RFID Based Real-time Train Tracking System for Sri Lanka

Y.R.R.P. Kumara

179470B

Dissertation Submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the Partial Fulfillment of the Requirements for Master of Science in Information Technology

June 2020

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 Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.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 Student Signature of Student

Date: _ _ _ _ _ _ _ _ _ _ _ _

Supervised by

Mr. B. H. Sudantha

---------- ----------

Name of the Supervisor Signature of the Supervisor

Date: _ _ _ _ _ _ _ _ _ _ _ _

Dedication

This research paper is dedicated to my family members who helped and support me lot during the study and also dedicated to my friends who helped me in various ways while the study.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Mr. B. H. Sudantha, Dean, Senior Lecturer, Faculty of Information Technology, University of Moratuwa for his support and guidance throughout the entire research.

Special thanks to Mr. D. K. Withanage, Former Dean, Senior Lecturer, Faculty of Information Technology, University of Moratuwa for sharing his computer network knowledge with me.

My parents, Mr. Y. R. Linton Leelarathne and Mrs. P. K. Abeydheera should receive special thank for their support given to me in various ways during the study.

Finally, I should thank lecture panel of MSc in Information Technology, Faculty of Information Technology, University of Moratuwa.

Abstract

During day-to-day train operations in Sri Lanka, unforeseen events may cause train schedule inconsistencies. This causes train delays a habitual experience for Sri Lankan train passengers. This requires a real time train tracking system for the train passengers, in order to locate trains that they wish to travel. Based on accurate monitoring of train positions, potential delays can be predicted and inform the train passengers at real time via a comprehensive web application. Using this proposed system, passengers will be able to find out exact last location of a train and predicted arrival time to a particular railway station.

The real-time train tracking system uses Radio Frequency Identification as its location tracking technology. Each train is equipped with two Radio Frequency Identification tags and each railway platform at railway stations is equipped with a Radio Frequency Identification reader. Tracked train location details will send using Arduino wireless communication technology to the central database server. Then passengers will be able to access the real time train tracking details via the web application. Passengers can select the starting railway station and destination railway station from the web application. After that, web application will display currently available train details and their predicted arrival time to starting railway station selected by the passenger.

The web application also facilitates the system users (Ceylon Government Railway users) to perform administrative tasks such as add/edit trains, railway station details and setting-up some system parameters.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Prolegomena

This chapter provides an overall introduction to the research RFID Based Real-time Train Tracking System (RTTS). By looking at the chapter, readers will be able to grab an idea about following sub topics of the research. Background and Motivation, Problem Statement, Hypothesis, Objectives, Methodology and Structure of the Research.

1.2 Background and Motivation

Sri Lanka railways operates approximately 396 daily trains and it has been the transportation platform for about 3.72 million daily passengers [1]. The above numbers show the importance of railway transportation in Sri Lanka. Railway operations are complex with many interacting processes depending on technical equipment, human behaviors and external environmental conditions. Due to the above reasons, train delays become habitual experience for Sri Lankan train passengers. Unfortunately for train passengers, they have to waste hours or several minutes until a particular train is arrived.

Currently there is no comprehensive method available for train passengers to get information about delayed trains and predicted arrival time to a particular railway station. Although there are few static train schedules available in the internet, visualizing or tracking trains in real time has become a niche in Sri Lanka. Since static train schedules are just a written set of arrival and departure times, when considering a delayed trains static train schedules can be useless.

In order to track trains in real time, there should be a time and cost effective solution. The solution also should be a user friendly one. Unless, the solution will become useless as static train schedules.

Considering all of the above factors, this research proposes an RFID based Real Time Tracking System for Sri Lankan train passengers. Proposed system is a web based application which will provide real time information about trains for train passengers from particular railway station to another railway station. Using the system, train passengers can select their starting railway station and destination railway station and then the system will show currently available trains and their predicted arrival times.

The Real Time Train Tracking System uses RFID as its basic tracking technology. The web application has developed using ASP.Net MVC technology. Programming languages used in here are C#.Net and Javascript. Database that has been used in the

system is MS Sql Server. The system is built as a prototype for demonstration purpose. In the prototype, Arduino is used as the interactive platform.

1.3 Problem Statement

The train schedule is such an important part of day-to-day train operations for train passengers. Basically there are two types of train schedules namely Static train schedules and Real time train tracking systems. Real time train tracking systems are currently not available for Sri Lankan train passengers.

Static train schedules do not provide interaction between passengers and real time train statuses. When a passenger wants to know exact trains that are available from a particular train station, static train schedules are clearly inappropriate.

Problem with Global Positioning System (GPS) is the "accuracy". GPS tracking technology required minimum of four satellites to accurately track an object. If at least one satellite get fails, then tracking information will become a mesh. Because of above reasons, RFID tracking becomes the winner over GPS.

1.4 Hypothesis

Accurate real-time train tracking niche in Sri Lanka can be fulfilled if there is an effective train tracking system using RFID location tracking technology and web based data manipulation and user application.

1.5 Objectives

- To study and analyze current situation of real time train tracking in Sri Lanka
- To study and compare train location tracking technologies
- To develop a prototype for demonstration purpose
- To develop a web based solution for real time train tracking in Sri Lanka
- To develop a dashboard for administration operations of the web application
- To provide real time train statuses and prediction time to train passengers

1.6 Structure of the Research

The research contains 8 chapters. Each chapter has brief introduction, content of the chapter and chapter summary.

First chapter provides helicopter view about the entire research. Second chapter compare and analyze previous related researches to the topic and identifies gaps to be fulfilled.

Chapter number three introduces the tools and technologies used in the research while chapter four provides an approach to RTTS. That discusses input, process, output model of RTTS as a solution for real-time train tracking problem in Sri Lanka. Chapter five is about the design of RTTS which include software, network and hardware design. Chapter six provides implementation procedure of RTTS. Chapter seven expresses the evaluation of research. Finally chapter eight includes conclusion and further work of RTTS.

1.7 Summary

This chapter contains brief information about the research. The chapter summarizes importance and background of the real time train tracking in Sri Lanka. This chapter also introduces the Real Time Train Tracking System (RTTS) as the solution with the introduction to the technologies used in the solution.

Next chapter is about brief history and overall view of RFID and real time tracking technologies.

Chapter 2

Background and Development of Real Time Train Tracking Technologies

2.1 Introduction

First chapter introduced characteristics of this research and all the technologies related to the research. This chapter widely reviews previous studies related to RFID and real time train tracking technologies.

This chapter reviews brief history and evaluation of real time train tracking technologies. In order to review, researcher uses many researches done related to real time tracking.

2.2 Brief History

Real time train tracking is an area which has wide range of studies done based on many different technologies. In this section, brief history about RFID technology usages and real time train tracking will be focused.

A research has done to use RFID as localization technology, way back in 2004 [2]. Even though the paper was focused on mobile robots and persons, it provides powerful basement to tracking objects using RFID technology.

RFID based positioning and location tracking of trains is one out of many RFID based applications in railway industry around the globe [3]. In this paper, operation and maintenance, train axle temperature measurement, track inspection system and wagon monitoring system are mentioned as some other RFID based applications in railway industry.

Model predictive control, i.e. real time predictive scheduling (RTPS) is another technology which uses to track trains at real time [4]. The paper contains the ways of improvement of train operations based on intelligent decision support.

Programming approaches such as Interactive Convex Programming (ICP), nonlinear programming approaches, i.e., Mixed-Integer Nonlinear Programming (MINLP) are also alternative technologies used for real time train scheduling [5].

In Sri Lanka, the trends of real time train tracking are on about lowest level. There was a research done to track trains at real time using GPS technology in Sri Lanka [6]. The main issue with this research is tracking details are only visible to control center staff of Sri Lanka railways. That means public does not get any direct advantage using the system. The other issue is technology that the researchers have used. When it

comes to real time train tracking, accuracy becomes a must have feature. GPS might not accurate under certain conditions such as tall buildings and big trees might block GPS signal and reduce the number of satellite in view, and ground reflected multipath signals affect the received signal of GPS receiver [7].

When considering the above researches, clear fact is that there have been lots of studies carried out regarding RFID technology and real time train tracking in the past. Still there are plenty of areas to research about RFID and real time train tracking technologies.

2.3 Current Trends of Train Tracking

Passengers are getting lots of direct and indirect benefits from a real time train tracking system. By identifying that, there are lots of researches and practical implementations executed researchers and solution providers these days. Recently there have been lots of studies done relevant to RFID and real time train tracking technologies. This section includes analysis regarding recent trends and developments on above technologies.

Couple of Italian researchers has conducted a research to localize trains along the railway track using passive RFID technology at UHF band [8]. In this research, researchers have set up UHF-RFID reader and passive RFID tags are set up along the railway track at some identified positions. Then they have used relatively less complex computational algorithm to the reader position with respect to the tags themselves.

An RFID-Based Technique for Train Localization with Passive Tags [8] provides considerably accurate train localization technique using RFID technology and some mathematical algorithm.

In the year 2014, a research named An In-depot Realtime Train Tracking System Using RFID and Wireless Mesh Networks [9] have been done in order to provide safety of public transportation. The research [9] uses RFID technology to localize the trains current position and wireless mesh network to transfer the position information to the control center.

Sri Lanka Railway and University of Colombo's School of Computing (UCSC) had implemented a train tracking system in year 2014 for coastal line train passengers [10]. The Railway Traffic Optimization System (RTOS) had limited only for coastal line train passengers and now the system is no longer in use.

Sri Lanka railway has developed a train schedule with association of Information and Communication Technology Agency (ICTA) [11]. This is an eService which provides static data regarding train schedule in Sri Lanka. The eService can access through the web and that provides the platform independent facility to the passengers. Since the

eService shows only the static data, which is not the most suitable train schedule for passengers who expected real time train details.

The above researches prove that there is considerable trend on real time train tracking around the globe. But when it comes to Sri Lanka, even though recently there were few researches conducted towards the real time train tracking, almost every research could not show successful results. The real time train schedule gap still available with demand.

2.4 Summary

This chapter reviews the history, evaluation and current trends relevant to real time train tracking and RFID technology. The review brief the technologies used and also discussed the advantages and disadvantages of the researches.

Chapter 3

Technology Adopted for Real-time Train Tracking System (RTTS)

3.1 Introduction

Real-time Train Tracking System (RTTS) uses various technologies for different tasks within the system. Also there are few third party tools used to integrate different technologies used. Technologies include both software and hardware tools and some network devices. This chapter describes each technology and tools used in the system.

3.2 Location Tracking Technologies

RFID is used as primary location tracking technology in the system.

3.2.1 Radio Frequency Identification (RFID)

RFID includes two main components namely RFID tag and RFID reader. Digital data are encoded into RFID tags and data can be read by RFID reader via radio waves. RFID reader consists of two components which are scanning antenna and transceiver [12]. RFID tag consists of transponder which activates from radio frequency waves send by RFID reader.

There are many applications of RFID are available including location tracking, product tracking in supermarkets, highway toll payments etc. apart from location tracking, wide range of RFID applications are available in railway industry [3].

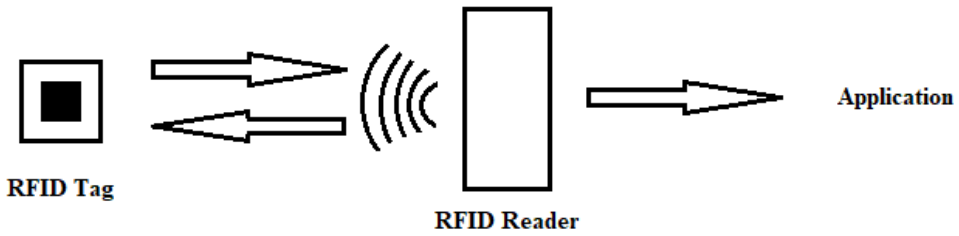

Figure 3. 1: How RFID Works

3.3 Data Transmission

Location data captured by RFID readers should transfer to the central database server. For data transmission purpose, the RTTS uses Arduino based wireless communication technology. Wireless communication technology is to make data transfer more efficient and easy [13]. HC-12 wireless serial port communication module is used to establish end-to-end data transmission process.

3.3.1 HC-12 Wireless Serial Port Communication Module

HC-12 is a wireless serial port communication module which has built in microcontroller in it [14]. The microcontroller can be configured using ATtention (AT) commands. Figure 3.2 shows an HC-12 module.

Figure 3. 2: HC-12 Wireless Serial Port Communication Module

3.4 Data Integration

Preparing collected RFID reader data to send from RFID reader side and preparing received data to store in the database belongs to data integration part. Arduino technology is used to accomplish data integration task. In RTTS, Arduino UNO board is used for both ends (Data collecting end and data receiving end) and RC522 module is used as RFID reader.

3.4.1 Arduino Technology

Arduino is an open-source platform which consists of programmable circuit board (microcontroller) and integrated development environment (IDE). Design goal of Arduino was to openly share details of microcontroller-based hardware design platforms to simulate the sharing of ideas and promote innovation [15]. Figure 3.3 represents Arduino UNO board used in the RTTS.

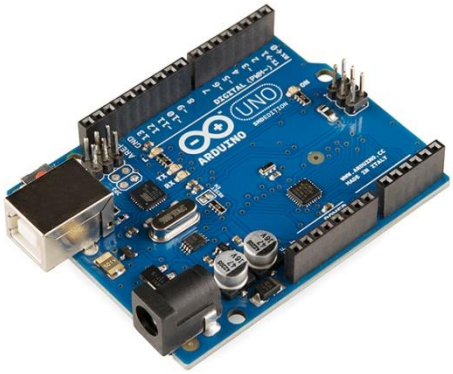

Figure 3. 3: Arduino UNO Board

3.4.2 RC522 Module

RC522 is short range RFID module. It creates 13.56MHz electromagnetic field to communicate with RFID tags. Low cost and low power consumption are certain advantages that can be obtained from RC522 module. Figure 3.4 represents an RC522 module.

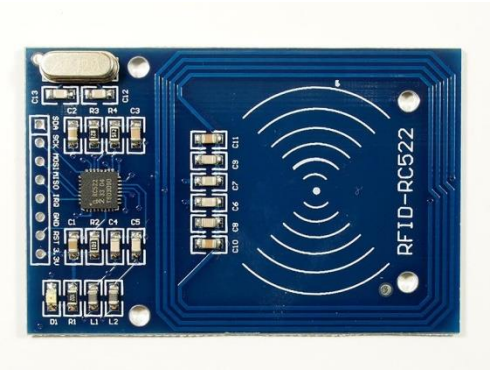

Figure 3. 4: RFID-RC522 Module

3.5 Data Storage

Microsoft SQL Server has used as database management system of the RTTS.

3.5.1 Microsoft SQL Server

Microsoft SQL Server is a database management system own by Microsoft Corporation. Sequential Query Language (SQL) is used to storing, manipulating and retrieving data from the MS Sql Server database.

3.6 System Backend

RTTS backend has developed using ASP.Net, MVC technology and C#.net has used as backend programming language.

3.6.1 ActiveX Server Pages (ASP.Net)

ASP.Net is a web development platform which contains tools, programming languages and libraries for web development [16].

3.6.2 Model View Controller (MVC)

MVC is a software development architecture which supports for many programming languages. Model represents the data retrieved from the database. View provides the interface for users to interact while Controller response for requests coming from the user.

3.7 System Frontend

Frontend of the system also referred as View of MVC architecture consists of frontend technologies such as jQuery, CSS and Bootstrap.

3.7.1 jQuery

jQuery is a library of client side scripting language called javascript. jQuery simplifies the javascript programming and provides cross-browser compatibility.

3.7.2 Cascading Style Sheet (CSS)

CSS defines the design of html elements in a web page. CSS allows create the design once and use it in multiple places.

3.7.3 Bootstrap

Bootstrap is a framework allows designers to create mobile-responsive web pages.

3.8 Chapter Summary

Chapter includes brief technical explanations of each technology and tool used in the RTTS.

Next chapter will discuss how RTTS provides the solution for real-time train tracking niche in Sri Lanka.

Chapter 4

An Approach to RTTS

4.1 Introduction

This chapter provides deep and clear understanding to the readers that how RTTS offer the complete solution for real-time train tracking problem in Sri Lanka. The solution has provided in order to fulfill the identified gaps in previous researches.

4.2 Hypothesis

Accurate real-time train tracking niche in Sri Lanka can be fulfilled if there is an effective train tracking system using RFID location tracking technology and web based data manipulation and user application.

4.3 Input

4.3.1 Collect Train Location Details

Primary input of the system is train location details. RFID technology is used to collect location details of trains. Each train is equipped with two passive RFID tags and each railway station is equipped with one RFID reader. Once train reach to a particular railway station, RFID reader captures the first RFID tag's details such as train number (train id), date, arrival time. Similar to that, once the second RFID tag passes RFID reader established in railway station, RFID reader captures train number (train id), date, and departure time.

4.3.2 Receive Requests from Users

User requests received via web application can be considered as secondary input to the system. User requests via web application contain passenger's start railway station and destination railway station.

4.4 Output

Real-time train tracking system can be considered as overall output of the research. The system includes prototype (RFID based location tracking), database and web application for both the system users and train passengers.

4.5 Process

RTTS whole procedure can be divided into set of sub tasks. Namely RFID based location tracking, data transmission, data integration and storage, and expose to passengers and system users via web application.

RFID based location tracking starts with setting-up two passive RFID tags in each train and one RFID reader in each railway platform in a railway station. Once a train is arrived to a platform of a railway station, first RFID tag on the train will be read by the RFID reader in railway platform as train arrival. When the second RFID tag is read, it will consider as train departure from that particular railway station.

Data transmission from railway stations to central server is done using the wireless communication. Each end (sender and receiver) there is a wireless communication modules are establish and each wireless module is connected to Arduino board.

Preparing the collected location tracking data to send and preparing received data to store is known as data integration. In order to achieve data integration task, here Arduino technology is used. Then the received data will be stored in the database.

Passengers can find out real-time train statuses via web application. If a passenger wants to start travelling from particular railway station to another railway station, then passenger can select “From” railway station and “To” railway station. Once passenger click on search button, the system will display up to three nearby trains that will arrive “From” railway station with predicted arrival times. System users/administrator (users from Ceylon Government Railway [CGR]) can also access the system via web application to perform administrative tasks such as maintain train, railway station, railway line details.

4.6 Users

Train passengers are the users who get direct benefits from the system. General public of Sri Lanka also get a chance to witness the real-time train statuses from the research.

4.7 Summary

This chapter explains how RTTS is providing the solution for real-time train tracking problem in Sri Lanka. The chapter includes input, output, process and users of RTTS and each section briefs in order to make understand overall system in clear and easy manner.

Next chapter includes the design of RTTS.

Chapter 5

Design of RTTS

5.1 Introduction

This chapter includes overall design of the RTTS. The design has broken into sections that include system architecture, Unified Modeling Language (UML) design, database design and User Interface (UI) design of web application. All the sections have been brief and figures are included when necessary.

5.2 RTTS Architecture

RTTS is a combination of different software and hardware components connected via wireless communication modules. Majority of system consists of high level technological components and it's somewhat difficult for general public understand the system. RTTS architecture provides overall understanding even for general public. Figure 5.1 displays the architecture of RTTS.

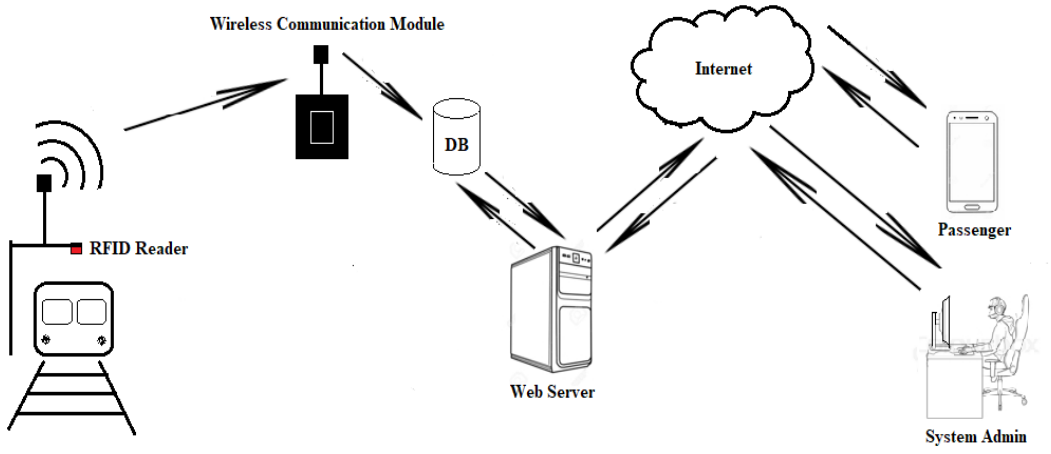

Figure 5. 1: RTTS Architecture

5.3 Unified Modeling Language (UML) Design

Unified Modeling Language (UML) is used to design the RTTS. UML is standard language for specifying, visualizing, constructing, and documenting the artifacts of software system [17]. Use-case diagram of the system is shown below.

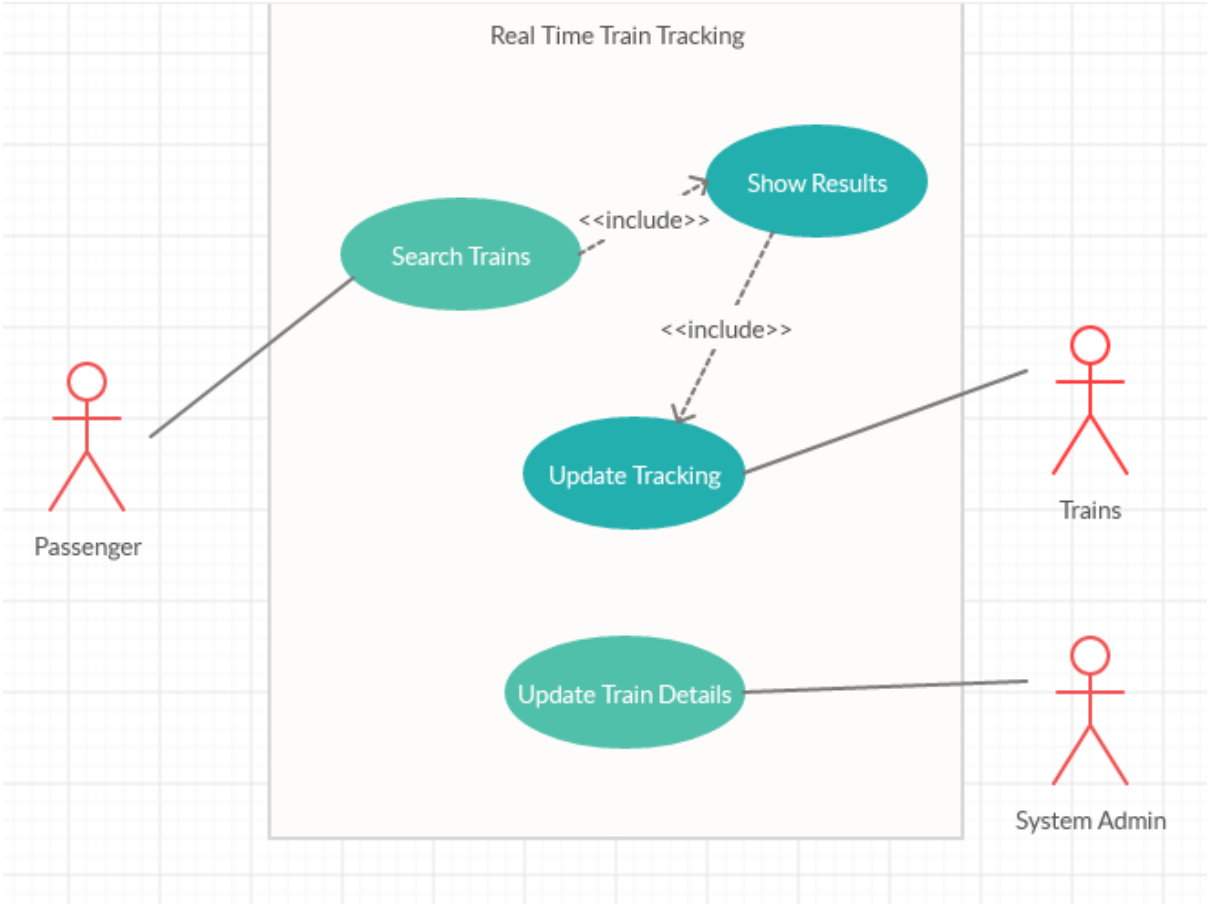

Figure 5. 2: Use-Case Diagram of RTTS

5.4 Database Design

Figure 5.3 represents the table structure of RTTS. As shown in Figure 5.3, the database has designed in simplified manner without making design complex.

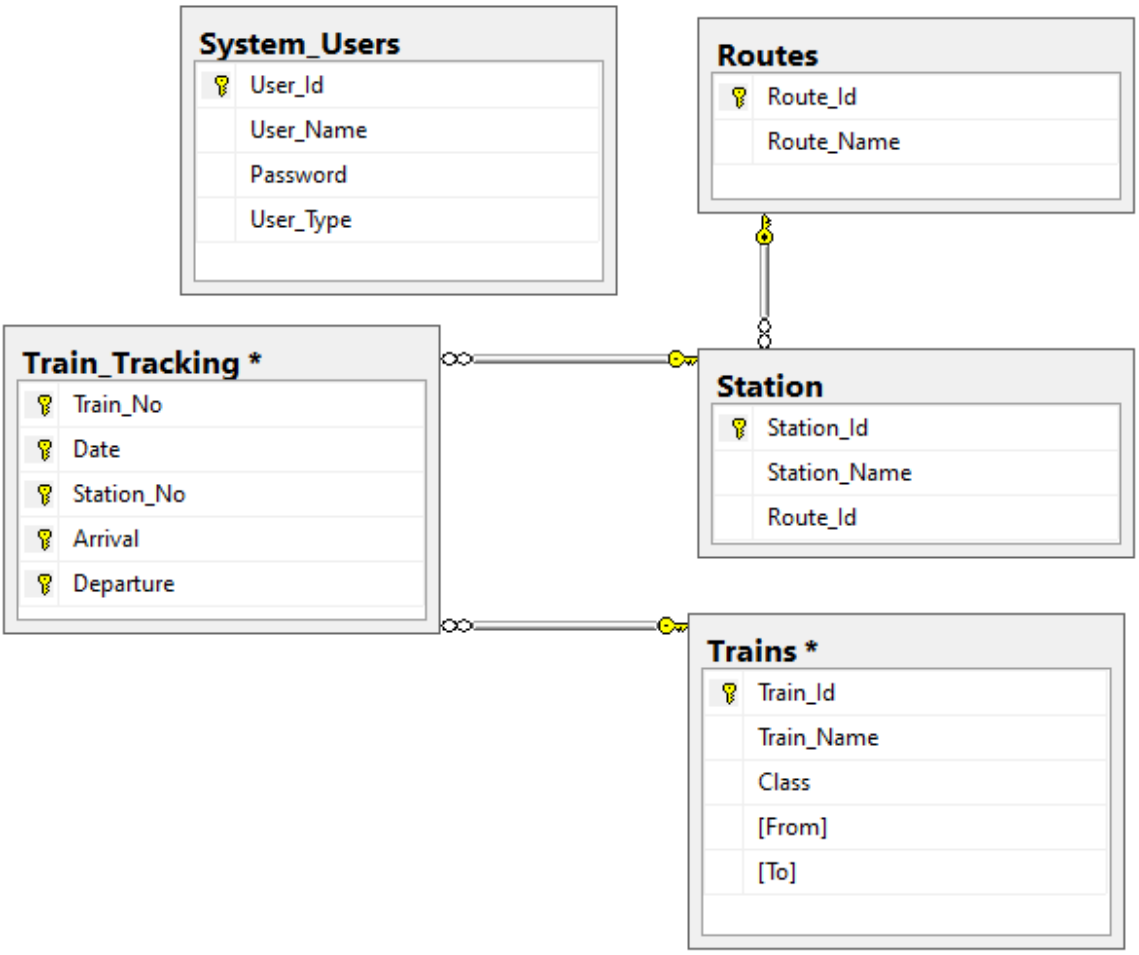

Figure 5. 3: Table Structure of RTTS

5.5 Summary

RTTS design has explained using different design aspects and different design technologies in the chapter.

Next chapter discusses practical implementation of RTTS according to the design provided in this chapter.

Chapter 6

Implementation of RTTS

6.1 Introduction

This chapter provides the implementation details of RTTS according to the design brief in the chapter 5. There are two main implementation phases available in RTTS, namely software implementation and hardware implementation. Both implementation phases are explained in this chapter including tools and technologies used to implement the RTTS.

6.2 Hardware Implementation

RTTS is a practical research. This means effectively developed software and hardware components are required to achieve expected results from the research. Therefor hardware implementation plays vital role in the research. Implement the hardware in real environment requires lots of resources such as financial resources, human resources. To minimize the above resource requirements and for demonstration purpose, hardware implementation is done as a prototype.

This section discusses the hardware prototype implementation of RTTS.

6.2.1 Location Tracking

RFID is used as location tracking technology of RTTS. In each train, two passive RFID tags are installed on the roof. First tag is installed in front side of the train and second tag is installed in rear side of the train. Comparison between active and passive RFID tags is showed in Table 6.1.

Table 6. 1: Comparison between Active and Passive RFID Tags

	Active RFID Tags	**Passive RFID Tags**
Power Source	Battery inside the tag	No battery inside the tag, powered by the RFID reader
Price	High	Comparatively low
Range Active	Long range	Short range
Active When	Always active	Active when powered by RFID reader
Lifetime	Comparatively shorter lifetime	Comparatively longer lifetime (Approximately 10 years)

Table 6.1, proves that passive RFID tags provide more benefits economically, power consumption wise and lifespan wise compared to active RFID tags.

Each railway track in a railway station is equipped with an RFID reader. RFID reader is mounted to a tower in order to read the RFID tags setup on the roof of the train. The RFID reader tower is situated in the middle of each railway platform. The reason for RFID tower to locate middle of railway platform is to read both the RFID tags on the train roof. Figure 6.1 represents how RFID reader and RFID tags are setup in railway platform and on the train.

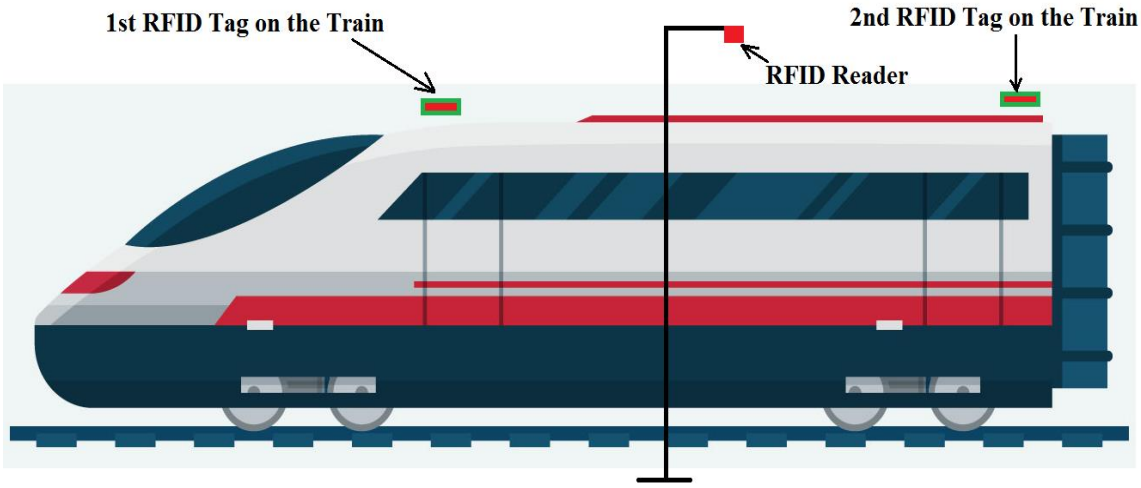

Figure 6. 1: RFID Reader and Tags Setup

6.3 Data Transmission

Transmitting location data captured by RFID readers to the central database server is known as data transmission. RTTS uses Arduino based wireless communication technology to accomplish data transmission process. HC-12 wireless serial port communication module is used to wirelessly transfer data between railway stations and central server.

6.4 Data Integration and Storage

6.4.1 Data Integration

Data integration is done using Arduino technology. For the demonstration of RTTS, Arduino UNO board has used. In the railway platform end, Arduino UNO board is

connected to an RFID-RC522 module while central server end Arduino UNO board connected to the database server.

6.4.2 Data Storage

Data received through Arduino UNO board are stored in a Microsoft SQL Server database. Database consists of many database objects including tables, stored procedures, triggers, etc.

6.5 Data Representation and Manipulation

Data representation refers to allowing train passengers to utilize the train tracking details via the web application. Data manipulation refers to allowing system users to access the system data via web application.

6.5.1 Data Representation

Front-end of the web application is developed using front-end technologies such as HTML, jQuery, CSS, and Bootstrap. Figure 6.2 expresses the main user interface for passengers (without search results).

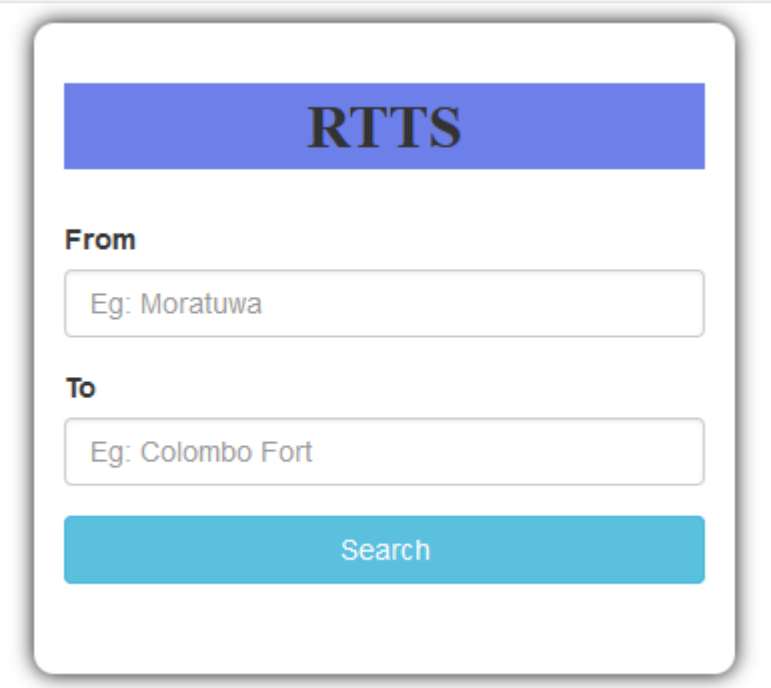

Figure 6. 2: Main User Interface for Passengers (Without Search Results)

Figure 6.3 expresses the main user interface for passengers (with sample search results).

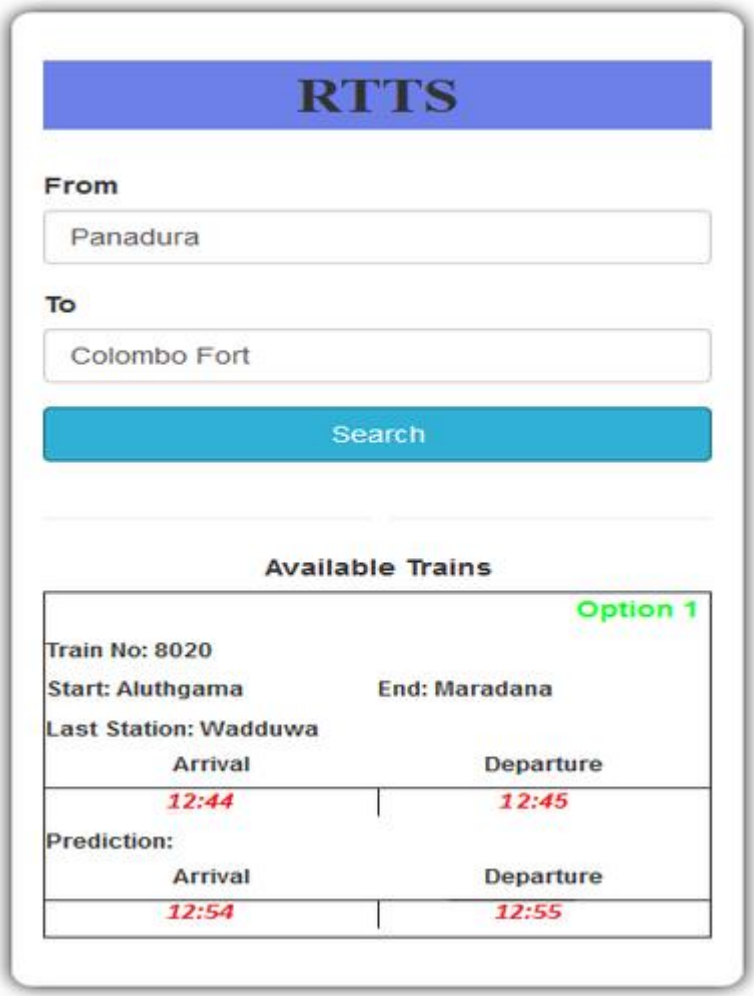

Figure 6. 3: Main User Interface for Passengers (With Search Results)

6.5.2 Data Manipulation

Data manipulation is done using an administrator panel. The administrator panel includes short-cuts to data manipulation sections such as setting up train details, setting up train station details, setting up railway network details, setting up train directions, access to the reports and other settings. By logging in to the system, the system users will be able to access each section on administrator panel based on their privileges.

Figure 6.4 shows the administrator panel of RTTS.

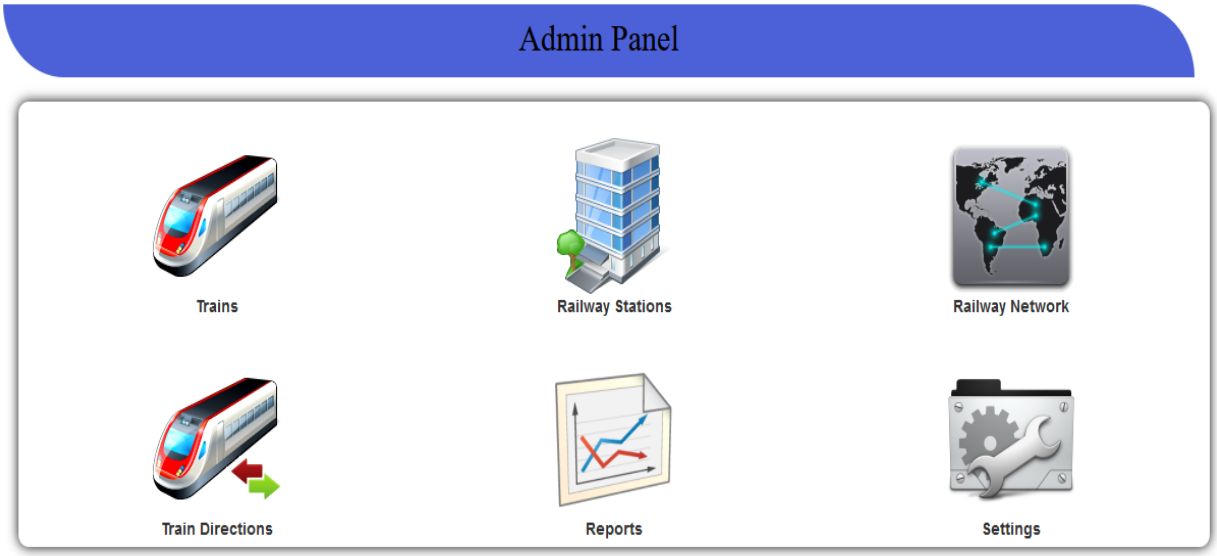

Figure 6. 4: Administrator Panel of RTTS

6.6 Summary

This chapter discussed implementation phase of RTTS. The implementation is done as hardware implementation and software implementation. All the tools and techniques used to implement the system have explained in the chapter.

Next chapter evaluates the RTTS using various aspects.

Chapter 7

Evaluation of RTTS

7.1 Introduction

Main purpose of the research is to provide a comprehensive web application to Sri Lankan train passengers for track the trains at real-time. Secondary purpose is to provide a web based interface for system users (Ceylon Government Railway users) to perform administrative tasks within the system. Within this chapter, results of the research will be evaluated in both passenger and system user perspectives.

7.2 Evaluation Methods

In order to evaluate the RTTS, the system should exposed to the both the system users. Namely train passengers and system users (Ceylon Government Railway users).

The RTTS has given to selected set of passengers in order to get a feedback about the RTTS passenger application. The system was publicly hosted in a server and sample set of data inserted into the database. Then a pre-defined set of questions in a questionnaire given to the passengers to answer. Questionnaire was designed to get passenger feedback in terms of functionality, usability and performance of RTTS's passenger application. Each question in the questionnaire, carried evaluation factors with assigned weighted marks to each.

Evaluation factors are shown below with assigned weighted marks in the brackets.

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

Following statistical method is used to analyze received feedback data from the passengers. Marks are calculated for each question in the questionnaire. Then average of the feature in each question is also taken. Critical line is defined and averages are compared with defined critical line.

No of users joined = N

No of questions on a feature = Q

Total marks per question = T

No of questions with answer Very Poor = VP

No of questions with answer Poor = P

No of questions with answer Average = A

No of questions with answer Good = G

No of questions with answer Excellent = E

Marks per one item = $(E * 10 + G * 8 + A * 6 + P * 3 + VP * 1)/N$

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

Critical line = 40%

7.3 Evaluation Results

Overall results of the evaluation process are represents using a table as well as line chart as below.

Table 7. 1: RTTS Passenger Application Evaluation Results

Application Feature	Feedback (%)
Functionality	72%
Usability	68%
Performance	78%
Average	**72.6%**

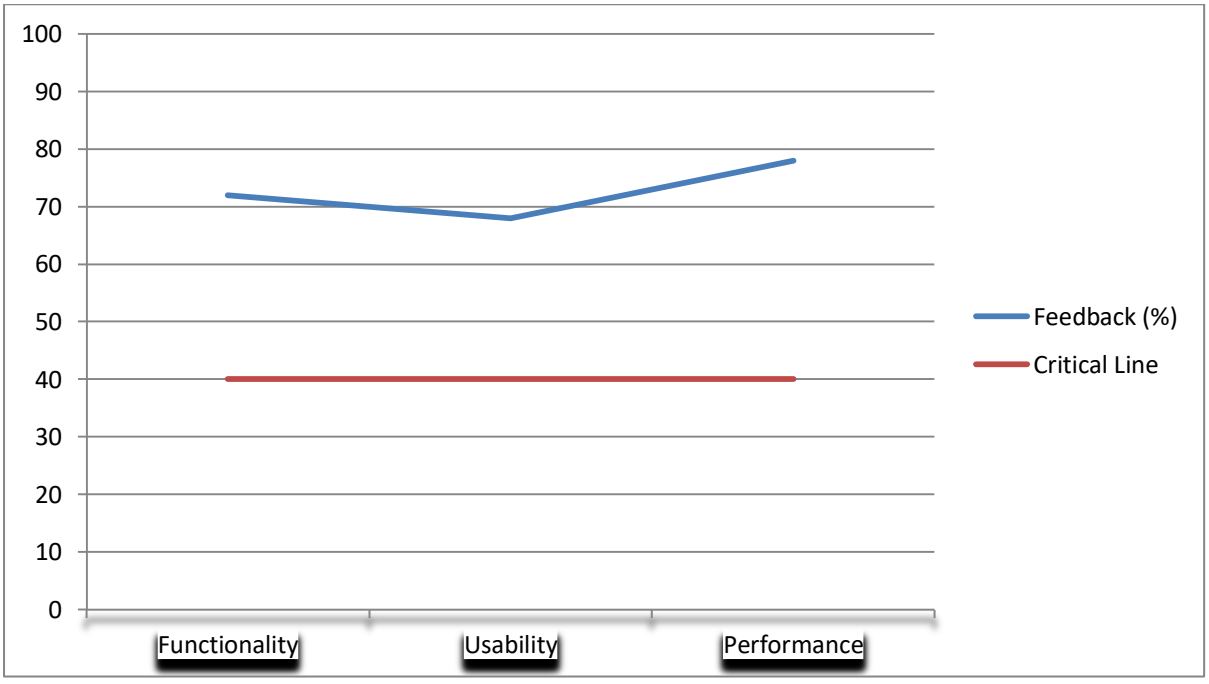

Figure 7. 1: RTTS Passenger Application Evaluation Results

Defined critical line is 40%. Results of overall evaluation features are clearly above the defined critical line. Therefore overall conclusion is that RTTS has achieved the expected results.

7.4 Summary

This chapter included overall evaluation process of RTTS. The results of the evaluation process have analyzed using a chart. Overall conclusion is that the RTTS has approached its target evaluation level.

Next chapter is the last chapter of the dissertation. That includes conclusion and further work of Real-time Train Tracking System.

Chapter 8

Conclusion and Further Work

8.1 Introduction

This chapter includes overall conclusion of the research, further work related to the study and limitations and obstacles identified during the study.

8.2 Overall Conclusion

This study is able to achieve expected evaluation level as shown in the chapter 7. Therefore the real-time train tracking niche in Sri Lanka has successfully fulfilled by the study.

The main location tracking technique used in the study is RFID technology. During the study, RFID is identified as accurate location tracking technology and identified comparative advantages over GPS location tracking [6].

Prototype is built for demonstration purpose of the study. The prototype is able to capture location of dummy trains, transmit location details to dummy central server over the dummy wireless network and store data on central database.

System users have given the facility to manipulate and utilize related data via the administrative panel. System users can perform functionalities such as setting up train details, setting up railway station details, setting up railway network details, setting up train directions (daily train operations) and other system related settings.

The passenger application is developed in order to allow train passengers access real-time train statuses from their browsers. Passenger application is achieved the target evaluation level in terms of functionality, usability and performance factors.

By considering above factors, the final conclusion of the study is that the overall study has achieved defined study objectives in chapter 1.

8.3 Further work

Application of the research findings and implement those in real world environment can be considered as extension to the study.

8.4 Limitations and Obstacles

Major limitation of the study is that RTTS is not available for public to access. That means the RTTS is not implemented in real world environment. Instead, it is developed for demonstration purpose only. Therefore the RTTS is exposed to limited amount of people (passengers).

To implement RTTS as publicly available service, engagement of different sections and coordination between different sections of Ceylon Government Railway (CGR) is required. That comes as major obstacle when considering currently available resources in Ceylon Government Railway (CGR).

8.5 Summary

This chapter discusses all the achievements of RTTS and compares achievements with study objectives. Actions required to enhance the study is included in the chapter. Also the identified limitations and obstacles are discussed within the chapter.

References

[1] Sri Lanka Railways, railway.gov.lk, About us, Overview. Available: http://www.railway.gov.lk/web/index.php?option=com_content&view=article&id=138&Itemid=176&lang=en. [Accessed: Jan. 10 2020].

[2] D. Hahnel, W. Burgard, D. Fox, K. Fishkin and M. Philipose, "Mapping and localization with RFID technology," *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, New Orleans, LA, USA, 2004, pp. 1015-1020 Vol.1.

[3] B. Malakar and B. K. Roy, "Survey of RFID applications in railway industry," *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*, Hooghy, 2014, pp. 1-6.

[4] P. Wang, Y. Zhou, J. Chen, Y. Wang and P. Wu, "Real-Time Train Scheduling and Control Based on Model Predictive Control," *2010 Second WRI Global Congress on Intelligent Systems*, Wuhan, 2010, pp. 47-50.

[5] Y. Wang, B. Ning, T. Tang, T. J. J. van den Boom and B. De Schutter, "Efficient Real-Time Train Scheduling for Urban Rail Transit Systems Using Iterative Convex Programming," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3337-3352, Dec. 2015.

[6] N. S. Gunasekara, "GPS Based Tracking System for Trains in Sri Lanka", M. S. thesis, Dept. IT, Univ. of Moratuwa, Moratuwa, Sri Lanka, 2006.

[7] E. Wang, W. Zhao and M. Cai, "Research on improving accuracy of GPS positioning based on particle filter," *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, Melbourne, VIC, 2013, pp. 1167-1171.

[8] A. Buffi and P. Nepa, "An RFID-based technique for train localization with passive tags," *2017 IEEE International Conference on RFID (RFID)*, Phoenix, AZ, 2017, pp. 155-160, doi: 10.1109/RFID.2017.7945602.

[9] Zongjian He, Yigui Luo and Junhao Zheng, "An in-depot realtime train tracking system using RFID and wireless mesh networks," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, 2014, pp. 2404-2409, doi: 10.1109/ITSC.2014.6958075.

[10] Sundaytimes, "Railway Traffic Optimization System (RTOS)", 2014.

[11] Sri Lanka Railways, railway.gov.lk, Train Schedule. Available: https://eservices.railway.gov.lk/schedule/searchTrain.action?lang=en. [Accessed: Mar. 15 2020].

[12] TechTarget, www.techtarget.com, IoTAgenda, Available: https://internetofthingsagenda.techtarget.com/definition/RFID-radio-frequency-identification. [Accessed: Mar. 15 2020].

[13] N. N. Lwin and K. T. Mya, "Analysis of Data Transmission on Wireless Processor to Processor Communication," *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, Nara, 2018, pp. 397-398, doi: 10.1109/GCCE.2018.8574493.

[14] D. P. Jose, A. L. D'Souza, A. A. Thomas and D. Daniel, "IoT Based Water Management Using HC-12 and Django," *2019 International Conference on Data Science and Communication (IconDSC)*, Bangalore, India, 2019, pp. 1-6, doi: 10.1109/IconDSC.2019.8816917.

[15] Steven F. Barrett, "Arduino I: Getting Started," in *Arduino I: Getting Started* , Morgan & Claypool, 2020.

[16] Microsoft, www.microsoft.com, .Net, What is ASP.Net, Available: https://dotnet.microsoft.com/apps/aspnet. [Accessed: Mar. 16 2020].

[17] TutorialsPoint, www.tutorialspoint.com, UML, Available: https://www.tutorialspoint.com/uml/index.htm. [Accessed: Mar. 19 2020].

Appendix A – Source Codes of RTTS

Source Code of RTTS Passenger Application

Front-end (Razor,JQuery,HTML)

```
@model  RealTimeTrainSchedule.Models.SearchTrains
@{
    ViewBag.Title = "Real Time Train Tracking System";
}

<script src="~/Scripts/jquery-3.4.1.min.js"></script>
<script src="~/Scripts/jquery-ui-1.12.1.js"></script>

<div class="container">
    <div class="col-sm-4">
    </div>
    <div class="col-sm-4">
        <div class="clsDivMainBorder">
            <div class="row" id="idDivMainTopRow">
                <label id="idLblMainTop">RTTS</label>
            </div>
            <div class="row" id="idDivMainSecondRow">
                <div class="form-group">
                    <label>From</label>
                    @Html.TextBoxFor(Obj => Obj.From, new { @class = "form-
control clsTxtMainSearch", id = "idTxtMainFrom", @placeholder = "Eg: Moratuwa"
})
                </div>
            </div>
            <div class="row">
                <div class="form-group">
                    <label>To</label>
                    @Html.TextBoxFor(Obj => Obj.To, new { @class = "form-
control clsTxtMainSearch", id = "idTxtMainTo", @placeholder = "Eg: Colombo
Fort" })
                </div>
            </div>
            <div class="row">
                <div class="form-group">
                    <button class="btn btn-info"
id="idBtnMainSearch">Search</button>
                </div>
            </div>
            <div class="row clsDivMainSearchResults">
                <hr />
                <label id="idLblMainTrAvb">Available Trains</label>
                <div id="idDivMainTS">
```

```
                    <div class="clsDivMainTrOpt" id="idDivMainTSRow1">
                        <div id="idDivMainTSOpt1">
                            <label class="clsLblMainOpts"
id="idLblMainOpt1">Option:</label>
                        </div>
                        <div id="idDivMainTSRow1Sub1">
                            <label class="clsLblMainTS"
id="idLblTrainNo1">Train No:</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow1Sub2">
                            <label class="clsLblMainTS"
id="idLblStart1">Start:</label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow1Sub3">
                            <label class="clsLblMainTS"
id="idLblEnd1">End:</label>
                        </div>
                        <div id="idDivMainTSRow1Sub4">
                            <label class="clsLblMainTS"
id="idLblLastStation1">Last Station:</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow1Sub5">
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainArr1">Arrival</label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow1Sub6">
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainDep1">Departure</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow1Sub7">
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainArrVal1"></label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow1Sub8">
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainDepVal1"></label>
                        </div>
                        <div id="idDivMainTSRow1Sub9">
                            <label class="clsLblMainTS">Prediction:</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow1Sub10">
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainPredArr1">Arrival</label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow1Sub11">
```

```
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainPredDep1">Departure</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow1Sub12">
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainPredArrVal1"></label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow1Sub13">
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainPredDepVal1"></label>
                        </div>
                    </div>
                    <div class="clsDivMainTrOpt" id="idDivMainTSRow2">
                        <div id="idDivMainTSOpt2">
                            <label class="clsLblMainOpts"
id="idLblMainOpt2">Option:</label>
                        </div>
                        <div id="idDivMainTSRow2Sub1">
                            <label class="clsLblMainTS"
id="idLblTrainNo2">Train No:</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow2Sub2">
                            <label class="clsLblMainTS"
id="idLblStart2">Start:</label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow2Sub3">
                            <label class="clsLblMainTS"
id="idLblEnd2">End:</label>
                        </div>
                        <div id="idDivMainTSRow2Sub4">
                            <label class="clsLblMainTS"
id="idLblLastStation2">Last Station:</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow2Sub5">
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainArr2">Arrival</label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow2Sub6">
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainDep2">Departure</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow2Sub7">
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainArrVal2"></label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow2Sub8">
```

```
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainDepVal2"></label>
                        </div>
                        <div id="idDivMainTSRow2Sub9">
                            <label class="clsLblMainTS">Prediction:</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow2Sub10">
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainPredArr2">Arrival</label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow2Sub11">
                            <label class="clsLblMainTS clsLblMainArrDep"
id="idLblMainPredDep2">Departure</label>
                        </div>
                        <div class="clsDivMainTwoColLeft"
id="idDivMainTSRow2Sub12">
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainPredArrVal2"></label>
                        </div>
                        <div class="clsDivMainTwoColRight"
id="idDivMainTSRow2Sub13">
                            <label class="clsLblMainTS clsLblMainArrDep
clsLblArrDepVal" id="idLblMainPredDepVal2"></label>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="col-sm-4">
    </div>
</div>

<div class="modal fade" id="idMdlError" role="dialog">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header clsMdlHdrError">
                <label id="idLblMdlHdrError">Error</label>
                <button class="close" data-dismiss="modal">&times;</button>
            </div>
            <div class="modal-body clsMdlBdyError">
                <label id="idLblError"></label>
            </div>
        </div>
    </div>
</div>

<script>
    var ErrMsg = '';
    var FSId = 0, TSId=0;

    $("#idTxtMainFrom").autocomplete({
```

```
        source: function (Request, Response) {
            $.ajax({
                url: '@Url.Action("GetSuggestions","Main")',
                dataType: 'Json',
                data: { Search: $("#idTxtMainFrom").val() },
                success: function (data) {
                    Response($.map(data, function (Item)
                    {
                        FSId = Item.FromId;
                        return { label: Item.From };
                    }));
                }
            });
        }
    });

    $("#idTxtMainTo").autocomplete({
        source: function (Request, Response) {
            $.ajax({
                url: '@Url.Action("GetSuggestions","Main")',
                dataType: 'Json',
                data: { Search: $("#idTxtMainTo").val() },
                success: function (data) {
                    Response($.map(data, function (Item)
                    {
                        TSId = Item.FromId;
                        return { label: Item.From };
                    }));
                }
            });
        }
    });

    $("#idBtnMainSearch").click(function ()
    {
        if (Validate())
        {
            $("#idLblError").text(ErrMsg);

            $("#idMdlError").modal("show");
        }

        LoadTrainTracking();
    });

    function LoadTrainTracking()
    {
        $.getJSON('@Url.Content("~/Main/LoadTrainTracking/")', { FromId: FSId,
ToId: TSId }, function (LstResults)
        {
            for (var i = 0; i < LstResults.length; i++)
            {
                var Val = parseInt(i) + 1;
                $("#idLblMainOpt1").text('Option ' + Val);
```

```
                var LblTrainNoId = '#idLblTrainNo' + Val;
                $(LblTrainNoId).text('Train No: ' + LstResults[i].Train_No);
                var LblStartId = '#idLblStart' + Val;
                $(LblStartId).text('Start: ' +
LstResults[i].Start_Station_Name);
                var LblEndId = '#idLblEnd' + Val;
                $(LblEndId).text('End: ' + LstResults[i].End_Station_Name);
                var LblLastStationId = '#idLblLastStation' + Val;
                $(LblLastStationId).text('Last Station: ' +
LstResults[i].Last_Station_Name);
                var LblArrValId = '#idLblMainArrVal' + Val;
                var HoursArr,MinutesArr,SecondsArr;

                HoursArr = LstResults[i].Arrival.Hours;
                MinutesArr = LstResults[i].Arrival.Minutes;
                SecondsArr = LstResults[i].Arrival.Seconds;

                $(LblArrValId).text(HoursArr + ':' + MinutesArr + ':' +
SecondsArr);

                var LblDepValId = '#idLblMainDepVal' + Val;
                var HoursDep, MinutesDep, SecondsDep;

                HoursDep = LstResults[i].Departure.Hours;
                MinutesDep = LstResults[i].Departure.Minutes;
                SecondsDep = LstResults[i].Departure.Seconds;

                $(LblDepValId).text(HoursDep + ':' + MinutesDep + ':' +
SecondsDep);

                var RealArrTime = (HoursArr < 10 ? '0' : '') + HoursArr + ':' +
(MinutesArr < 10 ? '0' : '') + MinutesArr + ':' + (SecondsArr < 10 ? '0' : '')
+ SecondsArr;

                var HoursExArr, MinutesExArr, SecondsExArr;

                HoursExArr = LstResults[i].Exp_Arrival_Curr.Hours;
                MinutesExArr = LstResults[i].Exp_Arrival_Curr.Minutes;
                SecondsExArr = LstResults[i].Exp_Arrival_Curr.Seconds;

                var ExArrTime = (HoursExArr < 10 ? '0' : '') + HoursExArr + ':'
+ (MinutesExArr < 10 ? '0' : '') + MinutesExArr + ':' + (SecondsExArr < 10 ?
'0' : '') + SecondsExArr;

                var TotalLateSec = (new Date("1970-1-1 " + RealArrTime) - new
Date("1970-1-1 " + ExArrTime)) / 1000;

                var LateHours = Math.floor(TotalLateSec / 3600);
                TotalLateSec -= LateHours * 3600;
                var LateMinutes = Math.floor(TotalLateSec / 60);
                var HoursPredArr, MinutesPredArr, SecondsPredArr;
                TotalLateSec -= LateMinutes * 60;
                HoursPredArr = parseInt(LstResults[i].Exp_Arrival_From.Hours) +
LateHours;
```

```
                MinutesPredArr =
parseInt(LstResults[i].Exp_Arrival_From.Minutes) + LateMinutes;
                SecondsPredArr =
parseInt(LstResults[i].Exp_Arrival_From.Seconds) + TotalLateSec;

                var DivId = '#idDivMainTSRow' + Val;
                $(DivId).show();
            }
        });
    }

    function Validate()
    {
        var FromVal = $("#idTxtMainFrom").val();
        var ToVal = $("#idTxtMainTo").val();

        var IsError = false;
        var ErrorMsg = '';

        if (FromVal == '' || ToVal == '')
        {
            IsError = true;
            ErrorMsg = '* From value or To value can\'t be empty\n';
        }
        else if (FromVal == ToVal)
        {
            IsError = true;
            ErrorMsg = '* From value and To value can\'t be same\n';
        }

        ErrMsg = ErrorMsg;
        return IsError;
    }
</script>
```

Front-end (CSS)

```
.clsDivMainBorder
{
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0px 0px 10px 0px #000;
    margin-top: 10px;
}
.clsLblMainTS
{
    font-size: 12px;
}
.clsLblMainOpts
{
    display: block;
    text-align: right;
    color: #00ff21;
```

```
    margin-right: 5px;
}
.clsLblMainArrDep
{
    display: block;
    text-align: center;
}
.clsLblArrDepVal
{
    font-style: italic;
    color: #ff0000;
}
.clsDivMainTrOpt
{
    border: 1px solid;
}
.clsDivMainTwoColLeft
{
    width: 50%;
    float: left;
}
.clsDivMainTwoColRight
{
    margin-left: 50%;
}
.clsMdlHdrError
{
    background-color: #87b7db;
    height: 40px;
}
.clsMdlBdyError
{
    background-color: #ffffff;
}

#idDivMainTSRow1, #idDivMainTSRow2
{
    display: none;
}
#idDivMainSecondRow
{
    margin-top: 20px;
}
#idLblMainTop
{
    font-size: 30px;
    font-family: 'Times New Roman', Times, serif;
    display: block;
    text-align: center;
    background-color: #6d7fe8;
}
#idBtnMainSearch
{
    width: 100%;
```

```
}
#idLblMainTrAvb
{
    display: block;
    text-align: center;
}
#idDivMainTSRow1Sub7, #idDivMainTSRow1Sub8, #idDivMainTSRow1Sub12,
#idDivMainTSRow1Sub13, #idDivMainTSRow2Sub7, #idDivMainTSRow2Sub8,
#idDivMainTSRow2Sub12, #idDivMainTSRow2Sub13
{
    border-top: 1px solid;
}
#idDivMainTSRow1Sub8, #idDivMainTSRow1Sub13, #idDivMainTSRow2Sub8,
#idDivMainTSRow2Sub13
{
    border-left: 1px solid;
}
#idLblMdlHdrError
{
    vertical-align: central;
}
#idLblError
{
    font-size: 12px;
    font-style: normal;
}
```

Back-end (C#.net)

```
using RealTimeTrainSchedule.Models;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Helpers;
using System.Web.Mvc;

namespace RealTimeTrainSchedule.Controllers
{
    public class MainController : Controller
    {
        RTTSEntities RTTS = new RTTSEntities();

        // GET: Main
        public ActionResult MainIndex()
        {
            return View();
        }
```

```
        public JsonResult GetSuggestions(string Search)
        {
            List<SearchTrains> LstStations = RTTS.Station.Where(Obj =>
Obj.Station_Name.Contains(Search)).Select(Obj => new SearchTrains
            {
                FromId = Obj.Station_Id,
                From = Obj.Station_Name
            }).ToList();

            return new JsonResult { Data = LstStations, JsonRequestBehavior =
JsonRequestBehavior.AllowGet };
        }

        public JsonResult LoadTrainTracking(int FromId, int ToId)
        {
            try
            {
                string ConnectionString =
System.Configuration.ConfigurationManager.ConnectionStrings["RTTSADO"].Connecti
onString;

                List<TrainTrackingModel> LstResults = new
List<TrainTrackingModel>();

                using (SqlConnection Con = new SqlConnection(ConnectionString))
                {
                    SqlCommand Com = new SqlCommand();

                    Com.Connection = Con;
                    Com.CommandType = CommandType.StoredProcedure;
                    Com.CommandText = "LoadTrainTracking";
                    Com.Parameters.Add(new SqlParameter { ParameterName =
"@FromStation", SqlDbType = SqlDbType.Int, Value = FromId });
                    Com.Parameters.Add(new SqlParameter { ParameterName =
"@ToStation", SqlDbType = SqlDbType.Int, Value = ToId });

                    Con.Open();

                    SqlDataReader DR = Com.ExecuteReader();

                    while (DR.Read())
                    {
                        TrainTrackingModel Obj = new TrainTrackingModel();

                        Obj.Train_No = DR.GetValue(0).ToString();
                        Obj.Date = DateTime.Parse(DR.GetValue(1).ToString());
                        Obj.Station_No = int.Parse(DR.GetValue(2).ToString());
                        Obj.Arrival =
TimeSpan.Parse(DR.GetValue(3).ToString());
                        Obj.Departure =
TimeSpan.Parse(DR.GetValue(4).ToString());
                        Obj.Last_Station_Name = DR.GetValue(5).ToString();
                        Obj.Start_Station_Name = DR.GetValue(6).ToString();
                        Obj.End_Station_Name = DR.GetValue(7).ToString();
```

```
                        Obj.Exp_Arrival_From =
TimeSpan.Parse(DR.GetValue(9).ToString());
                        Obj.Exp_Arrival_Curr =
TimeSpan.Parse(DR.GetValue(10).ToString());

                        LstResults.Add(Obj);
                    }
                }

                return Json(LstResults.ToList(), JsonRequestBehavior.AllowGet);
            }
            catch (Exception Ex)
            {
                throw Ex;
            }
        }
    }
}
```

Passenger Application SQL Stored Procedure

```
 USE [RTTS]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[LoadTrainTracking]
  @FromStation int, @ToStation int
AS
BEGIN
       DECLARE @FromTime time(7);

       SET @FromTime = DATEADD(MINUTE,0,GETDATE());
        ----------------------------------------------------

       SELECT Train_Tracking.Train_No,Date,Train_Tracking.Station_No AS
Station_No,Train_Tracking.Arrival AS Arrival,Train_Tracking.Departure AS
Departure,Station.Station_Name,Train_Operations.Start_Station,Train_Operations.
End_Station,dd.Station_Id,dd.Arrival,ss.actarr
       FROM Train_Tracking
       INNER JOIN (select Train_No,MIN(Station_No) as minno,max(Arrival) as
maxarr,max(Departure) as maxdep from Train_Tracking group by Train_No) tt
       on Train_Tracking.Train_No = tt.Train_No and Train_Tracking.Arrival =
tt.maxarr and Train_Tracking.Departure = tt.maxdep
       LEFT JOIN Station ON Train_Tracking.Station_No = Station.Station_Id
       LEFT JOIN Train_Operations ON Train_Tracking.Train_No =
Train_Operations.Train_No AND Opt_Date = CONVERT(VARCHAR(10), GETDATE(), 111)
       LEFT JOIN Stopping_Stations dd ON Train_Operations.Opt_Id = dd.Opt_Id
       left join (select hh.Arrival as actarr,hh.Opt_Id as op,hh.Station_Id as
si from Stopping_Stations hh) ss on dd.Opt_Id = ss.op and
Train_Tracking.Station_No = ss.si
       WHERE Date = CONVERT(VARCHAR(10), GETDATE(), 111) AND
Train_Tracking.Arrival <= @FromTime AND (Station_No > @FromStation AND
Station_No < @FromStation + 5) AND @FromStation = dd.Station_Id
END
```