Real Time Traveler Information System: An Efficient Approach to Minimize Travel Time Using Available Media

Mohammad Mahmudul Haque
ID# 01201083

Md. Abul Hasnat
ID# 01201024

Department of Computer Science and Engineering

December 2005

Declaration

In accordance with the requirements of the degree of Bachelor of Computer Science and Engineering in the division of Computer Science and Engineering, I present the following thesis entitled ‘Real Time Traveler Information System: An Efficient Approach to Minimize Travel Time Using Available Media ’. This work was performed under the supervision of Dr. Mumit Khan.

I hereby declare that the work submitted in this thesis is my own and based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of Supervisor	Signature of Author
-----------------------	---------------------------------------
Dr. Mumit Khan	Mohammad Mahmudul Haque

Acknowledgement

First of all, I would like to thank my supervisor, Dr. Mumit Khan. He gave us freedom to choose the thesis topic and complete guidance throughout its development. I was lucky enough to work under him for my thesis work. Although being extremely preoccupied with his busy schedule, he often showed much enthusiasm and took time and lot of pain to review my thesis work that enabled me to improve the system as well as adding new features. I learned plenty of useful things from his comments, revisions and discussions during this period.

I want to give my heartiest gratitude to Matin Saad Abdullah who gave us proper guideline for reviewing Use Cases. Many thanks to all of my friends for being with me specially to Ali Al Asadullah Mohammad Shafi who encourage me to work with this topic and Yea Hasib Md. Abu Bakar Siddique to support us providing some useful resources.

Last but not the least, thanks to the Almighty for helping me in every step of this work.

Abstract

The objective of this thesis project is to establish a Real Time Traveler Information System that will Support the operation of an Advanced Traveler Information System (ATIS). The system will monitor the current condition of each route and update automatically according to the collected information. It will also inform about any incident, weather condition and any special event in the road. The information collected from the routes will be analyzed in conjunction with the historical information. On the request from traveler it will generate real time information to guide travelers towards optimal decisions concerning their travel routes to avoid congestions and delay [1, 7]. This project incorporates unbiased-ness (best output) and consistency (expected output) into its core operations [1].

Table of contents

Introduction:

Traveling is everyday work in our life. But it is an annoying mater when it takes more than 2 or 3 times of required time and this happens almost everyday. There is no standard rule for traveling vehicles in the road and the road is not sufficient to accommodate all the vehicles. As a result everyday traffic jam occurs in the city. So, if we can minimize travel time by predicting the path which consumes less time to reach destination then obviously it will be smart work.

By analyzing the traffic data and some other parameter I built up such a system which acts smartly and give a path direction to the user with minimum cost. When a user gets a minimum cost (time) path direction before start his travel it will save his valuable time. The generated path is made by the system is related to less-congested road and traveler is encouraged to travel using those less-congested roads. As a result the whole city traffic condition will be improve. In our country the common accessible media are internet, cell phone and land phone. In this system a user can query through web, SMS or making a call to get a path direction.

The system has the capability to inform users about any incident, weather condition and any special event in the road. This information will be added to the path info for which the user requested if the path is related to such incidents. Moreover the information collected from the routes is analyzed and verified according to the historical information. If the traffic data is missing from the data source for a certain amount of time then the specific edge information will be updated according to the historical information. To maintain the historical information data validation agent works as a part of the system.

When user will wants to use web he has to reach to a specific web location. Then he has to give source and destination location to the system and also mention either “Time” or “Distance” parameter on which path direction will be generate. According to query user will get two output, one is text based path direction and other is visual road map.

User can use phone to get a verbal output from an operator. Operator will receive user request and manually handle the web interface to get output and then just tell the output to the user.

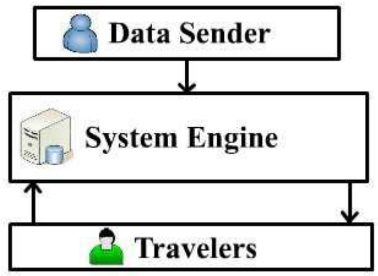


Figure: 1 Top Level View of the System

The top-level view or the black box of the overall system is shown in Figure: 1 that clearly explains the system. Further decomposition of each block will be shown in the design and analysis part. Here the data sender block represents all the data senders that include the traffic police, administrator, and also the system operator. The block at the middle represents the entire system. And the traveler's block represent all the users of the system that include both web and SMS users.

At present there is no such type of system exists in our country. But similar kind of system exists in other countries. Some research work has already done in the development of this system. A complete idea about the entire structure of similar system can be obtained from the "Survey" part of this report that provides all necessary information about existing Traveler Information Systems of different countries. We develop our system based on existing limited resources and keep open some field for further development of this service with additional resources. The "Limitations" and "Future Visions" part of this report provide a complete direction about further enhancement of this service when additional resources, equipments and technologies will be available.

At Design and Implementation phase of the service development we concentrate on the usage of the possible media and develop the system based on these. For each medium we develop separate application where all applications share a unique database and the common functionality of the central service engine. The web application is further decomposed into two parts: text based output provider and map based output provider which can be categorized as the GIS enabled service. The internal structure of the system engine is shown in Figure: 2 that represent the each separate application. The complete description about each application, the database and the central service engine development is available on the "Design" and "Implementation" part of this report.

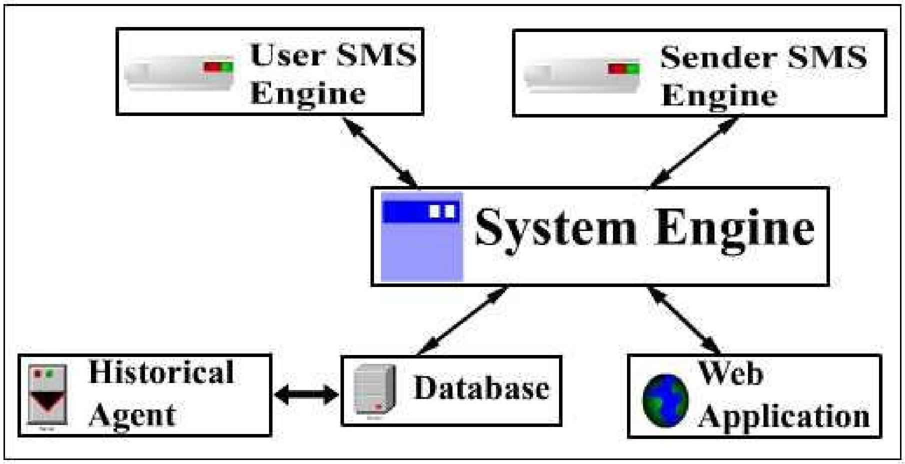


Fig: 2 Internal Structure of the System Engine

The entire system is designed and implemented by the collaboration of me and Md. Abul Hasnat. I have the total responsibility to implement “Historical data validation engine”. Md. Abul Hasnat was responsible to implement the SMS engine. Although both of us were responsible for different parts but each individual part was tested by both of us. To implement the entire system we followed Extreme Programming style of Software Engineering. The entire modules were tested in unit testing approach using NUnit GUI. Programming language C#, ASP.Net and Scalable Vector Graphics (SVG) was used to develop the entire system.

Survey on Similar System

Every system related with traveler information established till today, provide real time information about the road network condition and thus help travelers to travel through a path which decrease the travel time and congestion on the road. Initially the aim of such system was to distribute the information about road and traffic condition of the road network by broadcasting radio information through a national broadcast. We found such a system called CCISS which was introduced Italy during the 1990 world cup football. Such type of information was absolutely helpful for the travelers and also be able to make a good impact for improving the road traffic network congestion. So, later this system is introduced into their Toll Motorway Network, National Free Road and Freeway Network, and also in urban areas [5].

The system proves its effectiveness to the user (traveler) level. So many public [3, 5] and private (e.g. DELCAN, ILOG) companies are interested to establish this type of system and many research projects [1, 7] have done and also continuing in this area. With the advancement of technology, new equipments are introducing in this area. So day by day this system includes the advantages of latest available technology. Almost every system divides the entire task into 3 segments. First input/data collection segment, then data processing and the last one is to deliver the output. But the procedures applied in each segment are not always same. Different system employs different technology and model for this purpose.

Among these systems **DynaMIT** [1] propose a Real Time Traffic Estimation and Prediction System that will support the operation of an Advanced Traveler Information System (ATIS) and Advanced Traffic Management Systems (ATMS) at a Traffic Management Center (TMC). From **DynaMIT-P** [6] that is a planning version of DynaMIT, use dynamic traffic assignment (DTA) into their system. It split the time horizon into long-term, short-term, and within-day time horizons and established a model based on the time horizon. A related research project [7] is continuing in **University of Utah,** which involve ATIS to help drivers to make informed decisions. In 1992 Department of Transportation (**DOT**) of Los Angeles city implemented Automated Traffic Surveillance and Control (ATSAC) system [3], which was able to detect road incidents or tie-ups and alert emergency or repair vehicles for dispatch. A private company **DELCAN** [2] has implemented Advanced Traveler Information System (ATIS)/Traffic and Road

Information System under their Intelligent Transportation System (ITS) products. The other private company **ILOG** [4] implemented Real Time - Road traffic management which is intuitive graphical user interfaces (GUIs) based and claimed to be highly reliable. Their system enables real-time monitoring of traffic and on-screen interaction with data sources and signaling equipment. From the survey about present situation of road traffic management on **Area 3: Multiple Monitoring of Road Traffic** [5], idea about the road traffic monitoring system in different countries can be known where in each case the ultimate goal was to collect real time information of the route and provide that to the relevant users. In [8] ATIS is enhanced for emergency management and long term disaster recovery.

The input data to be collected is about the road network condition that includes congestion on the route, some real-time scenarios such as incidents, special events, weather conditions, highway construction activities and fluctuations in demand [1, 3, and 5]. Many systems set some historical data about the road traffic and other relevant incident information into their record [1, 5]. These are known as static data for the system. The dynamic/real time data about road traffic is collected by different input sources that include closed circuit video, sensors, cameras, inductive loop and infrared. Weather and pollution data is detected on street by road sensors [5]. Typically these input data sources are placed at potential congestion point on the road [3, 4 and 5]. Traffic data is collected on a fixed interval from the input sources. Incident and special event information are collected in many ways which include incident detection algorithm and manually from local police forces by voice data, emergency call system on motorways or, by mobile phone from any driver observing an incident [1, 5].

In the information collection and processing center that is the common national database [5], based on the input from various data sources the real time network condition is estimated [1]. The processing includes updating surveillance and O-D data, generate descriptive and prescriptive information [1]. Database queries are processed in a certain interval in different systems. In some cases the weather information comes from intermediately processed stations. These collected data also aggregated to update historical databases and processed to support traffic planning activities [5]. System where time horizon is split into long-term, short-term and within-day time horizons use output of one level as input for another level [6].

Finally, all the output data produced by different systems almost same in two areas. Route information for travelers according to their request [1] and traffic analyzed information for various parties of concern. These two areas also include any special events, weather and incident information [1, 3 and 5]. System can be able to display output in a well-organized, highly ergonomic graphical environment [4]. These outputs can be made available to the user via many different sources like telephone, VMS, local radio stations, text TV, SMS, WAP, Televideo, web based user interface and the like [5].

Design

In this phase we are going to discuss about the design methodology of the entire system in brief. We started our design task from requirement elicitation. From the primary requirements we got the use cases. Analyzing the use cases we obtained a block diagram of the system. Decomposing the block diagram we got individual components of the system. Then for each individual component we got several use cases. From the use cases we designed relevant class diagrams. Then we developed the Entity Relationship Diagram. The following subsections are briefly described in each phase of design.

Use Cases:

USE CASE: Real Time Traveler Information System

Primary Actor:

Travelers

Stakeholders and Interests:

System Admin

Data senders (traffic police)

Operator

Preconditions:

Everyone (except SMS user) has to access in either internet or phone

User (who use SMS) has to know query pattern and destination

Data sender (who use SMS) has to know input pattern and destination

Post conditions:

Generate reliable path directions to reach origin to destination

Generate MAP to visualize path direction

Give estimated time to reach destination

Main success scenario:

1. Traveler choose a medium (SMS, Web client or voice) by which he wants to make a query
2. Send his query through that medium (origin to destination)
3. System will generate list of reliable path direction, estimated time to reach destination and MAP information
4. System send this generated output to user through appropriate medium

Extensions:

*a System fails:

2a. invalid query: (invalid SMS pattern)

Reply with an error message and right query pattern

2b. query without sufficient data (missing origin or destination in web form)

Show an error message to user

2c. Ask for such destination which is unavailable

Operator will give him a suggestion

3a. Limited output generation

Generate only single path direction and estimated time for SMS client

4a. Long delay to send output

Nothing to do

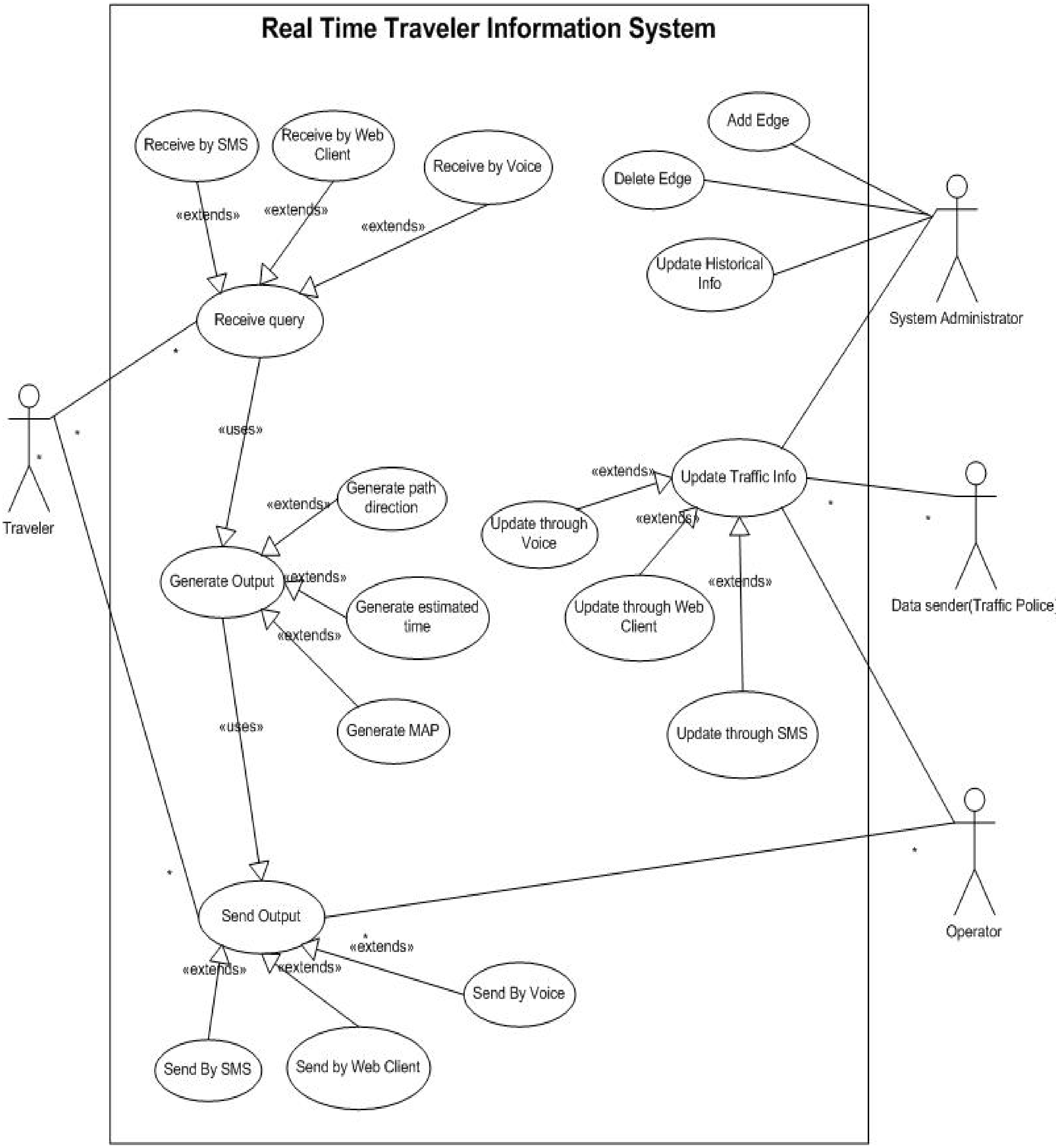


Figure:1 UML Diagram of RTTIS

USE CASE: Send query through SMS

Primary Actor:

Travelers

Stakeholders and Interests:

System Admin

Preconditions:

Traveler has to access compatible mobile phone

Traveler has to know query pattern

Traveler has to know destination/number (in which number the query should be send)

Post conditions:

System receive the query

Main success scenario:

1. Traveler take a mobile phone to write down the query
2. Write his desired query in predefined format
3. Send this query text to fixed number/node

Extensions:

*a System fails:

3a. Send invalid query

Reply an error message with correct query pattern to user

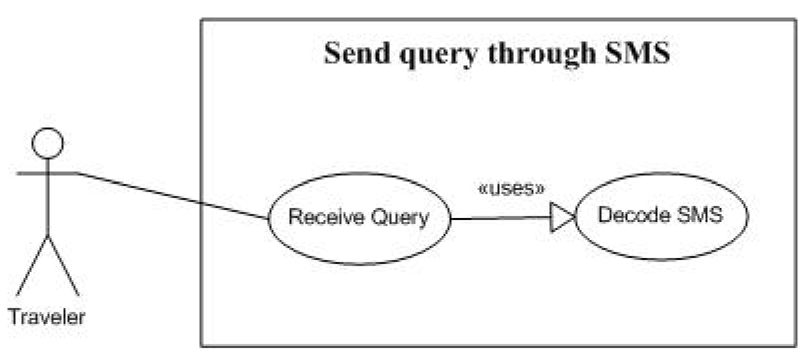


Figure:2 UML Diagram of “Send query through SMS”

USE CASE: Send query through web client

Primary Actor:

Travelers

Stakeholders and Interests:

System Admin

Operator

Preconditions:

Users have to have internet access

Users have to know the URL address

Post conditions:

System receive the query for generating path direction

Main success scenario:

1. User type the web address in browser application
2. User will get a web page from where he can make his query
3. Choose origin and destination from combo/list box
4. Click "Get Direction" button to send his query to the system

Extensions:

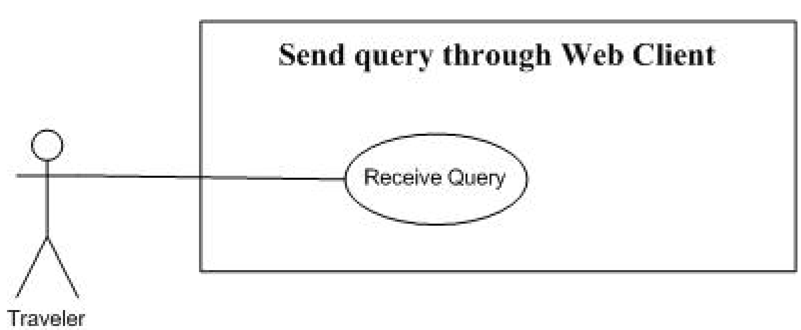
*a System fails:

1a. Type invalid URL

User will inform by browser output

3a. User can choose only origin or destination

Browser will give appropriate message



USE CASE: Send query through voice

Primary Actor:

Travelers

Stakeholders and Interests:

System Admin

Operator

Preconditions:

Users have to have phone access

Users have to know phone number of operator

Post conditions:

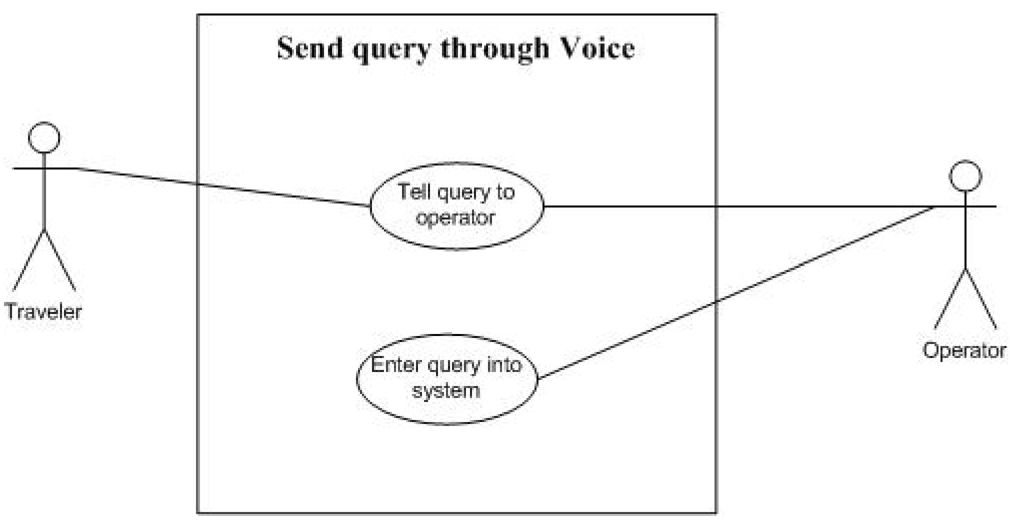
Operator receive oral query from user and send it to system through web client

Main success scenario:

1. User make a call to operator
2. Tell his query to operator
3. Operator listen user's query and identify origin and destination
4. Send this query through web client to system

Extensions:

4a System fails:



USE CASE: Receive output through SMS

Primary Actor:

Travelers

Stakeholders and Interests:

System Admin

Preconditions:

Traveler has to send a valid query through SMS

Post conditions:

Traveler gets a detail path direction

Traveler gets estimated time to reach destination

Main success scenario:

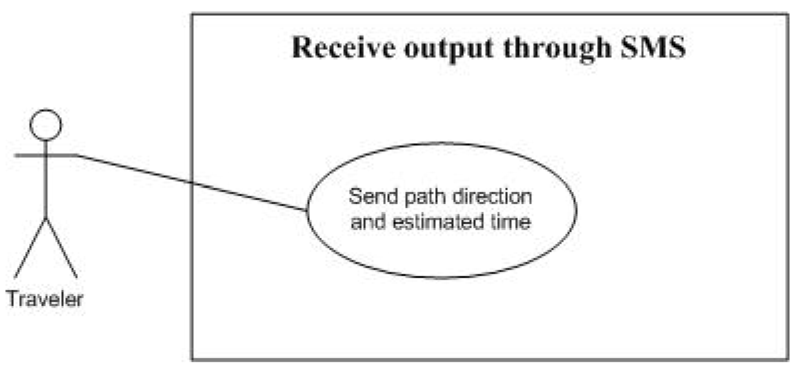
1. System apply “path reduction” on generated output
2. Send best path direction with estimated time to desired user (who initiate the query)

Extensions:

*a System fails:

2a. Long delay to send output

Nothing to do



USE CASE: Receive output through web client

Primary Actor:

Travelers

Stakeholders and Interests:

System Admin

Operator

Preconditions:

Traveler has to send a valid query through web client

Post conditions:

Traveler gets a list of detail path direction

Traveler gets estimated time to reach destination

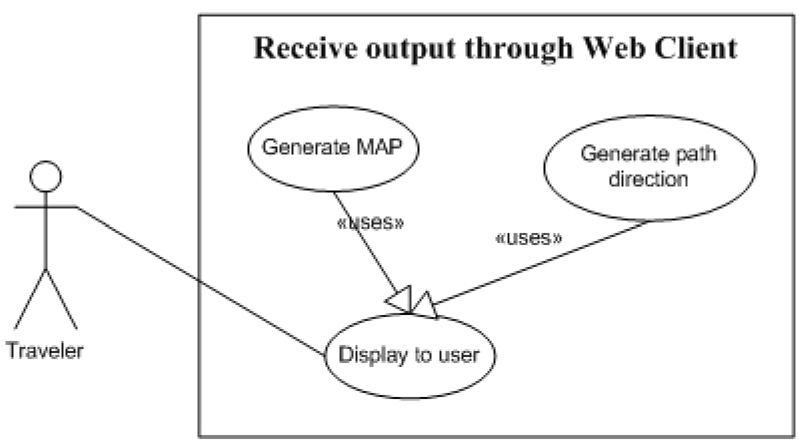
Traveler gets graphical MAP

Main success scenario:

1. System send generated list of path direction and MAP information to web client
2. Web client display the path direction and draw the MAP

Extensions:

*a System fails:



USE CASE: Receive output through voice

Primary Actor:

Travelers

Stakeholders and Interests:

System Admin

Operator

Preconditions:

Ask operator to get information

Operator has to input into web client

Post conditions:

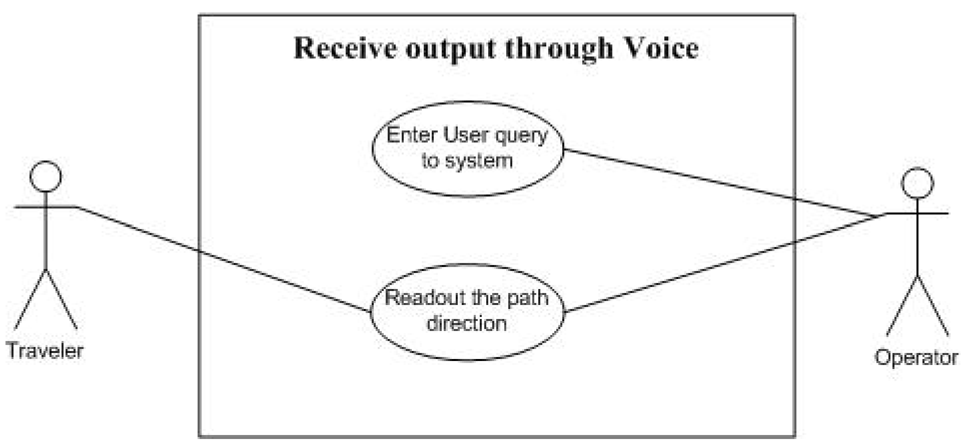
Operator readout the path direction for traveler

Main success scenario:

1. Operator receive output from web client
2. Readout this output for traveler in understandable format

Extensions:

*a System fails:



USE CASE: Update database through SMS

Primary Actor:

Traffic police

Stakeholders and Interests:

System Admin

Traveler

Operator

Preconditions:

Traffic police has to access in compatible mobile phone

Traffic police has to know SMS pattern

Traffic police has to know destination mobile phone/node number

Post conditions:

System receive the road traffic information in encoded form

System decodes the information and updates the database

Main success scenario:

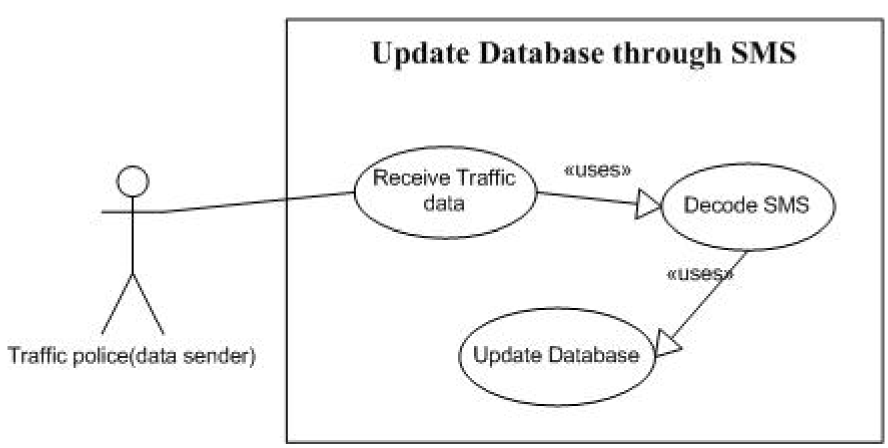
1. Traffic police observe the road traffic condition
2. Write a SMS about road traffic load
3. Send this SMS to server mobile phone at regular interval and at change of jam level
4. System receive this information in SMS format and decode it
5. Update the database according to decoded information

Extensions:

*a System fails:

4a. Invalid SMS: (invalid pattern or out of range jam level)

Reply to traffic police with SMS to inform about his invalid SMS



USE CASE: Update database through web client

Primary Actor:

Informer

Stakeholders and Interests:

System Admin

Traveler

Operator

Preconditions:

Web user has to enter into proper web address

Post conditions:

System receive the road traffic information

System updates the database

Main success scenario:

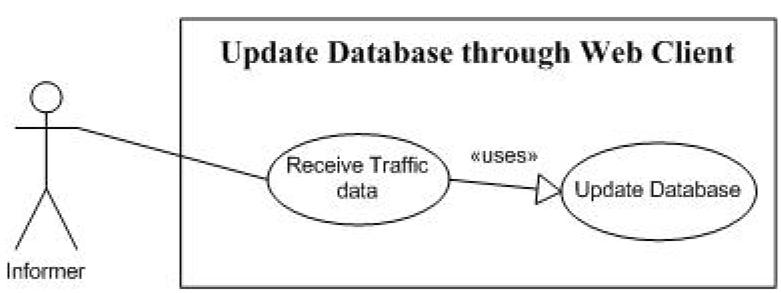
1. Informer observe the road traffic condition
2. Select proper edge/road
3. Specify the jam level and additional information (road status affected by weather, special events and incidents)
4. Send this to server by clicking send button
5. System receive this information
6. Verify the information in conjunction with historical information
7. Update the database according to the information

Extensions:

*a System fails:

4a. Sending improper data (field missing)

Browser will check and show appropriate error message



USE CASE: Maintain Historical Information

Primary Actor:

System Admin

Stakeholders and Interests:

System Admin

Operator

Preconditions:

System admin has to start the historical data validation tool

Post conditions:

Store traffic information into database

Agent updates the database

Main success scenario:

1. System admin start the historical data validation tool (Agent)
2. Set the agent either in manual or automated (specifying execution interval) mode
3. Click "Show Expired edge info" to apply validation check
4. Click "Execute validation agent" to apply heuristics

Extensions:

*a System fails:

4. Initially historical database is empty and no data to apply heuristics

Store some dummy data into heuristics database

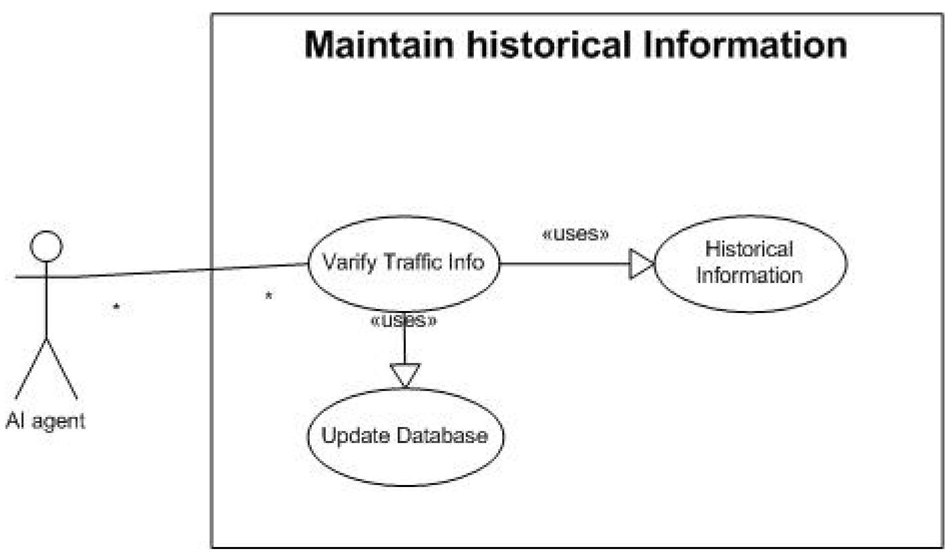


Fig: UML diagram for "Maintain historical information"

Block Diagram:

The block diagram of the system is shown in Figure: 1 where the system engine is the core part of the system that performs the major processing task. The system engine is supported to the components. The components are Web Application, Historical Agent, User SMS Engine and Data Sender SMS Engine.

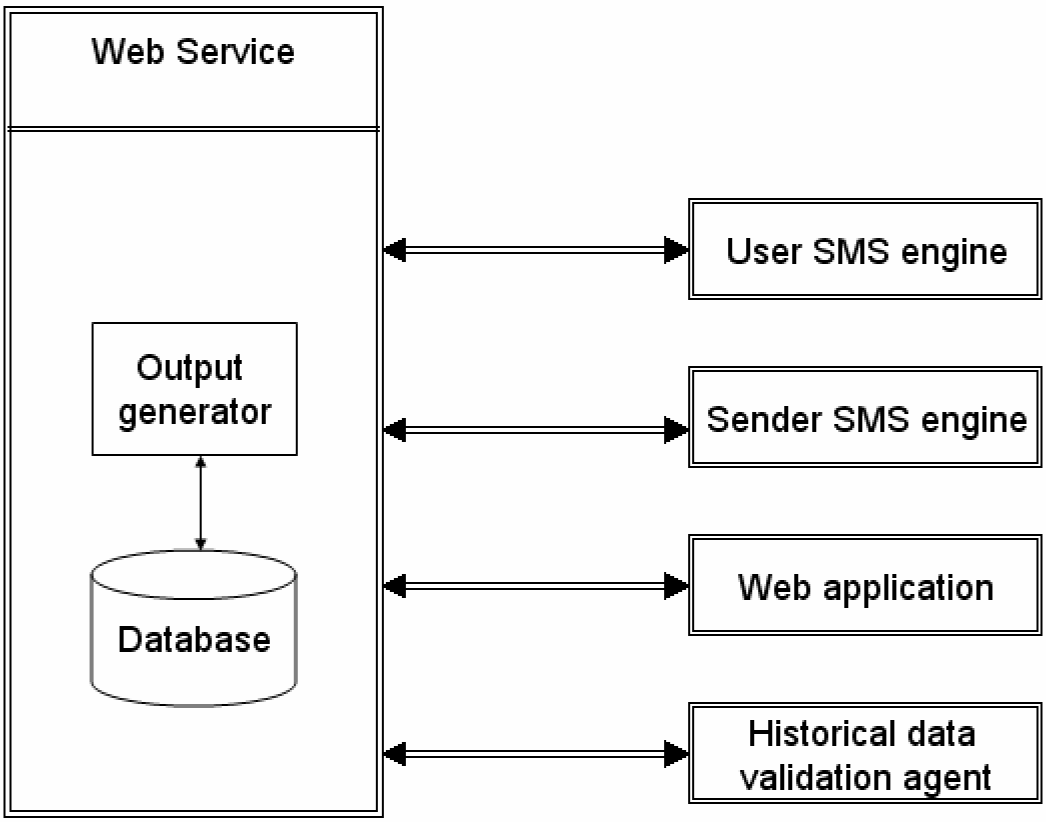


Figure: Block Diagram of the system

System Engine:

System engine performs some important task such as receiving query from web client or SMS user and takes action according to query to reply user. It also updates the database by collecting traffic information from data sender. The portion of the system which generates the output uses Dijkstra algorithm. Dijkstra is a constraint based shortest path algorithm. It works on road map (directed graph) where each road/edge has a weight. This weight is derived from constraints.

Constraints Building

Our first constraint is traffic jam level. We divide traffic jam level into 4 distinct jam levels and each jam level assign time with multiple of 3 minute.

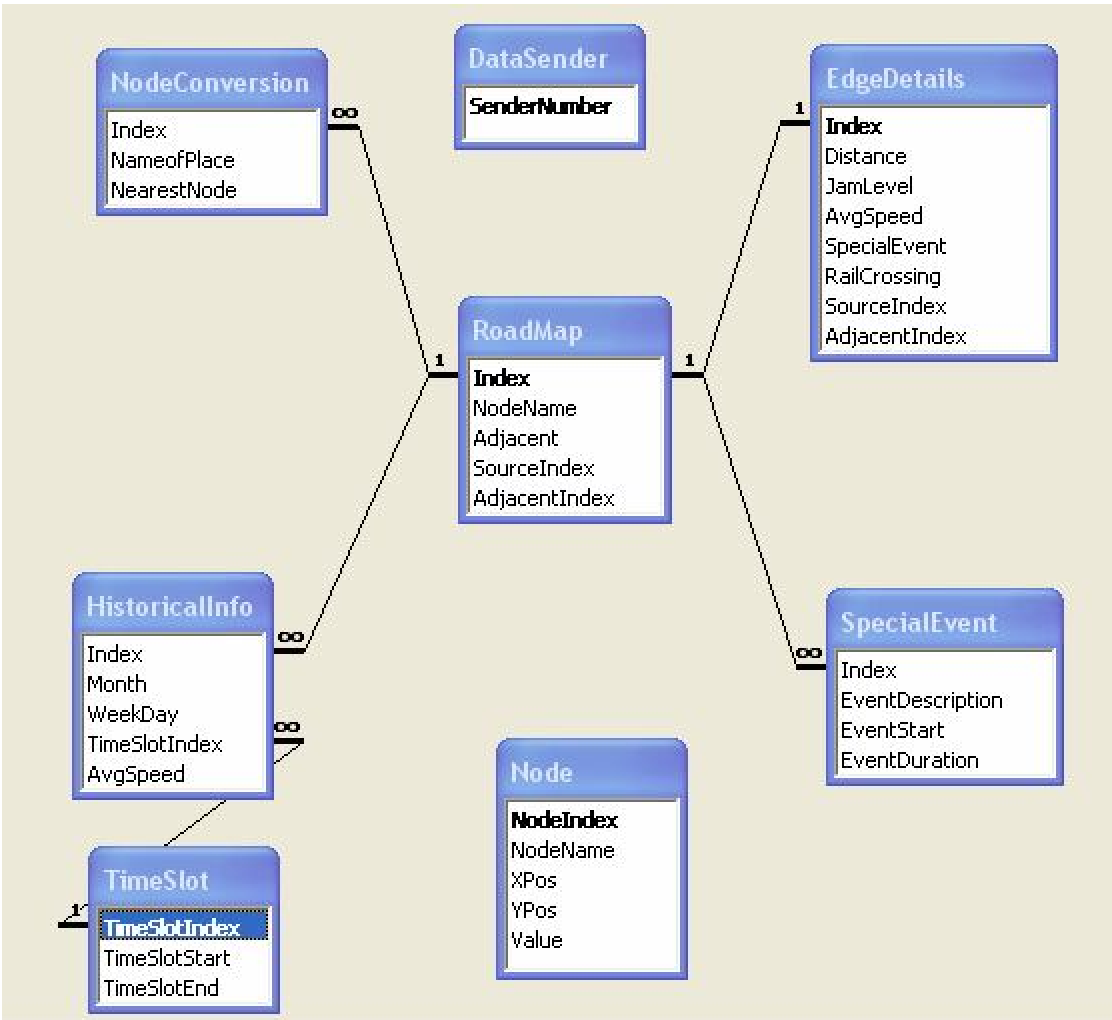
Jam level	Required Time
0	0 minute
1	3 minute
2	6 minute
3	9 minute

Other parameters are average speed on each edge and edge length (distance).
From the above mentioned parameters we derived another time variable.
For example:
Average speed 50 km/hour
Edge length 3 km then
Time required to traverse this edge = (50 * 3) / 60 ≈ 3 minute

So total time = jam level * 3 + traverse time
Using the described method we derived “time” constraint for each edge and assigned it as weight then we applied Dijkstra algorithm.

Database design:

After analyzing the requirements we design the entity relationship diagram.



Web Application

From the Use Cases we identified the major tasks for building the web application are:

- Generate Route Direction
- View Specific Node and Edge Information
- Send Traffic Information and
- Administrative Option

The desired route direction against specific origin and destination is generated under “Generate Route Direction" task. The generated direction can be viewed by either text mode or in visual (Map) mode. Information about any particular node or edge can get under "View Specific Node and Edge Information" task. Here user will get information in visual (Map) mode and they have the option to interact using mouse pointer by clicking and pointing. Using "Send Traffic Information" task user can update the edge information. Some administrative option is completed by the task "Administrative Option".

All these tasks use functionality of the Web Service that is working as the System Engine. To design each tasks we prepare class diagram, Interface and Dialogue Design and for map based service we draw a map. A brief description of the entire design task of the web application is given below. The choice of different tasks is shown in figure: 1.

Figure-1: Four Major Tasks

The class diagram of the web service and the web application is shown in figure 2 and 3. From the class diagram we can get idea about the structure and functionality of web service and web application.

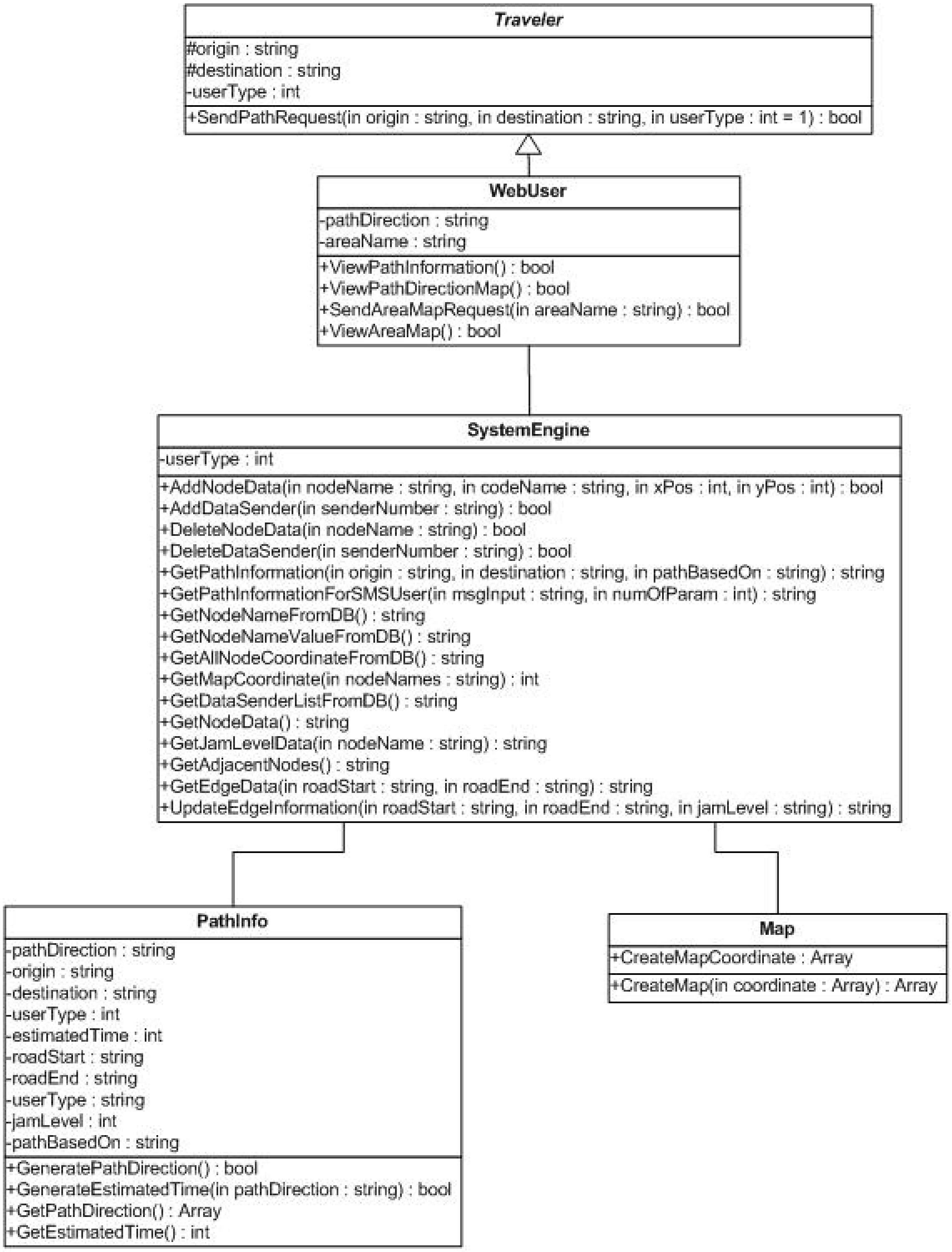


Figure-2: Class Diagram of Web Service

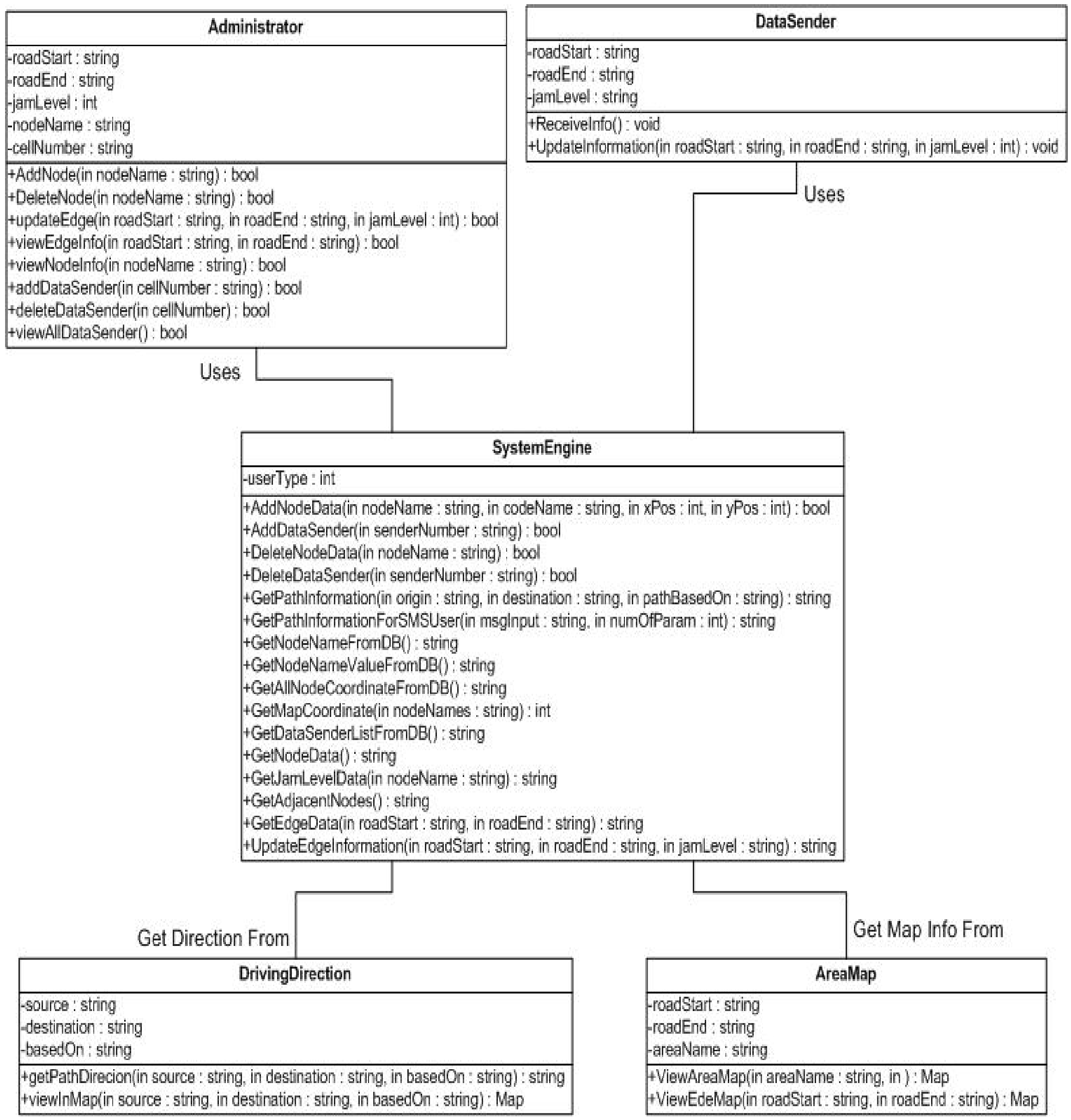


Figure-3 : Class Diagram of Web Application

Generate Route Direction

User will get the option to select origin, destination and the parameter to choose the path. The parameters are Time (default option) and Distance. User has further option to view the generated path in two different modes: "Text Based" and "Visual (Map) Based".

In the form the choice of origin and destination location is load on two dropdown lists. The parameter to generate path direction is differentiated using radio buttons and two Buttons are used to select the mode of output. This form is shown in Figure: 4.

Figure-4: Interface Design of “Generate Route Direction"

View Specific Node and Edge Information

User will get options to get specific information about any particular location (node) and edge (between two nodes) directly in map. User will get the opportunity to interact in the map using mouse pointer by clicking and pointing. In interactive mode user can get the latest visual picture of any location if the picture is available.

To perform this task user have to choose "View Area Map" link. In the form user will given three options:

- View Area in Map
- View Edge in Map
- Interactive Map

To view any particular node information use have to choose "View Area in Map" link. The form will then load where in a dropdown list all node names will be loaded. User has to choose desire node name from the loaded list and click on the button "SHOW". Same approach is applied to design the form for "View Edge in Map". To interact with Map user have to choose "Interactive Map" Link. The interface of the forms is shown in figure 5 and 6.

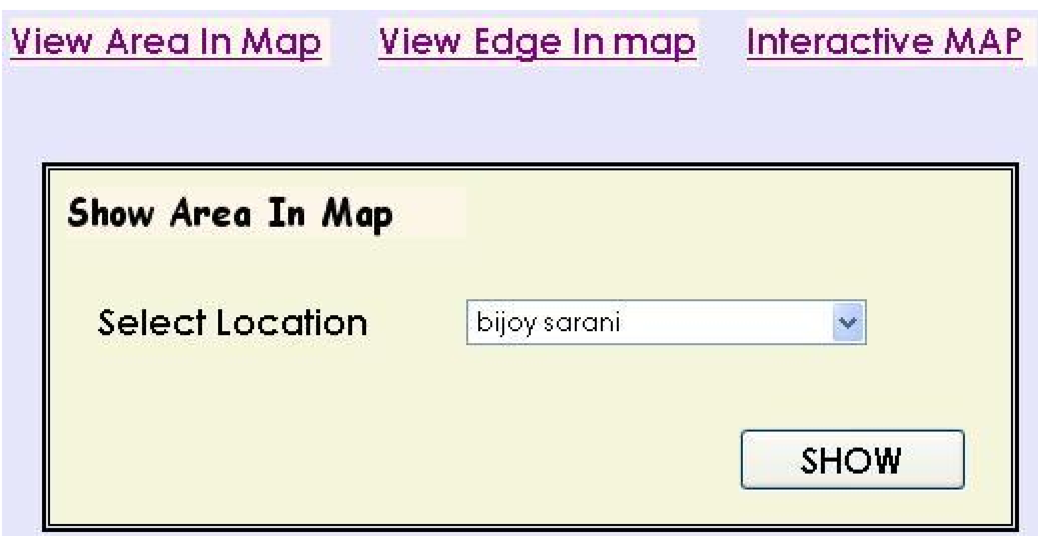


Figure-5: Interface Design of "Show Area in Map"

Figure-6: Interface Design of "Show Edge in Map"

The interactive Map is shown in figure 7.

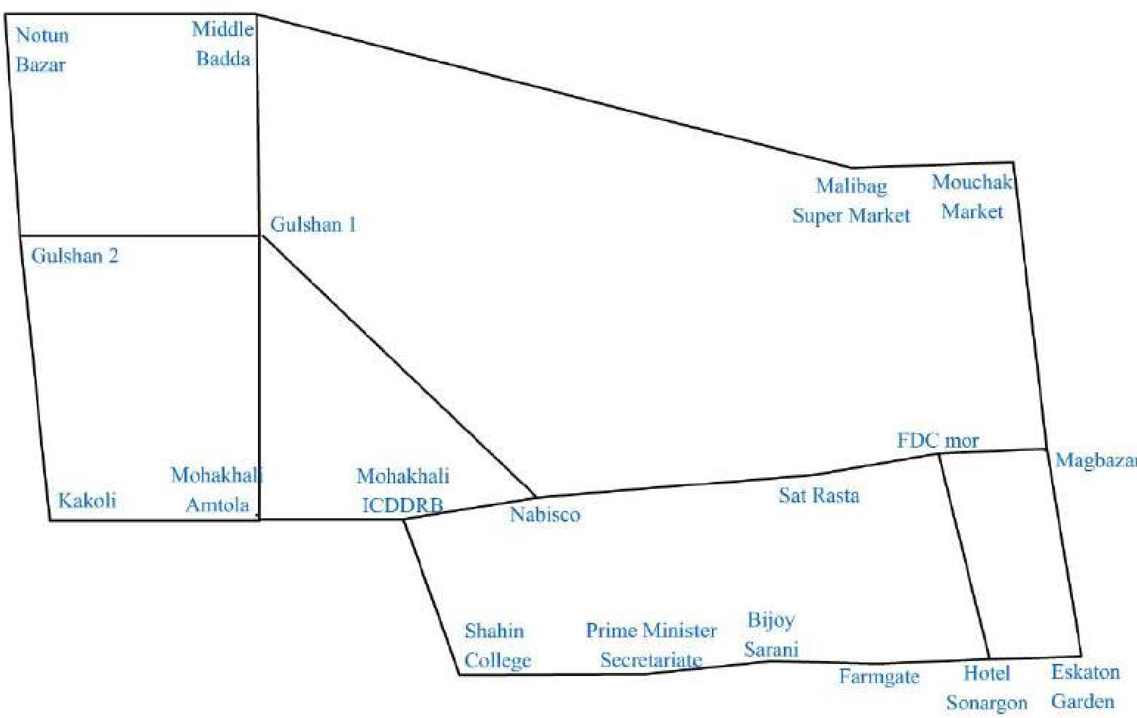


Figure-7: Interface of "Interactive Map"

Send Traffic Information

User will get option to send the traffic information of any particular edge. To perform this task user have to choose the link "Send Traffic Information".

In the form the edge will be loaded using two dropdown lists. In the first dropdown list the name of start node of the particular edge must be selected. Then the associated name of the nodes will be loaded in the second dropdown list. User has to select the jam level by mentioning it using another dropdown list that contains the possible jam levels. The form is shown in Figure-8.

Figure-7: Interface Design of "Send Traffic Information"

Administrative Option

The system administrator will perform the administrative tasks using this. This has the following subtasks:

- Add Node
- Delete Node
- View Node Information
- Add Edge
- Delete Edge
- Update Edge
- View Edge Information
- Add Data Sender
- Delete Data Sender
- View All Data Sender

The options for administrator are shown in Figure-8.

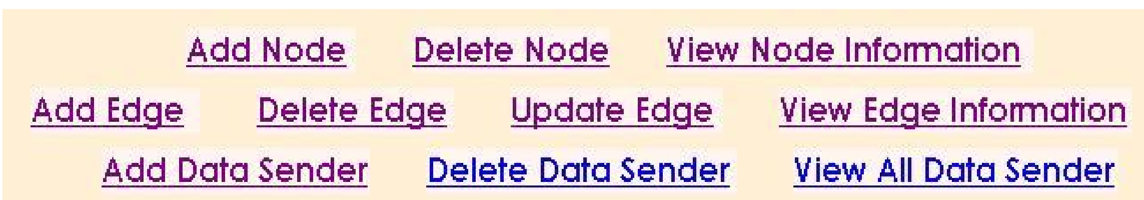


Figure-8: Option in Administrative task

To add a new node administrator have to select the link "Add Node". The form contains four text boxes where sequentially the name of the node, code value of the name, and the coordinate of the node must enter. This interface is shown in Figure-9.

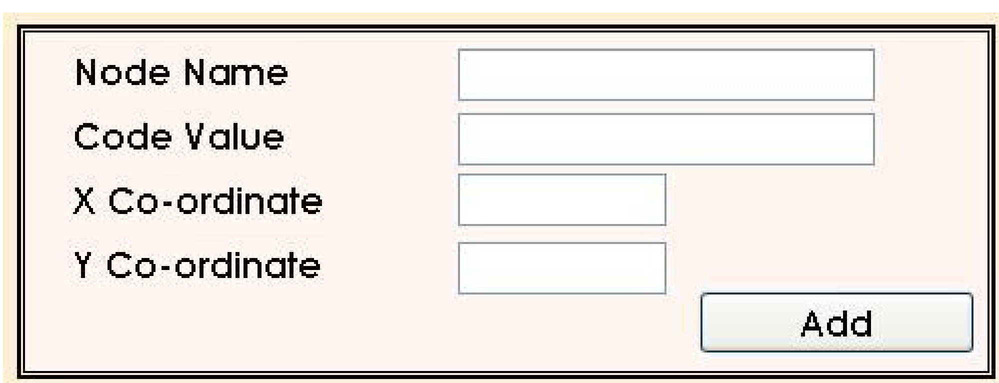


Figure-9: Interface Design of "Add Node" form

To delete a node administrator have to select the link "Delete Node". The form contains a dropdown list that loads all node names. The selected name will be deleted from the records in the server. This interface is shown in Figure-10.

Figure-10: Interface Design of "Delete Node" form

To view node information administrator have to select the link "View Node Information". The form contains a dropdown list that loads all node names. Two options are available to get the output. Text based output or Map based output option can be selected by clicking any of the two buttons "View Map" and "View Info". The information will be shown according to the selected name in the dropdown list. This interface design of the form is shown in Figure-11.

Figure-11: Interface Design of "View Node Information" form

To add an edge administrator have to select the link "Add Edge". The form contains two dropdown lists that load the name of all nodes. The node is added according to the selected names in the lists. This interface is shown in Figure-12.

Figure-11: Interface Design of "Add Edge" form

The procedure to delete is almost similar to add edge. Administrator has to select the link "Delete Edge" to delete an edge. The selected edge will be removed from the records in the database. The interface is shown in Figure-12.

Figure-12: Interface Design of "Delete Edge" form

The procedure to update the edge information is absolutely similar to the "Send Node Information" process described previously. Administrator has to select the link "Update Edge" to update the edge information. The interface is shown in Figure-13.

Figure-13: Interface Design of "Update Edge" form

The task associated to view the edge information is absolutely similar to the "View Edge Info" described previously. Administrator has to select the link "View Edge Information" to show the edge information. The interface is shown in Figure-14.

Figure-14: Interface Design of "View Edge Information" form

One of the administrative tasks associated with data senders is to add data sender. For this administrator has to select link "Add Data Sender". In this form there is a text box where the data sender's mobile number will be entered. The interface is shown in Figure-15.

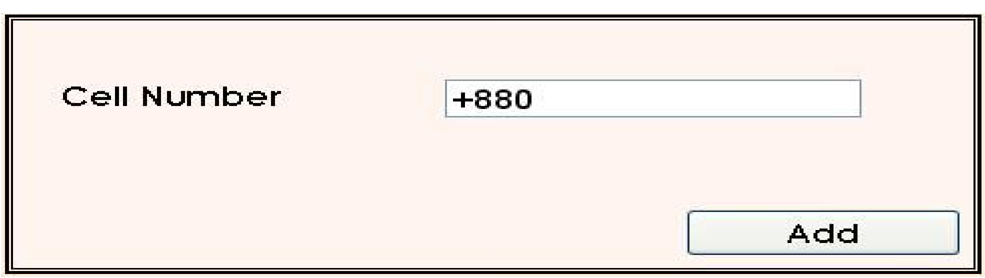


Figure-15: Interface Design of "Add Data Sender" form

To delete a data sender administrator has to select the link "Delete Data Sender". In this form a dropdown list will be loaded that will contain all data senders mobile number. The selected number in the list will be deleted from the list. The interface is shown in Figure-16.

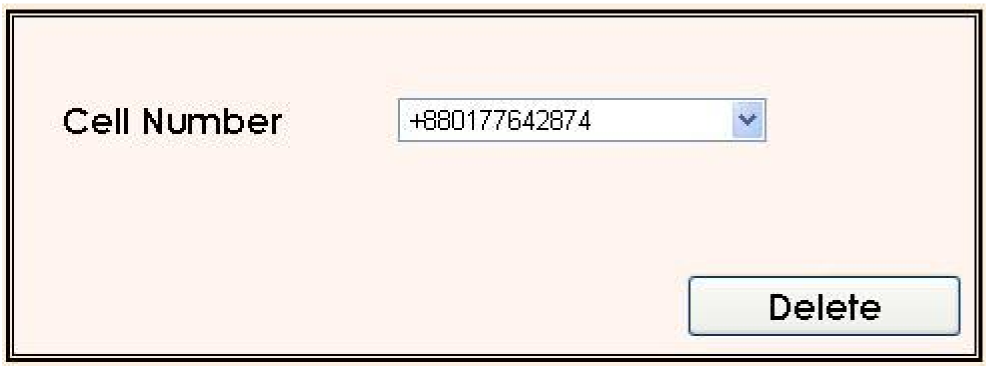


Figure-16: Interface Design of "Delete Data Sender" form

The link “View All Data Sender” provides the option to view all data senders’ number in the record. All number will be shown below the link.

SMS Application

The design of the entire SMS application is divided into two parts: "User SMS Engine" and "Data Sender SMS Engine" and these portions are design and implemented by Md. Abul Hasnat.

Historical Data validation Agent

Historical information is one of the important parts in this system. It is a collection of previous traffic information which is collected from different data sources. The historical database contains only partial information which is suitable. The 24 hour time slot of a day is divided into 7 slots depending on pick hour, off pick hour and some other parameters (e.g. working sector's opening & closing time) and database contains only one most update record for each slot.

Time slot (in 24 hour scale)	Time slot (in 12 hour scale)
00:00 – 07:00	12:00 am – 07:00 am
07:00 – 10:00	07:00 am – 10:00 am
10:00 – 13:00	10:00 am – 01:00 pm
13:00 – 16:00	01:00 pm – 04:00 pm
16:00 – 19:00	04:00 pm – 07:00 pm
19:00 – 22:00	07:00 pm – 10:00 pm
22:00 – 24:00	10:00 pm – 12:00 am

I build up an agent to maintain historical information using some heuristics. This agent stores only one traffic information for each of the above time slot. For example if the system get three traffic information for a particular edge in a day at time slot 07:00am – 10:00am then the agent store only 3rd traffic information for that edge and rest two are ignored. In this way the historical database are grown up.

Validity checking:

What will happen when any data source stops functioning or any particular edge traffic information can not be updated in database lack of real time data. The answer is user will get wrong path information. In this situation the agent uses a heuristics to find suitable traffic jam information and updates the traffic information for that edge. It checks the age of traffic data, if it is older then 8 hours then it searches last 7days record for that edge for running(when agent execute) time slot. Then it searches max frequency of traffic jam level among last 7 days record and replace this jam level with the previous one. Bellow one example shown using some snap shot when agent executes.

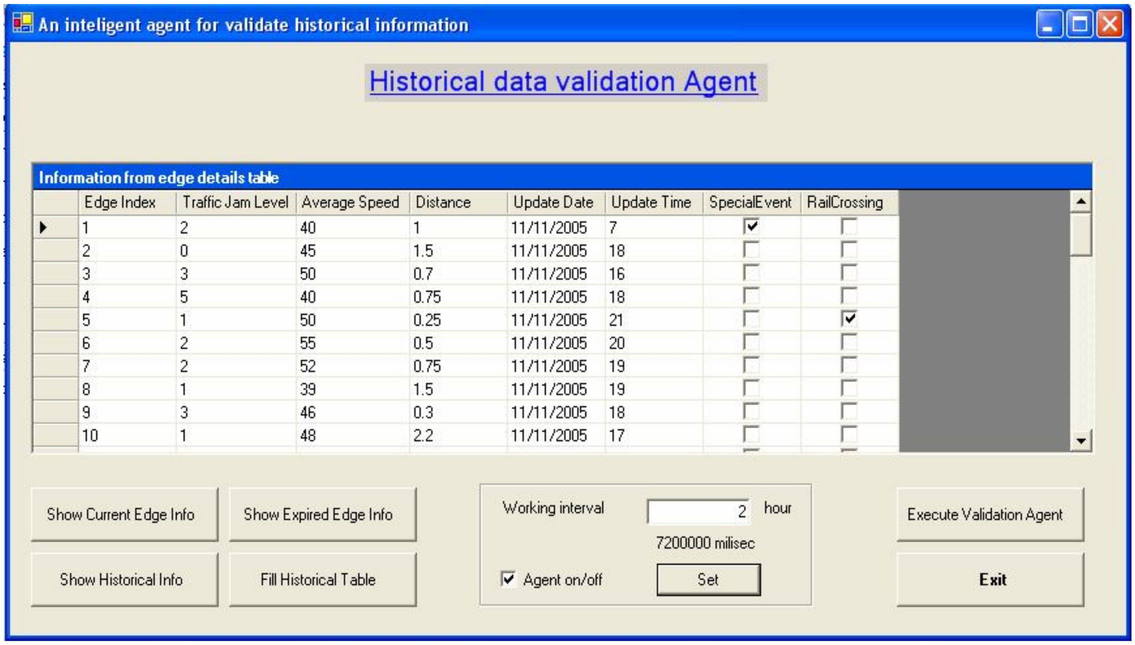


Fig: Shows the historical information of particular day (automated mode)

This agent works in two modes, one is manual mode and another is automated mode. In manual (when uncheck the Agent on/off checkbox) mode system admin does the validation check manually by clicking appropriate buttons. But in automated mode (when check the Agent on/off checkbox) system admin just assigns an interval on which agent executes at regular interval and performs validation check on each edge. After executing the agent finds out expire edge information (older than 8 hours) then apply heuristics.

For example the following figure shows edge 1's (gulshan1 to mohakhali amtola) traffic information where jam level is 2 and it recorded at 07:00 (24 hour scale). But the agent executes at 22:30 (24 hour scale) on the same day. Which means the record is older more than 15 hours.

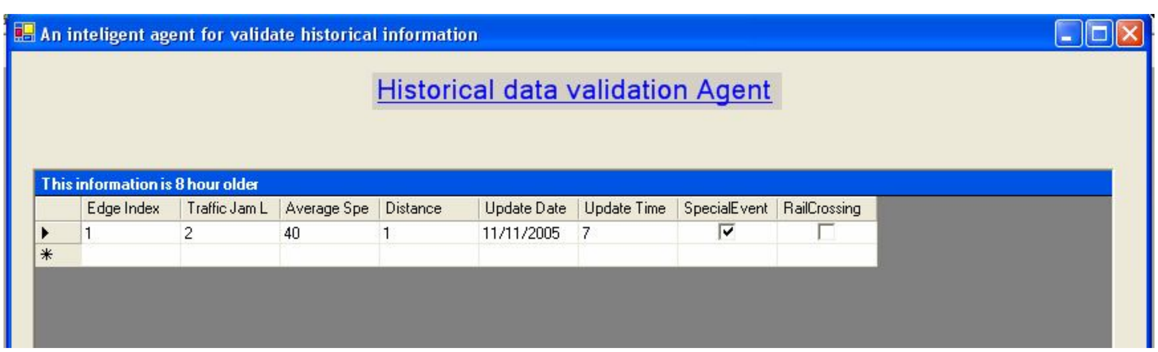


Fig: Shows one expire edge information (in manual mode)

Now it searches last 7 days record in time slot 22:00 – 24:00

Historical Info					
Index	**Date**	**WeekDay**	**StartTime**	**EndTime**	**JamLevel**
1	10-Nov-05	Friday	22	23	1
1	09-Nov-05	Thursday	22	23	2
1	08-Nov-05	Wednesday	22	23	1
1	07-Nov-05	Tuesday	22	23	1
1	06-Nov-05	Monday	22	23	1
1	05-Nov-05	Sunday	22	23	1
1	04-Nov-05	Saturday	22	23	2

Jam level **0** occur 0 times

Jam level **1** occur 5 times

Jam level **2** occur 2 times

Jam level **3** occur 0 times

Max frequency of jam level is **1**

So, by replacing the jam level of edge **1** is updated. Now it looks like below

Edge Index	Distance	Jam Level	Avg Speed	Last ChangeDate	Last ChangeTime	Special Event	Rail Crossing
1	1	1	40	11-Nov-05	22:30	True	False

Now the database containing updated information for edge 1 and user will get more accurate information.

Implementation

We implemented the system as web service so that any third party can use our system in their own application. The system is implemented in .NET platform using C# language. Database is designed with Microsoft Access 2000 and SVG (scalable vector graphics) used for map manipulation on run time. To implement the application for SMS Engines ActiveXpert SMS and Pager Toolkit from ActiveXperts Software is used that can send and receive SMS messages via a GSM modem and GSM phones. It supports windows COM port which can be used to connect with the GSM phone.

When we started coding, we also started testing process in parallel. First of all we wrote test cases for all classes which we will develop later. Then we coded each class one by one and also perform unit test for that class. For .Net platform (C#) we used “NUnit” as test framework and “Source Safe” for source control. “NUnit” is much the same as all the Extreme Programming test frameworks (xUnits). “NUnit” uses the "attribute" feature of .NET to identify tests and it allows write tests any .Net language. We coded the classes and tested it by “NUnit” simultaneously. “NUnit” has the capability to sense the changes of codes which made in .Net IDE.

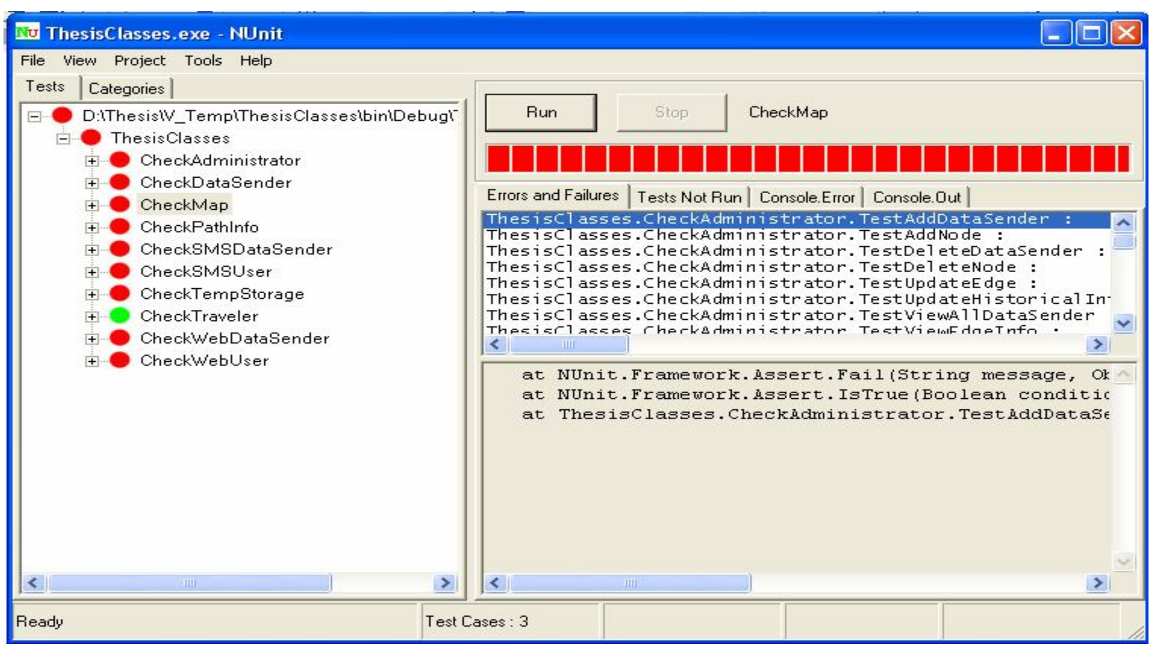


Fig: Testing classes with NUnit

Above snapshot is taken while we run test cases using “NUnit”. Red color means classes are not pass the test cases and Green color means classes pass the test cases.

Source safe is built-in tool which comes with Microsoft Visual Studio 6.0 but it can easily integrate with .Net IDE. To maintain our source code efficiently and keep different versions of code we used it through the whole implementation cycle.

We developed web service consumer using ASP.NET where user can choose source and destination location from a list box and click “Get Path” button to get a path direction.

To visualize the path direction we drew a JPEG map on which we highlighted the optimum path on the fly using SVG. SVG takes some coordinate (these coordinate generated by web service) and draw the path with some animation.

We also built up SMS engine to provide the service through SMS. This engine takes a SMS and then decodes it to pass Source and Destination to the web service.

Testing

Testing is the final part of the software development life cycle. Here we again used "NUnit" for testing classes and functions. We performed unit test, integration test and system test.

Unit test

NUnit performs unit test. Check the function's output for particular input.

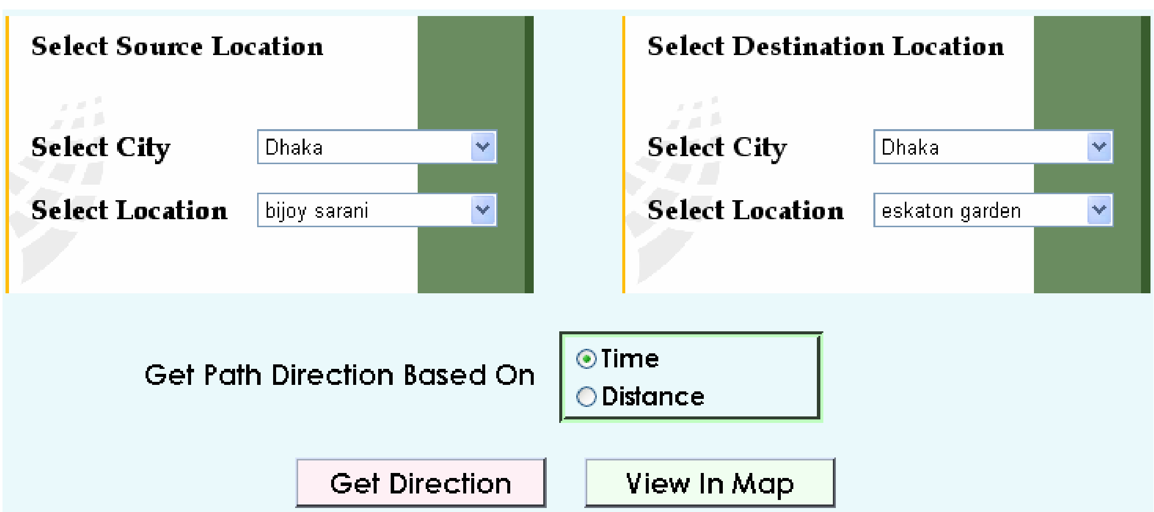
Integration test

Putting together all the modules in top-down fashion performs integration test.

System test

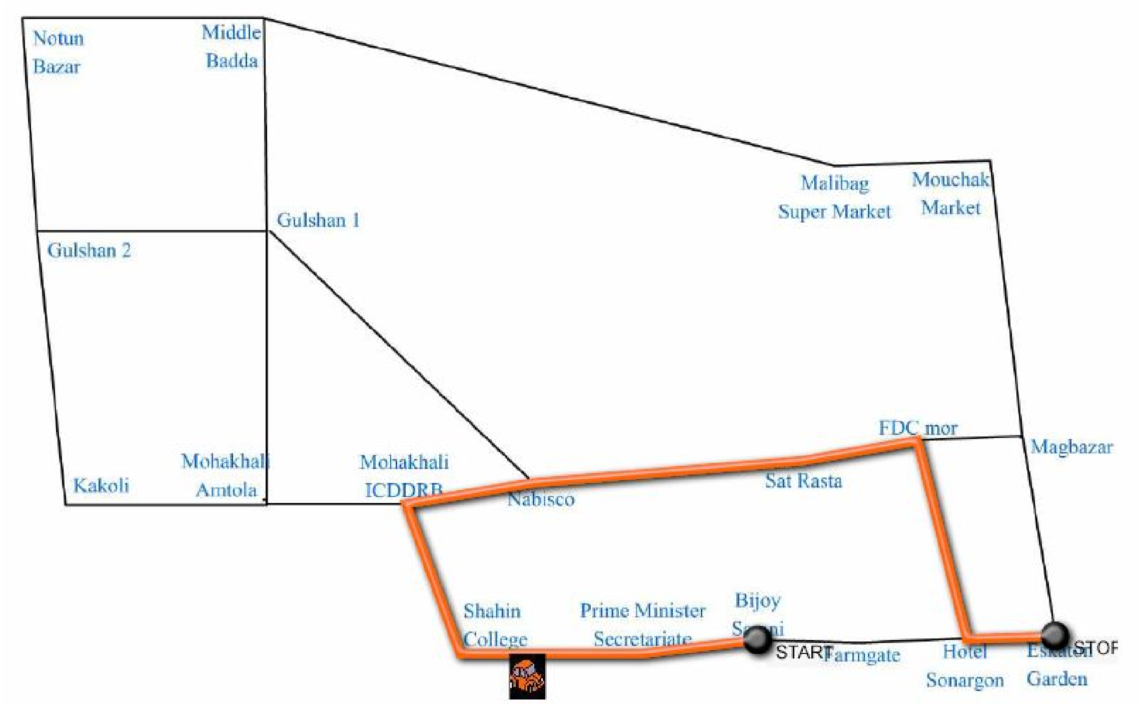
Putting all the programs together that a system comprises and test the system functionality. For example after integrates the whole system we perform a sample test to get path direction in text base output and in map.

After clicking "Get Direction" we get the following path direction with estimated time to each destination.



Source Location:	bijoy sarani
Destination Location:	eskaton garden
Path:	bijoy sarani-prime minister secretariate-shahin college-mohakhali icddrb-nabisco-sat rasta-fdc moor-hotel sonargon-eskaton garden
Estimated time to reach:	15 minutes.

After clicking “View in Map” we get the following map with path indication



The output we got above is exactly the desired output for source location “Bijoy Sarani” and destination location “Eskaton Garden”. So the system works perfectly.

Limitations of the System

Due to time limitation, unavailable resources, equipments and technologies some features cannot be implemented and some are not implemented with full power. A complete investigation on the existing process and comparison with the other Traveler Information Systems finds out the following limitations.

Data collection and manipulation

- Few Reliable Data Source: Now the reliable authorized data source is only the traffic police who send the traffic jam information according to their observation. Data can be obtained from other sources like travelers using web and voice (phone) media but these sources are not always reliable. There is no involvement of any public agency who can contribute to the process of reliable data collection.

- Manual Data Collection Process: Data collection procedure is completely manual and totally depends on sender's surveillance. There is no way to measure the accuracy of the collected data. No technology and equipments like video camera, sensors etc is integrated with the system to automate the data collection process.

- No effect for incidents in the path calculation: Information about the incidents and special events like road reconstruction, road block due to meeting is kept separate without effecting the output path calculation. So this may lead to the generation of suboptimal path.

- Incident and special event information obtain manually: Unlike some other ATIS applications, this system does not use any incident detection algorithm and similar approaches to detect the incidents and events. It totally relies on the data sources to collect this information.

- Lack of collaborations: It only maintains regular communication with the traffic polices who act as a data source for the system. Unlike other ATIS application it has no collaboration with the Road Transportation Authority for vehicle and route information, police stations for collecting incident and event information and other similar public agencies who can provide useful data for the system. It does not maintain any coordination and integration of data between different agencies and the system.

- No differentiation among vehicles: This system is unable to differentiae various types of vehicle and provide output based on the consideration of only one type of vehicle and use its average speed during path calculation.

- Inadequate Geographical Information: Right now the geographical coordinate information that is used in the map to locate different location is not properly organized. Only the direction of the locations is properly maintained. As GPS similar technologies are unavailable so we are not concerning about the actual coordinates.

Data output process:

- Limited media to broadcast information: The system is build up according to the highly available and usable Medias. Right now the system is phone (voice), web, and SMS service enabled. There is no GPS, WAP service, VMS, PDA and pager service.

- Lack of user interaction: The system can ensure limited user interaction into its core operation. In web based service the users must have to choose location from the available names in the dropdown list. Users are restricted to enter the origin and destination locations according to their own choice.

- Provide only specific node and edge info: At present the system is capable to provide information about limited nodes and edges where the number is quite small compared to the total number of locations of the entire city.

- Unable to control congestion: Unlike most of the ATIS, the system does not play any significant role to control the congestion in the road.

- No information about available transportation: The system is completely unable to provide any information about the available transportation and similar type of information like available places for parking etc ha is sometime essential for the travelers.

- Only one output path: The system suggests only one path direction as output against user request. No alternate (2^{nd} option) path information is provided. So sometime it is not sufficient for the travelers to take appropriate decision for their journey.

- Limited area coverage: At present this system covers only some specific portion of the city for testing purposes. So all locations will not be available right now.

- SMS cost in case of large output: Output length is variable for different choice of source and destination. The cost of SMS depends upon the length of output. Yet there is no suitable option to resolve this problem.

- Limited Usage of the Service: At present the system only perform to response user request for the shortest path for any specific source and destination. Unlike many other existing traveler information system this system has no usage on different purpose like application for emergency response and long term disaster recovery, traffic management and congestion control, finding out the vulnerable point of congestion etc.

Future Visions

As Future implementation our primary target to overcome the limitations of the existing service. Then we will focus on to add features that will ensure the service of the system more sophisticated and user friendly. This will enhance and improve the usability and reliability of the system to a higher degree. Keeping this in mind the following features may be added to the system:

Improve Data Collection Process

At present the data collection procedure is totally manual. So our target is to automate the process using appropriate Medias as an improvement. This can be done in the following ways:

- Use Image Analysis: The data collection procedure will be improved using image processing where the image will obtained from the closed circuit video or camera that is placed on each traffic point or potential congestion point in the road. This image will arrive in the server continuously after certain time interval. Then the image of each node will be processed and analyzed using image analysis tools to determine the traffic jam level of that node.

- Use Sensors: We can add special type sensors in the traffic point or in the potential congestion point of the road that can send some signals to the server at a regular time interval. These signals contain special meaning and analyzing those signals the traffic jam level can be identified using jam identification algorithm and update the database accordingly.

Enhance the Usability of the System by Including More Medias

The usability and reliability of the system will increase if we can enhance the service for more accessible Medias with the advancement of technologies and new equipments. So users can easily go through the service and benefited by it. The possible enhancements are discussed below.

- Interactive Map-based Service: The Map-based service will be enhanced so that user can get output by only interacting with the Map with the help of mouse pointer. Users don't have to write anything to get output from the service. To find desired output, users need only to locate their origin and destination using the interactive dynamic mapping features with "point-and-click". User can get several type of output using the same type features. The Map will have the property to show the real time data by continuous interaction with the central database.

- Incorporate Interactive Voice Response (IVR) Service: An Interactive Voice Response (IVR) service will allow users to interact with our system using the cell phone or telephone. IVR will use to enable the users to get output from the service. It also enables the data sender to provide traffic information into the system. It will help the users to avoid the complexity to write SMS. The output will be in Voice Message (VM) format. So the implementation of IVR system will be a replacement of existing Voice (phone) service.

- Ensure Support for PDA, Pagers & WAP enable service: We will implement our system to support the usage of PDA, Pagers & WAP service. Users with regular and web-enabled cell phones or personal digital assistants (PDA's) can access the service via wireless media. WAP web pages are served in a similar manner to the web pages used by Internet. Some scaled down web pages will be written specifically for WAP-enabled devices. Pager media will also support. Unlike portable computers, most PDAs began as pen-based, using a stylus rather than a keyboard for input. So the Interactive map-based service will be the most flexible and user-friendly service for PDA users.

- Incorporate GPS enabled service: GPS enable service can enhance the usability of the system to some significant degree. To use the service users have to send only the destination to the system. The current location (source) will be determined by the system itself from the coordinate of the source. So the system will be capable of generating the desired path direction only from the specified destination. If the GPS device is graphics enable then user will get a visual map that will show the path direction with the locations.

Conclusion

Here we present a complete Real Time Traveler Information System that facilitates the travelers to minimize their travel time by guiding them to travel through a shortest path to go their desired destination. This path will be calculated from the real time traffic data of the road that is collected by the system. Users can access this system through different available and flexible media. The comparison between the existing status of research and development of Traveler Information System and the system that we proposed indicate that yet we can enhance and improve the service through many different ways based on the availability and applications of current technology, equipments and resources. The limitations and future visions discussed here provide a complete guideline in these concerns.

References

1. Moshe E. Ben-Akiva, "DynaMIT - Real Time Traffic Estimation and Prediction System", Massachusetts Institute of Technology.
2. http://www.typo3.com
3. http://www.innovations.harvard.edu
4. http://www.ilog.com
5. http://www.trg.soton.ac.uk/rosetta/workareas/3_mmr/mmr_pr1_present.htm
6. Moshe Ben-Akiva, Haris N. Koutsopoulos, Joan Walker, "DynaMIT-P: Dynamic Assignment Model System for Transportation Planning", WCTR Conference Seoul, Korea, 2001.
7. Research Project: MPC-253, "Effectiveness of Traveler Information Systems", University of Utah.
8. David Lively, Osama Elhamshary, "Lessons Learned from Advanced Traveler Information Systems: Applications for emergency management and long term disaster recovery".
9. Naushad UzZaman and Mumit Khan, "A Double Metaphone Encoding for Approximate Name Searching and Matching in Bangla", Proc. The Fourth IASTED International Conference on Computational Intelligence, Calgary, Alberta, Canada, July 2005.