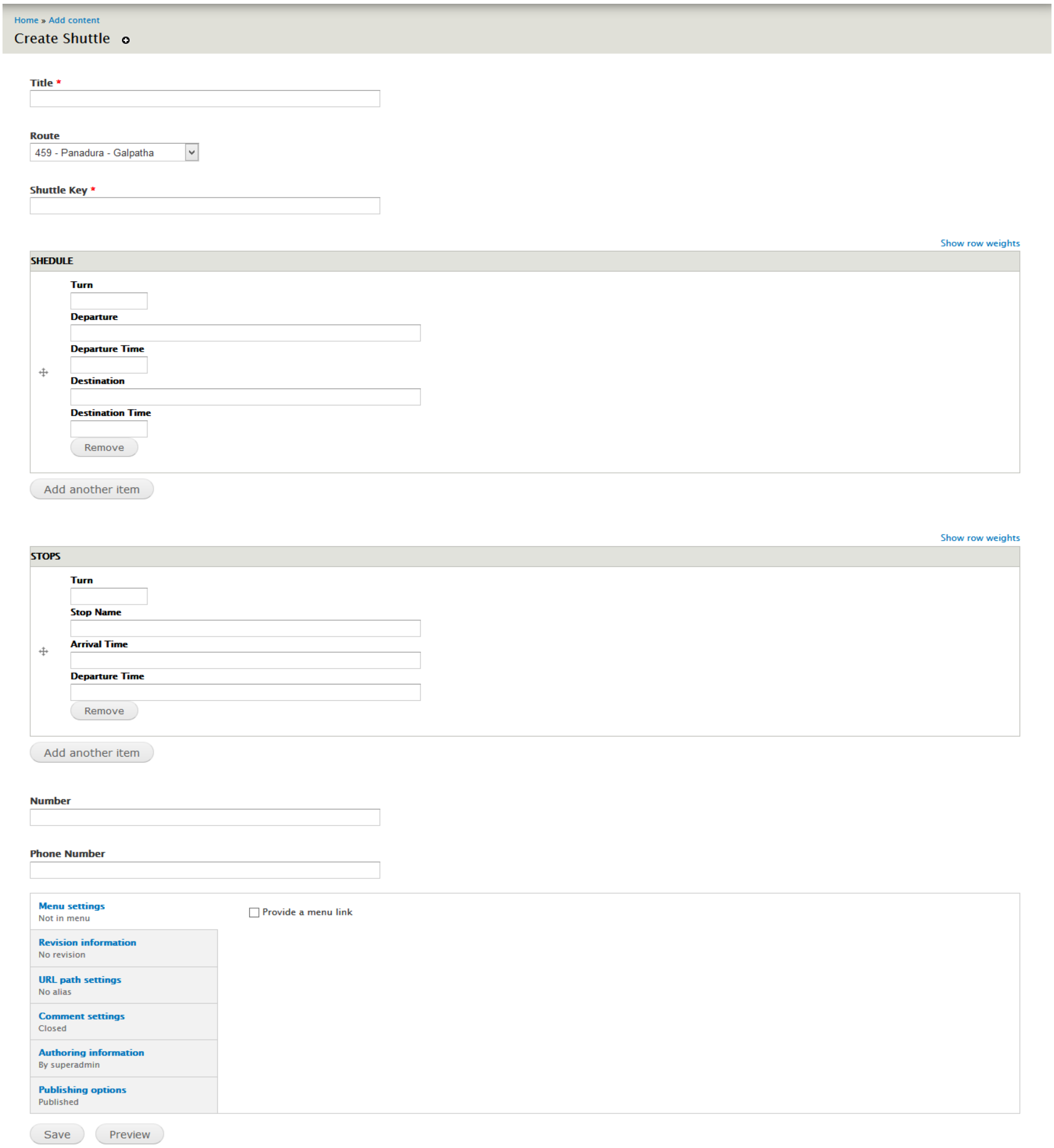


UPDATE OPTIONS

Delete selected content ▾ Update

□	TITLE	TYPE	AUTHOR	STATUS	UPDATED ▼	OPERATIONS
□	Panadura - Galpatha updated	Shuttle	superadmin	published	04/29/2017 - 02:14	edit delete
□	Shuttle 01 updated	Shuttle	superadmin	not published	04/17/2017 - 01:49	edit delete
□	Shuttle 02 updated	Shuttle	superadmin	not published	04/17/2017 - 01:49	edit delete
□	Test2 new	Shuttle	superadmin	not published	04/17/2017 - 01:49	edit delete
□	Galpatha - Panadura	Shuttle	superadmin	published	03/19/2017 - 13:55	edit delete
□	Test 3.1	Shuttle	superadmin	published	03/13/2017 - 12:55	edit delete

Figure 6.11 – Bus master data administration interface



Home » Add content

Create Shuttle

Title *

Route

459 - Panadura - Galpatha

Shuttle Key *

Show row weights

SHEDULE

Turn

Departure

Departure Time

Destination

Destination Time

Remove

Add another item

Show row weights

STOPS

Turn

Stop Name

Arrival Time

Departure Time

Remove

Add another item

Number

Phone Number

Menu settings
Not in menu

□ Provide a menu link

Revision information
No revision

URL path settings
No alias

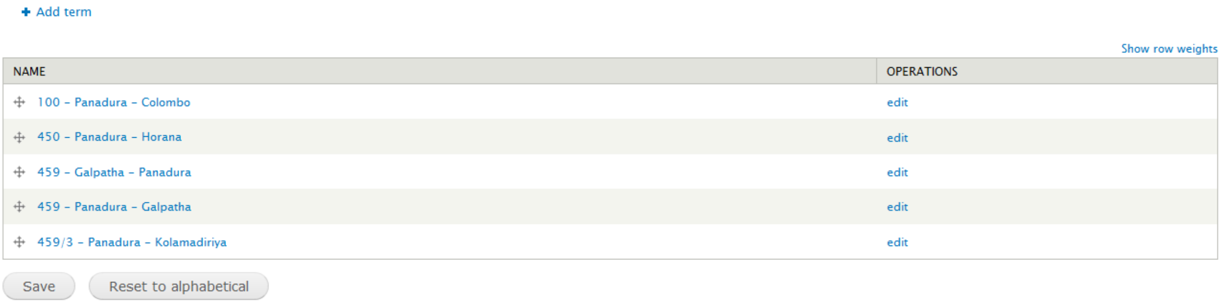
Comment settings
Closed

Authoring information
By superadmin

Publishing options
Published

Save Preview

Figure 6.12 - Adding bus and time schedule

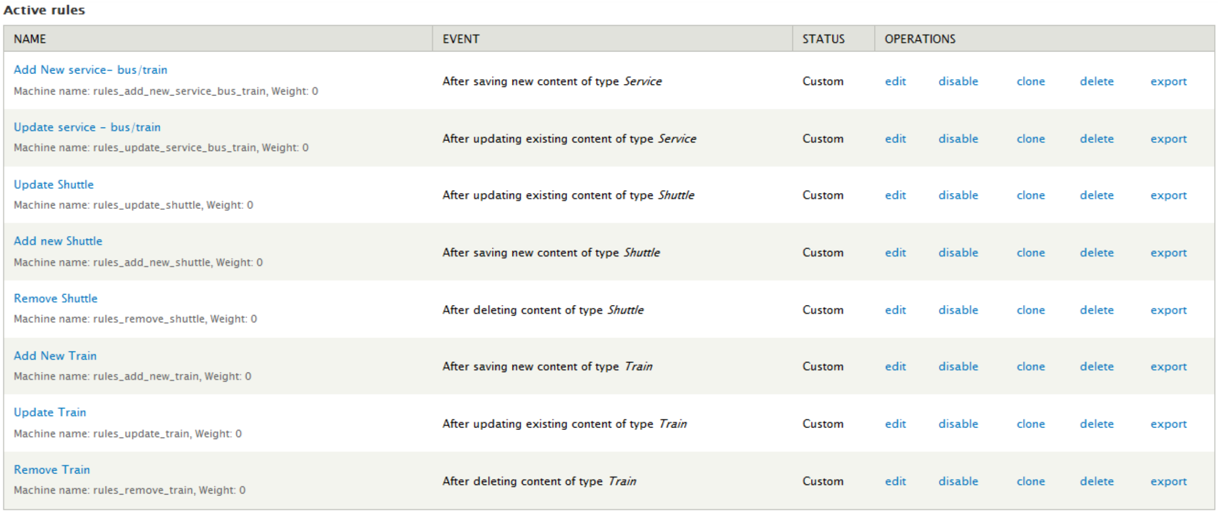


✚ Add term

Show row weights

NAME	OPERATIONS
✣ 100 – Panadura – Colombo	edit
✣ 450 – Panadura – Horana	edit
✣ 459 – Galpatha – Panadura	edit
✣ 459 – Panadura – Galpatha	edit
✣ 459/3 – Panadura – Kolamadiriya	edit

Figure 6.13 – Bus routes administration interface



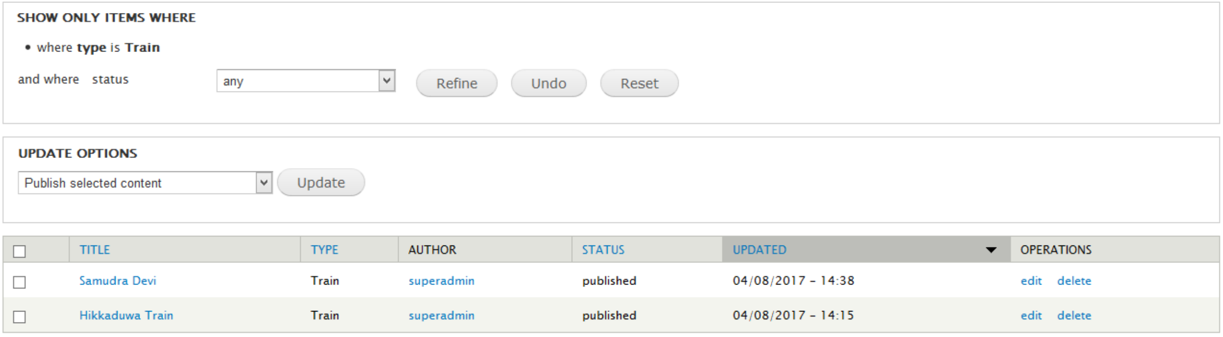
Active rules

NAME	EVENT	STATUS	OPERATIONS				
Add New service- bus/train Machine name: rules_add_new_service_bus_train, Weight: 0	After saving new content of type *Service*	Custom	edit	disable	clone	delete	export
Update service – bus/train Machine name: rules_update_service_bus_train, Weight: 0	After updating existing content of type *Service*	Custom	edit	disable	clone	delete	export
Update Shuttle Machine name: rules_update_shuttle, Weight: 0	After updating existing content of type *Shuttle*	Custom	edit	disable	clone	delete	export
Add new Shuttle Machine name: rules_add_new_shuttle, Weight: 0	After saving new content of type *Shuttle*	Custom	edit	disable	clone	delete	export
Remove Shuttle Machine name: rules_remove_shuttle, Weight: 0	After deleting content of type *Shuttle*	Custom	edit	disable	clone	delete	export
Add New Train Machine name: rules_add_new_train, Weight: 0	After saving new content of type *Train*	Custom	edit	disable	clone	delete	export
Update Train Machine name: rules_update_train, Weight: 0	After updating existing content of type *Train*	Custom	edit	disable	clone	delete	export
Remove Train Machine name: rules_remove_train, Weight: 0	After deleting content of type *Train*	Custom	edit	disable	clone	delete	export

Figure 6.14 – Admin interface to manage rules when managing master data

Appendix F – Train Tracking Module

Train Tracking Interfaces



SHOW ONLY ITEMS WHERE

- where **type** is **Train**

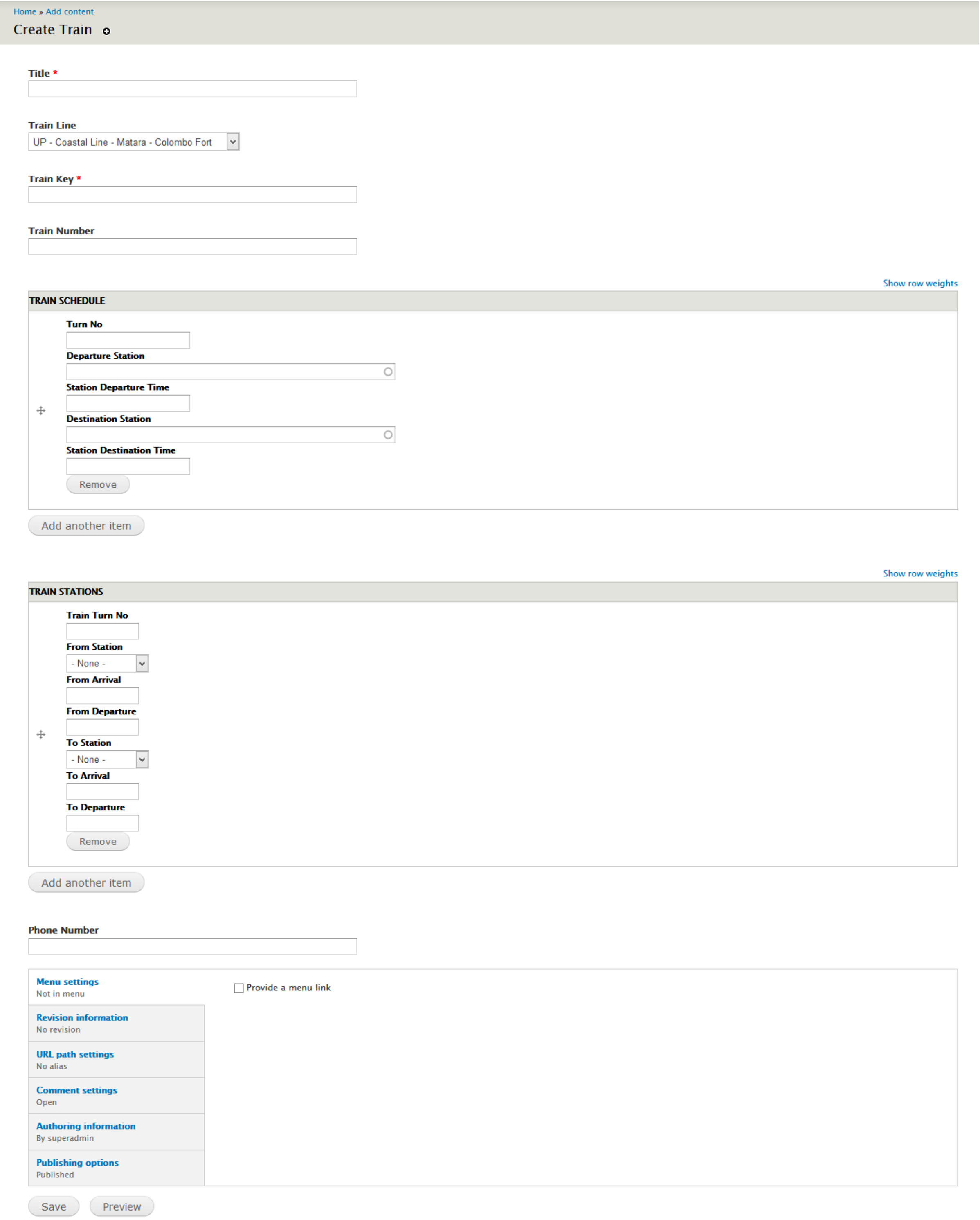
and where status: any

UPDATE OPTIONS

Publish selected content

□	TITLE	TYPE	AUTHOR	STATUS	UPDATED	OPERATIONS
□	Samudra Devi	Train	superadmin	published	04/08/2017 - 14:38	edit delete
□	Hikkaduwa Train	Train	superadmin	published	04/08/2017 - 14:15	edit delete

Figure 6.15 – Managing train admin interface



Home » Add content

Create Train

Title *

Train Line

UP - Coastal Line - Matara - Colombo Fort

Train Key *

Train Number

Show row weights

TRAIN SCHEDULE

Turn No

Departure Station

Station Departure Time

Destination Station

Station Destination Time

Remove

Add another item

Show row weights

TRAIN STATIONS

Train Turn No

From Station

- None -

From Arrival

From Departure

To Station

- None -

To Arrival

To Departure

Remove

Add another item

Phone Number

Menu settings
Not in menu

□ Provide a menu link

Revision information
No revision

URL path settings
No alias

Comment settings
Open

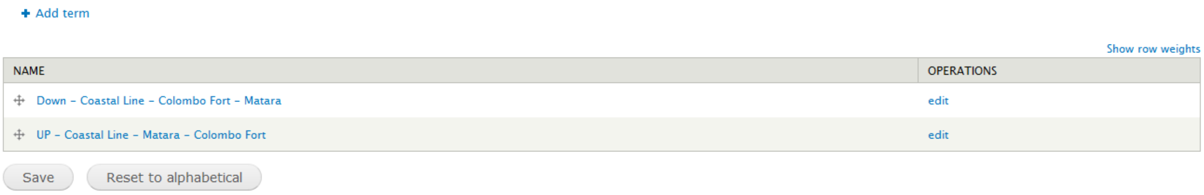
Authoring information
By superadmin

Publishing options
Published

Save

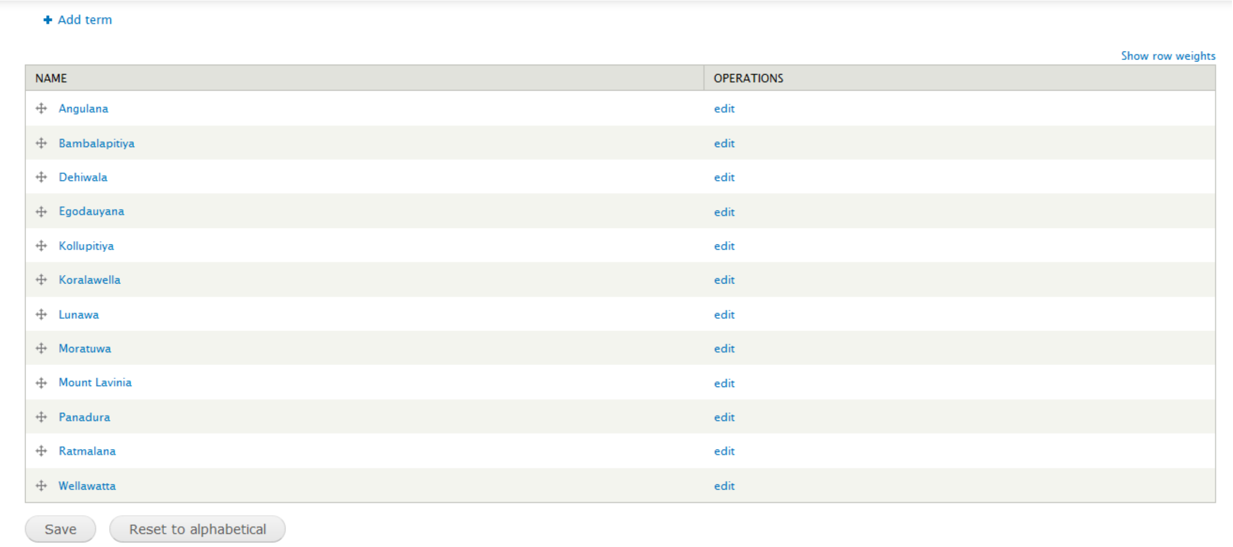
Preview

Figure 6.16 – Adding train and schedule to the system



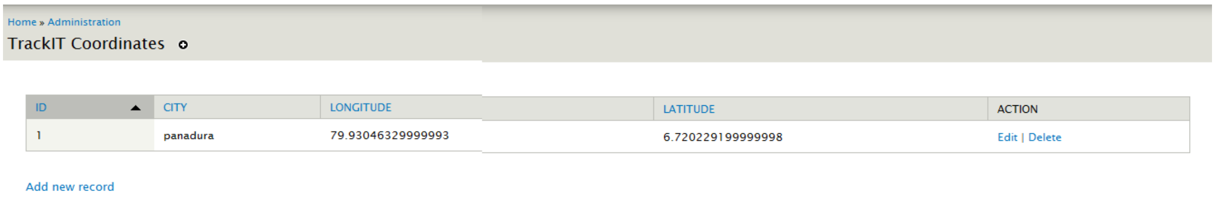
NAME	OPERATIONS
Down – Coastal Line – Colombo Fort – Matara	edit
UP – Coastal Line – Matara – Colombo Fort	edit

Figure 6.17 – Train Lines interface



NAME	OPERATIONS
Angulana	edit
Bambalapitiya	edit
Dehiwala	edit
Egodauyana	edit
Kollupitiya	edit
Koralawella	edit
Lunawa	edit
Moratuwa	edit
Mount Lavinia	edit
Panadura	edit
Ratmalana	edit
Wellawatta	edit

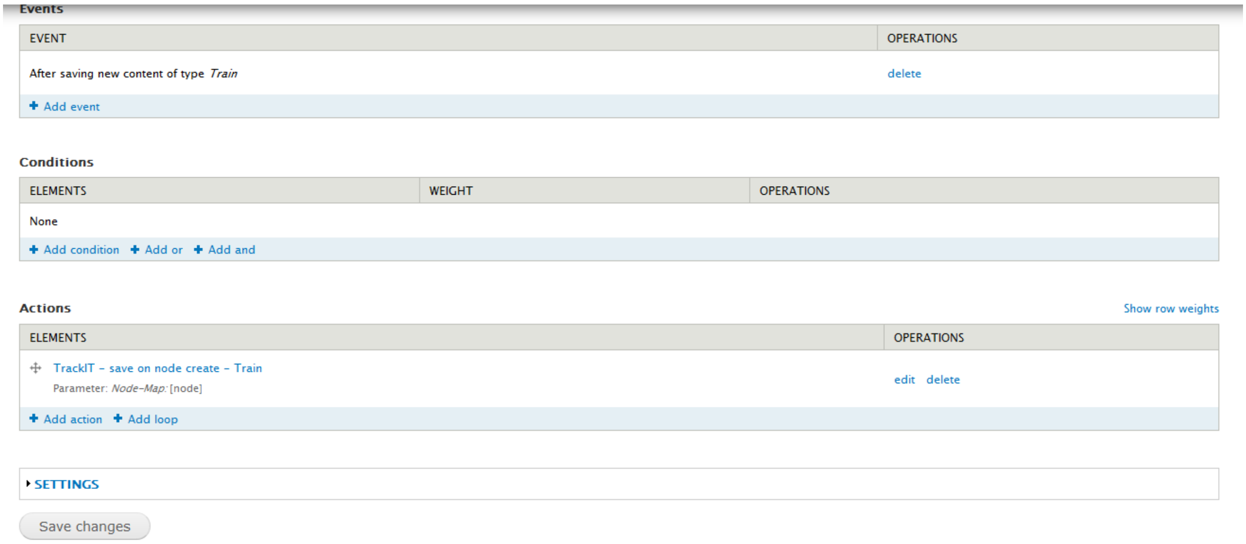
Figure 6.18 – Railway Stations



TrackIT Coordinates

ID	CITY	LONGITUDE	LATITUDE	ACTION
1	panadura	79.93046329999993	6.720229199999998	Edit | Delete

Figure 6.19 – Geo coordinates of stations



Events

EVENT	OPERATIONS
After saving new content of type *Train*	delete

✚ Add event

Conditions

ELEMENTS	WEIGHT	OPERATIONS
None		

✚ Add condition ✚ Add or ✚ Add and

Actions

Show row weights

ELEMENTS	OPERATIONS
✥ TrackIT – save on node create – Train Parameter: *Node-Map:* [node]	edit delete

✚ Add action ✚ Add loop

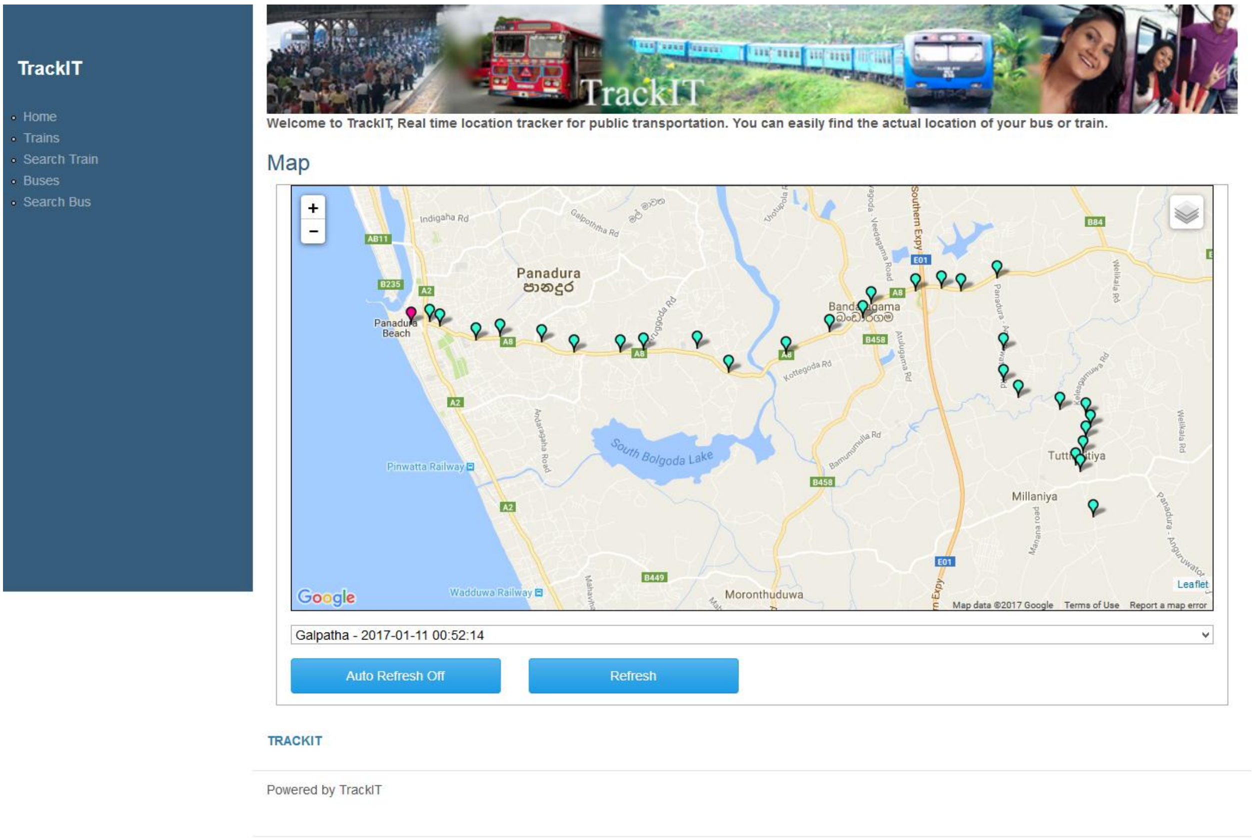
▸ SETTINGS

Save changes

Figure 6.20 – Rules configurations to handle train master data

Appendix G – Frontend Modules

Frontend Interfaces

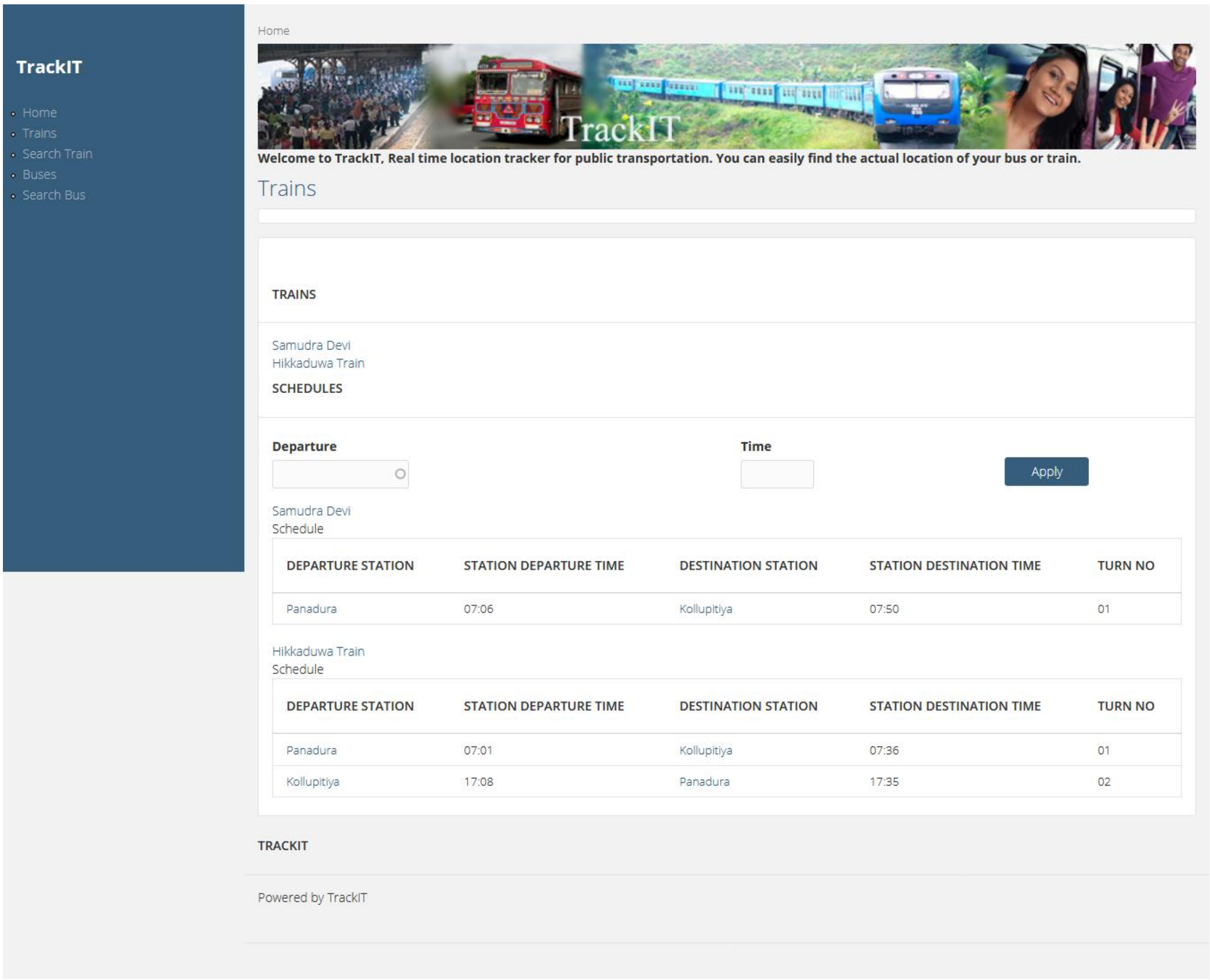


Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

Galpatha - 2017-01-11 00:52:14

Powered by TrackIT

Figure 6.21 – Home page for public users



Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

Trains

TRAINS

Samudra Devi
Hikkaduwa Train

SCHEDULES

Departure ○ Time Apply

Samudra Devi
Schedule

DEPARTURE STATION	STATION DEPARTURE TIME	DESTINATION STATION	STATION DESTINATION TIME	TURN NO
Panadura	07:06	Kollupitiya	07:50	01

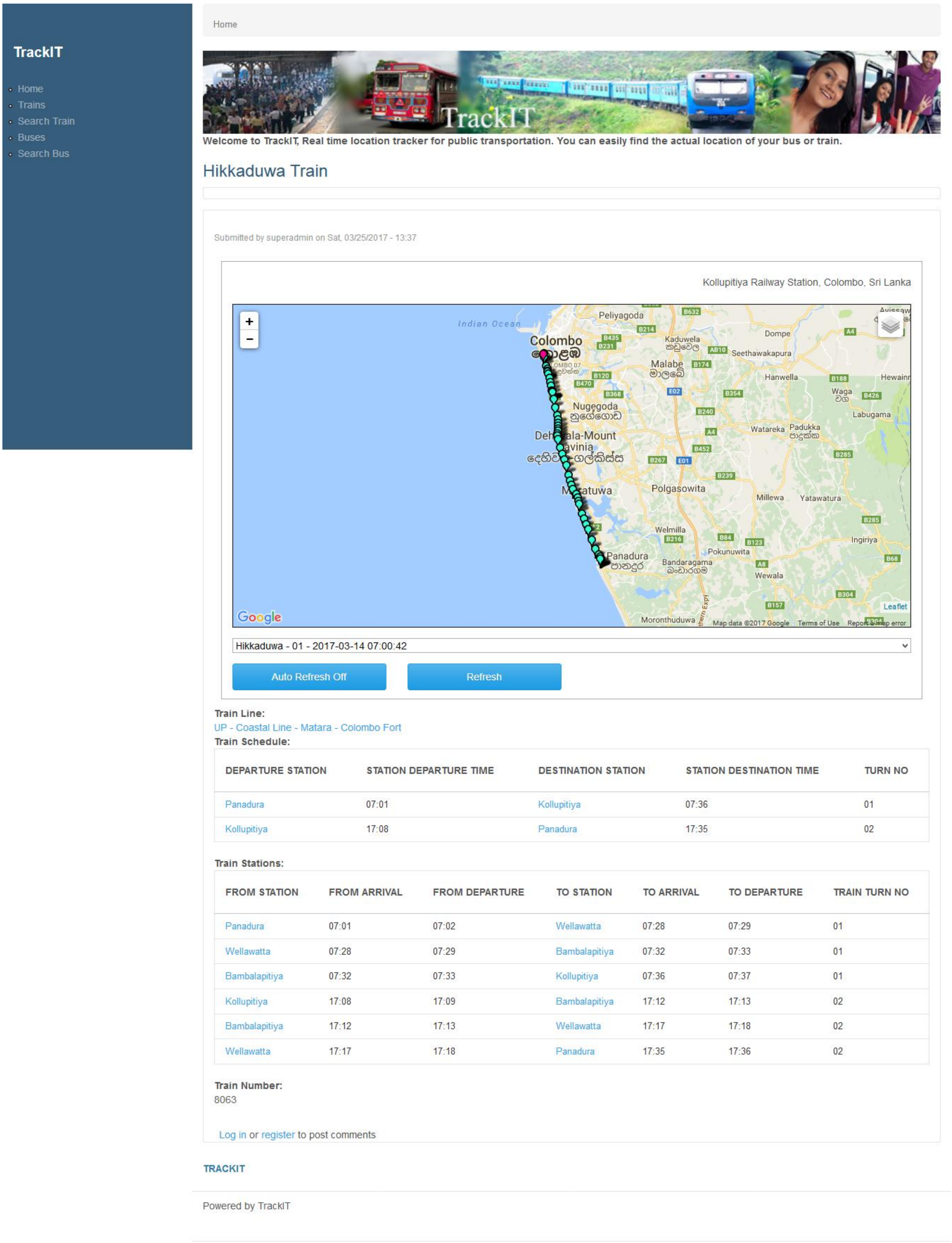
Hikkaduwa Train
Schedule

DEPARTURE STATION	STATION DEPARTURE TIME	DESTINATION STATION	STATION DESTINATION TIME	TURN NO
Panadura	07:01	Kollupitiya	07:36	01
Kollupitiya	17:08	Panadura	17:35	02

TRACKIT

Powered by TrackIT

Figure 6.22 – List of Trains with schedule



Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

Hikkaduwa Train

Submitted by superadmin on Sat, 03/25/2017 - 13:37

Kollupitiya Railway Station, Colombo, Sri Lanka

Hikkaduwa - 01 - 2017-03-14 07:00:42

Train Line:
UP - Coastal Line - Matara - Colombo Fort
Train Schedule:

DEPARTURE STATION	STATION DEPARTURE TIME	DESTINATION STATION	STATION DESTINATION TIME	TURN NO
Panadura	07:01	Kollupitiya	07:36	01
Kollupitiya	17:08	Panadura	17:35	02

Train Stations:

FROM STATION	FROM ARRIVAL	FROM DEPARTURE	TO STATION	TO ARRIVAL	TO DEPARTURE	TRAIN TURN NO
Panadura	07:01	07:02	Wellawatta	07:28	07:29	01
Wellawatta	07:28	07:29	Bambalapitiya	07:32	07:33	01
Bambalapitiya	07:32	07:33	Kollupitiya	07:36	07:37	01
Kollupitiya	17:08	17:09	Bambalapitiya	17:12	17:13	02
Bambalapitiya	17:12	17:13	Wellawatta	17:17	17:18	02
Wellawatta	17:17	17:18	Panadura	17:35	17:36	02

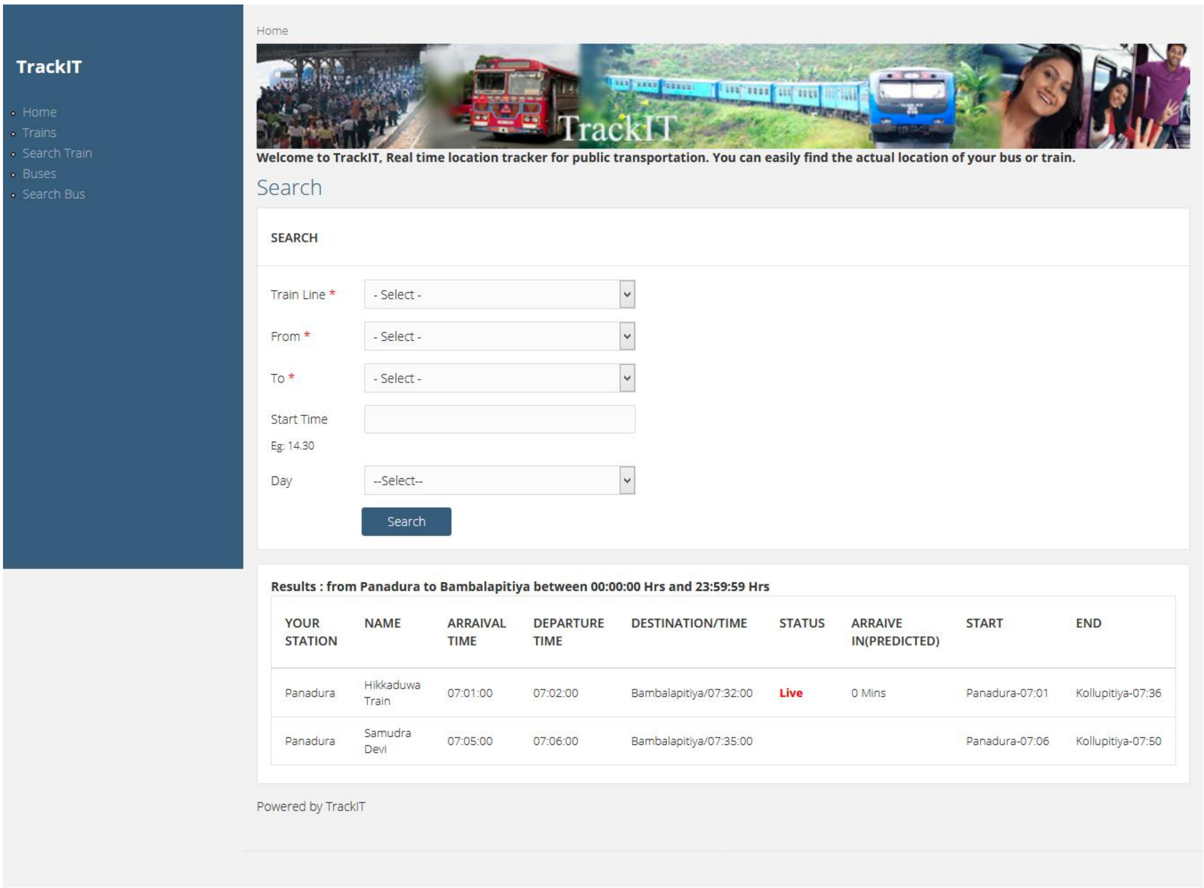
Train Number:
8063

Log in or register to post comments

TRACKIT

Powered by TrackIT

Figure 6.23– Live train details and timetable



Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

Search

SEARCH

Train Line * - Select -

From * - Select -

To * - Select -

Start Time

Eg: 14.30

Day --Select--

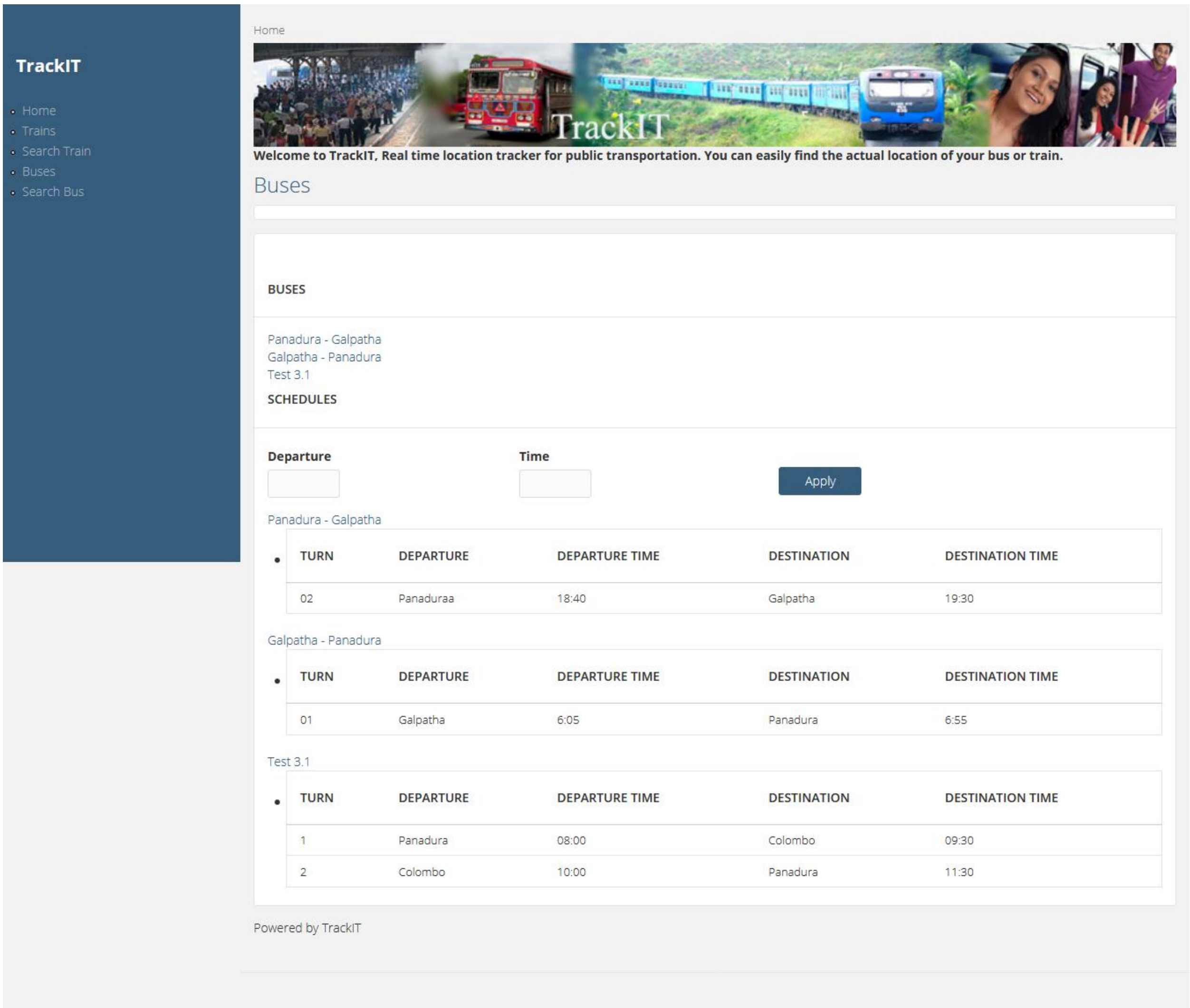
Search

Results : from Panadura to Bambalapitiya between 00:00:00 Hrs and 23:59:59 Hrs

YOUR STATION	NAME	ARRAIVAL TIME	DEPARTURE TIME	DESTINATION/TIME	STATUS	ARRAIVE IN(PREDICTED)	START	END
Panadura	Hikkaduwa Train	07:01:00	07:02:00	Bambalapitiya/07:32:00	Live	0 Mins	Panadura-07:01	Kollupitiya-07:36
Panadura	Samudra Devi	07:05:00	07:06:00	Bambalapitiya/07:35:00			Panadura-07:06	Kollupitiya-07:50

Powered by TrackIT

Figure 6.24 – Train Search and Search Results



Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

Buses

BUSES

Panadura - Galpatha
Galpatha - Panadura
Test 3.1

SCHEDULES

Departure **Time**

Panadura - Galpatha

TURN	DEPARTURE	DEPARTURE TIME	DESTINATION	DESTINATION TIME
02	Panaduraa	18:40	Galpatha	19:30

Galpatha - Panadura

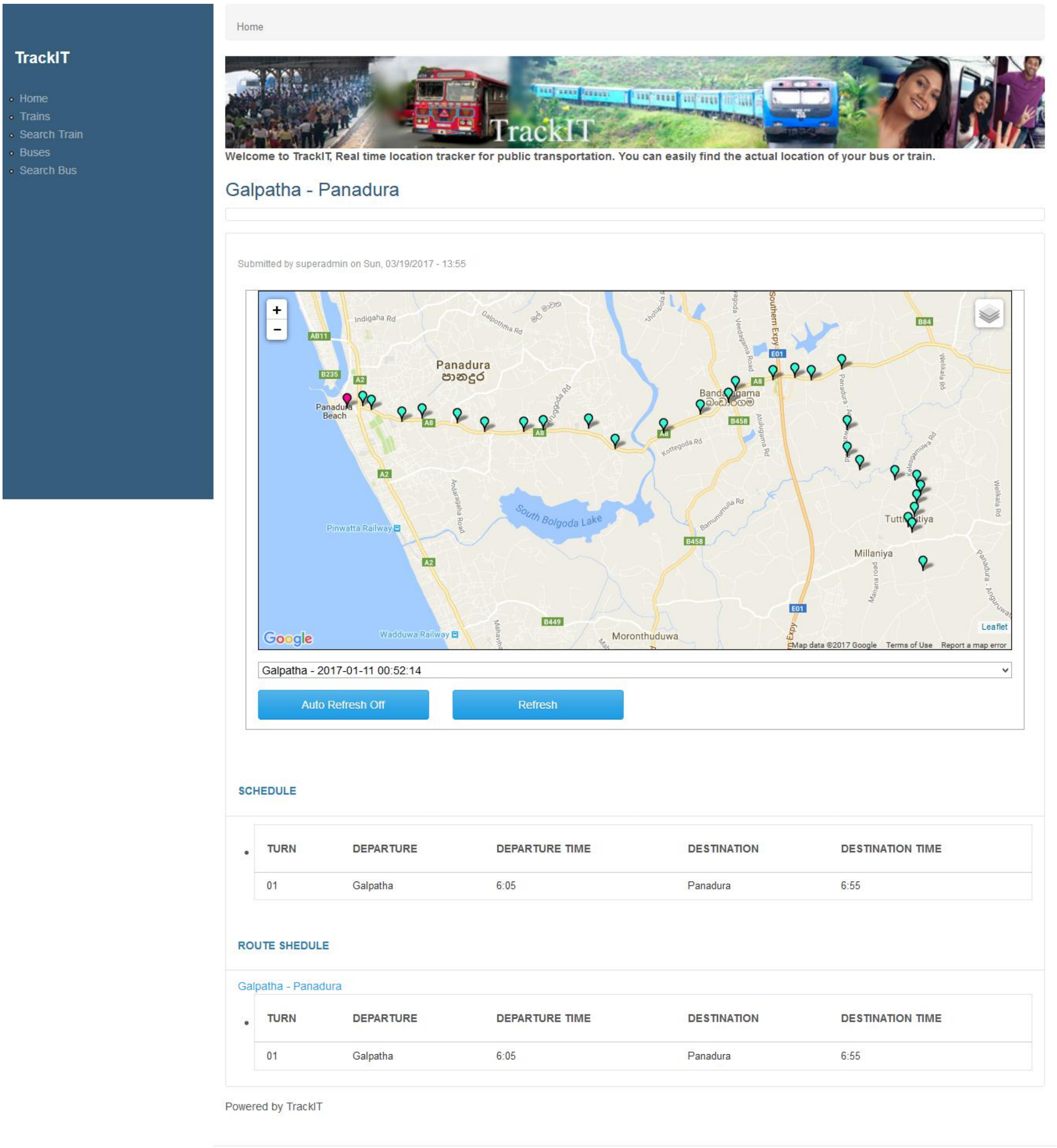
TURN	DEPARTURE	DEPARTURE TIME	DESTINATION	DESTINATION TIME
01	Galpatha	6:05	Panadura	6:55

Test 3.1

TURN	DEPARTURE	DEPARTURE TIME	DESTINATION	DESTINATION TIME
1	Panadura	08:00	Colombo	09:30
2	Colombo	10:00	Panadura	11:30

Powered by TrackIT

Figure 6.25 – List of buses with time schedule



Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

Galpatha - Panadura

Submitted by superadmin on Sun, 03/19/2017 - 13:55

Galpatha - 2017-01-11 00:52:14

SCHEDULE

TURN	DEPARTURE	DEPARTURE TIME	DESTINATION	DESTINATION TIME
01	Galpatha	6:05	Panadura	6:55

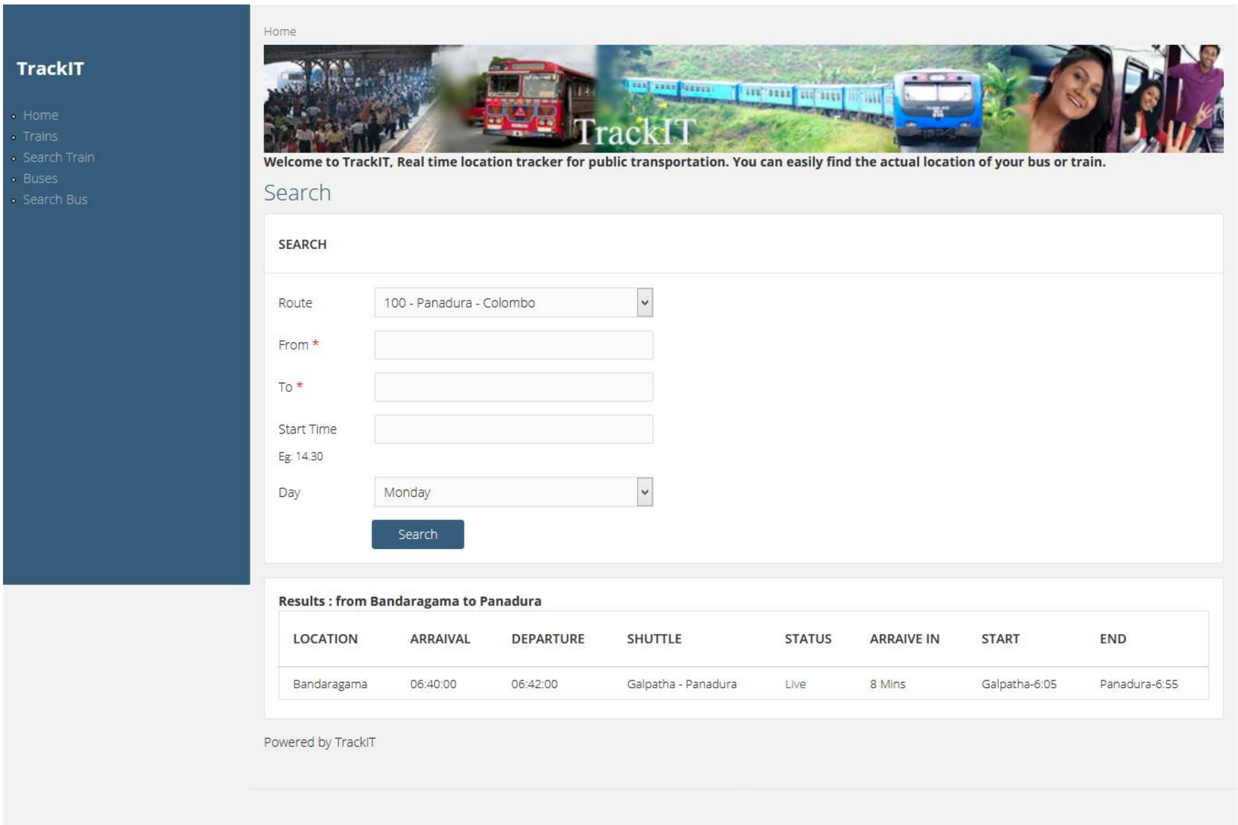
ROUTE SHEDULE

Galpatha - Panadura

TURN	DEPARTURE	DEPARTURE TIME	DESTINATION	DESTINATION TIME
01	Galpatha	6:05	Panadura	6:55

Powered by TrackIT

Figure 6.26 – Live map and schedule



Welcome to TrackIT, Real time location tracker for public transportation. You can easily find the actual location of your bus or train.

Search

SEARCH

Route: 100 - Panadura - Colombo

From *

To *

Start Time

Eg: 14.30

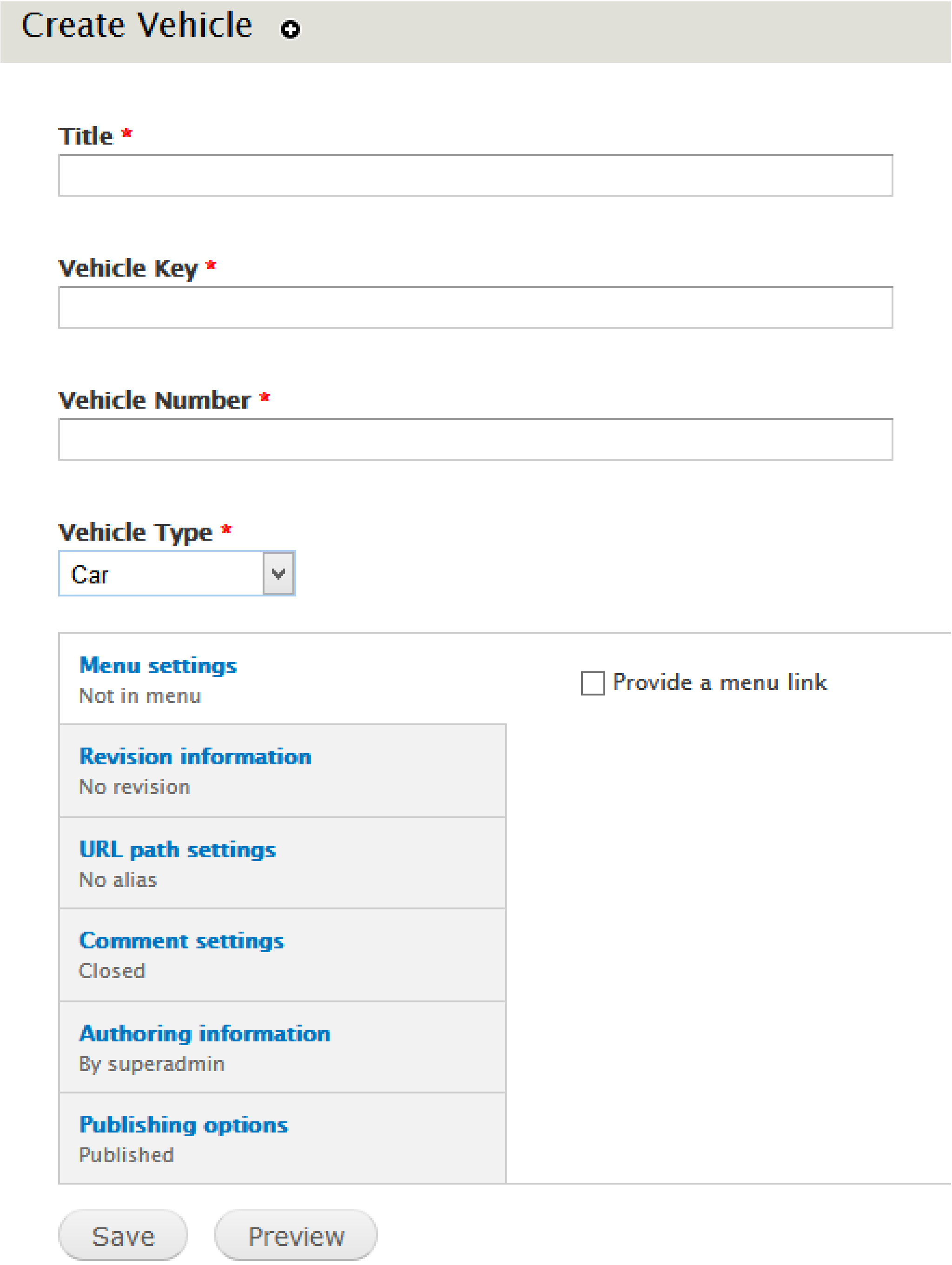
Day: Monday

Results : from Bandaragama to Panadura

LOCATION	ARRAIVAL	DEPARTURE	SHUTTLE	STATUS	ARRAIVE IN	START	END
Bandaragama	06:40:00	06:42:00	Galpatha - Panadura	Live	8 Mins	Galpatha-6:05	Panadura-6:55

Powered by TrackIT

Figure 6.27 – Bus Search and Results



Create Vehicle

Title *

Vehicle Key *

Vehicle Number *

Vehicle Type *
Car

Menu settings
Not in menu

□ Provide a menu link

Revision information
No revision

URL path settings
No alias

Comment settings
Closed

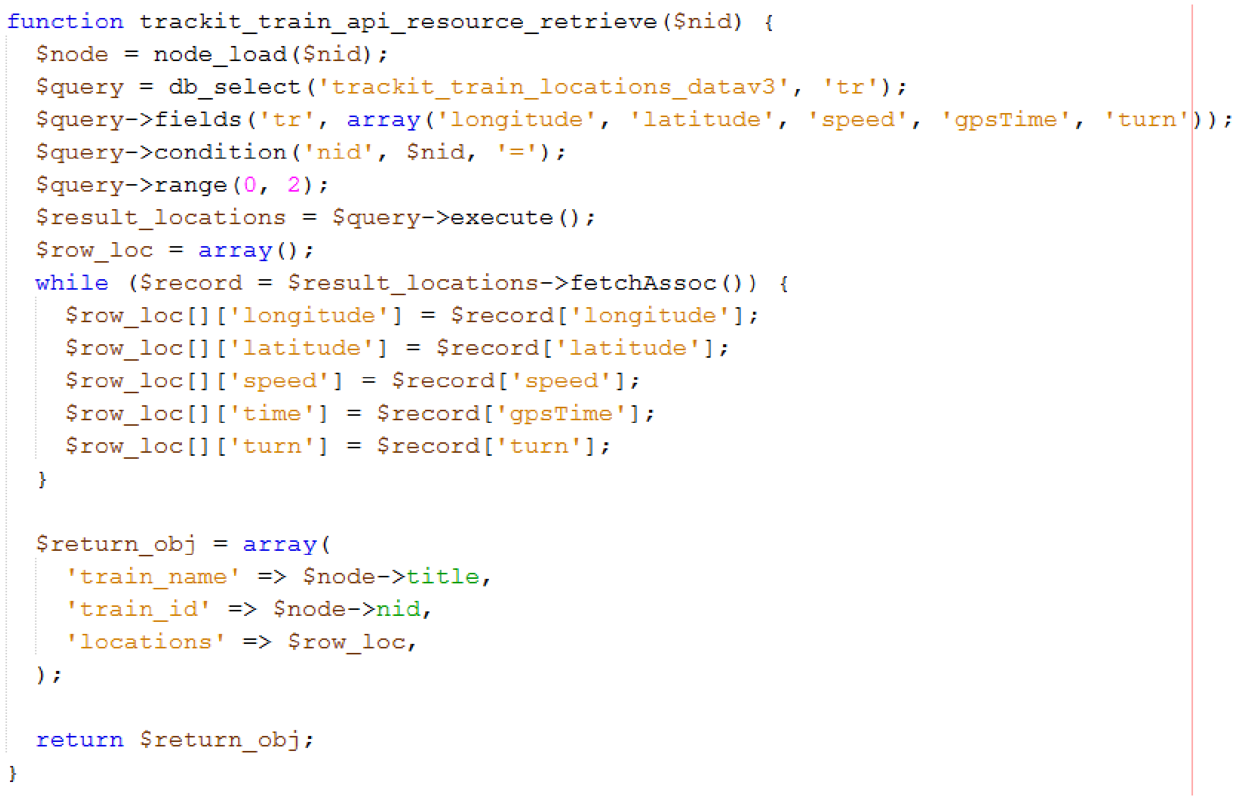
Authoring information
By superadmin

Publishing options
Published

Figure 6.28 – Add vehicle to system

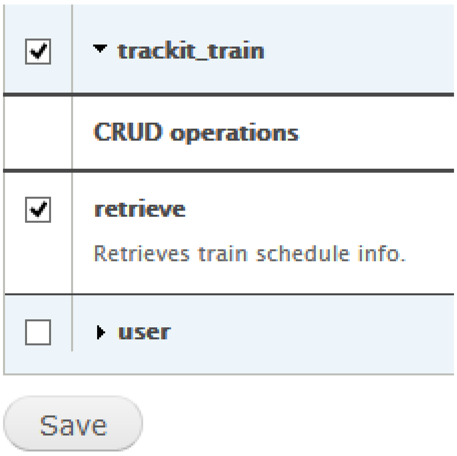
Appendix H – Services

Web service



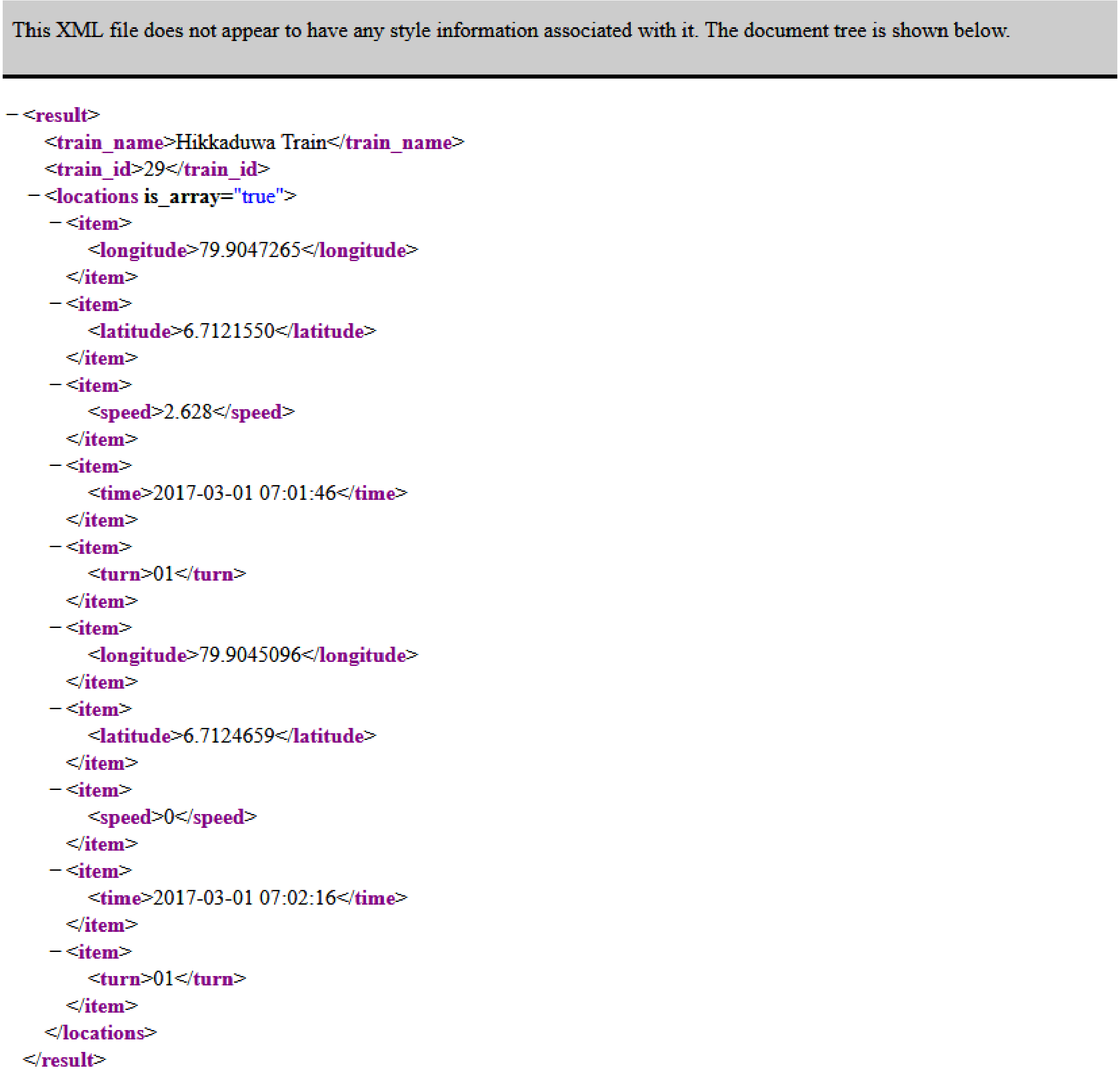
```
function trackit_train_api_resource_retrieve($nid) {
  $node = node_load($nid);
  $query = db_select('trackit_train_locations_datav3', 'tr');
  $query->fields('tr', array('longitude', 'latitude', 'speed', 'gpsTime', 'turn'));
  $query->condition('nid', $nid, '=');
  $query->range(0, 2);
  $result_locations = $query->execute();
  $row_loc = array();
  while ($record = $result_locations->fetchAssoc()) {
    $row_loc[]['longitude'] = $record['longitude'];
    $row_loc[]['latitude'] = $record['latitude'];
    $row_loc[]['speed'] = $record['speed'];
    $row_loc[]['time'] = $record['gpsTime'];
    $row_loc[]['turn'] = $record['turn'];
  }

  $return_obj = array(
    'train_name' => $node->title,
    'train_id' => $node->nid,
    'locations' => $row_loc,
  );

  return $return_obj;
}
```

Figure 6.29 – Part of a code to get location updates for web service



☑	▾ trackit_train
	CRUD operations
☑	retrieve Retrieves train schedule info.
□	▸ user

Figure 6.30 – Train data retrieval enabled in service



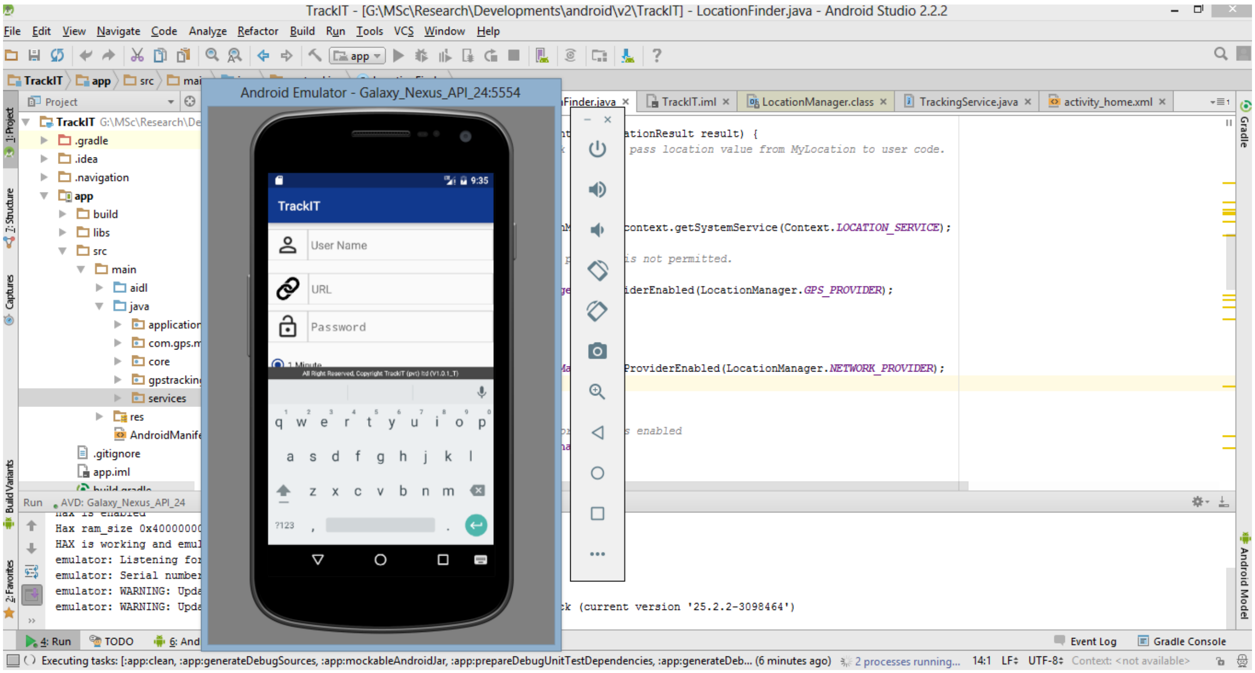
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <result>
    <train_name>Hikkaduwa Train</train_name>
    <train_id>29</train_id>
  – <locations is_array="true">
    – <item>
        <longitude>79.9047265</longitude>
      </item>
    – <item>
        <latitude>6.7121550</latitude>
      </item>
    – <item>
        <speed>2.628</speed>
      </item>
    – <item>
        <time>2017-03-01 07:01:46</time>
      </item>
    – <item>
        <turn>01</turn>
      </item>
    – <item>
        <longitude>79.9045096</longitude>
      </item>
    – <item>
        <latitude>6.7124659</latitude>
      </item>
    – <item>
        <speed>0</speed>
      </item>
    – <item>
        <time>2017-03-01 07:02:16</time>
      </item>
    – <item>
        <turn>01</turn>
      </item>
    </locations>
  </result>
```

Figure 6.31 – Web service output

Appendix I – Mobile Application

Development and Interfaces



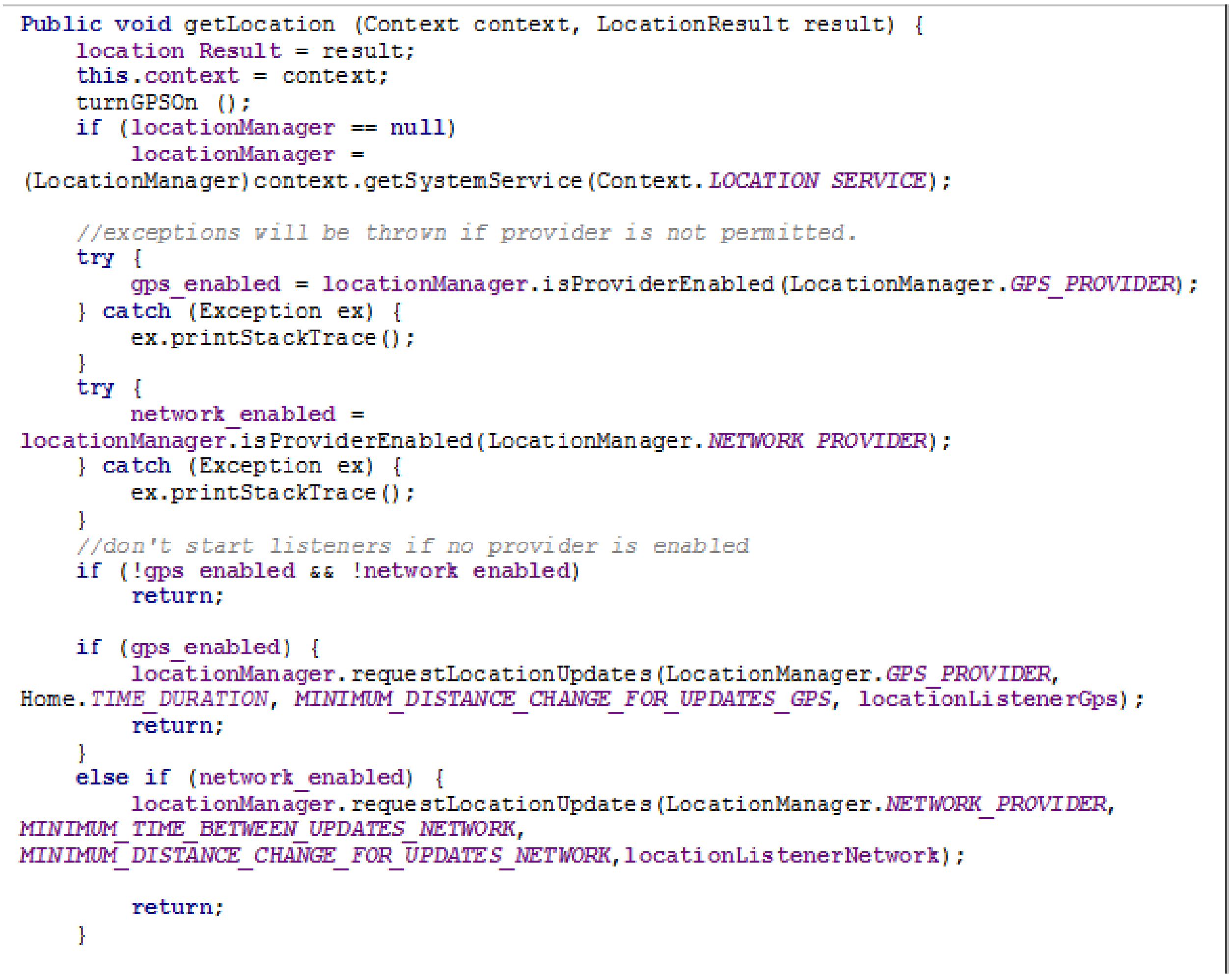
TrackIT - [G:\MSc\Research\Developments\android\v2\TrackIT] - LocationFinder.java - Android Studio 2.2.2

```
ationResult result) {
 pass location value from MyLocation to user code.

context.getSystemService(Context.LOCATION_SERVICE);

is not permitted.

iderEnabled(LocationManager.GPS_PROVIDER);

ProviderEnabled(LocationManager.NETWORK_PROVIDER);

s enabled
```

```
Hax ram_size 0x40000000
HAX is working and emul
emulator: Listening for
emulator: Serial number
emulator: WARNING: Upda
emulator: WARNING: Upda                    k (current version '25.2.2-3098464')
```

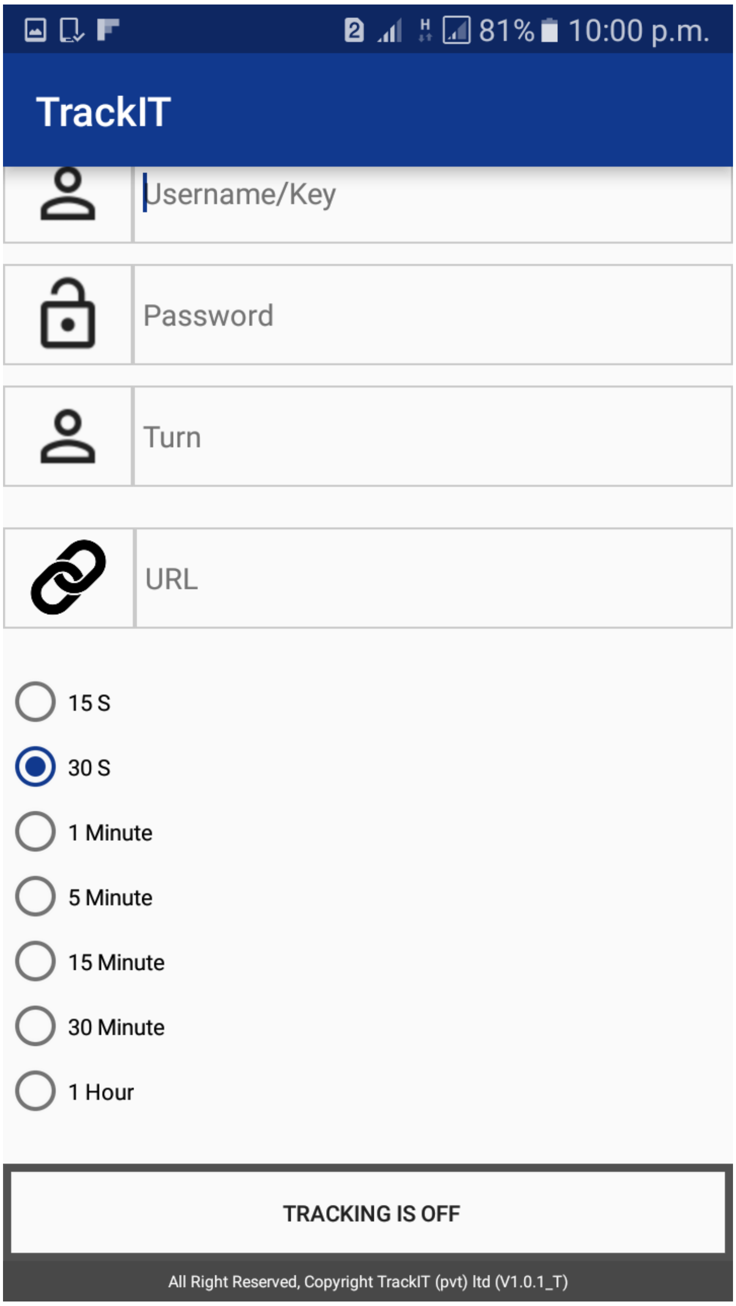
Executing tasks: [:app:clean, :app:generateDebugSources, :app:mockableAndroidJar, :app:prepareDebugUnitTestDependencies, :app:generateDeb... (6 minutes ago)   2 processes running...   14:1   LF   UTF-8   Context: <not available>

Figure 6.33 - Mobile Application development on Android Studio



```
Public void getLocation (Context context, LocationResult result) {
    location Result = result;
    this.context = context;
    turnGPSOn ();
    if (locationManager == null)
        locationManager =
(LocationManager)context.getSystemService(Context.LOCATION SERVICE);

    //exceptions will be thrown if provider is not permitted.
    try {
        gps_enabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    try {
        network_enabled =
locationManager.isProviderEnabled(LocationManager.NETWORK PROVIDER);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    //don't start listeners if no provider is enabled
    if (!gps enabled && !network enabled)
        return;

    if (gps_enabled) {
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
Home.TIME_DURATION, MINIMUM_DISTANCE_CHANGE_FOR_UPDATES_GPS, locationListenerGps);
        return;
    }
    else if (network_enabled) {
        locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
MINIMUM_TIME_BETWEEN_UPDATES_NETWORK,
MINIMUM_DISTANCE_CHANGE_FOR_UPDATES_NETWORK,locationListenerNetwork);

        return;
    }
```

Figure 6.34 – Code segment to get location updates



81% 10:00 p.m.

TrackIT

Username/Key

Password

Turn

URL

○ 15 S

● 30 S

○ 1 Minute

○ 5 Minute

○ 15 Minute

○ 30 Minute

○ 1 Hour

TRACKING IS OFF

All Right Reserved, Copyright TrackIT (pvt) ltd (V1.0.1_T)

Figure 6.35 - Mobile Application installed on phone

Appendix J – Evaluation

Evaluation Forms

TrackIT Main Web Application Evaluation in terms of usability

Selectable options- 1 - Very poor, 2 – Poor, 3 – Average, 4 - Good, 5 - Excellent

	Question	Selectable Answers
1	How would you categorize the system in terms of navigation through the system to entering, proceeding and closing the system?	1 / 2 / 3 / 4 / 5
2	Could you understand the system functionality at the first time use?	1 / 2 / 3 / 4 / 5
3	How would you categorize the system based on speed?	1 / 2 / 3 / 4 / 5
4	How would you categorize the usefulness of the system for your daily travelling needs?	1 / 2 / 3 / 4 / 5
5	How easy to access the system through your mobile device?	1 / 2 / 3 / 4 / 5

Table 7.1 – Evaluation form to validate usability

TrackIT Main Web Application Evaluation in terms of functionality

Selectable options- 1 - Very poor, 2 – Poor, 3 – Average, 4 - Good, 5 - Excellent

	Question	Selectable Answers
1	Does the system provide enough information in when viewing schedule?	1 / 2 / 3 / 4 / 5
2	Does it show live location when you navigate to relevant page?	1 / 2 / 3 / 4 / 5
3	How would you categorize the accuracy predicted time based on your search results based on your experience?	1 / 2 / 3 / 4 / 5
4	When comparing with existing systems for	1 / 2 / 3 / 4 / 5

	the similar purpose, how would you categorize this system	
5	How much do you satisfied the overall functionality provided by this system?	1 / 2 / 3 / 4 / 5

Table 7.2 – Evaluation form to validate functionality

TrackIT Main Web Application Evaluation in terms of Overall Impression

Selectable options- 1 - Very poor, 2 – Poor, 3 – Average, 4 - Good, 5 – Excellent

	Question	Selectable Answers
1	Does system provide the final result within your expected time scope with expected quality?	1 / 2 / 3 / 4 / 5
2	Did you encounter any difficulty while accessing the system functionalities, how would you categorize system based on this feature?	1 / 2 / 3 / 4 / 5
3	Look and feel when application to be used regular basis	1 / 2 / 3 / 4 / 5
4	How do you categorize the system based on familiarity?	1 / 2 / 3 / 4 / 5
5	Rate the system based on your overall satisfaction about the system	1 / 2 / 3 / 4 / 5

Table 7.3 – Evaluation form to validate overall impression

TrackIT Location updater mobile application - Evaluation in terms of usability

Selectable options- 1 - Very poor, 2 – Poor, 3 – Average, 4 - Good, 5 - Excellent

	Question	Selectable Answers
1	How would you categorize the mobile application in terms of navigation through the system to entering, proceeding and closing the system?	1 / 2 / 3 / 4 / 5
2	Could you understand the functionality at the first time use?	1 / 2 / 3 / 4 / 5
3	How would you categorize the system usability in terms of starting, stopping in each trip?	1 / 2 / 3 / 4 / 5
4	How application/error messages are understandable?	1 / 2 / 3 / 4 / 5
5	How to you categorize, your attention to start/stop app in each trip?	1 / 2 / 3 / 4 / 5

Table 7.4 – Evaluation form to validate usability of mobile application

TrackIT Location updater mobile application - Evaluation in terms of functionality

Selectable options- 1 - Very poor, 2 – Poor, 3 – Average, 4 - Good, 5 - Excellent

	Question	Selectable Answers
1	Does the application provide enough information when start tracking and stopping?	1 / 2 / 3 / 4 / 5
2	Rate the application based on expected output/behavior	1 / 2 / 3 / 4 / 5
3	How would you categorize the application functionality based on live updates sent to main system?	1 / 2 / 3 / 4 / 5
4	When comparing with existing applications, how would you categorize this app?	1 / 2 / 3 / 4 / 5
5	How much do you satisfied the overall functionality provided by this application?	1 / 2 / 3 / 4 / 5

Table 7.5 – Evaluation form to validate functionality of mobile application

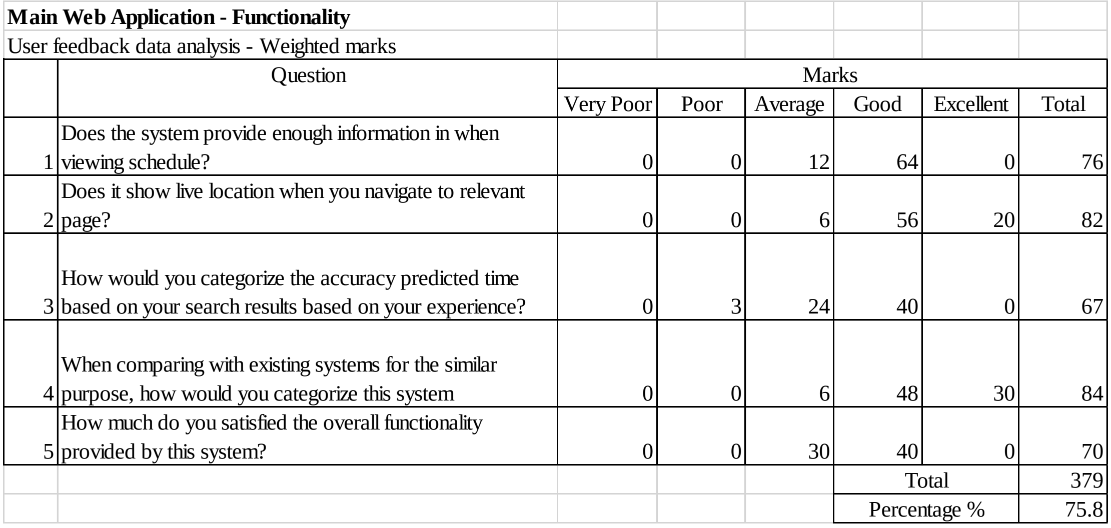
TrackIT Location updater mobile application - Evaluation in terms of Overall Impression

Selectable options- 1 - Very poor, 2 – Poor, 3 – Average, 4 - Good, 5 – Excellent

	Question	Selectable Answers
1	Does system provide the final result (Starting/Stopping) within your expected time scope with expected quality?	1 / 2 / 3 / 4 / 5
2	Did you encounter any difficulty while accessing the application functionality, how would you categorize system based on this feature?	1 / 2 / 3 / 4 / 5
3	Categorize the Look and feel when application to be used regular basis	1 / 2 / 3 / 4 / 5
4	How do you categorize the application based on familiarity?	1 / 2 / 3 / 4 / 5
5	Rate the application based on your overall satisfaction	1 / 2 / 3 / 4 / 5

Table 7.6 – Evaluation form to validate overall impression of mobile application

Evaluation Results

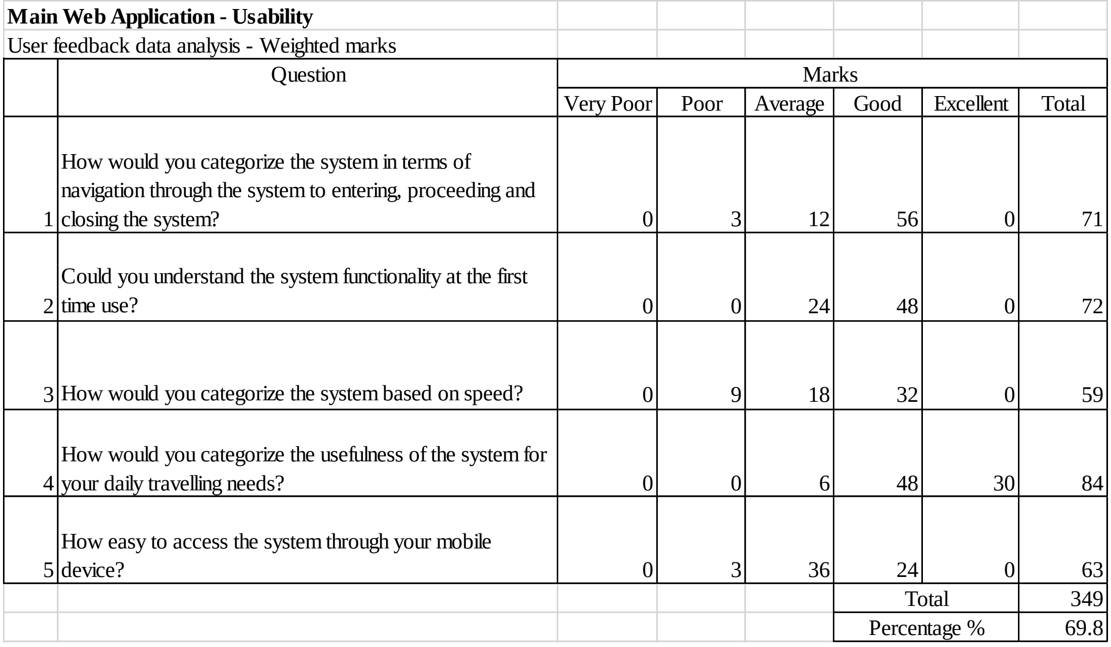


Main Web Application - Functionality

User feedback data analysis - Weighted marks

	Question	Marks					
		Very Poor	Poor	Average	Good	Excellent	Total
1	Does the system provide enough information in when viewing schedule?	0	0	12	64	0	76
2	Does it show live location when you navigate to relevant page?	0	0	6	56	20	82
3	How would you categorize the accuracy predicted time based on your search results based on your experience?	0	3	24	40	0	67
4	When comparing with existing systems for the similar purpose, how would you categorize this system	0	0	6	48	30	84
5	How much do you satisfied the overall functionality provided by this system?	0	0	30	40	0	70
					Total		379
					Percentage %		75.8

Table 7.7 – Evaluation results for main web application functionality

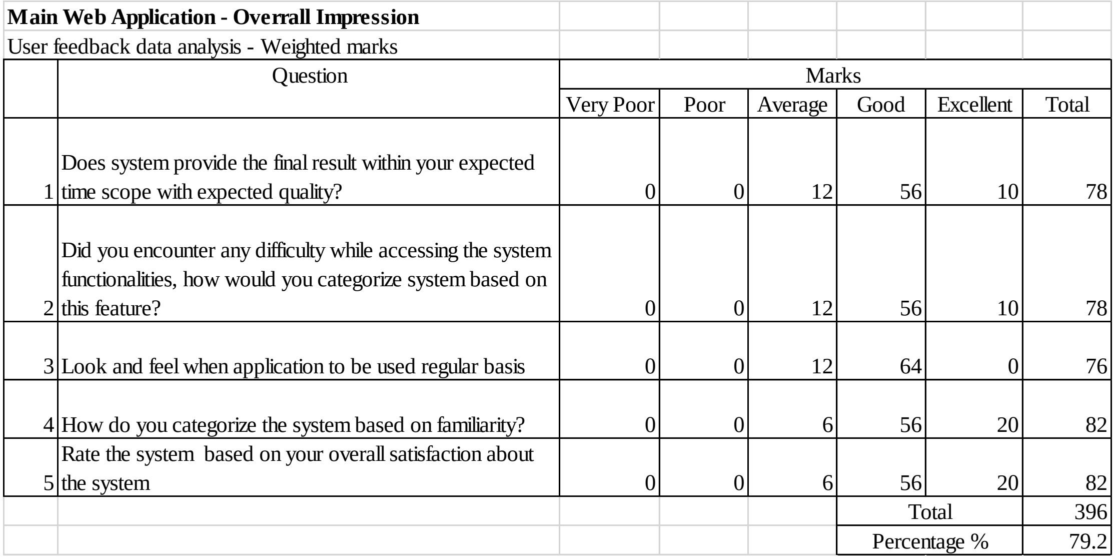


Main Web Application - Usability

User feedback data analysis - Weighted marks

	Question	Marks					
		Very Poor	Poor	Average	Good	Excellent	Total
1	How would you categorize the system in terms of navigation through the system to entering, proceeding and closing the system?	0	3	12	56	0	71
2	Could you understand the system functionality at the first time use?	0	0	24	48	0	72
3	How would you categorize the system based on speed?	0	9	18	32	0	59
4	How would you categorize the usefulness of the system for your daily travelling needs?	0	0	6	48	30	84
5	How easy to access the system through your mobile device?	0	3	36	24	0	63
					Total		349
					Percentage %		69.8

Table 7.8 – Evaluation results for main web application usability

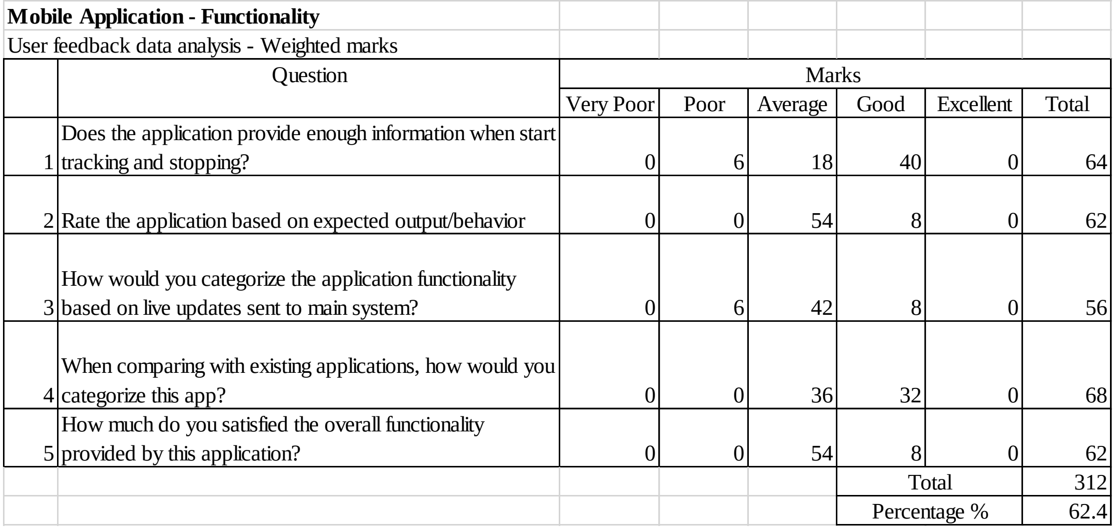


Main Web Application - Overall Impression

User feedback data analysis - Weighted marks

	Question	Marks					
		Very Poor	Poor	Average	Good	Excellent	Total
1	Does system provide the final result within your expected time scope with expected quality?	0	0	12	56	10	78
2	Did you encounter any difficulty while accessing the system functionalities, how would you categorize system based on this feature?	0	0	12	56	10	78
3	Look and feel when application to be used regular basis	0	0	12	64	0	76
4	How do you categorize the system based on familiarity?	0	0	6	56	20	82
5	Rate the system based on your overall satisfaction about the system	0	0	6	56	20	82
					Total		396
					Percentage %		79.2

Table 7.9 – Evaluation results for main web application overall impression

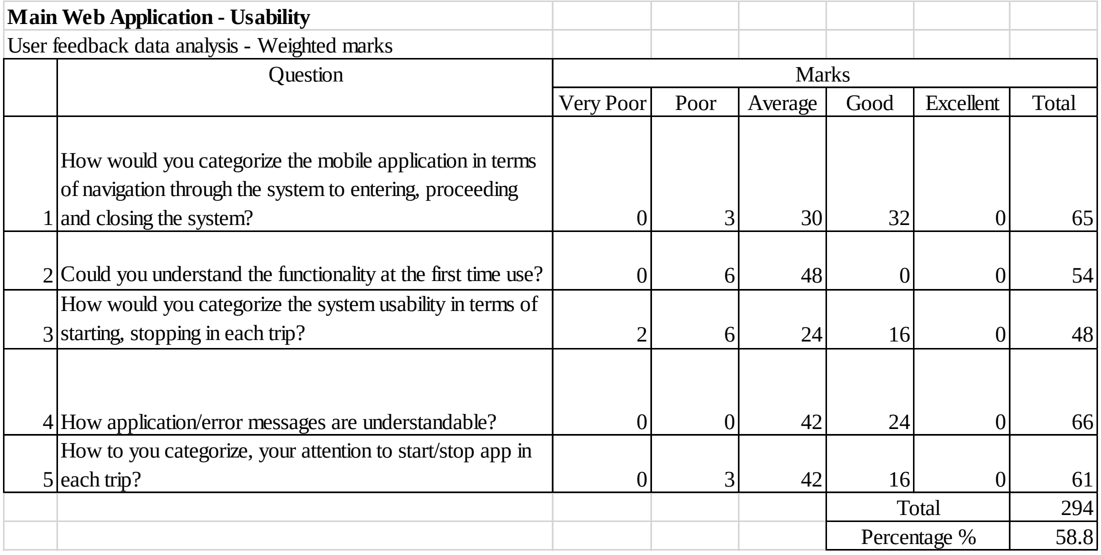


Mobile Application - Functionality

User feedback data analysis - Weighted marks

	Question	Marks					
		Very Poor	Poor	Average	Good	Excellent	Total
1	Does the application provide enough information when start tracking and stopping?	0	6	18	40	0	64
2	Rate the application based on expected output/behavior	0	0	54	8	0	62
3	How would you categorize the application functionality based on live updates sent to main system?	0	6	42	8	0	56
4	When comparing with existing applications, how would you categorize this app?	0	0	36	32	0	68
5	How much do you satisfied the overall functionality provided by this application?	0	0	54	8	0	62
					Total		312
					Percentage %		62.4

Table 7.10 – Evaluation results for mobile application functionality

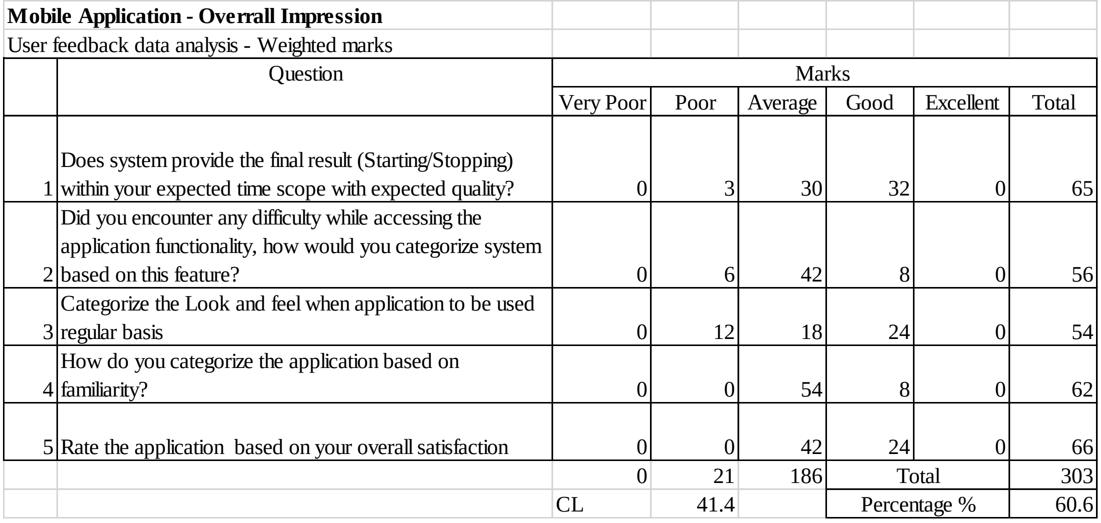


Main Web Application - Usability

User feedback data analysis - Weighted marks

	Question	Marks					
		Very Poor	Poor	Average	Good	Excellent	Total
1	How would you categorize the mobile application in terms of navigation through the system to entering, proceeding and closing the system?	0	3	30	32	0	65
2	Could you understand the functionality at the first time use?	0	6	48	0	0	54
3	How would you categorize the system usability in terms of starting, stopping in each trip?	2	6	24	16	0	48
4	How application/error messages are understandable?	0	0	42	24	0	66
5	How to you categorize, your attention to start/stop app in each trip?	0	3	42	16	0	61
					Total		294
					Percentage %		58.8

Table 7.11 – Evaluation results for mobile application usability



Mobile Application - Overrall Impression

User feedback data analysis - Weighted marks

	Question	Marks					
		Very Poor	Poor	Average	Good	Excellent	Total
1	Does system provide the final result (Starting/Stopping) within your expected time scope with expected quality?	0	3	30	32	0	65
2	Did you encounter any difficulty while accessing the application functionality, how would you categorize system based on this feature?	0	6	42	8	0	56
3	Categorize the Look and feel when application to be used regular basis	0	12	18	24	0	54
4	How do you categorize the application based on familiarity?	0	0	54	8	0	62
5	Rate the application based on your overall satisfaction	0	0	42	24	0	66
		0	21	186	Total		303
		CL	41.4		Percentage %		60.6

Table 7.12 – Evaluation results for mobile application overall impression