

Decision Support Traffic Controlling System

Pabasara Jeewanthi Wijerathne

139188F

LIBRARY UNIVERSITY OF MORATUWA, SRI LANKA MORATUWA

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfilment of the requirements of the Honours Degree of Bachelor of Science in Information Technology.

March 2016

004 "16



M 3177

TH 3177 + I DUD ROM (TH 3160 - TH 3180)

Declaration

We declare that this thesis is our 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 Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.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

R.

P. J. Wijerathne

Date: 27/04/2016

Supervised by

UOM Verified Signature

Signature of Supervisor

Mr. Saminda Premaratne

Name of Supervisor

Date: 27/04/2016

Dedication

This dissertation is dedicated to my beloved mother who gave me endless courage whenever I was discouraged and to my family and all friends who gave me all the support and resources to achieve my tasks successfully.

Acknowledgement

I pay my heartiest gratitude to my project supervisor Mr. Saminda Premaratne on behalf of giving valuable guidelines and procedures on how to complete a project successfully and encouraging me with a suitable direction.

I also thank and appreciate Mr. Dasun Samarasiri, Mr. Gratien de Almeda and Ms. Thimali Yasoda for their enormous support and providing me the knowledge and resources to carry out the project effectively.

My heartiest thank goes to my parents who encouraged me to make this project much reliable and productive one.

Finally thanks a lot for lecturers, staff members, all friends and everybody who devoted their time on encouraging and advising me to make this project a great success.

Abstract

In Sri Lanka, Traffic congestion has been a critical problem over the years especially in urban areas which occurs mostly in peak hours and hence it has badly affected day to day life of the people. It increases fuel wastage, vehicle maintenance due to being in the traffic and consequently become responsible for loss of man hours and higher air pollution which will put people's health in danger in near future. As per the background research, the facts available on this traffic management area are lesser. It is found that there is no effective and accurate data collection mechanism and data analysing methods are currently being in use.

The prime mission of "Decision Support Traffic Controlling System" is to provide a solution for the current problem of traffic congestion in Sri Lanka. When it comes to the existing researches, majority of them are fully based on the GPS devices and make the conclusions. The uniqueness of this system is, it not only focusing on GPS data, but also considers accidents, events and road development information. Therefore accuracy of the result of real time analysis is higher compared to the other existing systems. In addition to that, all above data collected for real time analysis are used in prediction module. Therefore accuracy of the predictions will get increased gradually. These predictions allow the users to arrange their future travel schedules avoiding high traffic areas and also it helps the traffic police in decision making.

The system contains mobile application, web application, common backend and a database. Required data will be gathered from Police, Road Development Authority (RDA), GPS devices and event organizers. Some of the technologies used for implementation are Java, Spring framework, MongoDB, Google API, Google Map Direction API, Minitab and Time series model.

Contents

•

Chapter 1 - Decision Support Traffic Controlling System -DSTCS 1
1.1 Introduction1
1.2 Background and Motivation1
1.3 Aim
1.4 Objectives
1.5 Summary
Chapter 2 - Existing Systems
2.1 Introduction
2.2 Existing Systems
2.3 Summary
Chapter 3 - Overview of Technology
3.1 Introduction9
3.2 Java as a programming language
3.3 Android to develop mobile application
3.4 Google Map API
3.5 Spring framework for backend development10
3.6 MongoDB as the database
3.7 Minitab for data analyzing
3.8 Google Direction API
3.9 Summery
Chapter 4 - System Behaviour
4.1 Introduction
4.2 User Interaction with the system
4.2.1 Road Users
4.2.2 Traffic Police
4.2.3 Road Development Authority (RDA)14
4.2.4 System Operator14
4.3 Summary
Chapter 5 - Analysis and Design
5.1 Introduction
5.2 System Architecture

5.2.1 RDA interface (web interface)	17
5.2.2 Event interface (web interface)	18
5.2.3 Incident interface (Mobile Application)	18
5.2.4 View current traffic level (Mobile Application)	18
5.2.5 Report module	18
5.2.6 Data collection module	18
5.2.7 Real time analysis module	19
5.2.8 Prediction module	19
5.2.9 GPS devices	20
5.3 Summary	20
Chapter 6 - Development Methodology	21
6.1 Introduction	
6.2 Web Interface	22
6.3 Mobile Application	22
6.4 Backend	22
6.5 Real time traffic analysis module	23
6.5.1 Equations	24
6.5.2 Find nearest road	28
6.5.3 Find traffic level	31
6.5.4 Generate data	32
6.6 Prediction module	34
Chapter 7 - Discussion	52
7.1 Introduction	52
7.2 Difference from similar systems	52
7.3 Future work	52
7.4 Evaluation plan	53
7.5 Summary	53
References	54
Appendix A - Abbreviations	55
Appendix B - Web Interfaces	56
Appendix C - Source Code	57

List of Figures

	Page
Figure 1.1 - Traffic Statistics - No. of Road Vehicles & Road Length	02
Figure 2.1 - Traffic Report for Sri Lanka	07
Figure 4.1 - Use case diagram	15
Figure 5.1 - System Architecture	16
Figure 5.2 - System Diagram	17
Figure 6.1 - Work breakdown structure	21
Figure 6.2 - Select location	22
Figure 6.3 - Vehicle density in specific area	23
Figure 6.4 - Find locations around selected point	28
Figure 6.5 - Select nearest road	29
Figure 6.6 - Define vertical area	29
Figure 6.7 - Define horizontal area	30
Figure 6.8 - Selected vehicles	30
Figure 6.9 - Find distance	31
Figure 6.10 - Road segments	32
Figure 6.11 - Generate GPS data	32
Figure 6.12 - Activity diagram of real time analysis module	33
Figure 6.13 - Data collected area	34
Figure 6.14 - Time series plot against Avg Speed of vehicle and Avg vehicle count	35
Figure 6.15 - Steps in time series analysis	36
Figure 6.16 - Time series plot of average vehicle count	37
Figure 6.17 - Trend analysis plot of average vehicle count	38
Figure 6.18 - The mean of the series	40
Figure 6.19 - The variance of the series	40
Figure 6.20 - The covariance of the series	40
Figure 6.21 - Autocorrelation Function for Avg Veh.Count	41
Figure 6.22 - Partial Auto Correlation Function for Avg. Veh. Count	42
Figure 6.23 - Time series plot after 1st differencing	43
Figure 6.24 - ACF of 1st differenced data	43
Figure 6.25 - PACF of first differenced data	44
Figure 6.26 - Fitted model Vs Actual Time series	48
Figure 6.27 - Residual plots for Fitted model	48

Figure 6.28 - PACF for Residuals of fitted model	49
Figure 6.29 - PACF of residuals of fitted model	49
Figure 6.30 - Forecast for future average vehicle count	50

List of Tables

	Page
Table 6.1 - Fitting ARIMA Model	47
Table 6.2 - Forecasts from period 1536	50

Decision Support Traffic Controlling System -DSTCS

1.1 Introduction

The "Decision Support Traffic Controlling System" provides a solution for the current problem of traffic congestion in Sri Lanka. It supports road users and traffic police to reduce traffic congestion using real time analysis of data and making predictions. The chapter 1 describes background and motivation, aim and objectives of the research.

1.2 Background and Motivation

In Sri Lanka Traffic congestion has become a very critical problem especially in urban areas occur mostly in peak hours which has badly affected day to day life of the people. It increases fuel wastage, wear and tear of vehicles while being responsible for loss of man hours and higher air pollution which put people's health in danger in near future.

According to Civil Engineering and transport expert Professor Amal S. Kumarage, the University of Moratuwa, main reasons for congestion were as below [1].

- Poor city planning
- Inappropriate public transport facilities
- Insufficient traffic system

He has stated that the country was losing 1.5% of the Gross Domestic Product (GDP) due to traffic congestion that creates a huge impact on Sri Lankan economy. Currently the vehicle population is 3.2 million in Sri Lanka. "If we expect the economy to grow at 8%, the country's vehicle population may increase to 5 million," he said adding that there will be 250 vehicles per 1,000 people which are likely to be concentrated around the city. Therefore the demand for road space will increase rapidly.



Road Length & Vehicle Population	2008	2009	2010
No. of Motor Vehicles (10,000)	3390993	3595068	3954311
Total Road Length (10,000 km)	11696.9	11696.9	12487.92
A Roads	3724.26	3724.26	4222
BRoads	7505.76	7505.76	7799
AB Road	466.92	466.92	466.92

Figure 1.1 - Traffic Statistics - No. of Road Vehicles & Road Length [2]

As per the road users' perspective, the unplanned and complicated construction of flyovers and highways would not address the prevailing conditions of heavy traffic in the city of Colombo. As an example the Dehiwala flyover bridge has failed to reduce the traffic on Galle Road. During peak hours traffic jam can still be seen even after the police made it one way for both lanes. This has shifted the congestion to other areas such as Kirulapana, Narahenpita and Maradana.

According to the Lt. Col. Chandana J.S Weerakoon who is a specialist in transportation systems and management, the transportation business in Sri Lanka is on a very low scale. There is no sound planning process being activated in a practical way due to the unavailability of accurate research data on the transportation sector. Therefore, he emphasizes that Ministry of Transportation should have insisted on forming an organization for research and data collection in order to reach careful planning, implementation and other decision-making in this regard [3].

In addition to that he has noted that in most Western countries, 12% GDP is earned by the transportation sector. Therefore, it is important that Government should be looking into this main sector in terms of development, economic management and facilitating the general public to save time. Then hopefully the government will be able to make more profits by providing safe and comfortable transportation mechanism while saving lot more energy. However in Sri Lanka there was no proper traffic controlling mechanism with accurate data collection techniques and data analysing methods ever existed. This has badly affected to the country's economy and became a bad influence in time and resource management. The proposed Decision Support Traffic Controlling System (DSTCS) is introduced to address such problems in a better manner while providing practical solutions for them.

1.3 Aim

Provide a solution for the current problem of traffic congestion in Sri Lanka by developing the proposed system "Decision Support Traffic Controlling System" that support road users and traffic police to reduce traffic congestion using real time analysing of data and predictions.

1.4 Objectives

- To identify current issues with road users and police regarding traffic congestion in Sri Lanka.
- To discover what stimulates traffic flow and understand the cities/areas with frequent traffic congestions.
- To develop an Android mobile application and web base application to gather required data from traffic police, road users, event organizers, GPS devices and Road Development Authority (RDA).
- To discover most suitable equations to find level of the traffic in particular area based on GPS device data, construction zones, events, accidents or other road disruptions Real time analysis.
- To introduce a suitable model for predictions to help traffic police, RDA and road users in decision making and future planning.

1.5 Summary

As per the background research there is no proper traffic controlling mechanism available in Sri Lanka and it creates a huge impact on country's economy The "Decision Support Traffic Controlling System" identifies existing issues in current plans used to control traffic, cities/areas with frequent traffic congestions, current problems with road users and police and provides practical solution for them. It supports road users for finding traffic level in particular area and helps traffic police, RDA and road users in decision making and future planning. The chapter 2 explains similar researches done and how DSTCS differs from them.

Existing Systems

2.1 Introduction

The chapter 1 described the background and motivation, aim and objectives of the research. In the section of "background and motivation" explained current issues occurred due to traffic congestion in Sri Lanka and other two sections explained in terms of goal and objectives of the "Decision Support Traffic Controlling System". This chapter discussed similar researches done and how DSTCS differs from them. Some of the existing researches are SATNAV system, T-Navi, Sirasa FM – Broadcast Media, "road.lk's" Traffic Application, CityGuide Live Traffic Information System and ANT System.

2.2 Existing Systems

In present days there are few traffic alert systems in Sri Lanka to support road users by providing real time traffic alerts.

SATNAV System

Dialog Plc makes real time traffic alerts through SATNAV system. This system delivers automated and voice supported navigation with the help of GPS data. Main functions of it are providing live traffic alerts, breakdown or road construction alert and special alerts made by Sri Lanka police such as road closure, detours etc. It is developed with a detailed map of Sri Lanka and 3D building views are also available in this map. Voice support is available in three main languages that are English, Sinhala or Tamil. In addition to that, this system supports a range of mobile devices because of its sophisticated technology [4]. However this system can be considered as a commercial product and no data analysis and prediction methods included.

T-Navi

Another GPS system called T-Navi is introduced by Mobitel to provide road users with live updates on traffic jams or road congestions. Mainly T-Navi solution is available in three formats.

- Mobile application for SMART phones
- PND (Portable Navigation Device) to be fitted in vehicles
- Software SD card for 2DIN devices

However it is a requirement that both PND and the software SD Cards should be purchased from Mobitel branches. This system consists of 3D moving map and voice guidance components. Map indicates static yet dynamic places of interest. Voice guidance is available in English as well as in Sinhala. The map automatically gets updated when new roads come up. The major advantage of T-Navi system is that it offers turn-by-turn direction to users and at the same time it directs users to required destination through routes based on traffic condition. However this system works in SMART phones on IOS (APPLE OS), Windows Mobile, Android and Symbian operating systems only and no data analysis and prediction mechanism is used.

There are more examples from foreign countries which have already addressed this problem. In Switzerland, they have developed a system which is very much similar to T-navi solutions in Sri Lanka. The target audience is only SMART mobile phone users [5].

"road.lk's" Traffic Application

The road.lk's traffic application provides real-time traffic updates by frustrated tweeps are grabbed off Twitter, marked on a map and pushed right to Android smartphone. The users can check on the particular website or visit road.lk's FaceBook page. Road.lk entitles to provide real-time traffic updates all over the country and this end has a search feature built in to show the traffic alerts by locality. The updates of the system depend on the Twitter population which is significantly lesser in the outlying regions [6].



Figure 2.1 - Traffic Report for Sri Lanka [7]

CityGuide Live Traffic Information System

CityGuide Live Traffic Information System is introduced by Myriad Digital through the CityGuide GPS Navigator. This system allows the road users to avoid traffic jams, unpredicted one way road plans as well as avoid any unexpected sudden traffic plan changes and road blocks put up by police. In this system all the information is updated automatically so that users may get quick updates during the drive. In addition to these features, one of the other advantages is if there are any unforeseen emergency changes in roads, road traffics etc., will be instantly communicated to the CityGuide users. Key features of CityGuide are,

- The map with latest updates and 3D buildings
- Free and automatic live traffic updates
- Instant routing error corrections/Automatic map updates
- Option to add dynamic point of interests (DPOI) such as bad roads, police, incident, danger, text and ability to share this information with other users.

When it comes to hardware requirements, there is navigating equipment which should be fixed to vehicle. This equipment receives relevant information through the satellite system instantly and this navigator has the facility of getting live traffic data directly from their server [8]. However since this is a commercial product buying the application may cost the user about Rs 36,000 and also no data analysis and prediction methods are used.

ANT System

ANT system is introduced as a result of the research conducted in Moratuwa University that can be used by any mobile phone user (both SMART and non-SMART phone users). The traffic condition evaluation is carried out through the GPS data collected from GPS devices fixed in cabs during a specific time period. The core output of the system is to send a message to the users' mobile phone to make them aware of the road traffic situations, best possible alternative routes, anticipated cost, time and distance to avoid traffic. Besides, ANT system also performs cost, distance and velocity calculations before informing the best and optimal alternative route though alerts [9]. However no prediction module is integrated with this system.

As per the ground research information I have collected, the facts available on this traffic management area are lesser. It is found that there is no effective and accurate data collection mechanism and data analysing methods are currently being is used. However currently many more researchers working on this area and various systems are already being developed. Yet it is a present requirement that a technically sophisticated system needs be developed to address traffic congestion issues and especially to predict potential traffic congestions in future allowing the users to arrange their future travel schedules avoiding predicted traffic areas and also it helps the traffic police for decision making. Therefore this research introduce most suitable Time Series model for data analysing and decision making. Required data will be gathered from Police, Road Development Authority (RDA), GPS devices and event organizers.

2.3 Summary

Currently many researchers are being conducted on this area and several systems are already being developed. However Sri Lanka it is in a critical situation since traffic is getting increased daily in an alarming rate. By addressing problems in existing systems and current traffic plans, DSTCS introduces a Time Series model for data analysing and decision making. It allows the users to arrange their future travel schedules avoiding predicted traffic areas and traffic police officers are able to use predictions for decision making. The chapter 3 describes the technologies used to implement DSTCS.



Overview of Technology

3.1 Introduction

The existing systems and how "Decision Support Traffic Controlling System" is differing from them is explained in chapter 2. This chapter describes what are the technologies have been selected and why and how those are used in the system. The chosen technologies are Java, Spring framework, MongoDB, Google Map API, Google Direction API, Android, Minitab and Time series model.

3.2 Java as a programming language

Java is the programming language that is being used due to below reasons,

- Many open source tools are available such as IDEs, web servers and build tools. Among them I use, Eclipse as IDE, Appache tomcat as web server and Maven as the build tool.
- Java is well supported with other technologies that I used here, such as Google API and Android.
- There are many support documents available in web, therefore it is much convenient to discover and learn solutions for programming issues that may occur during the work.
- One of the main advantages of Java is good performance. It is very important to use a language which has adequate performance since this system has complex logics and algorithms.

3.3 Android to develop mobile application

Day by day Android mobile usage market is getting higher and it is less expensive when compared with other operating systems supported phones such as iOS/iPhones. Hence this system is critical in terms of performance, used native android for mobile application development. In addition to that Google provides well supported Map API for Android and DSTCS is needed to integrate Google Map with Android mobile application.

3.4 Google Map API

Google Map API provides web and Android API to visualize maps and access rich mapping features such as accurate directions, street views and more importantly it is free for wide variety of use cases. Thus Google Map API is well suited for this project. There are well prepared API documents, tutorials and learning materials are available for Java and also it is more flexible and well support customization. The Google map is integrated to facilitate road users, police, system operator and RDA for selecting locations.

3.5 Spring framework for backend development

Spring is one of a framework which use MVC (Model View Controller) pattern that separate presentation from business logic. It helps to "wire" different components together and simplify the code. The DSTCS consists with complex logics and algorithms used for real time analysis and prediction. Therefore main purpose of using Spring framework is to reduce the complexity of the program and make the overall project less complex. Along with Spring framework used JSP, Servlet, XML and JSON for developments. MongoDB is easy to integrate and configure with Spring, that provides the MongoTemplate class to handle the database related operations [10]. In addition to that there are many learning materials are available in web.

3.6 MongoDB as the database

The MongoDB is open-source NoSQL Document Oriented database and it save data in the form of document and collections like JSON. MongoDB is consider as general purpose database and well suited for Web, Mobile and Real-Time Analytics. Here the related information is stored together for fast query access through the MongoDB query language. For real time analysis it is very important to have a speed data access, because when a road user requests on the traffic level, the system should be able to do the analysis based on the data collected and saved in the database at once. Here, the response time of road user request is very important. Further the amount of data saved in the database is getting increased as the prediction module system collects data on daily basis and MongoDB is capable of handling these kinds of problems. In MongoDB the structure of records can be changed simply by adding new fields or deleting existing ones. This gives ability to represent hierarchical relationships, to store arrays, and other more complex structures easily. MongoDB is also designed with high availability and scalability features. It does not require DBA (Database Administrator) to modify schema when application changes. It saves implementation and maintenance time and cost. In addition to that it provides some features such as Rich Data Model, Dynamic Schema, Data Locality and Auto-Sharding which are not provided by other databases like MySQL [11].

3.7 Minitab for data analyzing

The Minitab is a statistical data analysis tool developed by Pennsylvania state university. The main reason for using Minitab in this research is its rich set of statistical analysis methods. In this research there is a greater need of analysis and identifying data and patterns in large scale traffic data sets. Using Minitab we can easily analyze data sets for certain properties. Also it can be used to identifying patterns in data sets. In the context of this research we have used a traffic data set regarding to Denmark to build a traffic prediction model. We used Minitab to analyze that data set for its applicability to feed it to a time series model. Also we use Minitab to identifying traffic patterns in that dataset.

3.8 Google Direction API

Google direction API is an API provided by Google to obtain route details. The API encapsulate various data properties regarding to a route between given locations such as road name, road steps, road polylines, estimated time for arrive to destination etc. The primary reason for using direction API is it's easiness of usage. Also this API provides much of the data that we need for our various research requirements. At most of the times we can decompose the original data received by the API to required levels easily and that is also a major reason to use this API. In the context of this research user select a particular location for to get traffic level regarding to that location. We use direction API to get the road details in between that location and another custom location given by the system. As the road details API gives us the road steps and road step polylines. We use road step polylines to determine the validity of the location which is selected by the user.

3.9 Summery

The Java, Spring framework, MongoDB, Google Map API, Google Direction API, Android, Minitab and Time series model are the technologies used for the system implementation. These technologies are selected based on their attributes and appropriateness for the system. The chapter 4 will describe system users with inputs, outputs and how users interact with the system.

Chapter 4

System Behaviour

4.1 Introduction

The chapter 3 explained technologies used for the system implementation and why they are chosen based on which attributes. This chapter describes system users, their inputs to the system, output from the system and their interaction with the system. Main users of the system are police, road users, RDA (Road Development Authority) and system operator.

4.2 User Interaction with the system

4.2.1 Road Users Inputs to the system

Road users can request current traffic level in a specific area through the map integrated in mobile application. If they want to know traffic level in a future date they can request it via web interface.

Outputs from the system

Road users can see current traffic level in particular area via mobile application and also predictions on potential traffic congestions with construction zones, events or other road disruptions etc. in future to schedule their future travel plans via web interface.

4.2.2 Traffic Police Inputs to the system

Incident information

The traffic police are able to use mobile application to update the system on accidents, incidents, road closure and etc. with traffic level of that area. The system uses this information for real time analysis.

Select areas to predict traffic

The System provides a web interface for Police to select areas to predict traffic and these areas are considered when choose zones to collect GPS data for predictions.

Outputs from the system

Reports including predictions

Future predictions help traffic police to make future plans. Through the reports provided by the system they can be aware of which places are vulnerable for more traffic congestion and around what time, and decide on how many traffic police officers should be assigned to such areas to control it etc.

4.2.3 Road Development Authority (RDA) Inputs to the system

Road development information

RDA is able to provide data regarding on going and future constructions or developments through web interface provided by the system. Those information will be considered when performing real time analysis and in prediction module.

Outputs from the system

• Reports including predictions

RDA can use these reports to plan their future constructions in areas with high traffic congestion.

4.2.4 System Operator Inputs to the system

• Event information

Event organizers can provide information regarding the event (when/where/expected traffic level and affected area) that they are going to organize to the system operator who is allowed to enter this information to the system via web interface. In advanced the system uses this information for real time analysis and predations.

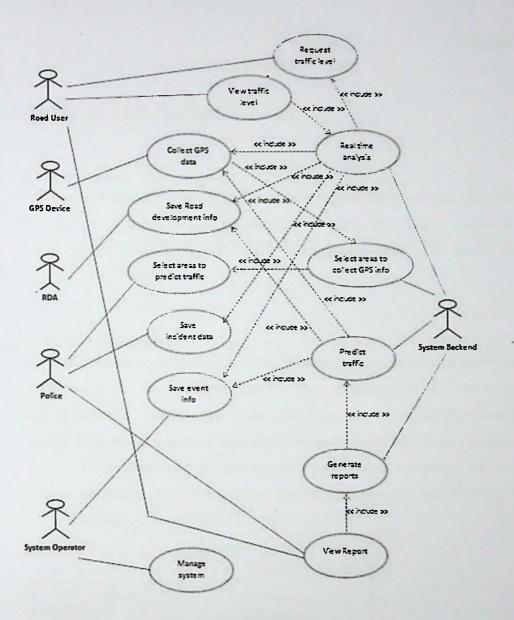


Figure 4.1 - Use case diagram

4.3 Summary

The police, road users, RDA (Road Development Authority) and system operator are main users of DSTCS and they provide various inputs to the system and received output from the system. The users interact with the system via mobile application and web application. The chapter 5 describes design of the "Decision Support Traffic Controlling System" with the system architectural diagram.

Analysis and Design

5.1 Introduction

The chapter 4 described system users and their inputs/outputs to/from the system and how they work together with the proposed system. This chapter discusses about design of the "Decision Support Traffic Controlling System" with the architectural diagram.

5.2 System Architecture

The DSTCS consists of mobile and web applications connected to a same backend. These two applications are mostly used for data gathering purpose. The information such as number of vehicles in particular area of a road at a time (density), speed of the vehicle, flow rate and etc. are discovered using collected data from registered GPS devices. The backend contains four modules such as data collection, real time data analysis, prediction and report creation. The backend connects with database to save and retrieve data.

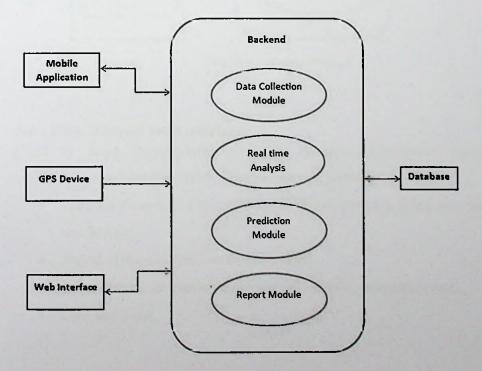


Figure 5.1 - System Architecture

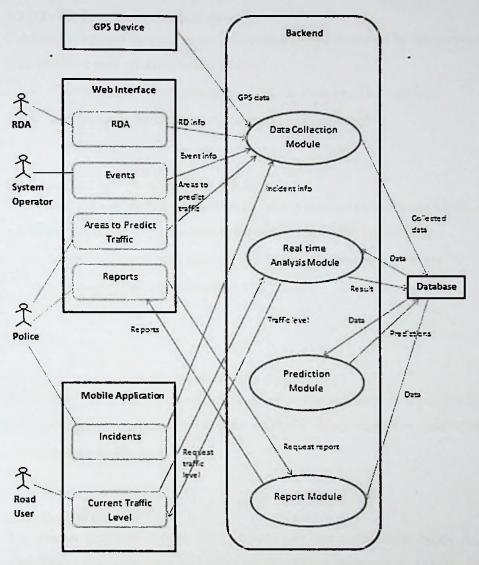


Figure 5.2 - System Diagram

5.2.1 RDA interface (web interface)

Used by Road Development Authority to enter information regarding road constructions and developments. This interface contains below fields,

- Location (from/to) of the construction/development is going on longitude and latitude
- Period of the construction/development
- Level of the construction/development Highly affected or not
- Date and Time

5.2.2 Event interface (web interface)

Used by the system operator to save data on an event provided by an event organizer. This interface may contains below fields,

- Event details event name, description, location, date and time
- Traffic area (from/to) longitude and latitude based

5.2.3 Incident interface (Mobile Application)

The Police can use mobile application to inform incidents such as accidents, closed roads, developments/constructions and events. The involving fields are,

- Type accident/ road closure/ development/ construction/ event
- Location (from/to) longitude and latitude
- Date and Time

5.2.4 View current traffic level (Mobile Application)

Road users can request current traffic level in specific areas via mobile application and the system performs real time analysis using equations and displays the corresponding traffic level.

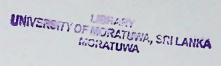
5.2.5 Report module

The system generates reports using predictions and conclusions based on collected data. The police, road users and RDA are able to use these reports to make their future plans.

5.2.6 Data collection module

This is a backend process which collects all the system input data and saves in the database. The system inputs are,

- GPS data of vehicles Collected form GPS devices
- Event data Provided by event organizers
- Incident data Sent by police
- Road development data- Provided by RDA



5.2.7 Real time analysis module

When road users request traffic level in specific area of the road, system starts to collect data from GPS devices which are attached into vehicles moving in a particular area. Then based on this GPS data, system will find out number of vehicles in particular area at this time (density), speed of the vehicles, flow rate and etc. Simultaneously it retrieves other relevant data such as events, incidents and road developments related to the specific area according to the time selected by road user. By applying all above data to the appropriate equations system will find out traffic level of that specific area.

5.2.8 Prediction module

The police officers are able to request areas that they want to have predictions. For these selected areas system collect vehicle GPS data periodically and save in database. When system is making predictions it uses these collected GPS data as historical data and other relevant information such as road constructions and events. In order to predict future traffic it uses time series model.

Examples of possible system predictions, outcomes and their usages,

Predictions and outcomes,

- How traffic varies (in which level) in day wise (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday) at which time period.
- Where is high traffic in which day at which time.
- Which areas traffic level is getting reduced and which areas are getting it increased.

Usages,

- Road users are able to decide which roads they should avoid to get rid of traffic in future days.
- Police officers can decide how many traffic police officers to be assigned to which area on which day at which time and be able to make traffic plans.
- The Police officers can make conclusions such as,
 - Which areas traffic police work is managed properly.

19

TH3177

- Which areas need more traffic police, need to have proper traffic plan or need of making enhancements to the existing plan.
- Which areas are in need of more attention.

5.2.9 GPS devices

The information such as number of vehicles in a particular area of a road at a time (density), speed of the vehicle, flow rate and etc. are found using collected data from registered GPS devices.

5.3 Summary

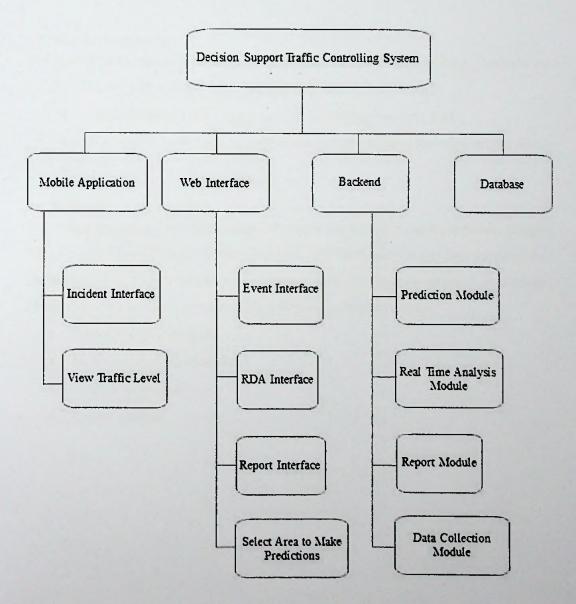
The DSTCS is designed to have two major applications such as mobile and web based applications and both connected to a common backend which implements four main modules called data collection, real time analysis, prediction and report creation. The chapter 6 explains how this design and architecture will be implemented.

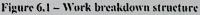
Chapter 6

Development Methodology

6.1 Introduction

The chapter 5 described the design of the proposed system. This chapter expresses how this design is going to implement with equations, models, diagrams, screenshots and source code of some interfaces.





6.2 Web Interface

The web interface is developed using technologies such as Java, HTML, JSP, Google Javascript API and Eclipse IDE. Some of the developed web interfaces are depicted in Appendix B and source code is shown in Appendix C. Followings are the main interfaces in web application,

- Event information used by system operator to insert event data provided by event organizers.
- Select area to predict traffic police defining area to predict traffic and system use this information to choose areas to collect GPS data.

6.3 Mobile Application

The mobile application is developed using technologies such as Java, Android and Eclipse IDE. System will get implemented with below interfaces,

- Incident information use by police for adding incident data.
- View traffic level use by road users to make request and view traffic level.

6.4 Backend

The technologies used for backend development are Java, Spring framework, Eclipse IDE, Google Map API and Google Direction API. Both mobile and web applications are integrated to a backend which is connected with the database implemented using MongoDB. The backend consists with following modules such as;

- Data collection module
- Real time analysis module
- Prediction module
- Report module

6.5 Real time traffic analysis module

Road users are able to request current traffic level by selecting a specific location on map given with mobile application.

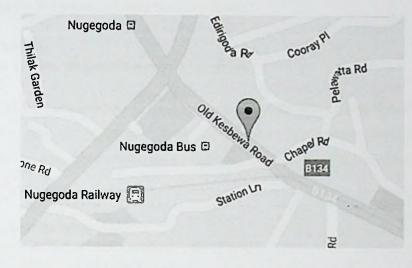


Figure 6.2 - Select location

When system received that request it will check whether this location is on one of a road or off road. This is done using **Direction Map API** provided by Google and some of **Equations**. The system defines the area which covers selected location on the road and then it finds out the density of this area (number of vehicles in this area) and average speed of these vehicles as depicted in Figure 6.3.

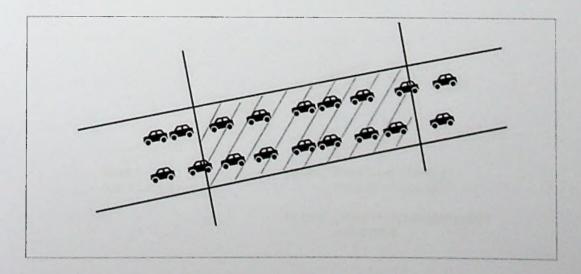


Figure 6.3 - Vehicle density in specific area

This has been done using collected data from GPS devices. Since there is no any real time GPS device data available, the system uses backend process to dynamically generate real time GPS device data for a specific area and save that data in to a database. System uses these data to find out the traffic level of the selected location. Following are the equations used for calculations.

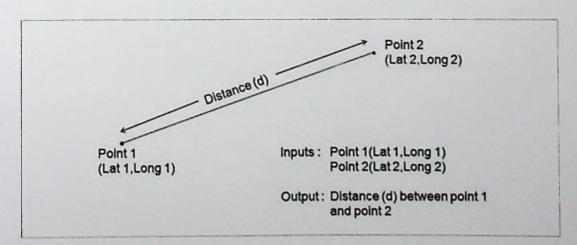
6.5.1 Equations Equation 1: Find distance between 2 locations.

The system uses the 'haversine' formula to calculate the great-circle distance between two points. That is, the shortest distance over the earth's surface giving an 'as-thecrow-flies' distance between the points ignoring any hills they fly over [12].

Haversine formula: $a = \sin^2(\Delta \phi/2) + \cos \phi 1 \cdot \cos \phi 2 \cdot \sin^2(\Delta \lambda/2)$ $c = 2 \cdot a \tan^2(\sqrt{a}, \sqrt{1-a})$ $d = R \cdot c$

φ is latitude λ is longitude R is earth's radius (mean radius = 6,371km) Angles need to be in radians.



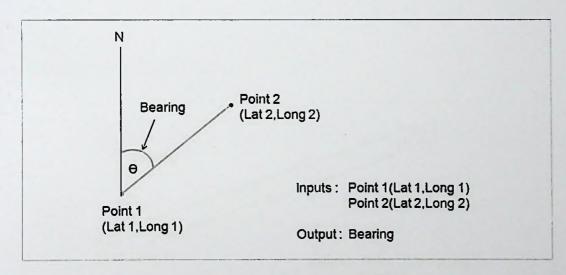


Equation 2: Find bearing value.

This formula is used for finding initial bearing (sometimes referred to as forward azimuth) which if followed in a straight line along a great-circle arc will take you from the start point to the end point [12].

Formula: $\theta = atan2(\sin \Delta \lambda \cdot \cos \varphi^2, \cos \varphi^1 \cdot \sin \varphi^2 - \sin \varphi^1 \cdot \cos \varphi^2 \cdot \cos \Delta \lambda)$ φ_{1,λ_1} is the start point φ_{2,λ_2} the end point $\Delta \lambda$ is the difference in latitude

Ex:

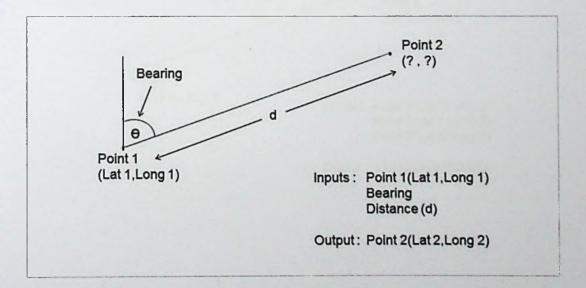


Equation 3: Find destination point by giving distance and bearing from start point.

Given a start point, initial bearing, and distance, this will calculate the destination point and final bearing travelling along a (shortest distance) great circle arc [12].

Formula: $\varphi 2 = a \sin(\sin \varphi 1 \cdot \cos \delta + \cos \varphi 1 \cdot \sin \delta \cdot \cos \theta)$ $\lambda 2 = \lambda 1 + atan2(\sin \theta \cdot \sin \delta \cdot \cos \phi 1, \cos \delta - \sin \phi 1 \cdot \sin \phi 2)$ φ is latitude λ is longitude θ is the bearing (clockwise from north) δ is the angular distance d/R d being the distance travelled R the earth's radius

Ex:





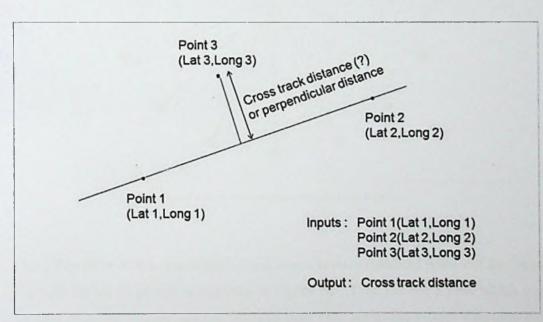
Equation 4: Find cross track distance.

The distance of a point from a great-circle path is called cross track distance [12].

Formula: $dxt = asin(sin(\delta 13) \cdot sin(\theta 13 - \theta 12)) \cdot R$

 δ_{ι_3} is (angular) distance from start point to third point θ_{ι_3} is (initial) bearing from start point to third point θ_{ι_2} is (initial) bearing from start point to end point *R* is the earth's radius

Ex:



6.5.2 Find nearest road

Use Equation 3 to find out 16 points around user selected location as shown in Figure 6.4. Then system discovers the 16 roads between each point and user selected point by Google Direction API.

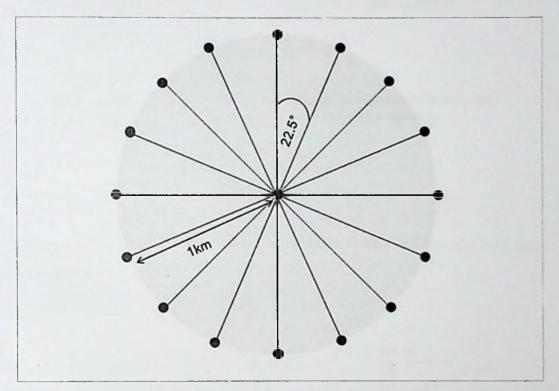
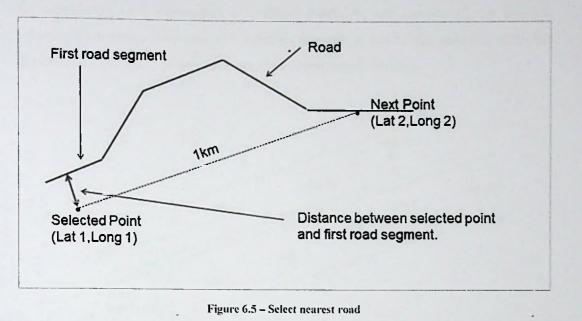


Figure 6.4 - Find locations around selected point

Using Equation 4, the system finds the distance between selected point and first road segment for all 16 points as depicted in Figure 6.5. Then choose a road which has shortest distance. This distance should less than 25m, otherwise system considers the selected location is not on the road as well as not near to any road.



Then select the vehicles which have c < +20m && c > -20m and d is calculated using Equation 4.

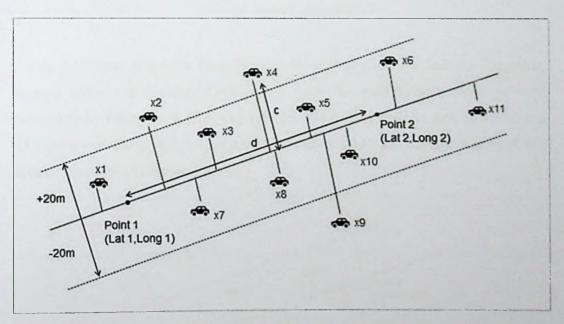


Figure 6.6 - Define vertical area

As shown in Figure 6.6 the x3, x5, x7, x8, x10 vehicles are in within the area and x1, x2, x4, x6, x9, x11 vehicles are in out of the area.

By considering (c < +20m && c > -20m) condition, can remove x2, x4 and x9 vehicles. However x1, x6 and x11 vehicles are still in out of the selected area. As depicted in Figure 6.7 system removes above mentioned vehicles.

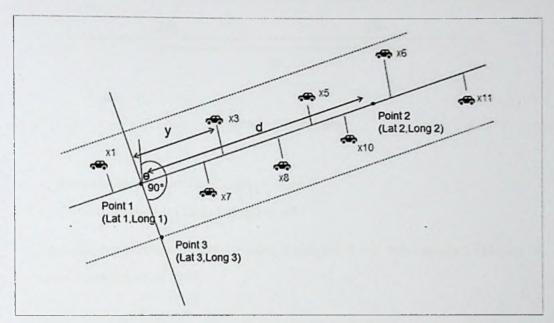


Figure 6.7 - Define horizontal area

Using Equation 3, system finds Point 3 by giving point 1's latitude/ longitude, bearing angle and d value. Then find y value for each vehicle which selected previously by Equation 4. Choosing vehicles which has y < d can remove x1, x6 and x11. Now remaining vehicles are x3, x5, x7, x8 and x10 and vehicle density of this area is 5 as shown in Figure 6.8.

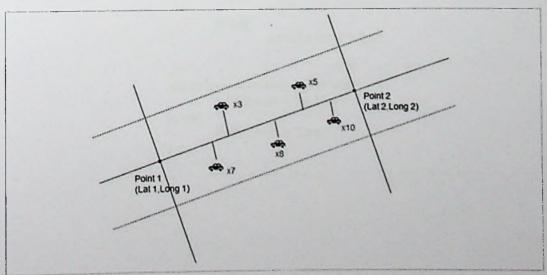


Figure 6.8 - Selected vehicles

6.5.3 Find traffic level

After selecting relevant vehicles, the velocity of these vehicles is calculated by considering previous record of each vehicle.

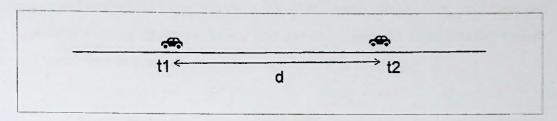


Figure 6.9 - Find distance

- Previous location = (Lat1, Long1) -> p1
- Current location = (Lat2, Long2) -> p2

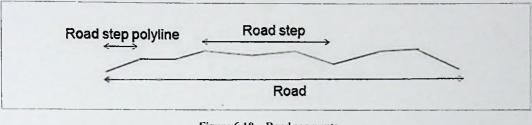
Find distance (d) between p1 and p2 using Equation 1 and then calculate velocity of each vehicle as shown in below,

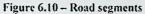
Velocity = distance (d) / (t2 - t1)

Then system calculates average velocity of selected vehicles. Based on this average velocity and vehicle density system will find out the traffic level.

6.5.4 Generate data

The system's backend process generates real time GPS device data for 100 vehicles in Nugegoda area. For that 4 roads have been selected in Nugegoda area with their start points and end points. Using Direction map API system collects road, road steps and road steps polyline details (including start and end points) for these selected 4 roads and saved them in database.





By assuming all vehicles start from one of the first polyline of a road the backend process generates first 100 vehicle's GPS location data. Thereafter in each 500milliseconds backend process updates these vehicles current location by changing latitude and longitude. To find out next location of the vehicle it used Equation 3. Likewise system generates real time GPS data as shown in Figure 6.11.

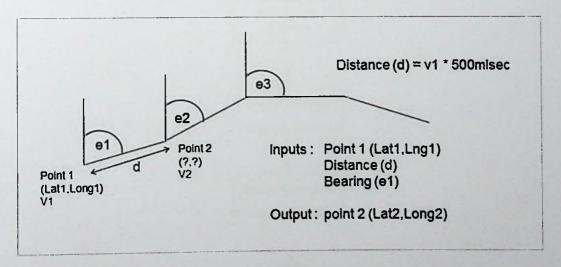


Figure 6.11 - Generate GPS data

In this research I did not consider GPS data only, but also take events, road developments and incidents into account. System use same equations and logic to check whether event or road development or incident occurs in selected area. If so system displays these details as additional information along with traffic level.

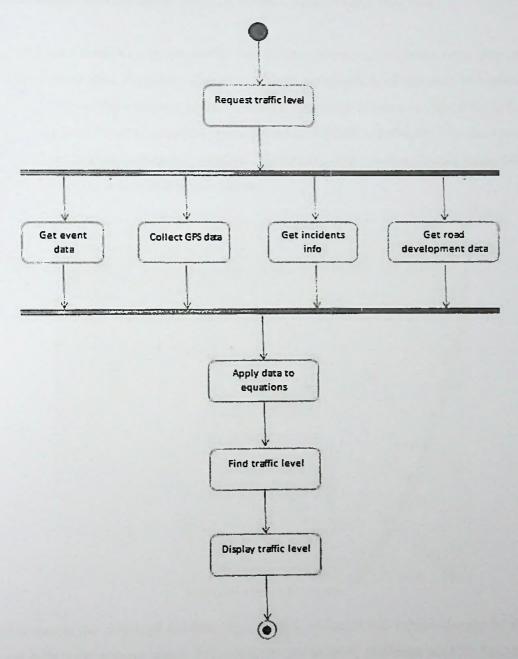


Figure 6.12 - Activity diagram of real time analysis module

6.6 Prediction module

In this module will introduce suitable time series model to predict future traffic. Time series modelling is the method which deals with time based data use for predictions, forecasting and to derive hidden insights to make informed decision making. Time series models are very useful models when have serially correlated data.

In Sri Lanka there is no proper traffic data collection mechanism being used, thus no traffic related data available. Therefore I used a collection of datasets of vehicle traffic, observed between two points in Aarhus, Denmark (Shown in Figure 6.13) for a set duration of time (5 minutes) over a period of 9 months (February, March, April, May, June, August, September, October, November). They were collected these data using sensors located in different locations.

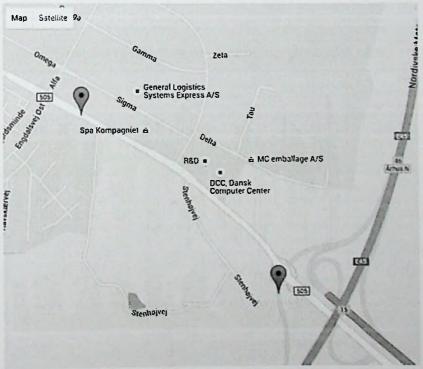


Figure 6.13 - Data collected area

This dataset has details of the area where data is collected and vehicle density of the area with their average speed. Following are the primary attributes used to find the suitable model for predictions,

- Average speed
- Vehicle count (vehicle density of the area)

- Date and time
- Longitude and latitude of origin point
- Longitude and latitude of destination point
- Distance between origin and destination

This dataset has more than 60000 data as they collected data for a set of 5 minutes duration over a period of 9 months. To reduce data volume I divided dataset in day wise (Monday to Sunday) and calculated average of vehicles and speed in hourly for each month. That formatted dataset is applied to model.

There are two attributes available to predict about the traffic.

- 1. Average vehicle count
- 2. Average speed of vehicles

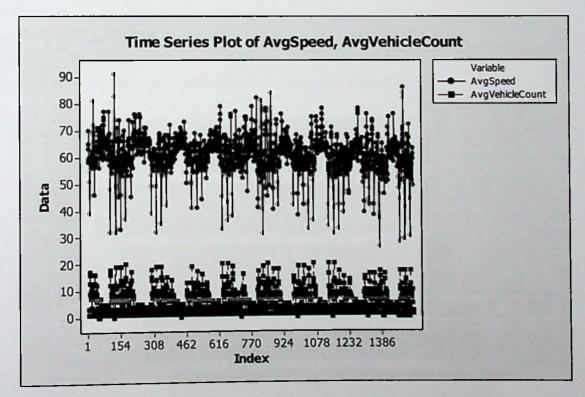


Figure 6.14 - Time series plot against Avg Speed of vehicle and Avg vehicle count

According to the Figure 6.14 there is a relation between average vehicle count and average speed of a vehicle. It shows when no of vehicle count increases, average speed reduces. However average speed of a vehicle can be influence by so many

reasons such as weather, incidents and etc. other than traffic, which cannot be get into consideration. Therefore I consider average vehicle count for predict future traffic.

Assumptions

There's not much back ground information are available for the data set. Therefore I made following assumptions,

- There are no external factors to affect the data set during 9 months (ex:-school vacations, holidays, weather and etc.)
- Average vehicle count for each hour will represent the behaviour of traffic throughout the hour.
- Average vehicle count represents the traffic of that particular area.

Time Series Analysis

Following are the steps I followed to do the time series analysis.

. VIS	ualize the time series
2. 9	Stationarize the series
	3. Plot ACF/PACF charts and find optimal parameters
	4. Build the ARIMA model
	5. Make Predictions

Figure 6.15 - Steps in time series analysis

Step 1: Visualize the Time Series

It is essential to analyse the trends, seasonality or random behaviour in the series prior to building any kind of time series model. The software MINITAB 14 is used to analyse data and find the most suitable time series model for future traffic predictions. Seasonal ARIMA model has been fitted for the data set, because the dataset is linear and seasonal in behaviour.

To fit the seasonal ARIMA model series should be stationary. In cases where the stationary criterion is violated, the first requisite becomes to stationarize the time series and then try stochastic models to predict this time series. There are multiple ways of bringing this stationarity. Some of them are Detrending, Differencing and Seasonality.

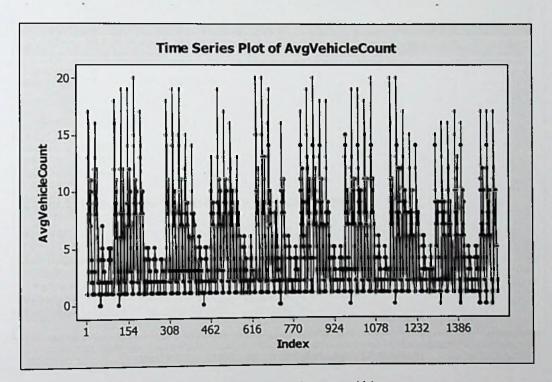


Figure 6.16 - Time series plot of average vehicle count

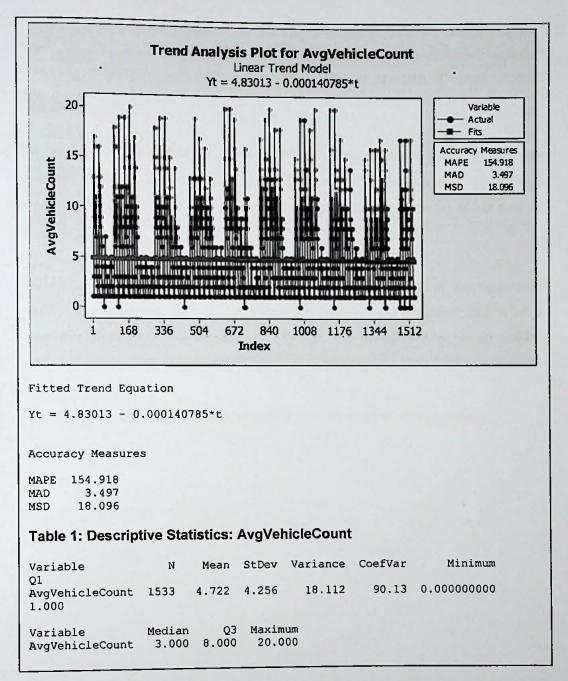


Figure 6.17 - Trend analysis plot of average vehicle count

According to the Figure 6.16 (Time series plot), can notice that there is seasonality behaviour in the dataset. In Figure 6.17 (trend analysis plot) can see a very small decreasing trend in the average vehicle count over the time. According to the descriptive statics the MAPE, MAD and MSD values are high. Those statistics are used to compare the fits of different forecasting and smoothing methods. It can used to compare the fits obtained by using different methods. For all three measures, smaller values usually indicate a better fitting model. The Mean absolute percentage error (MAPE) - is a measure of prediction accuracy of a forecasting method in statistics and expresses accuracy as a percentage of the error [13].

The Mean Absolute Deviation (MAD) - is the average distance between each data value and the mean. It expresses accuracy in the same units as the data, which helps conceptualize the amount of error. Outliers have less of an effect on MAD than on MSD [14].

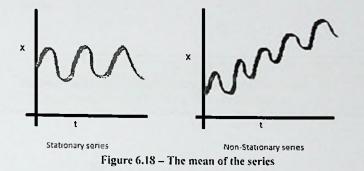
The Mean squared deviation (MSD) - is measures the average of the squares of the errors or deviations, that is, the difference between the estimator and what is estimated. It commonly-used measure of accuracy of fitted time series values. Outliers have a greater effect on MSD than on MAD [13].

Therefore need to remove seasonal variations as well as trend to fit a model.

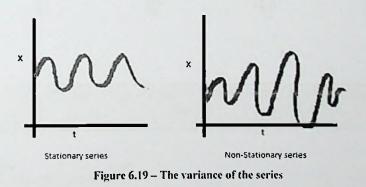
Determine stationarity and seasonality

There are three basic standards for a series to be classified as stationary series.

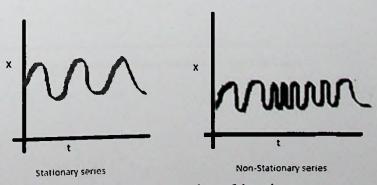
1. The mean of the series should not be a function of time rather should be a constant. In the Figure 6.18, the left hand graph satisfying the condition whereas the graph in red has a time dependent mean.

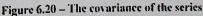


2. The variance of the series should not be a function of time as depicted in Figure 6.19.

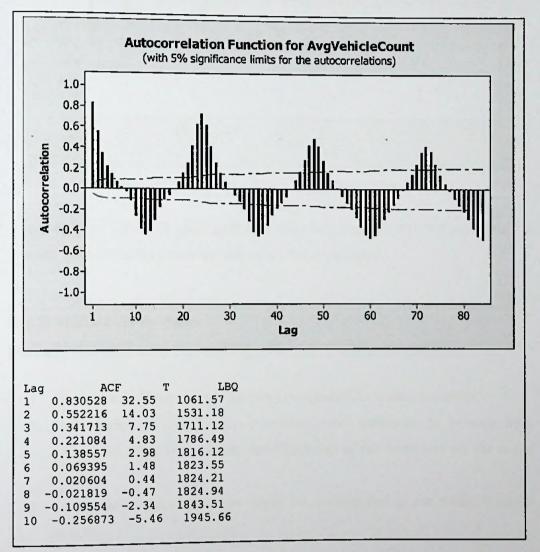


3. In the Figure 6.20, the spread becomes closer as the time increases in 'red series'. Hence, the covariance is not constant with time.





The Autocorrelation function (ACF) is one of the widest used tools in time series analysis. It is used to determine stationarity and seasonality. Partial Autocorrelation Function (PACF) is a conditional correlation between two variables under the assumption. The differences in ACF and PACF among models are useful when selecting models. Using time series can determine stationarity, but the ACF helps to ascertain this indication.







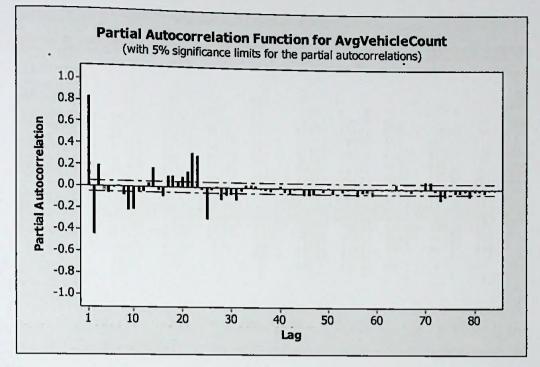


Figure 6.22 - Partial Auto Correlation Function for Avg. Veh. Count

As depicted in Figure 6.21 and Figure 6.22 there is non-stationarity and seasonality in data set. Therefore need to convert this series into stationary.

Step 2: Stationarize the Series

There are three commonly used techniques to make a time series stationary:

- 1. Detrending Here, remove the trend component from the time series.
- 2. Differencing This is the commonly used technique to remove nonstationarity. Here try to model the differences of the terms and not the actual term.
- 3. Seasonality Seasonality can easily be incorporated in the ARIMA model directly.

The differencing method is used for remove non-stationarity in this series.

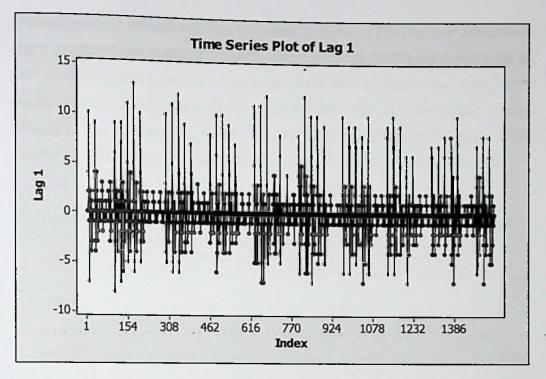


Figure 6.23 - Time series plot after 1st differencing

After 1st differencing Figure 6.23 has clearly indicated that less seasonality when compare with time series plot in before differencing (Figure 6.16). Therefore now data seems stationary. Using ACF and PACF can ascertain it further.

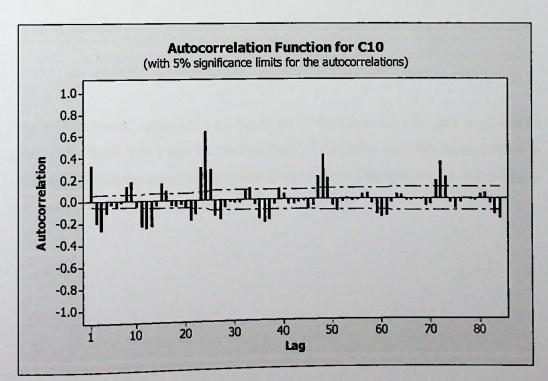


Figure 6.24 - ACF of 1st differenced data

A correlogram of the centred data is plotted in figure 6.24. The two dotted lines, nearly parallel to the x-axis of the plot, denote the 95% confidence interval for the ACF (autocorrelation function). If the value of the ACF is within these lines, then it can be considered to be negligible or equal to zero. This ACF shows this series is stationary. The correlogram shows that the ACF has considerably large values repetitively, after 1 spike it goes to negative values. Therefore AR degree is AR(1).

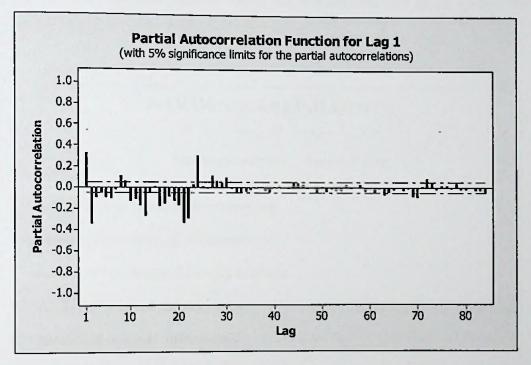
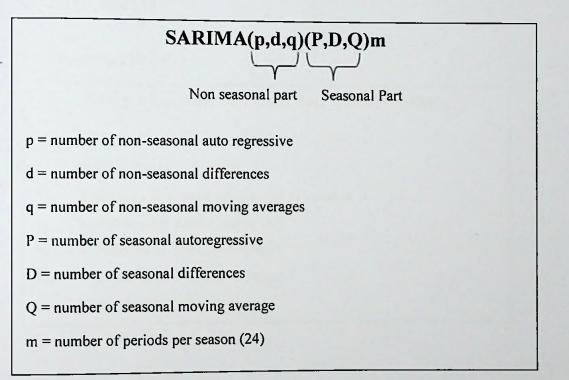


Figure 6.25 - PACF of first differenced data

The PACF (partial autocorrelation function) of the centred traffic data is plotted as well. This Figure 6.25 confirms the stationary of the series from the correlogram it is quite evident that the zero-mean process is also non-stationary in nature. According to the PACF, MA degree is MA(1).

Step 3: Find Optimal Parameters

ARIMA model is the most widely used model to analyse time series like this. Also it is proven it gives more accurate predictions compared other methods. This time series also has seasonality in nature. Therefore I used seasonal ARIMA model as fitted model. There are two types of seasonal models, additive and multiplicative. Here the multiplicative seasonal ARIMA model (p, d, q)(P, D, Q) is used, since seasonal effect is changing. In the multiplicative model the non-seasonal part (p, d, q) and the seasonal part (P, D, Q) are multiplied together.



Step 4: Build ARIMA Model

According to the ACF and PACF, parameters are identified as bellow,

AR (1) (p=1)MA (1)(q=1)d = 0P=0Q=1D=1

According to the identified parameters model can be derived as bellow for predict future traffic,

SARIMA(1,0,1)(0,1,1)

The forecasting equation is,

 $\hat{Y}_{t} = \mu + Y_{t-12} + \phi_{I}(Y_{t-1} - Y_{t-13}) - \theta_{I}e_{t-1} - \Theta_{I}e_{t-12} + \theta_{I}\Theta_{I}e_{t-13}$ $e_{t} \quad \text{Random Error}$ $\hat{Y}_{t+1} \quad \text{Forcasting Avg no of vehicles for t+1 period}$ $\mu \quad \text{Coefficient of Auto Regression}$ $\theta_{I} \quad \text{Coefficient of Moving Averages}$ $\theta_{24} \quad \text{Coefficient of Moving Averages}$ Time periode

Fitting Seasonal ARIMA Model

Final Estimates of Parameters Type Coef SE Coef Т P AR 1 0.6536 0.0237 27.53 0.000 MA -0.3674 1 0.0291 -12.61 0.000 SMA 24 0.8737 0.0132 66.20 0.000 Constant -0.000967 0.008008 -0.120.904 Differencing: 0 regular, 1 seasonal of order 24 Number of observations: Original series 1536, after differencing 1512 Residuals: SS = 4189.08 (backforecasts excluded) MS = 2.78 DF = 1508 Modified Box-Pierce (Ljung-Box) Chi-Square statistic Lag 12 24 36 48 Chi-Square 93.4 253.5 279.1 315.3 DF 8 20 32 44 0.000 - 0.000 0.000 0.000 P-Value

Table 6.1 - Fitting ARIMA Model

The first-order autoregressive coefficient, the seasonal coefficient and the first-order moving average coefficient are all significant at the 5% alpha level, indicating that this might be an efficient model. The sum of squares that measures the sum of the squared differences between each original data point and its estimated value using this SARIMA model are quite small. Also the Ljung-Box Chi-Square statistics show no correlations between points with a difference of 12 or 24 lags including the seasonal coefficient has factored these out.

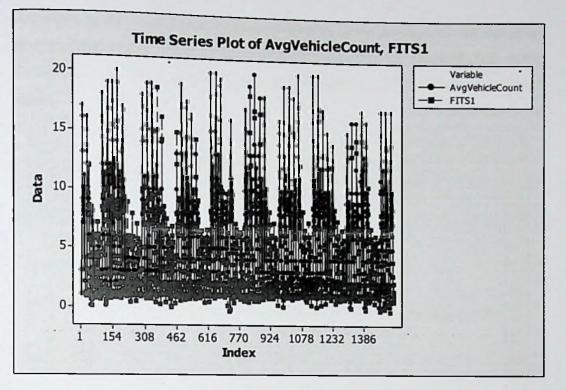


Figure 6.26 - Fitted model Vs Actual Time series

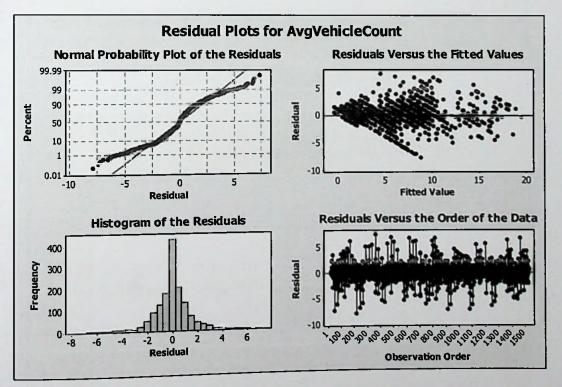


Figure 6.27 - Residual plots for Fitted model

According to the above Figure 6.26 and Figure 6.27 the assumptions are met quite well, Residual plot is normal, variance is constant, and Data set is centred. This can be further check by PACF (Figure 6.28) and ACF (Figure 6.29) for residuals of fitted model.

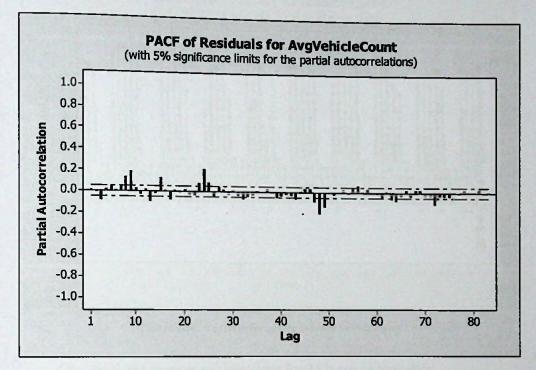


Figure 6.28 - PACF for Residuals of fitted model

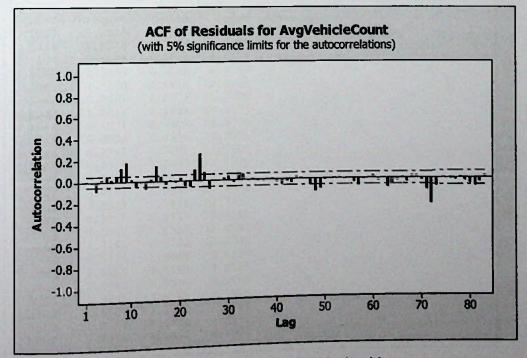
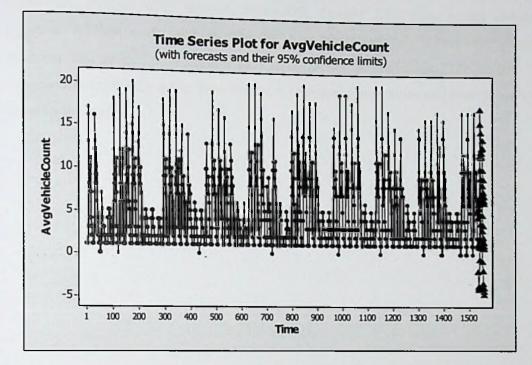


Figure 6.29 - PACF of residuals of fitted model

Step 5: Make Predictions



Forecast future traffic using SARIMA(1,0,1)(0,1,1) model.

Figure 6.30 - Forecast for future average vehicle count

Below is the predicted average vehicle count for 24 hours in next date.

95 Percent Limits Period Forecast Lower Upper Actual 1537 1.1810 -2.0864 4.4484 1538 0.9070 -3.7626 5.5767 1539 0.9588 -4.1947 6.1124 1540 1.3285 -4.0185 6.6755 1541 4.7860 -0.6415 10.2135 1542 10.2993 4.8377 15.7608 1543 11.9714 6.4954 17.4474 1544 9.1907 3.7086 14.6729 1545 6.8223 1.3375 12.3071 1546 6.2741 0.7882 11.7600 1547 6.3526 0.8662 11.8390 1548 6.2913 0.8047 11.7779 1549 6.7197 1.2300 12.2064 1550 7.2363 1.7496 12.7231 1551 8.3319 2.8451 13.8166 1552 7.3180 1.8312 12.8047 155					
$\begin{array}{cccccccccccccccccccccccccccccccccccc$					
	1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558	1.1810 0.9070 0.9588 1.3285 4.7860 10.2993 11.9714 9.1907 6.8223 6.2741 6.3526 6.2913 6.7197 7.2363 8.3319 7.3180 4.7165 3.5419 2.0299 1.4942 1.5102 1.4697 1.0184	Lim Lower -2.0864 -3.7626 -4.1947 -4.0185 -0.6415 4.8377 6.4954 3.7086 1.3375 0.7882 0.8662 0.8047 1.2330 1.7496 2.8451 1.8312 -0.7702 -1.9449 -3.4568 -3.9926 -3.9766 -4.0170 -4.4684	its Upper 4.4484 5.5767 6.1124 6.6755 10.2135 15.7608 17.4474 14.6729 12.3071 11.7600 11.8390 11.7779 12.2064 12.7231 13.8186 12.8047 10.2033 9.0287 7.5167 6.9809 6.9969 6.9565 6.5052	Actual

Table 6.2 - Forecasts from period 1536

6.7 Summary

The DSTCS is implemented using several technologies such as Java, Spring framework, Eclipse IDE, Minitab, MongoDB, Google API, Time series model (Seasonal ARIMA model) and etc. The Real time analysis module is developed using equations and defined most suitable time series model for predictions. The chapter 7 expresses how DSTCS differ from similar researches, future plans and how going to evaluate the system.

Chapter 7

Discussion

7.1 Introduction

In chapter 6 described the implementation of the system with design. This chapter explains how DSTCS differ from similar researches, future development, enhancements and how to evaluate the system.

7.2 Difference from similar systems

Most of the existing researches are fully based on the GPS devices and current data and make the conclusions. DSTCS is not only focusing on GPS data, but also consider accidents, events and road development information provided by police, event organizers and RDA. Therefore accuracy of the result of real time analysis is high. In addition to that and according to the DSTCS design, all above data collected for real time analysis are used in prediction module. Therefore accuracy of the predictions will get increased gradually. These predictions allows the users to arrange their future travel schedules avoiding high traffic areas and also it helps the traffic police and RDA for decision making.

7.3 Future work

The real time traffic analysis module has been implemented based on generated GPS device data, since there is no proper traffic data collection mechanism in Sri Lanka. Therefore this module can be improved to use with actual data collected on time.

To increase the accuracy of the predictions, system can be used results of the real time analysis and other data collected from different sources such as event information, GPS data and road development/construction data. The prediction module has been implemented using data set obtained from Denmark, as I explained before no traffic data available in Sri Lanka. As well as when predict traffic, system neglect below conditions.

- Time ranges regarding to the school holidays
- Time ranges regarding to the public holidays
- Events
- Incidents
- Weather

To have high accuracy of the predictions need to consider above neglected conditions when preparing the data set. As well as to apply this research model to Sri Lanka the researchers need to use actual data set which relevant to the Sri Lanka.

The project requires completing implementation of basic system functions including both web and mobile applications in order to be get practical usability. At that time this analysis and prediction module should integrated with the system. Then prediction results can be shown using reports which can be used by Road users and Police.

7.4 Evaluation plan

If this research continues to its implementation, all modules will be tested separately (unit testing) and once the system development is completed the system testing will be carried out by integrating all the modules together. Further real time analysis and prediction modules should be tested against accuracy, reliability and efficiency.

7.5 Summary

The decision support traffic controlling system is designed to makes predictions and does the real time analysis based on several factors such as GPS data, accidents, events, incidents and road development information provided by police, event organizers and RDA. Thus accuracy of the outputs of this system are high when compare with other similar systems.

References

[1] Massive Rs 32 bln loss due to traffic congestion: Transport expert, Sunday Times, http://www.sundaytimes.lk/110313/BusinessTimes/bt03.html

[2] Traffic Statistics - No. of Road Vehicles & Road Length, http://www.police.lk/index.php/traffic-statistics/113

[3] Remedies for traffic congestion, http://www.ft.lk/2012/04/25/remedies-for-trafficcongestion/

[4] SATNA, https://www.dialog.lk/mobile-vas-miscellaneous-satnav/

[5] Mobitel Partners With T-Navi To Offer The Latest Traffic Navigation Solution, http://www.businesstoday.lk/article.php?article=3963

[6] Road.lk: Sri Lanka's answer to real-time traffic updates. http://readme.lk/road-lksri-lankas-answer-real-time-traffic-updates/

[7] Traffic reports for Sri Lanka, https://lk.roadlk.com/traffic/

[8] Traffic congestion: an important economic issue, http://www.ceylontoday.lk/22-8532-news-detail-traffic-congestion-an-important-economic-issue.html

[9] GPS Assisted Traffic Alerting and Road Congestion Reduction Mechanism, http://www.ip-sl.org/procs/2013/ips113-12.pdf

[10] Spring MVC and MongoDB, http://findnerd.com/list/view/Spring-MVC-and-MongoDB/2240/

[11] MongoDB, https://www.mongodb.com/compare/mongodb-mysql

[12] Movable Type Scripts, http://www.movable-type.co.uk/scripts/latlong.html

[13] Minitab 17 Support, http://support.minitab.com/en-us/minitab/17/topic-library/ modeling-statistics/time-series/time-series-models/what-are-mape-mad-and-msd/

[14] Mean Absolute Deviation, http://mathbits.com/MathBits/TISection/Statistics1/MAD.html

Appendix A

Abbreviations

DSTCS	- Decision Support Traffic Controlling System
GPS	- Global Positioning System
IDE	- Integrated Development Environment
GDP	- Gross Domestic Product
RDA	- Road Development Authority
PND	- Portable Navigation Device
SD	- Secure Digital
OS ·	- Operating System
DPOI	- Dynamic Point of Interests
API	- Application Program Interface
HTML	- Hypertext Markup Language
JSP	- Java Server Page

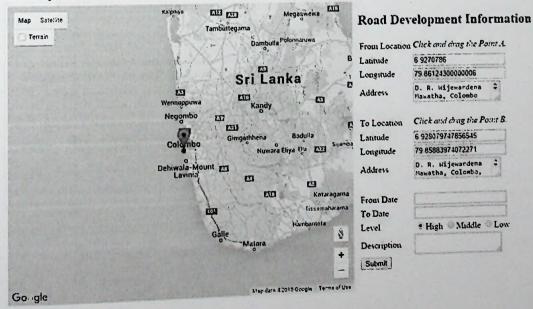
Appendix B

Web Interfaces

This is the web interface created for system operator to insert event information provided by event organizers.



This interface is used by the Road Development Authority (RDA) to insert road development information.



Appendix C

Source Code

Equations

_	
	//Equation 1
	public double distance(
	double lat1,
	double lon1,
	double lat2,
	double lon2,
	String unit) {
	double theta = lon1 - lon2;
	double dist = Math.sin(deg2rad(lat1)) * Math.sin(deg2rad(lat2)) +
	Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2)) * Math.cos(deg2rad(theta));
	dist = Math.acos(dist);
	dist = rad2deg(dist);
	dist = dist * 60 * 1.1515;
	if ("K".equalsIgnoreCase(unit)) {
	dist = dist * 1.609344;
	} else if ("N".equalsIgnoreCase(unit)) {
	dist = dist $*$ 0.8684;
	}
	return (dist);
	}
	//Equation 2
	//Equation 2 public double bearing(
	-
	double lat1,
	double lon1,
	double lat2,
	double lon2) {
	double latitude $l = deg2rad(lat1);$
	double latitude2 = deg2rad(lat2);
	double longDiff = deg2rad(lon2 - lon1);
	double y = Math.sin(longDiff) * Math.cos(latitude2);
	double $x = Math.cos(latitude1) * Math.sin(latitude2) - (longDiff);$
	Math sin(latitude1) * Math.cos(latitude2) * Math.cos(longDiff),
	return (rad2deg(Math.atan2(y, x)) + 360) % 360;
	}

```
//Equation 4
  private double getCrossTrackDistance(
       double pLinePoint | Latitude,
       double pLinePoint I Longitude,
       double pLinePoint2Latitude,
       double pLinePoint2Longitude,
       double pPointLatitude,
       double pPointLongitude) {
    double ret = 10000;
    double bearingAngleBetweenLinePoint1AndLinePoint2 = 0.0;
    double bearingAngleBetweenLinePointIAndPoint = 0.0;
    double distanceFromLinePoint | ToPoint = 0;
    long R = 6371:
    try {
       bearingAngleBetweenLinePoint1AndLinePoint2 = bearing(pLinePoint1Latitude,
pLinePoint1Longitude.
            pLinePoint2Latitude, pLinePoint2Longitude);
       distanceFromLinePoint1ToPoint = distance(pLinePoint1Latitude, pLinePoint1Longitude,
            pPointLatitude, pPointLongitude, "K");
       bearingAngleBetweenLinePointIAndPoint = bearing(pLinePointILatitude,
pLinePoint1Longitude,
            pPointLatitude, pPointLongitude);
       ret = Math.asin(Math.sin(distanceFromLinePoint1ToPoint / R) *
            Math.sin(deg2rad(bearingAngleBetweenLinePoint1AndPoint -
bearingAngleBetweenLinePoint1AndLinePoint2))) * R;
     } catch (Exception e) {
       e.printStackTrace(); //To change body of catch statement use File | Settings | File Templates.
     }
     return ret;
  }
  //Equation 3
  public double[] findNextPointLatLonUsingCurrentLatLonBearingAndDistance(
       double pLat1,
       double pLon1,
       double pBearing,
       double distance) {
     double point2[] = new double[2];
     point2[0] = 0;
     point2[1] = 1;
     long R = 6371;
    double latitude | = deg2rad(pLat1);
    double longitude1 = deg2rad(pLon1);
    double bearing = deg2rad(pBearing);
     point2[0] = Math.asin(Math.sin(latitudel) * Math.cos(distance / R) +
          Math.cos(latitude1) * Math.sin(distance / R) * Math.cos(bearing));
     point2[1] = longitude1 + Math.atan2(Math.sin(bearing) * Math.sin(distance / R) *
          Math.cos(distance / R) - Math.sin(latitude1) * Math.sin(point2[0]));
Math.cos(latitude1),
     point2[0] = rad2deg(point2[0]);
     point2[1] = rad2deg(point2[1]);
     return point2;
```

58

}

The EventController

package com.dstcs.controller;

import java.util.Date; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Controller; import org.springframework.ui.ModelMap; import org.springframework.web.bind.annotation.ModelAttribute; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RequestMethod; import org.springframework.web.servlet.ModelAndView; import com.dstcs.common.util.MapUtil; import com.dstcs.domain.EventInfo; import com.dstcs.domain.User; import com.dstcs.mongo.repositories.EventInfoRepository; import com.dstcs.mongo.repositories.UserRepository; @Controller public class EventController { @Autowired private EventInfoRepository eventInfoRepository; @Autowired private UserRepository userRepository; @RequestMapping(value = "/eventinfoform", method = RequestMethod.GET) public ModelAndView getEventInfoForm() { return new ModelAndView("eventInfoForm", "command", new EventInfo()); } @RequestMapping(value = "/saveeventinfo", method = RequestMethod.POST) public String saveEventInfo(@ModelAttribute("SpringWeb")EventInfo eventInfo, ModelMap model) { double distance = MapUtil.calculateDistance (eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.gettLatitude(), eventInfo.gettLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLongitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLatitude(), eventInfo.getfLatitude(), eventInfo.getfLatitude(), eventInfo.getfLatitude(), eventInfo.getfLongitude(), eventInfo.getfLatitude(), eventInfo.getfLatitude(),e(),eventInfo.gettLongitude(),'K'); System.out.println("distance : "+distance); eventInfo.setDistance(distance); eventInfo.setDate(new Date()); User user = userRepository.findByUsername(username); eventInfo.setUser(user); eventInfoRepository.save(eventInfo); model.addAttribute("msg","Successfully saved ..!!"); model.addAttribute("command", new EventInfo()); return "eventInfoForm"; } }

The RoadDevelopmentController

package com.dstcs.controller;

import java.util.Date; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Controller; import org.springframework.ui.ModelMap; import org.springframework.web.bind.annotation.ModelAttribute; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RequestMethod; import org.springframework.web.servlet.ModelAndView; import com.dstcs.common.util.MapUtil; import com.dstcs.domain.RoadDevelopmentInfo; import com.dstcs.mongo.repositories.RoadDevelopmentRepository; @Controller public class RoadDevelopmentController { @Autowired private RoadDevelopmentRepository roadDevelopmentRepository; @RequestMapping(value = "/roaddevelopmentform", method = RequestMethod.GET) public ModelAndView getRoadDevelopmentForm() { return new ModelAndView("roadDevelopmentForm", "command", new RoadDevelopmentInfo()); } @RequestMapping(value = "/checkroaddevelopment", method = RequestMethod.GET) public ModelAndView checkroaddevelopment() { return new ModelAndView("checkrd"); } @RequestMapping(value = "/saveroaddevelopmentinfo", method = RequestMethod.POST) public String saveRoadDevelopmentInfo(@ModelAttribute("SpringWeb")RoadDevelopmentInfo roadDevelopmentInfo, ModelMap model) { double distance = MapUtil.calculateDistance (roadDevelopmentInfo.getfLatitude (), roadDevelopmentInfo.getfLongitude (), road(),roadDevelopmentInfo.gettLatitude(),roadDevelopmentInfo.gettLongitude(),'K'); roadDevelopmentInfo.setDistance(distance); roadDevelopmentInfo.setDate(new Date()); roadDevelopmentRepository.save(roadDevelopmentInfo); model.addAttribute("msg","Successfully saved ..!!"); model.addAttribute("command", new RoadDevelopmentInfo()); return "roadDevelopmentForm"; 3 }

Load Google map

</script>

```
<script type="text/javascript">
             var $fp1 = $( ".filthypillow-1" );
initializeCalendar($fp1);
             var $fp2 = $( ".filthypillow-2" );
             initializeCalendar($fp2);
             function initializeCalendar($fp){
                     var now = moment().subtract( "days", 1);
                               $fp.filthypillow( {
                                minDateTime: function() {
                                  return now;
                                }
                               });
                               $fp.on( "focus", function( ) {
                                $fp.filthypillow( "show" );
                               });
                               $fp.on( "fp:save", function( e, dateObj ) {
    $fp.val( dateObj.format( "MMM DD YYYY hh:mm A" ) );
                                $fp.filthypillow( "hide" );
                               });
             }
 google.maps.event.addListener(marker1, 'dragstart', function() {
          updateMarkerAddress('Dragging ... ', "from");
 });
 google.maps.event.addListener(marker1, 'drag', function() {
             updateMarkerPosition(marker1.getPosition(),"from");
 });
 google.maps.event.addListener(marker1, 'dragend', function() {
             geocodePosition(marker1.getPosition(),"from");
 });
 google.maps.event.addListener(marker2, 'dragstart', function() {
             updateMarkerAddress('Dragging ... ', "to");
 });
 google.maps.event.addListener(marker2, 'drag', function() {
           updateMarkerPosition(marker2.getPosition(),"to");
 });
 google.maps.event.addListener(marker2, 'dragend', function() {
             geocodePosition(marker2.getPosition(),"to");
 });
 google.maps.event.addListener(marker2, 'dragstart', function() {
           updateMarkerAddress('Dragging ... ', "to");
 });
 google.maps.event.addListener(marker3, 'drag', function() {
          updateMarkerPosition(marker3.getPosition(),"evt");
 google.maps.event.addListener(marker3, 'dragend', function() {
 });
          geocodePosition(marker3.getPosition(),"evt");
 google.maps.event.addListener(marker3, 'dragstart', function() {
             updateMarkerAddress('Dragging ... ', "evt");
 });
}
// Onload handler to fire off the app.
google.maps.event.addDomListener(window, 'load', initialize);
```

```
function geocodePosition(pos,loc) {
 geocoder.geocode({
  latLng: pos
 }, function(responses) {
  if (responses && responses.length > 0) {
   updateMarkerAddress(responses[0].formatted_address,loc);
  } else {
   updateMarkerAddress('Cannot determine address at this location.',loc);
 });
3
function updateMarkerPosition(latLng,loc) {
          if(loc == 'from'){
                   document.getElementById('fLatitude').value = latLng.lat();
                   document.getElementById('fLongitude').value = latLng.lng();
          }else if(loc == 'to'){
                    document.getElementById('tLatitude').value = latLng.lat();
                    document.getElementById('tLongitude').value = latLng.lng();
          }else if(loc = 'evt'){
                    document.getElementById('eLatitude').value = latLng.lat();
                    document.getElementById('eLongitude').value = latLng.lng();
          ì
}
function updateMarkerAddress(str,loc) {
          if(loc = 'from'){
                    document.getElementById('fAddress').value = str;
          }else if(loc == 'to'){
                    document.getElementById('tAddress').value = str;
          }else if(loc == 'evt'){
                    document.getElementById('eAddress').value = str;
          3
}
function initialize() {
 var latLng1 = new google.maps.LatLng(6.785277702704765, 80.11392854687506);
 var latLng2 = new google.maps.LatLng(6.775369668926126, 80.28181337353521);
 var latLng3 = new google.maps.LatLng(7.048031607682675, 80.20490907666021);
 var map = new google.maps.Map(document.getElementById('mapCanvas'), {
  zoom: 8,
   center: latLng1,
   mapTypeId: google.maps.MapTypeId.ROADMAP
 });
 var marker I = new google.maps.Marker({
          position: latLng1,
          title: 'Point A',
          map: map,
          draggable: true
 });
 var marker2 = new google.maps.Marker({
             position: latLng2,
             title: 'Point B',
             map: map,
             draggable: true
  });
 var marker3 = new google.maps.Marker({
             position: latLng3,
             title: 'Event',
             map: map,
             draggable: true
  });
```