TE
228.3
.I37
2010

Advanced Traveller Information System (ATIS) Using GPS/GIS

By

Bassim Ibrahim

A project presented to Ryerson University
in partial fulfillment of the requirements for the degree of
Master of Engineering
Graduate Program in Electrical and Computer Engineering
Toronto, Ontario, Canada, 2010

©Bassim Ibrahim, 2010

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

Author's Declaration

I hereby declare that I am the sole author of this project. I authorize Ryerson University to lend this project to other institutions or individuals for the purpose of scholarly research.

[redacted]

I further authorize Ryerson University to reproduce this project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

[redacted]

Abstract

Vehicle arrival time is one of the most important factors of Intelligent Transportation System (ITS). The accurate transit travel information is important because it attracts additional customers and increases the satisfaction of transit users. A passenger waiting for a train or a bus, a person waiting for a cab, a customer waiting for a courier to come to his/her home to pickup or deliver a package, a business office waiting for a truck for goods and a home user waiting for his/her shipment which he/she did online shopping are few examples of how important is vehicle arrival time in different areas of the life. Most of the companies are investing a lot of money to improve their system for better, faster and reliable customer service. As the cost of ITS component has been decreased, Automatic Vehicle Location (AVL) system, which is one component of ITS, has become more widely used. Many transit agencies use AVL system to track their vehicles in real-time. Tracking system technology was made possible by the integration of three technologies: Global Positioning System (GPS), Global System for Mobile Communication (GSM) and the Geographical Information System (GIS).

This project shows a detailed research in the area of Automatic Vehicle Location and implements a low cost vehicle tracking system using GPS and GPRS. The system reads the current position, speed and direction using GPS, the data is sent via GPRS service from GSM network towards a server using TCP/IP protocol and the server saves this information to the database on regular time interval. The web based application then uses this data and calculates the approximate arrival time. The system allows a user to view the present position of the vehicle using Google map and calculates the arrival time. Also bus location can be monitored in real-time, by Route supervisors. This will allow these supervisors to make better service-adjustment decisions because they will be able to see how the route is operating. The test bed was a bus route running in the downtown of Toronto.

Acknowledgment

First of all, I would like to thank my supervisor Dr. Lian Zhao, for her technical knowledge and insight, and for her encouragement and thoughtful guidance in the preparation of this project. She has given me many stimulating suggestions and valuable comments that helped me to finish the project. I appreciate her inspiration and help. She always tries to find how to help and encourage her students. I would like to thank the members of my project committee, Dr. Truman Yang and Dr. Yifeng He for their insights and guidance in one way or another.

I would like to thank and extend my heartfelt gratitude to my faithful friend and classmate Arfan Mohammed for his help and for the pleasant time we spent during the development of this project.

I want to express my sincere appreciation to my mother, my brother and sisters for their help and encouragement to finish my study. I would like to thank my wife, my sons, Osama, Mohammed. I couldn't finish my study without their patience, support and encouragement.

CONTENTS

List of FIGURES:

List of TABLES:

CHAPTER I

1 INTRODUCTION

The Advanced Traveler Information System (ATIS) delivers real-time information of the traffic conditions to the public and provides real-time information to travelers, so that the congestions in the city would be reduced. Also this accurate transit arrival time information attracts additional transit ridership and increases the user satisfaction [1-2].

Automatic Vehicle Locating (AVL) is an advanced method of remote vehicle tracking and monitoring. Each vehicle is equipped with an AVL unit that receives signals from the GPS satellites. The GPS receiver determines its current location, speed and heading. This data can be stored or directly transmitted to an operating centre. Current position can be displayed on a PC in digital maps. Most popular Automatic Vehicle Location is based on GPS (Global Positioning System) and GSM (Global standard of mobile communication).

Automatic Vehicle Location (AVL) provides up-to-date location information for emergency vehicles, delivery trucks, freight trucks, service vehicles etc. The AVL system consists of a GPS receiver on the truck or vehicle, a communications link between the vehicle and the dispatcher, and PC-based tracking software for dispatch. The communication system is usually a cellular network similar to the one used by our cellular phones.

There are three types of traveler information: pre-trip information, in-terminal information, and in vehicle information [1]. These are real-time information which includes: arrival time, departure time, current vehicle location, speed, and delay. Real-time information is very valuable to transit users, as users can decide whether to wait at transit stops or seek another travel mode.

1.1 PROJECT OBJECTIVEES

The purpose of this project was to implement a BUS arrival time using GPS/GIS technology, on the platform of Google map, which can provide relevant bus routes information with bus arrival time to users, also to allow the dispatch center to monitor buses in real time. A bus arrival time, bus location, bus route are calculated on the server side and then the processed data are retrieved and presented to the user and the dispatcher with the help of Google map application. The system software is developed using C#.Net and Java script languages and SQL database server is used for storing data.

1.2 MOTIVATIONS

The aim of this project is to develop an application that predict the arrival time for TTC (Toronto Transit Commission) buses and provide better service adjustment decisions for the benefit of customers. The current system doesn't provide real time next-bus information to the passengers. Passengers depend on the bus-schedule at shelters/stations. Also TTC- Route supervisors use traditional technology to monitor bus location and adjust service from a center location; this technology is known as the Communication Information System (CIS). This system was designed in the 1970's using technology which has significant limitations in terms of accuracy and the amount of information it provides [11]. With the existing TTC system customers will spend more time in case of bus service disruption and can't adjust their travel plans accordingly, no real-time connecting route information, and no real-time vehicle diagnostics.

1.2.1 Reliability

By using the new technology, TTC will be an attractive system for additional ridership. Passengers will rely more on TTC services; also customer service will be improved to allow Route Supervisors to make quicker and more-effective service adjustment decisions. This will improve the congestion in the city, reduce operating and maintenance cost, respond more quickly to emergency situation, and help cleaning environment.

1.2.2 User-friendly

With the new system, passengers have real-time information at a stop/station about when the next bus will arrive. Also next vehicle information is available on the internet. With this information, customers plan their TTC trips in advance.

1.3 Project Framework

The aim of this project is to develop an application that predict the arrival time for TTC buses using GPS/GIS technology. Actual data collected in Toronto, ON, were used in this project as a test bed. The test bed is route 81N, which runs in congested area in Toronto. The GPS provides bus information, location, speed, time, direction...etc. The bus route has 6 bus stops and 4.9Km long. The suggested project framework is shown in FIGURE 1-1.

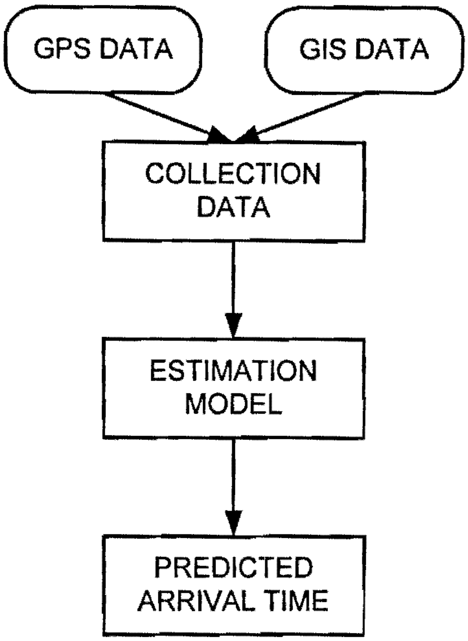


FIGURE 1-1 The Project Framework

1.4 PROJECT CONTRIBUTION AND OUTLINE

1.4.1 Project Contributions

The need for a model or technique to predict transit arrival time using AVL data is increasing. Many transit agencies have adopted Automatic Vehicle Location systems to track their vehicles in real-time. Providing travel time to the passengers is the major part of ATIS. The provision of traveler information can extend the ridership and increase the passengers' satisfaction.

This project uses real-time information and can provide accurate prediction of bus arrival time considering the traffic conditions. This would be very useful information for passengers and transit agencies.

The project consists of a client /server application developed using Microsoft .Net technologies with TCP/IP socket programming to establish a standard AVL system to store and retrieve the geo-location of the buses. A Microsoft .Net based web service was also developed as part of the project to communicate to the database. This web service is hosted on the web server and is behind the firewall to avoid any SQL injection from the hackers. A centralized web based application was developed using Microsoft .Net. ASP.Net and Java Script were used on client side while all objects, controllers and the models were developed in C#. Administrators, dispatchers and the passengers use this centralized application for their own interest. The website communicates to the database through the web service which in turn communicates to the database via ADO.Net.

1.4.2 Project Outline

This project is organized as following:

- Chapter 2 covers the background information on Intelligent Transportation Systems and Automatic Vehicle Location Systems. The information involves a discussion of various categories of ITS, techniques used in a standard AVL system, their pros and cons and the preferred technique that suits our requirement. This chapter also describes the communication media (wireless network), NMEA (National Marine Electronics Association) protocol, client/server TCP/IP communication protocol.

- Chapter 3 presents system design. All objects including client/server application, web service, website, database and stored procedures are discussed here in details. It covers each and every object individually and how it is dependant and relates to other objects. It also discusses the Google map and JavaScript APIs for GIS and travel time estimation algorithm.

- Chapter 4 provides the simulation results with step by step processing and execution.

- Chapter 5 concludes recent work and summarizes the future direction of this project.

CHAPTER II

2 BACKGROUND KNOWLEDGE

2.1 INTELLIGENT TRANSPORTATION SYSTEM (ITS)

Intelligent Transportation System (ITS) is the application of advanced information, control, and communications technology to surface transportation system. The objectives of ITS are to improve the efficiency, safety, and to reduce the environmental impact of the movement of goods and people on the surface transportation system. Intelligent Transportation Systems contain four elements: the vehicle, the user, the infrastructure, and the communications system as shown in FIGURE 2-1. The ITS will enhance the transportation system performance in:

- Traveler information & route guidance
- Incident management
- Traffic control
- Commercial vehicle electronic clearance
- Collision avoidance systems
- Electronic payment services
- Fleet management

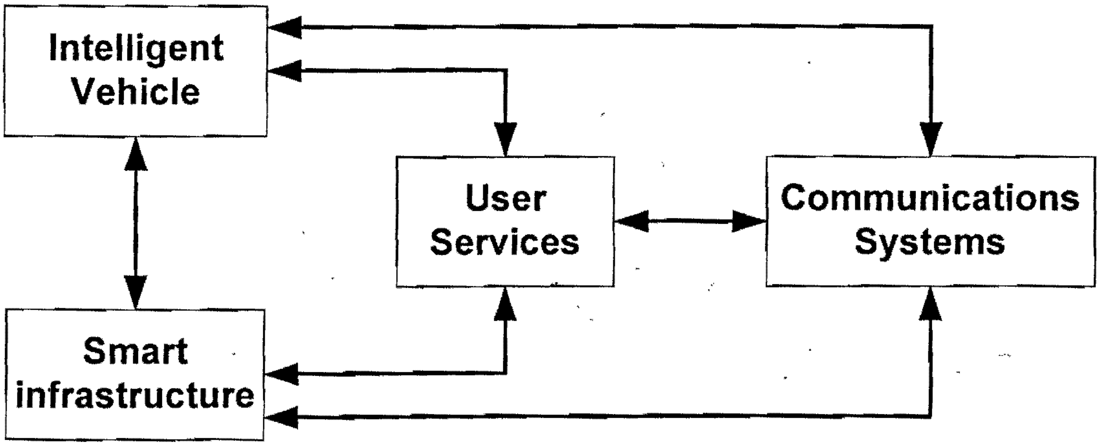


FIGURE 2-1 ITS Components

2.1.1 Intelligent Transportation Systems Categories

There are six inter-related ITS areas:

1. **Advanced Traffic Management Systems (ATMS)**

   ATMS are technologies that monitor, control, and manage traffic on streets and highways. Traffic control centers gather and report traffic information to, support transportation planning, manage traffic incident, apply policy/enforcing traffic regulations, etc. Message signs provide current information on traffic conditions and suggest alternative routes. These systems respond to breakdown and accidents sites by sending tow, service, and emergency vehicles.

2. **Advanced Traveler Information Systems (ATIS)**

   ATIS are technologies that assist travelers to improve the convenience and efficiency of travel by providing more information to make decisions about route choices, estimate travel times, and avoid congestion. These systems provide pre-trip information like trip planning services and public transit route and schedule information that is available on line.

3. **Advanced Vehicle Control Systems (AVCS)**

   AVCS technologies enhance the control of vehicles to improve safety and critical functions such as vision enhancement systems, adaptive cruise control, automatic collision avoidance systems, lane departure warning systems, pre-crash resistant deployment.

4. **Advanced Public Transportation Systems (APTS)**

   APTS technologies used to improve the efficiency, availability, effectiveness, and economics of public transportation systems. Such technologies include fleet monitoring and dispatch systems, on-board displays for operators and passengers, real-time displays at bus stops, and intelligent fare collection systems. APTS help to make public transportation a more attractive for passengers by providing the arrival time and departure status of buses and trains.

5. **Commercial Vehicle Operations (CVO)**

   CVO technologies include the operations associated with moving freight and passengers in commercial vehicles such as trucks and commercial fleets. To regulate these operations as well as activities related to safety assurance, commercial vehicle credentials and tax administration, freight management, commercial vehicles fleet management, and automated roadside safety operation.

6. Advanced Rural Transportation Systems (ARTS)

ARTS are designed to solve the problems arising in rural zones. Examples include: demand responsive transit (DRT) software in transit systems and mobile communication systems for emergency response and highway maintenance services.

2.2 AUTOMATIC VEHICLE LOCATION (AVL) SYSTEMS

Automatic Vehicle Location (AVL) systems, which is a part of ITS, calculate the real-time location of any vehicle equipped with a Global Positioning Satellite (GPS) receiver. They are used in tracking fleet, police, transit, ambulances, and military purposes; FIGURE 2-2 shows a schematic diagram of an AVL system used in transit agencies. The figure shows the standard AVL system with GPS signal arriving from a satellite to the car/truck. Data are then transmitted to the transit center with the use of radio or cellular communications and can be used immediately for daily operations.

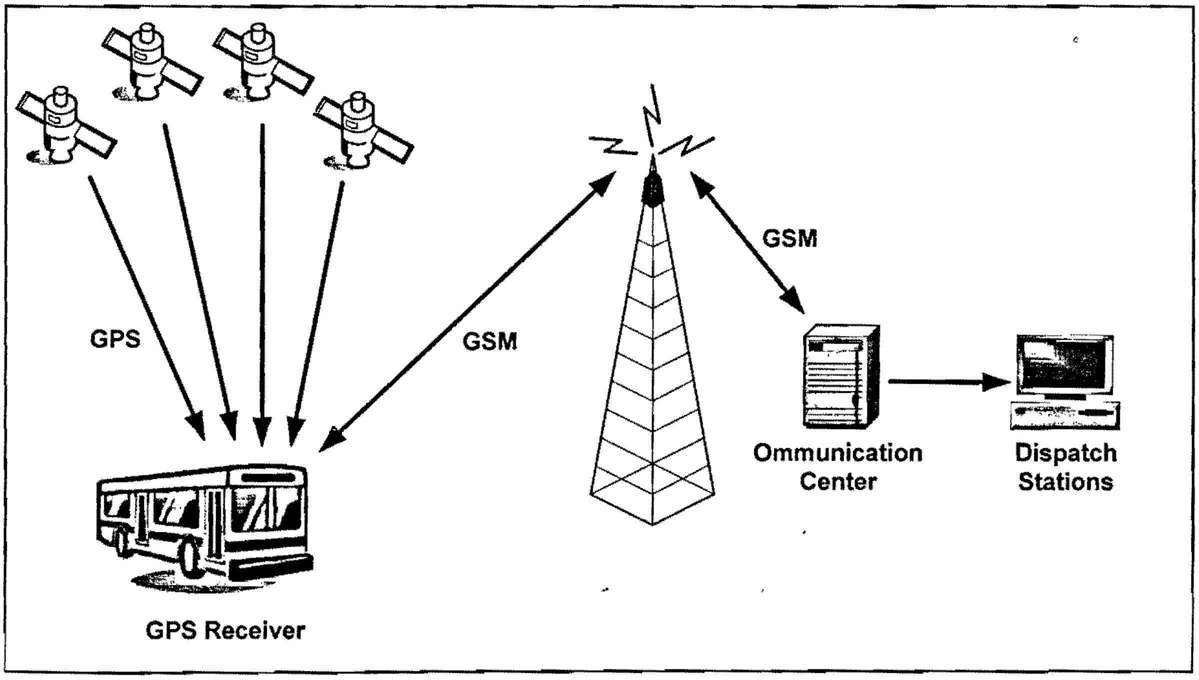


FIGURE 2-2 Basic AVL System

AVL system is a computer-based vehicle tracking system that includes a location technology and a communication method to transmit data from the vehicle to the dispatch center. This data can be stored or directly transmitted to the dispatch center. Currently, the dispatch center uses digital maps integrated with GIS (Geographical Information System) to display the current position of the vehicle.

AVL system has been used by both military and civilian organization to provides up-to-date location information for emergency vehicles, delivery trucks, freight trucks, service vehicles (bus, cab, train) etc. The AVL system consists of a GPS enabled hardware installed on the vehicle, a wireless communications network (GPRS or GSM) link between the vehicle and the dispatch center, and a window based tracking application for dispatch.

2.2.1 Benefits of AVL systems:

The benefits of AVL systems are as follows:

- Provide customers with real-time information.
- Reduce passenger wait time.
- Plan schedule and routes, and improve schedule reliability.
- Improve response times to incidents and emergencies.
- Reduce operating and maintenance costs.
- Improve communications between supervisors, dispatchers, and operators.
- Enhance safety and security

With these advantages gained by using AVL systems as well as their cost has rapidly dropped, many transit agencies are planning to install AVL systems.

2.2.2 Vehicle Location Technologies

The location technologies used in AVL systems are one of the following, or may combine two of these technologies:

- Global Positioning System (GPS);
- Signpost and Odometer interpolation, both active and passive;
- Ground-Based Radio, such as Loran C; and
- Dead Reckoning

2.2.2.1 Global Positioning System:

GPS is a U.S. government operated network of earth-orbiting satellites and ground control stations, and it the most reliable and accurate choice to detect location for Intelligent Transportation System (ITS) as it provides the location of a vehicle with accuracies of about 25-30 feet [3]. More than 90% AVL systems are being implemented using this technology. There are 24 satellites participating in this network, distributed in 6 Orbital planes, each plane has 4 satellites as shown in FIGURE 2-3 [4]. Each GPS satellite continually transmits position information. These signals are picked up by a receiver onboard the bus. To obtain an accurate location and time calculation, four of GPS satellites should be in the range of the receiver as shown in FIGURE 2-4 [4].The accuracy and reasonable cost of GPS makes it the most attractive and appealing option. Foliage, tall buildings, and tunnels can block the satellite signal, and at times satellite signals do not reach specific locations. Some systems use dead reckoning in combination with GPS to overcome such issues.

The GPS receiver measures the signal travel time from the GPS satellite, then it multiplies this time by the velocity of light to obtain the range between the satellite and the receiver [5].

Once the position is calculated, the software in the GPS hardware transforms the geocentric into geographic co-ordinates using World Geodetic System 84 format (WGS84), (Latitude, Longitude, height), calculates the direction and velocity of the vehicle, then saves this information using the National Marine Electronics Association-0183 (NMEA -0183). NMEA-0183 Standard input/output messages protocol could be in different types:

- GGA (GPS Fix Data, fixed data for the Global Positioning System)
- GLL (Geographic Position – Latitude/Longitude)
- GSA (GNSS DOP and Active Satellites, degradation of accuracy and the number of active satellites in the Global Satellite Navigation System)
- GSV (GNSS Satellites in View, satellites in view in the Global Satellite Navigation System)
- RMC (Recommended Minimum Specific GNSS Data)
- VTG (Course over Ground and Ground Speed, horizontal course and horizontal velocity)
- ZDA (Time & Date)

The Recommended Minimum Specific (RMC) GNSS data format is shown in TABLE 2-1[6]:

FIGURE 2-3 GPS Satellites [4]

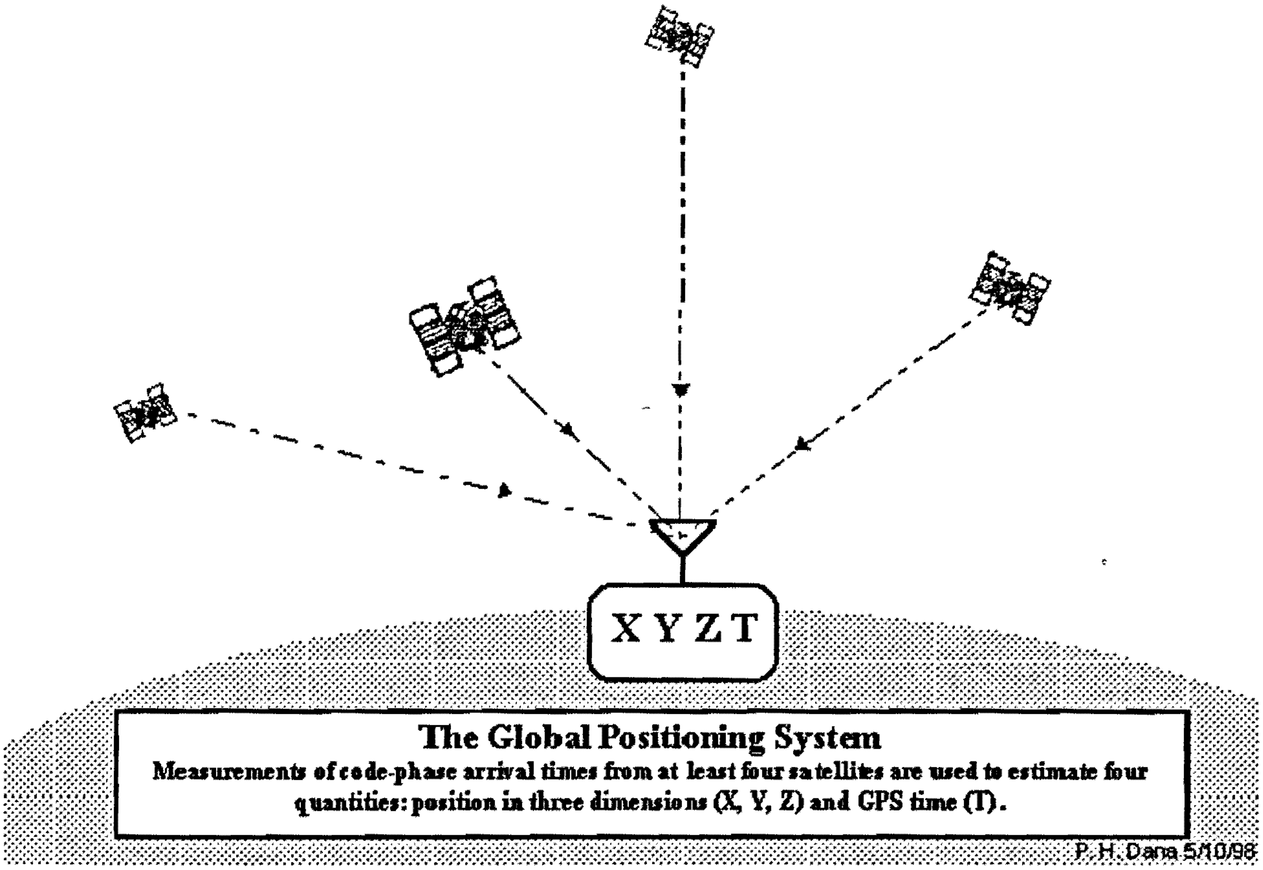

XYZT
The Global Positioning System
Measurements of code-phase arrival times from at least four satellites are used to estimate four quantities: position in three dimensions (X, Y, Z) and GPS time (T).
P. H. Dana 5/10/98

FIGURE 2-4: Four satellite are needed to calculate the position [4]

TABLE 2-1 RMC data format

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header
UTC (time)	101841	hhmmss.ss	Universal Time co-ordinate

Status	A		A = data valid or
			V = data not valid

Latitude	43.680148	ddmm.mmmm	

N/S Indicator	N		N = North or
			S = South

Longitude	-79.345025	ddmm.mmmm	

E/W Indicator	E		E = East or
			W = West

Speed Over Ground	1.2	Knots	
Course Over Ground	15.7	Degrees	True
Date	100810	ddmmyy	

Magnetic Variation		Degrees	E = East or
			W = West

Mode indicator			
Checksum			
<CR> <LF>			End of message termination

2.2.2.2 Dead-Reckoning:

Dead-reckoning sensors use direction/bearing and distance/speed to measure heading (direction) and distance to a new position from a previous position. Odometers, compasses and inertial platforms, such as gyroscopes and accelerators, are dead-reckoning sensors. Typically, these systems act as a backup to another AVL system. Such systems are relatively inexpensive and self-contained on the bus. Dead reckoning, however, has a number of drawbacks. For example, uneven surfaces and hills can compromise the positioning information. Should the vehicle leave a fixed route, its location will no longer be known since there will be no waypoints off the fixed route. Also, accuracy degrades with distance traveled, and regular recalibration is required.

2.2.2.3 Signpost:

Most early AVL systems used signpost technology. This technology composed of an on-board short-range communication device and an infrastructure mounted beacon. System comprises of a receiver mounted on the bus, while transmitters are placed along the route. The bus picks up a low-powered signal from these transmitters as it passes by and the mileage is noted from odometer. When the bus reports its location, the distance from the last pole is used to locate the vehicle's position on a route. The system can be run in reverse, with the transmitter on the bus and multiple receivers mounted along the route. There are some drawbacks of such systems. For example, if the bus needs to leave the route then there will be no information about the bus; so most agencies prefer to have a receiver on the bus. Similarly, creation of new routes requires the placement of new transmitters, and the system needs intensive maintenance due to the relatively high number of transmitters and receivers involved. This approach needs more investment.

2.2.2.4 Radio Navigation:

These systems are defined as any location technology that relies on radio signal to determine position. LORAN-C (Long Range Aid to Navigation), which is maintained by the U.S. under the direction of the Cost Guard, is the most common type of land-based radio location. A LORAN C uses a low frequency wave that is generated by several towers that are synchronized to each other. Each station will transmit in an order and at exact time in relation to others, forming a pattern. A LORAN receiver on a vehicle makes time measurement of signals transmitted from at least three stations. These measurements determine how far away the vehicle is from the fixed locations transmitting the signals. Applying the fixed location of the transmitters from which the signals were received, the receiver uses triangulation to calculate the vehicle's latitude and longitude. Although such systems are very simple but, they are subject to some major drawbacks. For example, overhead power lines or power substations can cause signal interference, and signal reception is typically very poor in canyons.

TABLE 2-2 shows the advantages and disadvantages of different location technologies.

TABLE 2-2 Advantages and Disadvantages of Location Technologies

Type	Advantages	Disadvantages
GPS	Accurate Global coverage Moderate cost per vehicle	Signal attenuation by foliage and tunnels Subject to multi-path errors
Dead reckoning	Inexpensive Self-contained on vehicle (no infrastructure costs) Only odometer needed (if on-route is assumed)	Accuracy degrades with distance traveled Requires direction indicator and maybe map matching for off-route use. Corrupted by uneven road surfaces, steep hills, or magnetic interference
Signpost and odometer	Low in-vehicle cost No blind spots or interference Repeatable accuracy	Requires well-equipped infrastructure No data outside of deployed infrastructure Frequency of updates depends on density of signpost
Radio Navigation/ Location	Moderately accurate Costly	signals are susceptible to electromagnetic distortion by power lines, neon signs, railroad tracks, etc.

2.2.3 Global System for Mobile Communication (GSM)

A communication system is needed to establish a communication link between the vehicle and the dispatch center. The two most common data transmission technologies are: polling and reporting [1, 7]. With polling technology, the computer at the dispatch center asks each bus for its location at regular intervals. With the reporting method, each bus reports its location at a few specified locations or when the bus is found to be off-schedule by some pre-defined tolerance.

GSM is a digital cellular communication system that is the most popular second generation (2g) digital cellular system in the world. There are four main versions of GSM system, depending on the operating frequency. These are: (a) GSM 900, (b) GSM 1800, (c) GSM 1900 and GSM 450. One of the popular GSM services is the Short Message Service (SMS) that allows users to send and receive point-to-point text messages up to a few tens of bytes. The SMS service provides a basic tool to transfer data used to estimate position or co-ordinates of the mobile station [8, 9]. Because of the limited data transfer in the GSM, new technologies have been developed on top of GSM such as the (2.5g), General Packet Radio Service (GPRS), and the third generation (3g) cellular system Universal Mobile Telecommunications System (UMTS). GPRS, which is often referred to as 2.5g, is emerged to optimize the Internet/Intranet access capabilities. GPRS is a new IP-based technology of packet data transmission in a mobile communications network [8]. The main features of GPRS technology [8]:

- GPRS is a packet-based network of data transmission.
- GPRS mobile devices are always connected to the network which gives the possibility of bidirectional communications between the client and server on the Internet instantaneously.
- The subscriber will be able to pay only for the data transmission and not for the time period of subscription to the network. This advantage decreases the cost of AVL system dramatically instead of using SMS whose charging algorithm is based on connection time rather than amount of traffic (Packets) transferred.

2.3 GEOGRAPHICAL INFORMATION SYSTEM (GIS)

Geographic Information System (GIS) is a computer based tool for mapping and analyzing things that exits and events that happen on earth. A geographic information system (GIS) captures, stores, analyzes, manages, and displays geographical reference information, i.e. data identified according to their locations. Technically, GIS is geographic information systems which includes mapping software and its application with remote sensing, land surveying, aerial photography, mathematics, photogrammetry, geography, and tools that can be implemented with GIS software [25]. Work on GIS began in late 1950s, but the first GIS software came only in late 1970s from the lab of the Environmental Systems Research Institute (ESRI). Canada was the pioneer in the development of GIS as a result of innovations dating back to early 1960s.

A GIS can perform complicated analytical functions and then present the result visually as maps, tables or graphs. Most of the information we have about our world contains a location reference, placing that information at some point on the globe. For example, when rainfall information is collected, it is important to know where the rainfall is located. This is done by using a location reference system, such as longitude and latitude, and perhaps elevation. Comparing the rainfall information with other information, such as the location of marshes across the landscape, may show that certain marshes receive little rainfall. This fact may indicate that these marshes are likely to dry up, and this inference can help us make the most appropriate decisions about how humans should interact with the marsh. A GIS, therefore, can reveal important new information that leads to better decision making.

"Every object present on the Earth can be geo-referenced"[25], is the fundamental key of associating any database to GIS. Here, the term 'database' is a collection of information about things and their relationship to each other and 'geo-referencing' refers to the location of a layer or coverage in space defined by the co-ordinate referencing system (X, Y and Z co-ordinates).

2.3.1 Advantages and Applications of GIS

A GIS helps to answer questions and solve problems in a way that is quickly understood and easily shared, it has been an effective tool for implementation and monitoring of municipal infrastructure. GIS is being used in various fields [26].The use of GIS has been in vogue primarily due to the advantages mentioned below:

- Can cope with larger amounts of data
- Can cover large study areas (the whole world if necessary)
- Can conveniently select any sub-study area
- Can cope with unlimited and frequent edits and changes
- More robust and resistant to damage
- Faster and more efficient
- Requires less person time and money

Following is a few example applications of GIS:

GIS is used in urban planning, housing, transportation planning architectural conversation, urban design, and landscape.

A GIS is used in street network applications such as, vehicle routing and scheduling, location and site selection, and disaster planning. Cities and utilities use GIS every day to help them map and inventory systems, track maintenance, monitor regulatory compliance, or model distribution analysis, transformer analysis, and load analysis.

GIS is used every day to help protect the environment. As an environmental professional, one can use GIS to produce maps, inventory species, measure environmental impact, or trace pollutants. The environmental applications for GIS are almost endless.

Geologists use GIS every day in a wide variety of applications. GIS can be used to study geologic features, analyze soils and strata, assess seismic information, or create 3-dimensional displays of geographic features.

GIS can help in zoning, sub-division plans review, land acquisition, environment impact analysis, nature quality management and maintenance. Also GIS can locate underground pipes and cables for maintenance, planning, tracking energy use.

2.3.2 GIS Location and Coordinates

Any location on Earth is described by identifying the X-Y earth coordinates at a time and place [29]. The most common coordinate system is longitude and latitude. A longitude (shown as vertical lines) is the angular distance, in degrees, minutes, and seconds, of a point east or west the Greenwich Meridian. The latitude is defined as the angular distance in degrees, minutes, and seconds, of a point north or south the Equator. Lines of latitude are often referred to as parallels. The concept of longitude and latitude is shown in FIGURE 2-5.

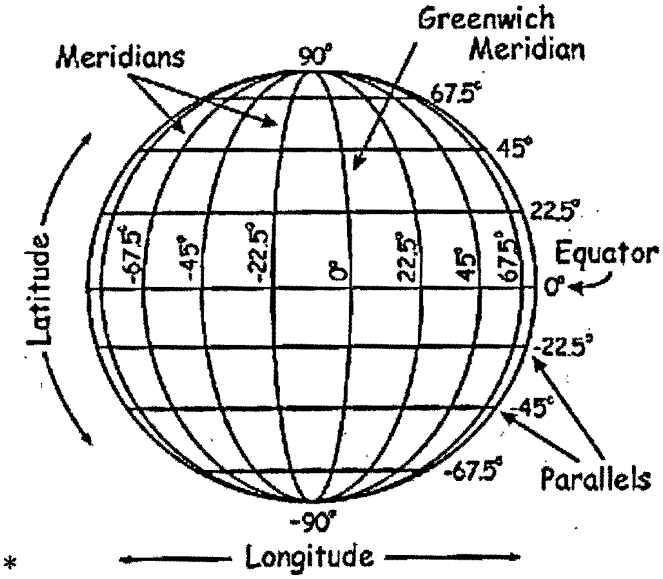


FIGURE 2-5 Latitude / Longitude Concepts [29]

A third dimension can be added for elevation, resulting in (X, Y, Z) coordinates. For GIS, three coordinates allow 3-D displays and analysis. Such a display could be useful for architects, builders, and city planners.

A geographic coordinate system enables every location on the Earth to be specified in three coordinates, using mainly a spherical coordinate system. The Earth is not a sphere, but an irregular shape approximating an ellipsoid; the challenge is to define a coordinate system that can accurately state each topographical feature as an unambiguous set of numbers.

2.3.2.1 Geo-coding and Reverse Geo-coding

- ***Geo-coding***

Geo-coding is the process of finding associated geographic coordinates (often expressed as latitude and longitude) from other geographic data, such as street addresses, or zip codes (postal codes). With geographic coordinates the features can be mapped and entered into Geographic Information Systems, or the coordinates can be embedded into media such as digital photographs via geo-tagging. Converting an address to a specific point location on a map can serve as a valuable tool in many applications from simple data analysis to business and customer management to distribution techniques.

- ***Reverse Geo-coding***

Reverse geo-coding is the process of using latitude and longitude coordinates to find specific geographical features or street addresses [28]. This permits the identification of nearby street addresses, places, and/or aerial subdivisions such as neighbourhoods, county, state, or country. Combined with geo-coding and routing services, Reverse geo-coding is considered an important step in many navigation technologies, like on-board vehicle navigation and Enhanced 911 service rely on reverse geo-coding to estimate the location of destinations.

The process of standard geo-coding and reverse geo-coding go together. Using standard geo-coding techniques, the beginning and endpoint of a street have exact latitude and longitude coordinates laid out. Geo-coding software lays out the street in a mapping program, and the user enters the first and last street address for the area. Using reverse geo-coding, the program extrapolates the street numbers of the houses along the mapped road using coordinates and relative position.

2.4 BUS ARRIVAL PREDICTION ALGORITHM

The accurate prediction of travel time is critical in ITS transit application. There are several commonly used bus arrival prediction algorithms in the market. This project is not focusing on the evaluation of each algorithm; a brief investigation of these algorithms is useful [2].

2.4.1 Historical Data Model

Historical data model predicts travel time for a given time period using the average travel time for the same time period obtained from a historical data base. These models assume that traffic patterns are cyclical and the ratio of the historical travel time on a specific link to the current travel time reported in real-time will remain constant. The procedure requires an extensive set of historical data and it is difficult to install the system in a new setting.

2.4.2 Regression Models

Regression models are conventional approaches for predicting travel time. Regression models predict a dependent variable with a mathematical function formed by a set of independent variables. To establish a regression model, the dependent variables need to be independent. This requirement limits the applicability of the regression model to the transportation areas because variables in transportation systems are highly inter-correlated.

2.4.3 Kalman Filtering Models

Kalman filter is a set of mathematical equations that provide an efficient computational (recursive) solution of the least-square method. The filter is very powerful in several aspects: it supports estimation of past, present, and future states [31, 32].

2.4.4 Artificial Neural Network Models

Due to their ability to solve complex non-linear relationships, artificial neural network models (ANNs) have been developed for transportation since the early 1990s [2]. ANN models had better results than those of existing link travel time techniques, including a Kalman filtering model, a historical profile. While other models are dependent on cyclical traffic data patterns or need independence between dependent and independent variables, ANNs do not require that variables be uncorrelated and/or that they have a cyclic pattern. ANN is good at pattern recognition, prediction, classification, etc. ANNs have two stages, training and testing. During the training stage, inductive learning principles are used to learn patterns from a training set data. There are two types of learning processes used: unsupervised and supervised learning. In unsupervised learning, the network attempts to classify the training set data into different groups based on input patterns. In supervised learning, the desired output from output layer neurons is known, and the network adjusts the weight of connections between neurons to produce the desired output. During this process, the error in the output is propagated back from one layer to the previous layer by adjusting the weights of the connections. This is called the back-propagation method, which is the most frequently used technique in transportation applications. The learning process of ANNs can be continuous so that the models can adapt to changes in environmental characteristics. In other words, ANN models can be considered dynamic prediction models because they can be updated and modified using new online data [2].

2.4.5 Mathematical Model

Mathematical models developed by researchers to calculate the bus arrival time using factors that affect the arrival time. As an example Ojili [12] developed a model to calculate arrival time by breaking the route into one-minute time zones along the bus route. After locating the current bus location with respect to these zones, the arrival time is predicted by counting the estimated number of the one-minute time zones between the current location and the given stop. This algorithm ignored the effect of congestion and variation in dwell times by the time of the day and the day of the week. Lin and Zeng [13] used bus location, schedule information, schedule adherence, and waiting time at time-check stops in their algorithm. The algorithm couldn't consider traffic congestion and dwell time at stops. They collected their data using GPS that provides data every forty six seconds on average. In this research, the longest time interval was almost seven minutes. These long intervals of location data could result in inaccurate travel time predictions. Li Weigang and Marlon W. [14] developed a model that estimates the speed of the vehicle based on an empirical proposal and the calibration during the practical experiment, without considering dwell time and schedule information.

CHAPTER III

3 SYSTEM DESIGN

This chapter covers the complete project in details. It provides detailed information about the architecture, various modules and how these modules are integrated to become a complete system. In the first section of this chapter, the test bed and data collection are detailed.

3.1 TEST ROUTE AND DATA COLLECTION

3.1.1 Test Route

The test bed is route 81N, Downtown Toronto, which runs in congested area in Toronto. The GPS provides bus information, location, speed, time, direction…etc. The bus route has 6 bus stops and 4.9Km long as shown in FIGURE 3-1 and FIGURE 3-2. TABLE 3-1 shows the distance between stops.

TABLE 3-1 Distance between Stops in Route 81N

Stop number	Distance (m)	Accumulated Distance
1	0	0
2	730	730
3	440	1170
4	450	1620
5	2150	3770
6	1220	4990
	average= 623.75	Total=4990

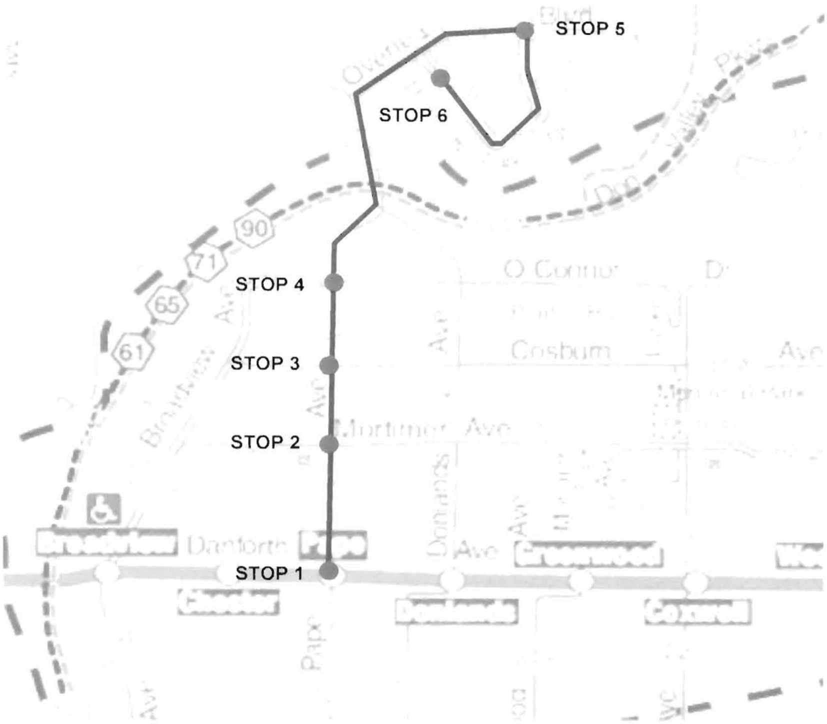


FIGURE 3-1 Map of Route 81N (TTC map)

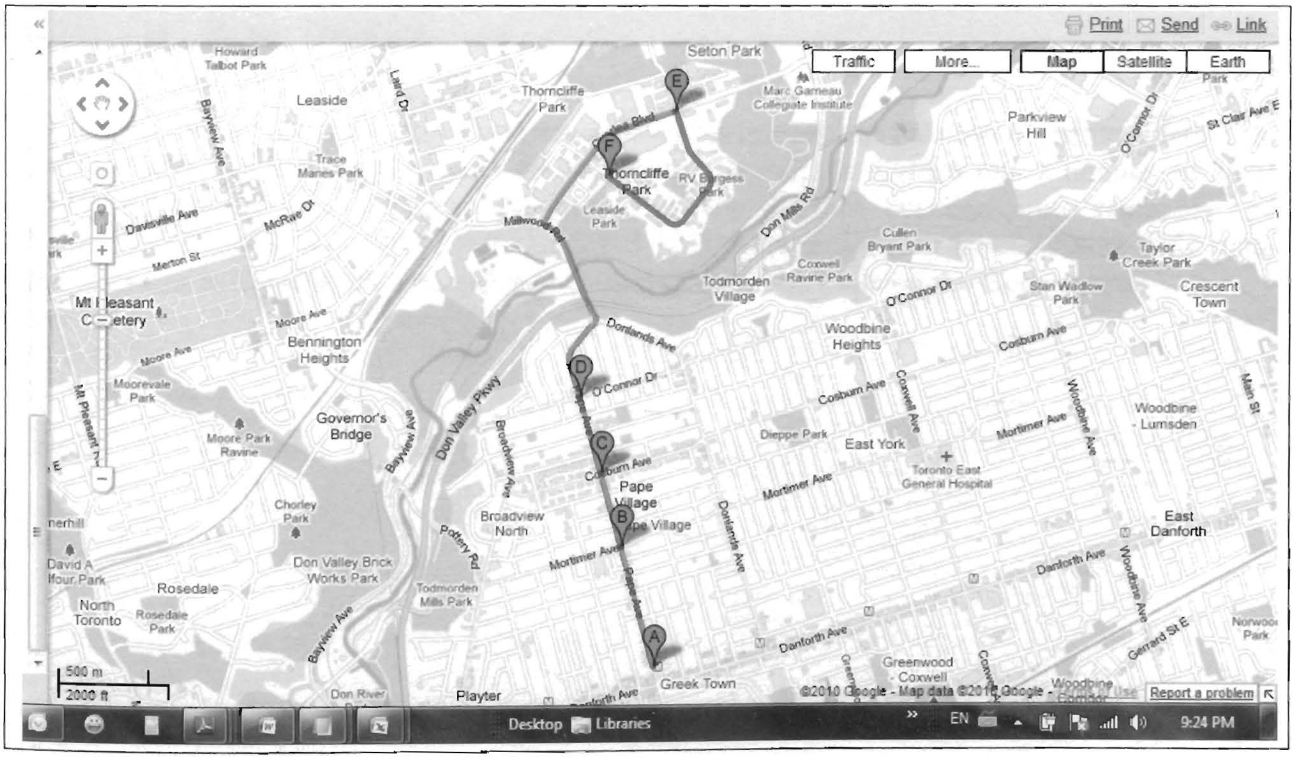


FIGURE 3-2 Route 81 N (Google map)

3.1.2 Data Collection

In this project we collected Bus information by GPS. Bus schedules, route information are collected from TTC [15], this data is available every 3 month and clustered by time period, as shown in TABLE 3-2. According to this time period, there are different bus schedules [15]. The bus location data have the current time at specific bus stops, and the time data determined by GPS unit were used to decide in which time period a bus is located.

TABLE 3-2 Time period definitions

MONDAY TO FRIDAY		SATURDAY AND SUNDAY	
Morning Peak	6:00 am to 9:00 am	Early Morning	6:00 am to 8:00 am
Midday	9:00 am to 3:00 pm	Morning	8:00 am to 12:00 noon
Afternoon Peak	3:00 pm to 7:00 pm	Afternoon	12:00 noon to 7:00 pm
Early Evening	7:00 pm to 10:00 pm	Early Evening	7:00 pm to 10:00 pm
Late Evening	10:00 pm to 1:00 am	Late Evening	10:00 pm to 1:00 am
Overnight	2:00 am to 5:00 am	Overnight	2:00 am to 5:00 am (9:00 am Sundays)

3.1.2.1 Bus Route Tracking

Google map was used to display the distance from the current bus location to the destination. In order to make the algorithm follows the bus route; the route is divided into a number of short intervals (segments) as shown in FIGURE 3-3. The distance between route segments and location information are shown in TABLE 3-3 with average segment length 208m. This route segmentation is saved in the database and is used to draw bus route on the Google map.

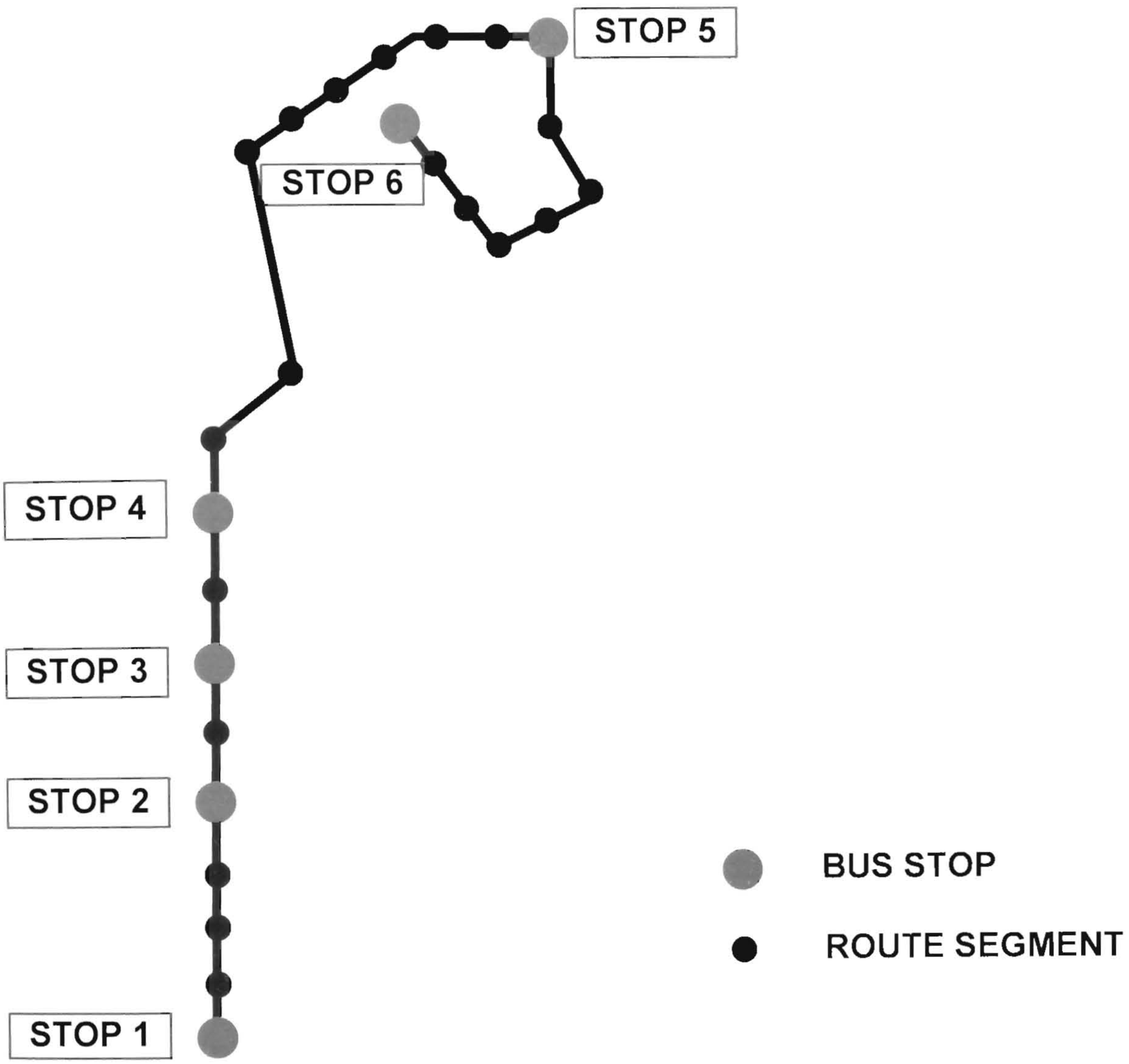


FIGURE 3-3 Bus Route Segmentation

TABLE 3-3 Route Segments

Route_Id	Stop_Id	Segment_Id	Latitude	Longitude	Segment Length (m)	Starting Point (m)	Ending Point (m)
81N	1	0	43.68	-79.345025	110	0	110
81N	1	1	43.68	-79.3454161	200	110	310
81N	1	2	43.682	-79.3460566	260	310	570
81N	1	3	43.684	-79.3468976	160	570	730
81N	2	0	43.686	-79.3474298	200	730	930
81N	2	1	43.687	-79.3480889	220	930	1150
81N	3	0	43.689	-79.3488096	220	1150	1370
81N	3	1	43.691	-79.3495432	220	1370	1590
81N	4	0	43.693	-79.3502584	250	1590	1840
81N	4	1	43.695	-79.3509633	190	1840	2030
81N	4	2	43.697	-79.3495107	700	2030	2730
81N	4	3	43.702	-79.3529367	200	2730	2930
81N	4	4	43.703	-79.3515544	240	2930	3170
81N	4	5	43.705	-79.3497525	200	3170	3370
81N	4	6	43.706	-79.34806	200	3370	3570
81N	4	7	43.707	-79.3457184	92	3570	3662
81N	4	8	43.707	-79.3446308	140	3662	3802
81N	5	0	43.707	-79.3433605	170	3802	3972
81N	5	1	43.706	-79.3426945	86	3972	4058
81N	5	2	43.705	-79.3420478	170	4058	4228
81N	5	3	43.704	-79.3412969	200	4228	4428
81N	5	4	43.702	-79.3427511	200	4428	4628
81N	5	5	43.702	-79.3447317	210	4628	4838
81N	5	6	43.703	-79.3467806	160	4838	4998
81N	6	0	43.704	-79.3483279			
					Avg.=208 m		**Total= 4998m**

3.2 ARRIVAL TIME ALGORITHM

A mathematical algorithm was developed to calculate the estimated arrival time based on the real-time and historical data. Following activities are performed by the System /user to calculate /display the bus arrival time.

1. Input Data (AVL)

   - Client side application continuously communicates to the GPS and transmits this data to the server after fixed interval of time (10 seconds, this interval is configurable).
   - Server calls web service to communicate to the database. It sends:
     - Time, speed, and Geo-location of the vehicle.
     - Current Stop number travelled by the bus.
     - Current segment between the current stop to the next stop.

2. User browse website to check the bus arrival time:

   - User selects route and stop number.
   - System displays bus route in Google map.
   - User clicks on Arrival Time button.
   - System has destination stop number. It checks all buses with stop number less than the destination. If it finds more than one bus then it checks the bus which has travelled more segments. This bus is considered to be the closet one to the destination.

3. System applies Mathematical Algorithm:

   - System calculates the time taken by the bus from the current bus-location to the next bus-stop.
   - If the next stop is the destination stop then the system shows this time as arrival time to the user.
   - If the next stop is not the destination then it calculates the time from the next stop to the destination (this is done by using Bus-schedule data stored in the database).
   - Finally, the system draws on Google map the current location of the bus, destination stop, distance between source and destination, current speed of the bus and the time taken by the bus to reach the destination.

3.2.1 Arrival time Estimation

Basic flowchart of the algorithm is shown in FIGURE 3-4.

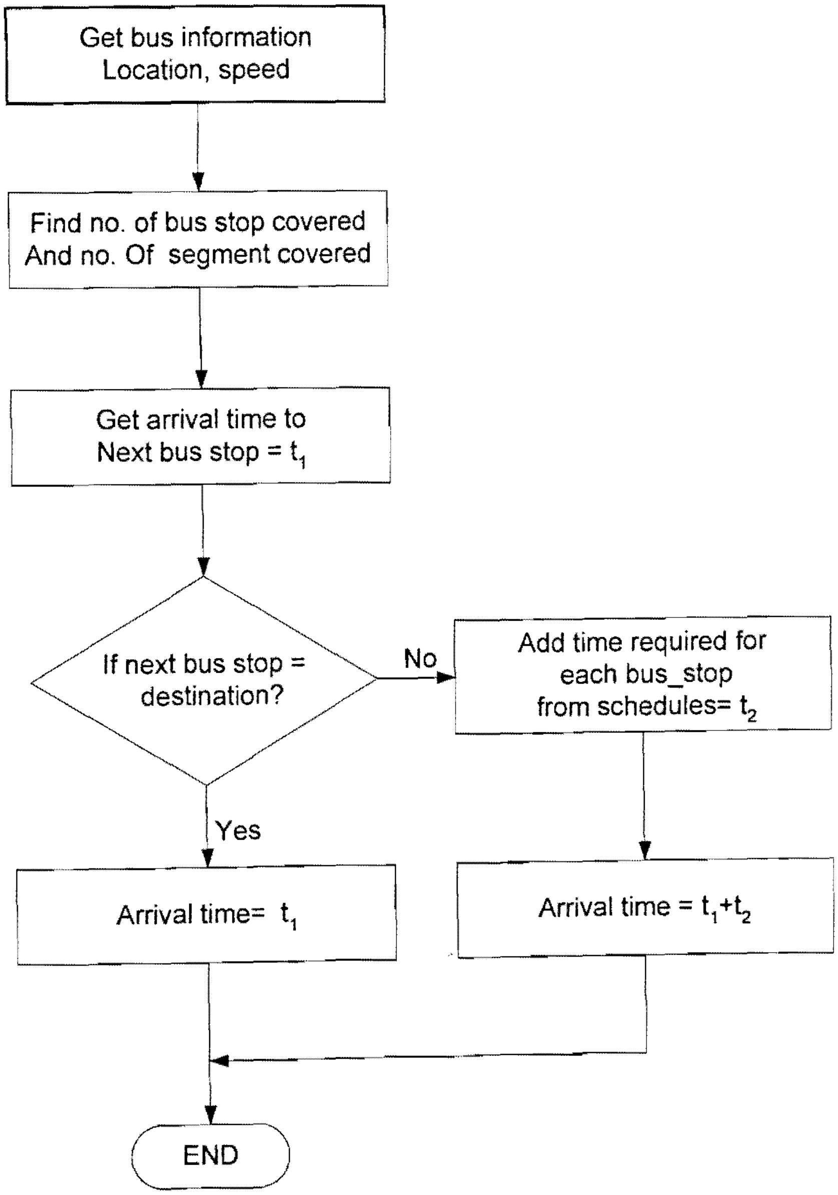


FIGURE 3-4 Mathematical Algorithm Flowchart

To calculate the arrival time from the current bus location to the nearest bus-stop:

$$t_i = l/v \tag{1}$$

Where,

t_i = travel time to the nearest bus stop i

l = the distance from current bus location to the nearest bus stop

v = current bus speed obtained from the GPS

In particular case if the bus speed is zero, $v = 0$, the travel time, t, would be infinity. This may occur for the number of reasons, for example, when the bus arrives at a bus stop or stops at traffic signals. To solve this problem, the average speed of the bus is considered which is obtained from the historical data during similar periods and the current bus speed.

$$v_{avg} = \frac{v + v_a}{2} \tag{2}$$

Where,

v_{avg} = average bus speed (obtained from historical data)

v = Current bus speed obtained from the GPS

v_a = Average bus speed from the historical data

Arrival time to the destination stop k can be calculated as shown in Equation (3).

$$A_T = t_i + \sum_{j=i+1}^{K} t_j \tag{3}$$

Where,

A_T : arrival time to the destination.

t_j : link travel time (from bus-schedules database).

k : bus-stop destination.

FIGURE 3-5 illustrates the link travel time (t_j) , and arrival time (A_T)

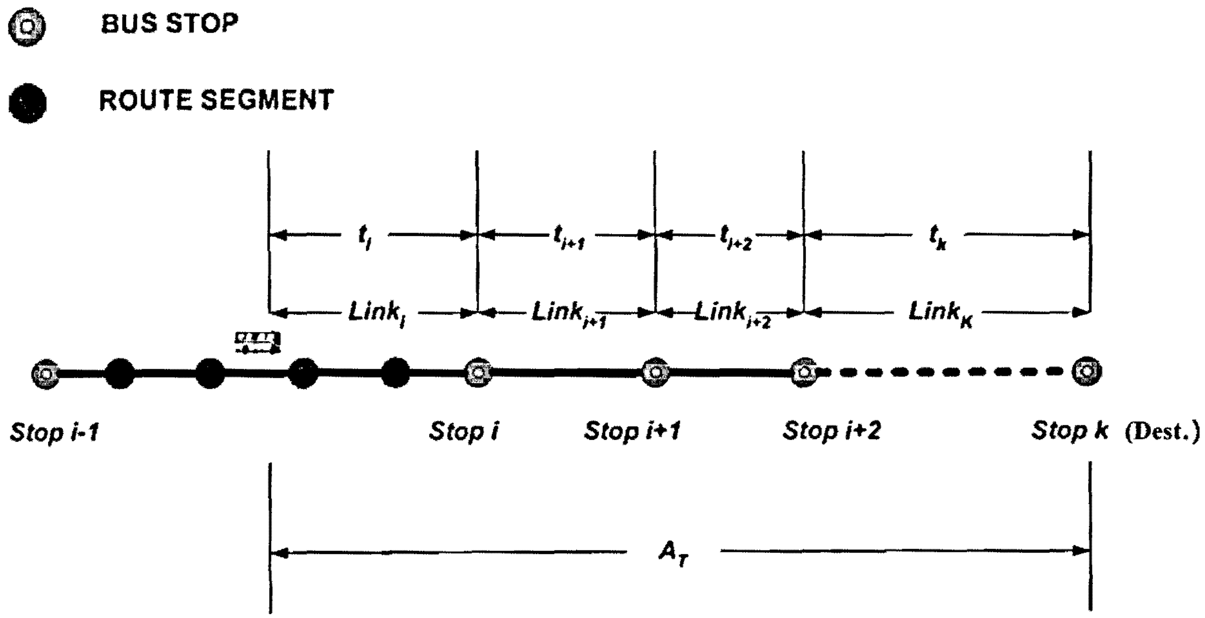


FIGURE 3-5 Arrival time, link travel time

3.3 ARCHITECTURE

The system was designed on MVC architecture. This is n-tier architecture comprises of Presentation, Business and Data layers. Presentation layer is an interface to the user/customer and provides all GUIs and user control validations. Business layer has all business rules objects. Data layer is an interface to the database. Business layer validates/apply business rules and pass the request to data layer which in turns communicates to the database to send/receive information. System architecture is shown in FIGURE 3-6:

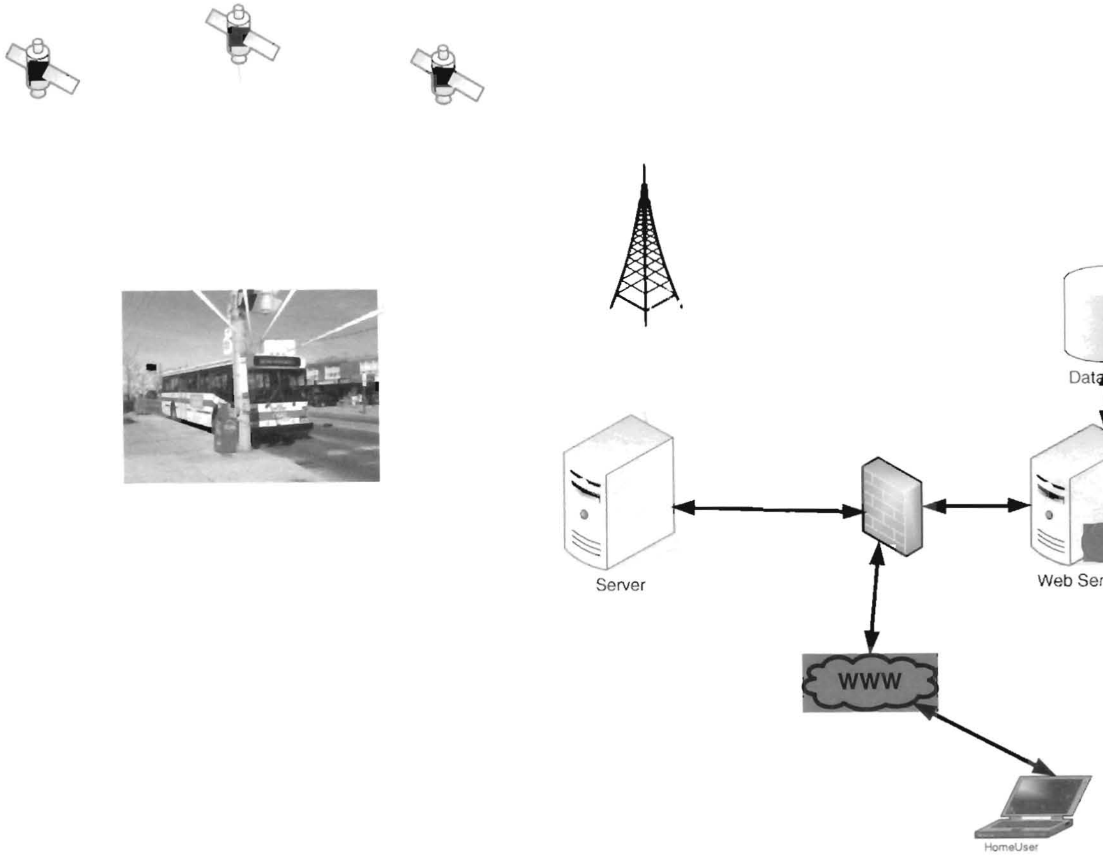


FIGURE 3-6 System Architecture

3.3.1 Modules/Development Technologies

TABLE 3-4 shows various modules of the standard ATIS system and the technologies used to develop these modules.

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

TABLE 3-4 ATIS Models

Module	Role	Technology
Client	A GPS enabled device capable of wireless connectivity resides in the courier's truck	Microsoft .Net (C#, WinForms), TCP/IP Sockets, GPS communication
Server	Accepts connection requests from the clients, start threads. Communicate to the web service to send/receive data to/from the database	Microsoft .Net (C#), TCP/IP Sockets, Multithreading
Web Service	Data layer which communicates to the database via ADO.Net, calls stored procedure from the database to send/receive data	Microsoft .Net (C#, ADO.Net)
Website	A centralized application that administrator, dispatcher and users/customers need to use to perform tasks and view real-time activities	Microsoft .Net (ASP.Net, C#), Google Map, Google Programming APIs, JavaScript
Database	Centralized data storage media use to store data for the whole system	Microsoft SQL Server 2005 (T-SQL, Stored Procedures)

3.4 COMPONENTS

3.4.1 Server

Server is responsible to control all clients, handshake, and broadcast and send/receive messages (tokens) to/from the clients. Server communicates to client and the database as shown in FIGURE 3-7.

1. Client
   Communication happens through TCP/IP protocol. All clients connect to the server through the same port (8001) and IP (IP address of the server, local-host in current scenario).

2. Database
   Communication happens through a web service. The web service is data layer and communicates to the database directly. All read/write functionalities to/from the database are performed through the web service. This web service is behind the DMZ (demilitarized

zone) and is necessary to secure the system so that database server is not exposed to the outer world.

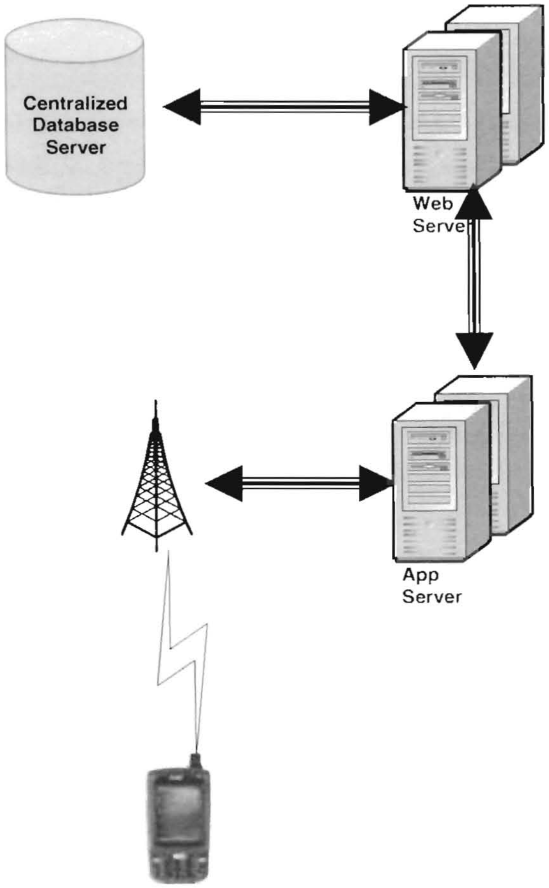


FIGURE 3-7 Client / Server communication via wireless

Server continuously polls the port 8001 for any new connection. Whenever a client connects to the server through this port, server calls web service to write this information into the database table(s) for logging information.

3.4.1.1 TCP Connection Establishment Process

Before client sends GPS data to the server, a TCP connection must be set up between them. This process, usually called connection establishment, involves an exchange of messages that transitions both devices from their initial connection state (CLOSED) to the normal operating state (ESTABLISHED).

3.4.1.2 Connection Establishment Functions

The connection establishment process actually accomplishes several things as it creates a connection suitable for data exchange:

- Contact and Communication: The client and server make contact with each other and establish communication by sending each other messages. The server usually doesn't even know what client it will be talking to before this point, so it discovers this during connection establishment.
- Vehicle Number: Each client sends vehicle number as user ID to the server when it wishes to make a connection for data transmission.
- Parameter Exchange: Certain parameters that control the operation of the TCP.

3.4.1.3 Control Messages

TCP uses control messages to manage the process of contact and communication. A set of control flags in the TCP header indicates whether a segment is being used for control purposes or just to carry data. The first byte was introduced as data type which informs the client /server of the type of the data.

The data transmission is controlled by the following values of the first byte:

- 0: It indicates that the client is requesting for login.
- 1: It indicates that the client is requesting to logout.
- 2: It indicates that the client is sending data.
- 3: It indicates that the system is sending a MESSAGE.

Second byte indicates the number of records contains in the packet.

3.4.1.4 Start-up Flow

The start-up process of the server is show in FIGURE 3-8.

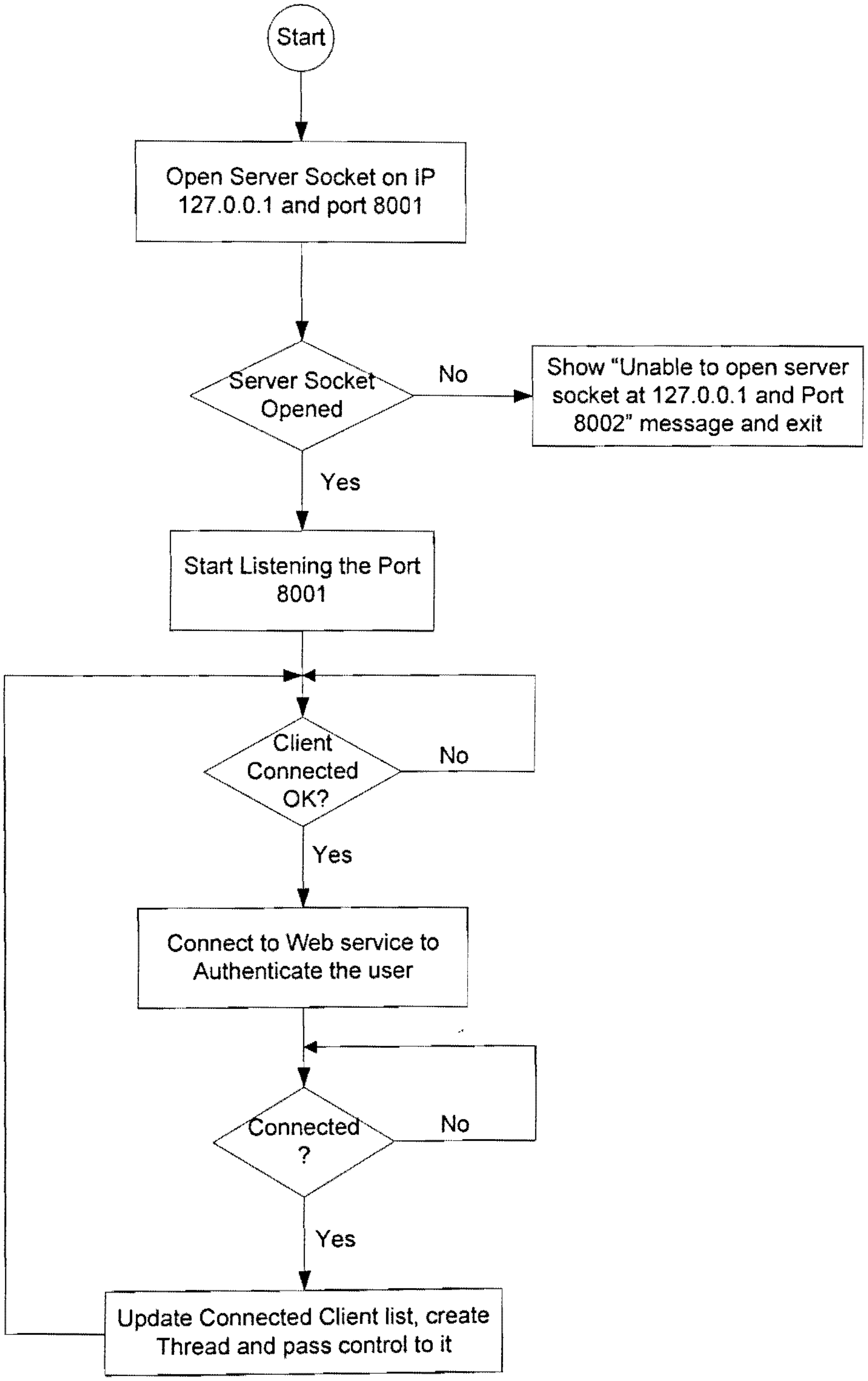

Start
Open Server Socket on IP 127.0.0.1 and port 8001
Server Socket Opened
No
Show "Unable to open server socket at 127.0.0.1 and Port 8002" message and exit
Yes
Start Listening the Port 8001
Client Connected OK?
No
Yes
Connect to Web service to Authenticate the user
Connected ?
No
Yes
Update Connected Client list, create Thread and pass control to it

FIGURE 3-8 Server Start-up Flow

3.4.2 Client

This application resides in the hardware device (mobile device like scanner). It connects to the server on specific port using TCP/IP based .Net application. This hardware unit is GPS enabled so it can either communicate to the 4 satellites or to the cell tower to get geo-location, speed and direction of the vehicle and sends this information to the server. Client must have to sign on before using the system. User's credentials are authenticated by the server. This application is responsible for sending geo-location whenever requested by the server. FIGURE 3-9 below gives high level of wireless communication between clients and server.

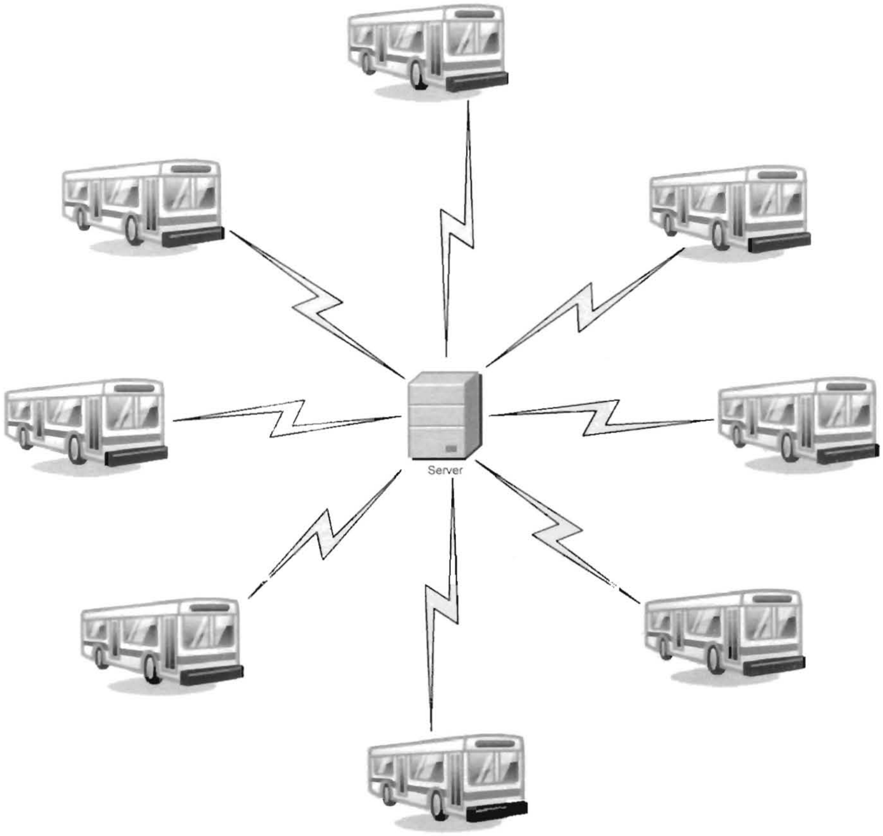


FIGURE 3-9 Wireless communication

3.4.2.1 Mobile Device

This device is designed and manufactured by Motorola (formerly Symbol) as shown in FIGURE 3-10. It has GPS and wireless capabilities. It communicates to the GPS via COM port and can connect to the wireless world using GSM/GPRS network. The client side application resides in this device, continuously reads COM port and receives geo-information from the GPS and send to the server in the predefined data format

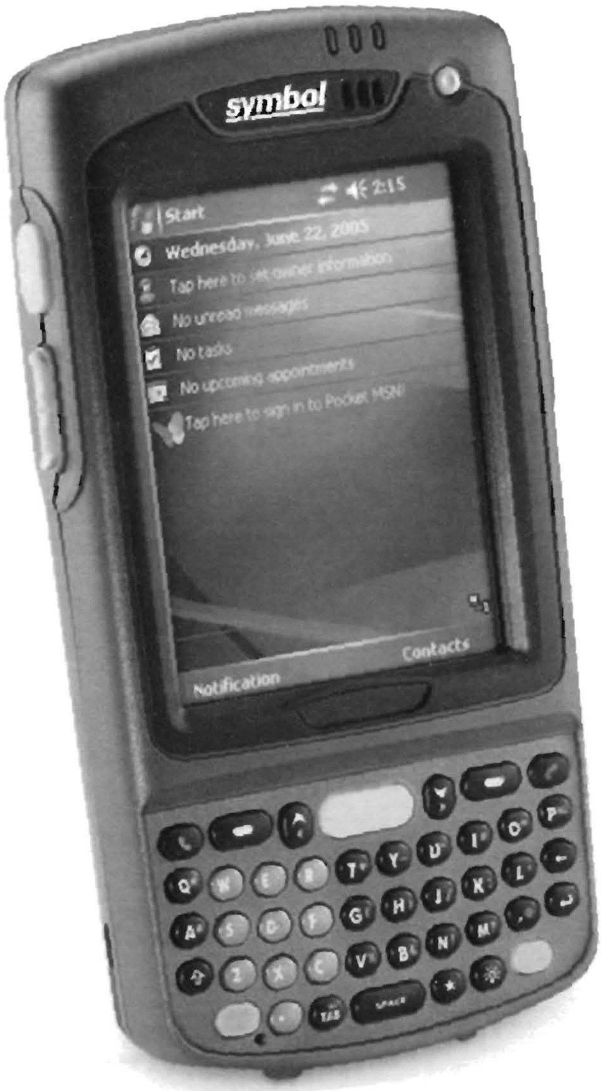


FIGURE 3-10 Mobile Device used by a BUS

3.4.2.2 Login Flow

User authentication process is shown in FIGURE 3-11.

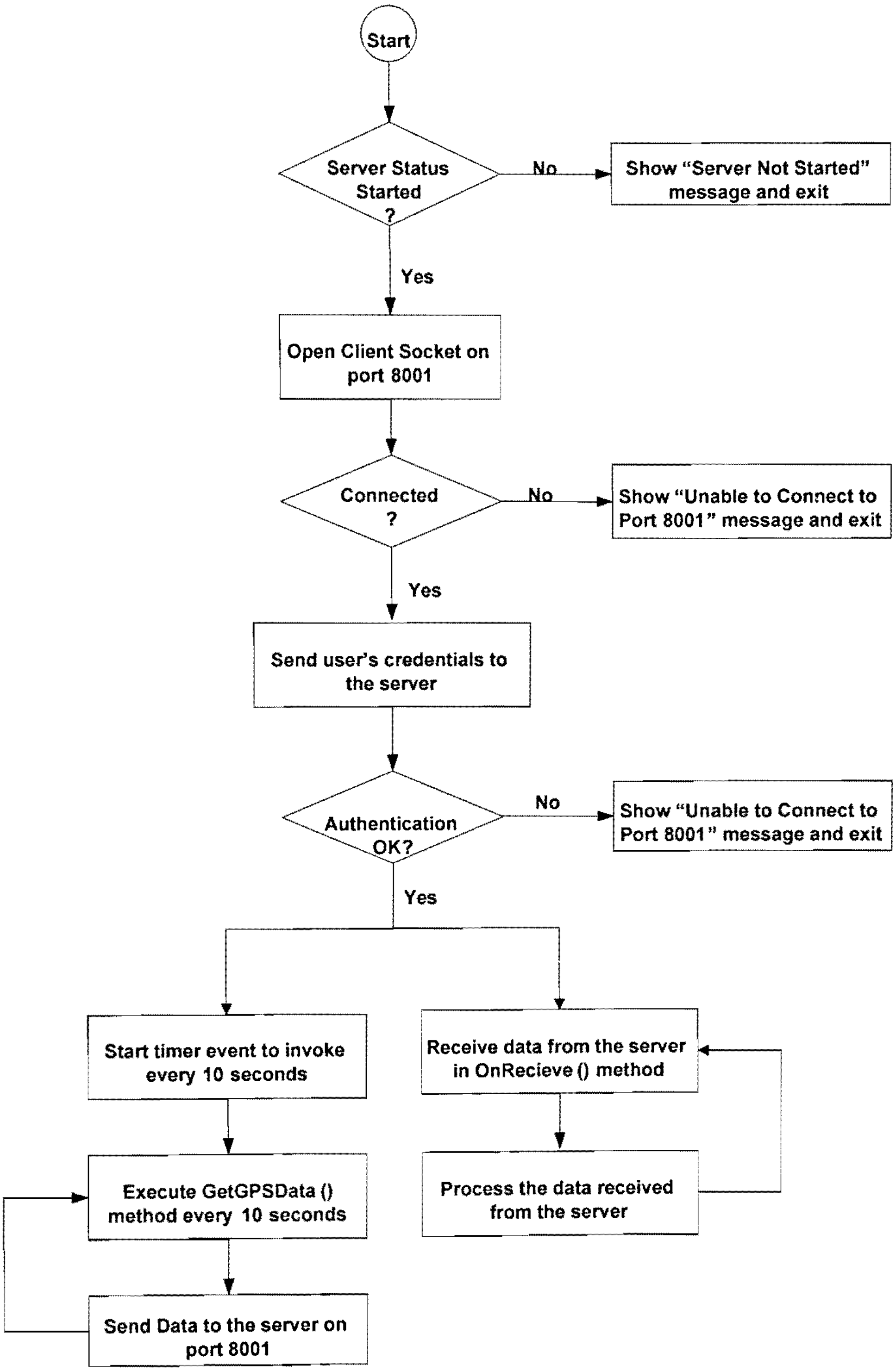

Start
Server Status Started ?
No
Show “Server Not Started” message and exit
Yes
Open Client Socket on port 8001
Connected ?
No
Show “Unable to Connect to Port 8001” message and exit
Yes
Send user’s credentials to the server
Authentication OK?
No
Show “Unable to Connect to Port 8001” message and exit
Yes
Start timer event to invoke every 10 seconds
Execute GetGPSData () method every 10 seconds
Send Data to the server on port 8001
Receive data from the server in OnRecieve () method
Process the data received from the server

FIGURE 3-11 Login Flow Diagram

3.4.2.3 Data Protocol

TABLE 3-5 Client / Server data protocol

Character Position	Data Type	Description	Possible Values
1	Integer	It indicates data type	0 = LOGIN, 1 = LOGOUT, 2 = DATA, 3 = MESSAGE
2	Integer	It indicates the total number of records contained in the packet	1 -- 5
3	String	Data separated by comma	

TABLE 3-6 Client / Server data protocol examples

Type	Example
LOGIN	0,1,AXPX643;*****
LOGOUT	1,1 AXPX643
DATA	2,2,56.33;34.22;60;N,57.29;34.11;60;N
MESSAGE	3,2, HI, CAN YOU PLEASE SEND SOME WITH A SCANNER AS I AM HAVING PROBLEM WITH THIS DEVICE. I AM AT PAPE/DANFORTH INTERSECTION

3.4.3 Web Service

The most sensitive part of the whole system that makes sure the communication happens through secure channel. This service is installed on the web service and is behind the firewall to set up security measures. It receives requests from the server, passes to database through stored procedures, retrieves the result and sends back to the server. It also receives requests from the dispatch administration application and sends updated status back. The dispatch application in turn updates various status windows.

According to the World Wide Web Consortium (W3C), a Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface that is described in a machine-processable format such as Web Service Description Language (WSDL). Other systems interact with the Web service in a manner prescribed by its interface using messages, which may be enclosed in a Simple Object Access Protocol (SOAP) envelope, or follow a Representational State Transfer (REST) approach. These messages are typically conveyed using Hyper Text Transfer Protocol (HTTP), and are normally comprised of

Extensible Mark-up Language (XML) in conjunction with other Web-related standards. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and Universal Description, Discovery and Integration (UDDI) open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a Graphical User Interface (GUI). Web services instead share business logic, data and processes through a programmatic interface across a network. Developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users.

Web services allow different applications from different sources to communicate with each other without time-consuming custom coding, and because all communication is in XML, Web services are not tied to any one operating system or programming language. For example, Java can talk with Perl; Windows applications can talk with UNIX applications. FIGURE 3-12 shows how the data exchange happens among various tiers and web service.

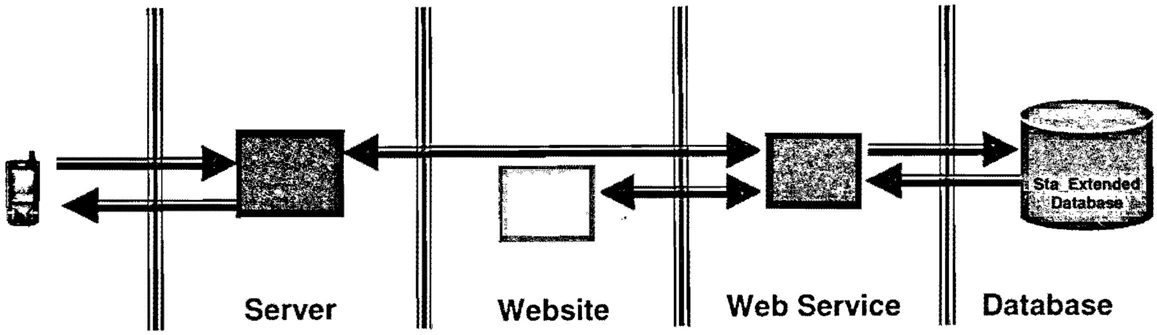


FIGURE 3-12 Data Exchange between tiers and database

3.4.3.1 Process Flow

Communication between web service and database is shown in FIGURE 3-13.

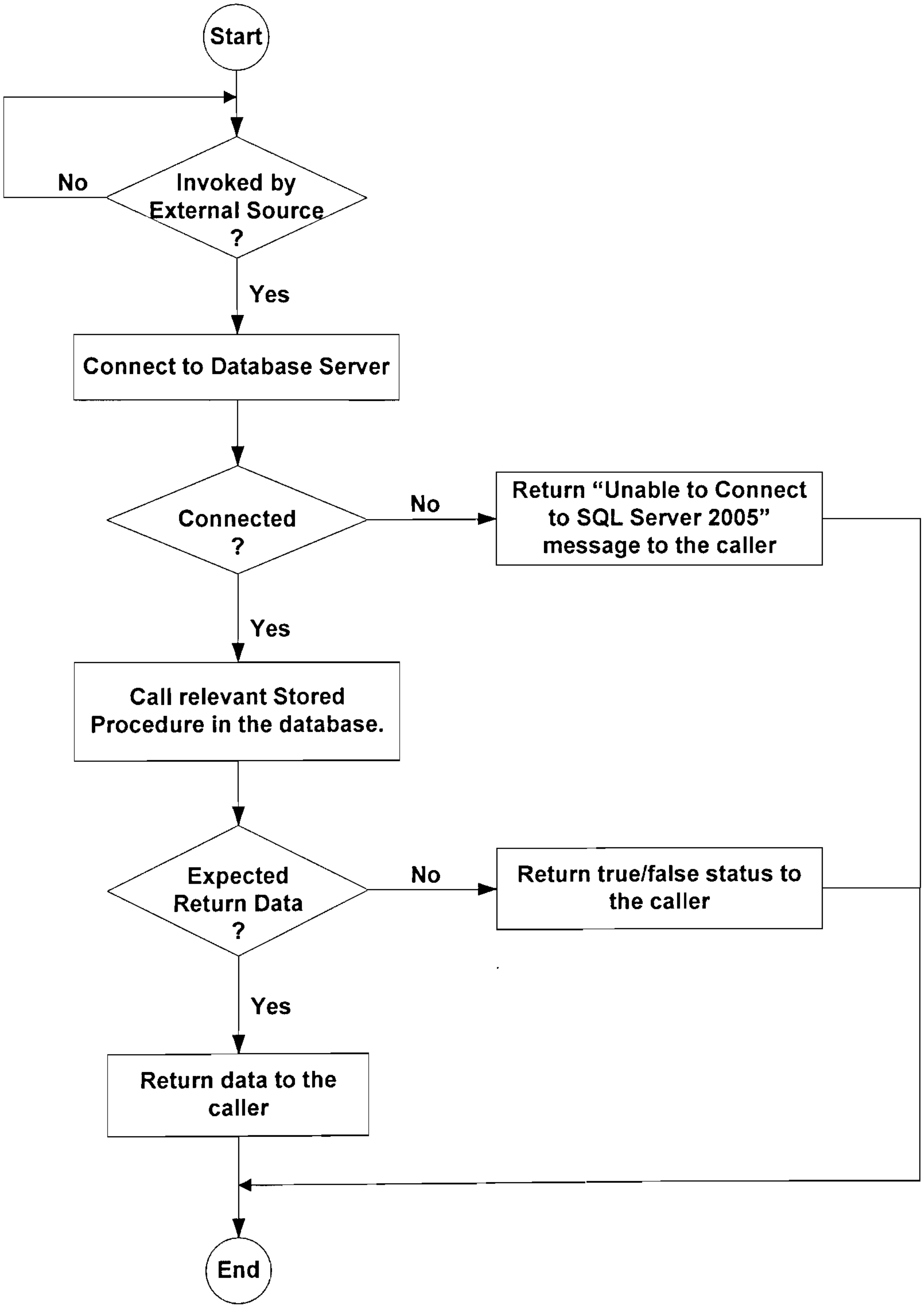

Start
No
Invoked by External Source ?
Yes
Connect to Database Server
Connected ?
No
Return "Unable to Connect to SQL Server 2005" message to the caller
Yes
Call relevant Stored Procedure in the database.
Expected Return Data ?
No
Return true/false status to the caller
Yes
Return data to the caller
End

FIGURE 3-13 Process Flow Diagram

3.4.4 Web Site

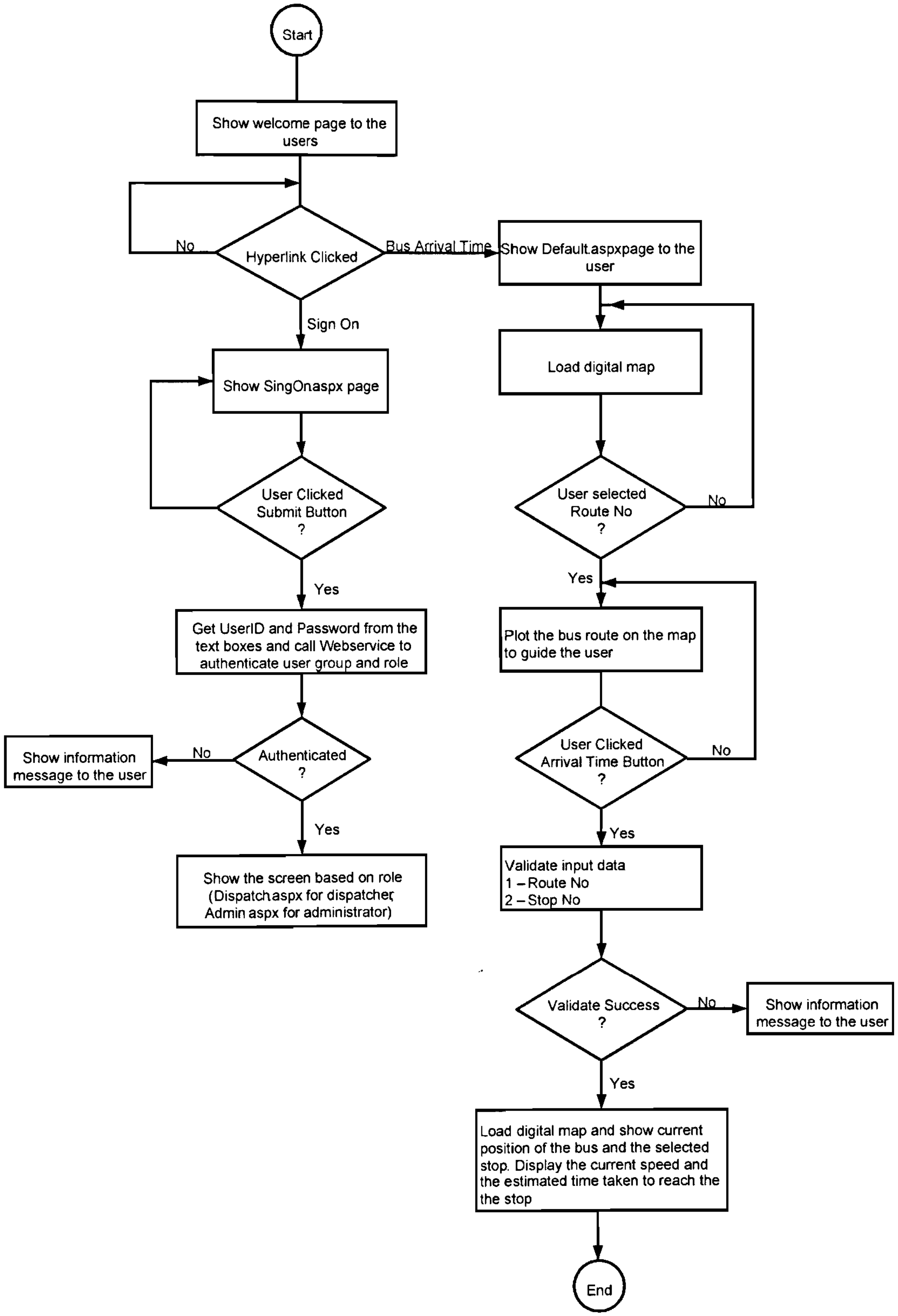
The main function of this application is to provide a central mechanism for the users/passengers, administrators and dispatchers. This is a real time monitoring as well as administration tool which is used by the administrators to administrate the system. It continuously communicates to the web service for any new or updated status. Administrator can add new user if new employee joins the company, modify an existing user if there is change in address, phone etc. and can delete an existing user if the user left the company. Administrator can view all user currently logged-in the system. He can view a user (bus) on digital map in real time and can send or receive text messages.

This application uses personalization to give access to the users. Each user belongs to a group and each group is assigned a role (user, administrator, and dispatcher). It authenticates the user credentials and displays the web pages according to the user group and role assigned.

The passenger is only interested to know the bus arrival time so user credentials are not required to login to the system. Passengers can just browse the website, enters route /stop Id to get the next bus arrival time. The system provides current bus location, speed, direction and the estimated time to reach the stop. It uses Google map to visually display the bus and stop locations.

3.4.4.1 User Role Authentication

User group and role authentication is shown in FIGURE 3-14.



Start
Show welcome page to the users
No
Hyperlink Clicked
Bus Arrival Time
Show Default.aspxpage to the user
Sign On
Load digital map
Show SingOnaspx page
User Clicked Submit Button ?
User selected Route No ?
No
Yes
Yes
Get UserID and Password from the text boxes and call Webservice to authenticate user group and role
Plot the bus route on the map to guide the user
Show information message to the user
No
Authenticated ?
User Clicked Arrival Time Button ?
No
Yes
Yes
Show the screen based on role (Dispatch.aspx for dispatcher, Admin.aspx for administrator)
Validate input data
1 – Route No
2 – Stop No
Validate Success ?
No
Show information message to the user
Yes
Load digital map and show current position of the bus and the selected stop. Display the current speed and the estimated time taken to reach the the stop
End

FIGURE 3-14 User Authentication

3.4.5 Database

The most popular database servers are Oracle and SQL Server 2005 with normalized database. Currently, most of the applications are being developed using Microsoft technologies so SQL server is very popular database server. The web service communicates to SQL server through stored procedure to avoid any SQL injection. All jobs, vehicle locations, client messages and clients logged-in status are continuously updated in the database. Database is the most important entity of the AVL system. It stores all the information coming from the vehicles. The dispatch application uses this data and displays the vehicle location in real time mode. The archive data helps in generating various history reports. TABLE 3-7 shows a list of database tables created for data storage and retrieval.

TABLE 3-7 : Database Objects (Tables)

Table Name	Columns	Brief Description
dbo.AlertHistory	[group_id] [create_date] [DATETIME] [user_id] [alert_id] [update_date]	The table is used to store history of the alerts generated by the system. These alerts are mostly variances which can happen
dbo.Alerts	[alert_id] [alert_name]	The table is used to store summary of the alerts generated by the system.
schedule		
dbo.BusSchedules	[ScheduleType] [PeriodType] [StopId] [StopName] [ArrivalTime] [DATETIME] [StartingTime]	The table is used to store the bus schedules for each stop of the route
dbo.ClientDetails	[ConfirmationNo [FirstName] [LastName] [Address] [City] [State] [Zip] [HomePhone] [CellPhone]	The table is used to store detailed information about the customer upon creating a pickup request
dbo.DeviceInfo	[DeviceId] [DeviceName]	Static table that shows various type of supported devices (GPS, Scanner, PC)
dbo.Groups	[GroupId] [RoleId] [Details]	The table is used to store various groups that a newly created user can belong to. User is belong to a group and group is assigned role
dbo.Jobs	[JobId] [WirelessStatusId] [UserId]	The table is used to store job information. A newly created pickup job goes under various steps before completion

	[DispatcherId] [JobStatusId] [JobDate] [CreateDate] [AcceptedBy] [Instructions] [ConfirmationNo] [ScheduleTime]	
dbo.JobStatus	[JobStatusId] [Details]	The table is used to store the current status of the job (Not Assigned, Assigned, Sending, Pending, Sent, Viewed, Rejected, Completed)
dbo.Messages	[FromUser] [ToUser] [MessageRead] [CreateDate] [ReadDate] [ProcessedDate] [MessageDetails [MessageType]	The table is used to store the messages send to/from the users (couriers or bus drivers)
dbo.Roles	[RoleId] [Details]	Group table is linked with this table as Role is assigned to the group rather than user itself
dbo.Routes	[RouteId]	The table is used to store the route information
dbo.Schedules	[RouteId] [StopId] [ScheduleType] [SchedulySubType] [ArrivalTime] [StartTime]	The table is used to store the schedules at various time and various days. This is master level table
dbo.ScheduleSubType	[ScheduleSubType] [Description]	This is static and master table to Schedule and it stores ScheduleSubType information
dbo.ScheduleTypes	[ScheduleType [Description]	This is static and master table to Schedule and it stores ScheduleType information
dbo.SendEmail	[group_id] [send_email_id [email_address]	The table is used to store the emails sending to the administrators, dispatchers (for future use)
dbo.Stops	[RouteId] [StopId] [Details] [Latitude] [Longitude] [Distance]	The table is used to store to stops information corresponding to the route
dbo.StopsSegments	[RouteId] [StopId] [SegmentId] [Location] [Latitude] [Longitude] [Distance]	The table is used to store the segments information. Each stop is divided into various segments to keep track of the buses and estimating the correct arrival time
dbo.Users	[UserId] [Password] [Home_Phone] [MobilePhone] [Address1] [Address2]	The table is used to store the users profiles.

	[City] [State] [ZipCode] [GroupId]	
dbo.UsersOnline	[UserId]	The table is used to store the online users that are currently connected to the system via wireless
dbo.VehicleData	[VehicleId] [Longitude] [Latitude] [Speed] [Direction] [CurrentAddress] [CreateTime] [DeviceId] [RouteId] [CurrentStopId] [CurrentSegmentId] [TripStatus]	The table is used to store the GPS data coming for the Client application.
dbo.WirelessStatusType	[WirelessStatusId] [Description]	The table is used to store wireless status type (Not Assigned, Assigned, Wireless Sent, Completed)

TABLE 3-8 shows a list of stored procedures. These procedures are invoked by the web service to communicate to the database.

TABLE 3-8: Database Objects (Stored Procedures)

Id	Stored Procedure Name	Brief Description
1	dbo.[ATIS_AddUser]	Called from data layer to add a new user
2	dbo.[ATIS_CalculateArrivalInformation]	Called from data layer to calculate the bus arrival time to the next stop
3	dbo.[ATIS_CreatePickup]	Called from data layer to create a pickup as requested by the customer. Administrator or dispatcher can perform this activity
4	dbo.[ATIS_DeleteUserInfo]	Called from data layer to delete an existing user from the system. Administrator or dispatcher can perform this activity
5	dbo.[ATIS_GetCoordinates]	Called from data layer to get geo-coordinate of the segments within each bus stop
6	dbo.[ATIS_GetPickupDetails]	Called from data layer to get pickup details (address, scheduled pickup time etc.).
7	dbo.[ATIS_GetPickups]	Called from data layer to get the pickup details. It can be in any status. Administrator or dispatcher can perform this activity
8	dbo.[ATIS_GetPickupStatus]	Called from data layer to get the wireless status Id of the pickup (Delivered to the courier or still within the system)

9	dbo.[ATIS_GetRole]	Called from data layer to get the role of the user. This is called from website.
10	dbo.[ATIS_GetRoutes]	Called from data layer to get all the routes within the system
11	dbo.[ATIS_GetStops]	Called from data layer to get all the stops within a route
12	dbo.[ATIS_InsertData]	Called from data layer to insert GPS data coming from the client
13	dbo.[ATIS_InsertMessageDetails]	Called from data layer to insert message details. Administrator or dispatcher can perform this activity
14	dbo.[ATIS_RetrieveAllUsers]	Called from data layer to get a list of users that exist in the system. Administrator or dispatcher can perform this activity
15	dbo.[ATIS_RetrieveData]	Future Use
16	dbo.[ATIS_RetrieveOnlineUsers]	Called from data layer to get a list of online users. Administrator or dispatcher can perform this activity
17	dbo.[ATIS_RetrieveUserInfo]	Called from data layer to get the details of the user. Administrator or dispatcher can perform this activity
18	dbo.[ATIS_RetrieveVehicleInfoByRouteAndStop]	Called from data layer to get Vehicle information based on route Id and stop Id
19	dbo.[ATIS_RetrieveVehicleInfoByUser]	Called from data layer to get Vehicle information based on user Id
20	dbo.[ATIS_SendMessage]	Called from data layer to insert message details for the server application to send to the online user
21	dbo.[ATIS_TraceClosestBus]	Called from data layer to trace the closest bus based on segment ID and stop ID
22	dbo.[ATIS_UpdateUserInfo]	Called from data layer to modify details of an existing user from the system. Administrator or dispatcher can perform this activity
23	dbo.[ATIS_ValidateUser]	Called from data layer to validate user’s credentials. It is called by the server upon receiving wireless connection from the user

CHAPTER IV

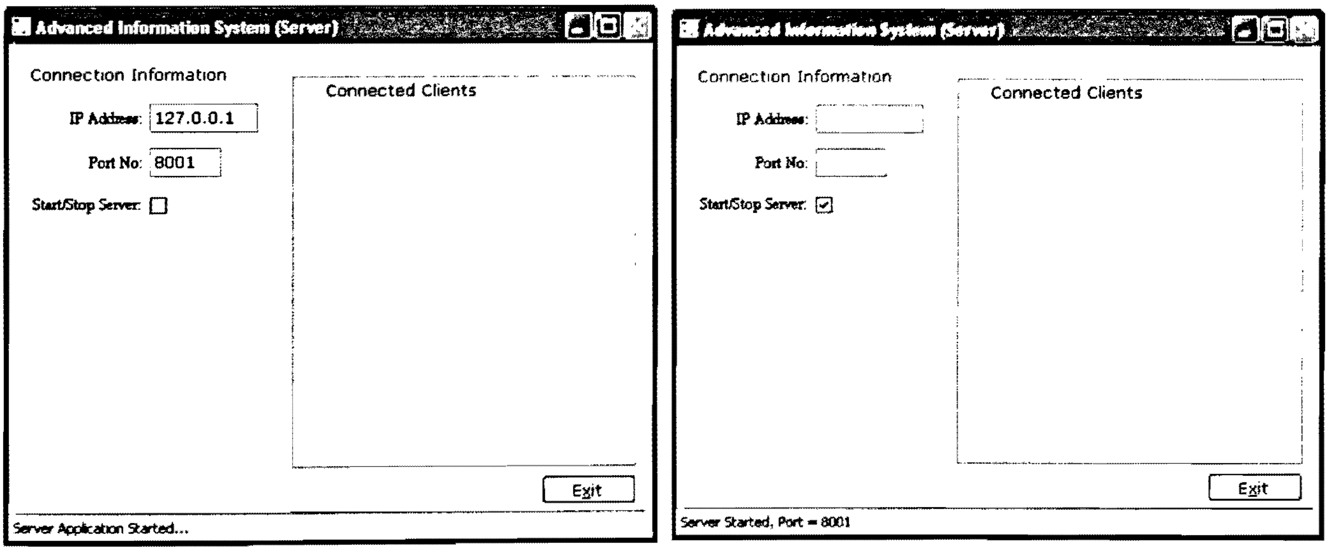
4 DEVELOPMENT / SIMULATION

4.1 CLIENT / SERVER

Client / Server are TCP/IP based applications with extensive use of multithreading. Server side application communicates to the clients on port 8001. It continuously monitors the input connection requests from the clients. As soon as a client connects, it starts a thread and passes control to it and entertains other request from new clients. Client side application is TCP/IP based application with two types of communication interfaces: GPS interface on COM port to read geo-location and TCP/IP interface to the server to send this data certain period of time.

C# (Microsoft .Net) was used to develop this client /server application. In this application, .Net Windows Forms were used for GUI, TCP/IP sockets for communication between client and server and multithreading to control and entertain multiple clients connection. In any client /server application, clients need to make connection to the server, so first step is to start the server side application.

FIGURE 4-1 refers to the server start-up process with local host and on the port 8001. Server is ready to entertain the connection from the port 8001 and accepts client requests

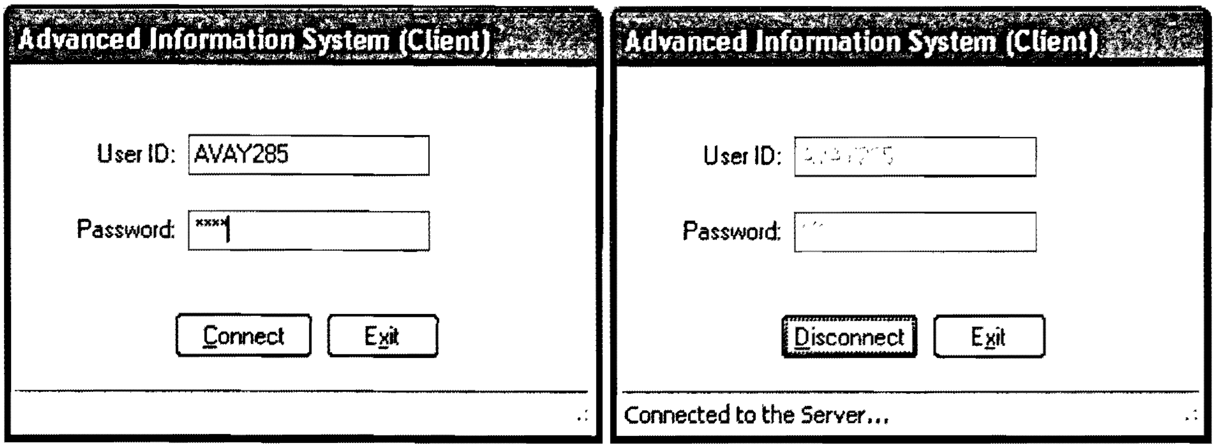


IP Address: 127.0.0.1
Port No: 8001
Start/Stop Server: □
Server Application Started...

IP Address:
Port No:
Start/Stop Server: ☑
Server Started, Port = 8001

FIGURE 4-1 Server Start-up

Client can be started to establish a TCP/IP connection with the server. User is asked for credentials and click on Connect button. The application connects to the server on the port 8001 and sends login request as shown in FIGURE 4-2.



Advanced Information System (Client)

User ID: AVAY285

Password: ****

Advanced Information System (Client)

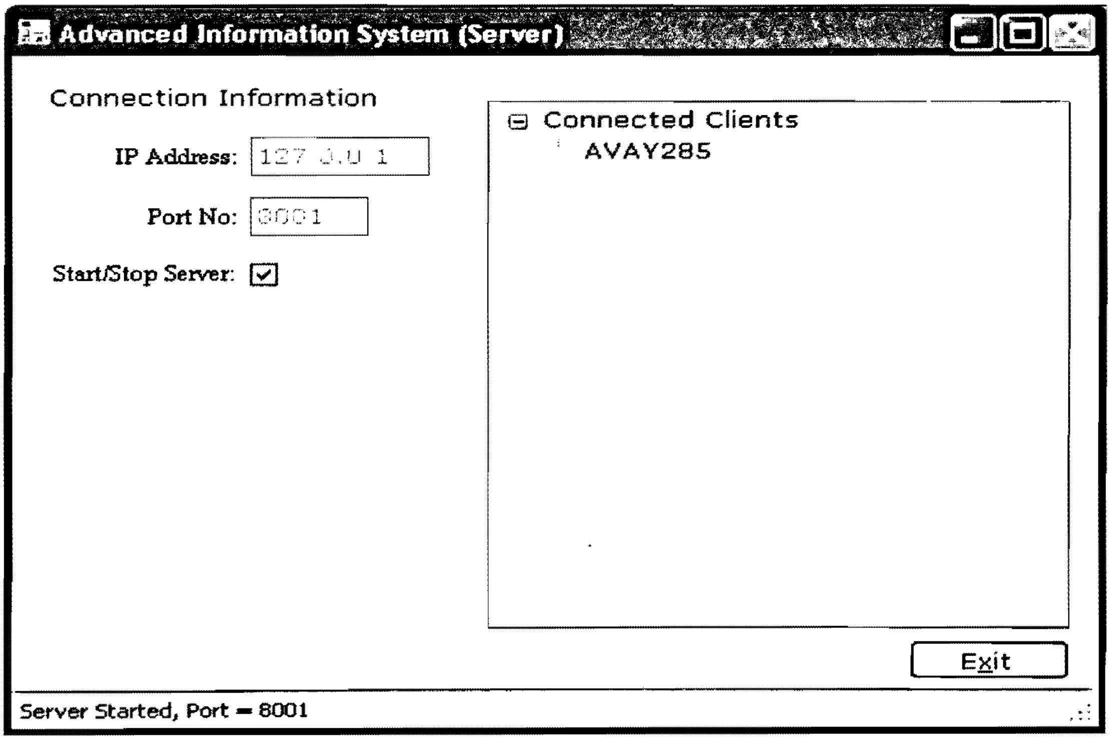
User ID: [illegible]

Password: [illegible]

Connected to the Server...

FIGURE 4-2 Client / Server Connection

Server authenticates the client credentials. It allows the connection only after verifying user id and password of the client. If everything goes well, it shows the client under the connected clients as shown in FIGURE 4-3.



Advanced Information System (Server)

Connection Information

IP Address: 127.0.0.1

Port No: 8001

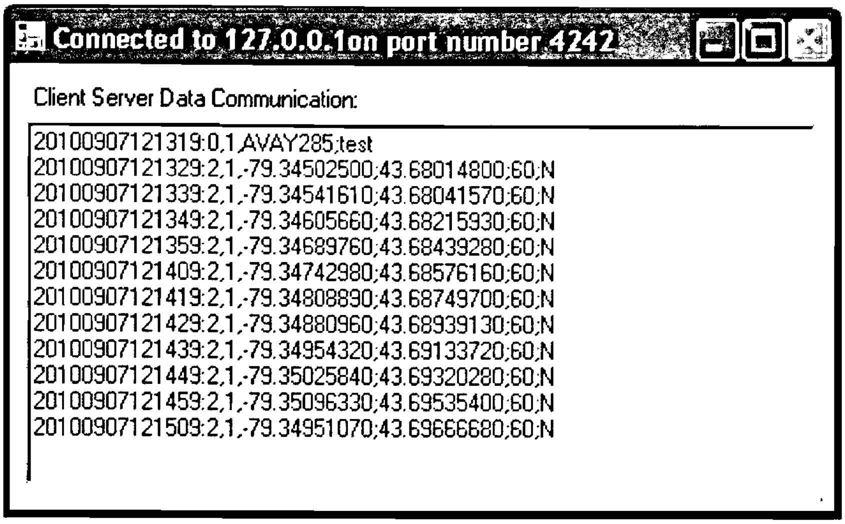
Start/Stop Server: ☑

Connected Clients
AVAY285

Server Started, Port = 8001

FIGURE 4-3 Server Real-time Status

Client is ready to communicate to GPS and collects geo-information. It sends this information to the server after fix interval of time as shown in FIGURE 4-4.



Client Server Data Communication:

```
201009071213l9:0,1,AVAY285;test
20100907121329:2,1,-79.34502500;43.68014800;60;N
20100907121339:2,1,-79.34541610;43.68041570;60;N
20100907121349:2,1,-79.34605660;43.68215930;60;N
20100907121359:2,1,-79.34689760;43.68439280;60;N
20100907121409:2,1,-79.34742980;43.68576160;60;N
20100907121419:2,1,-79.34808890;43.68749700;60;N
20100907121429:2,1,-79.34880960;43.68939130;60;N
20100907121439:2,1,-79.34954320;43.69133720;60;N
20100907121449:2,1,-79.35025840;43.69320280;60;N
20100907121459:2,1,-79.35096330;43.69535400;60;N
20100907121509:2,1,-79.34951070;43.69666680;60;N
```

FIGURE 4-4 Client / Server Data Communication

4.2 WEB SITE (USER + DISPATCH APPLICATION)

Website is Microsoft .Net based application that hosts on the web server. This is centralized application used by the customers, users, administrators and the dispatchers. Each user has own credentials and the application validates and authenticates the user credentials, group it belong to and the roles assigned to the group. The application offers a main welcome page to everyone and routes the traffic to the specific pages based on roles.

This is the most important and complicated part of the architecture as it uses state of the art technologies likes Google map and Google APIs. This n-tier application is based on the most popular MVC architecture and was developed by using various tools ASP.Net, C#, Java Script, Google map, Google APIs, web service and the database. The presentation layer that provides GUI to the user was developed in ASP.Net, Google map, Google APIs, JavaScript while business layer was developed in C# and the web service was used in data layer. It starts with the welcome page. User is only interested to check the arrival time for a bus while administrators and dispatchers can login with their credentials to perform tasks as shown in FIGURE 4-5.

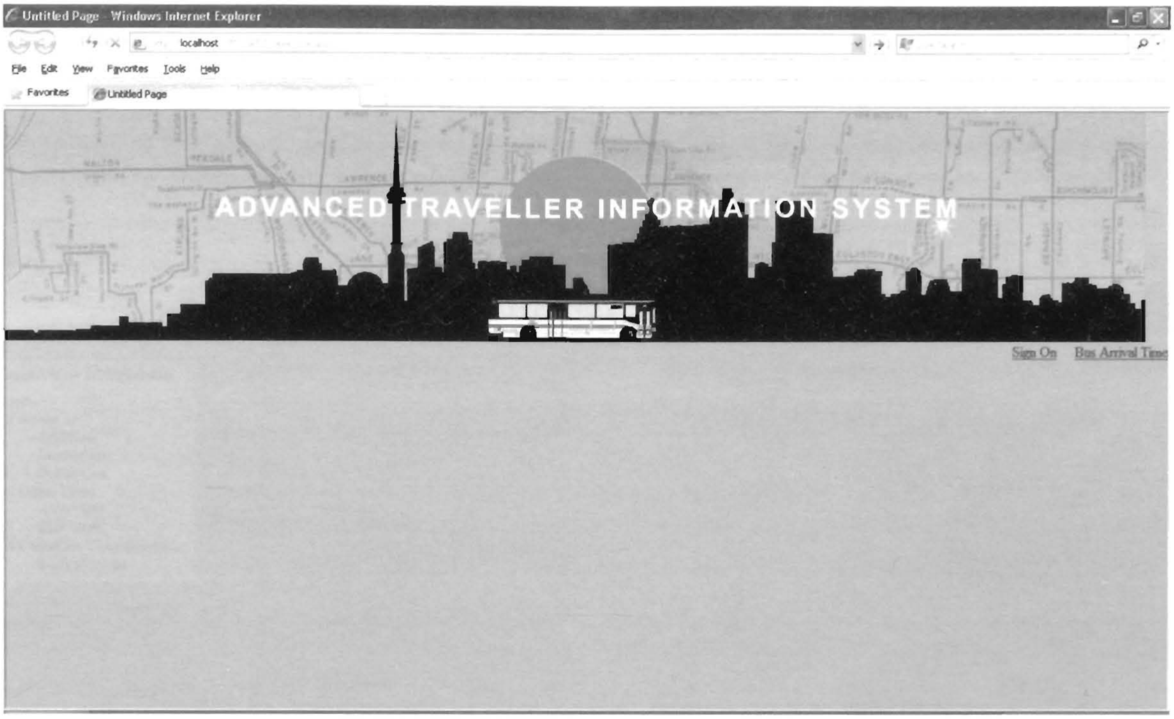
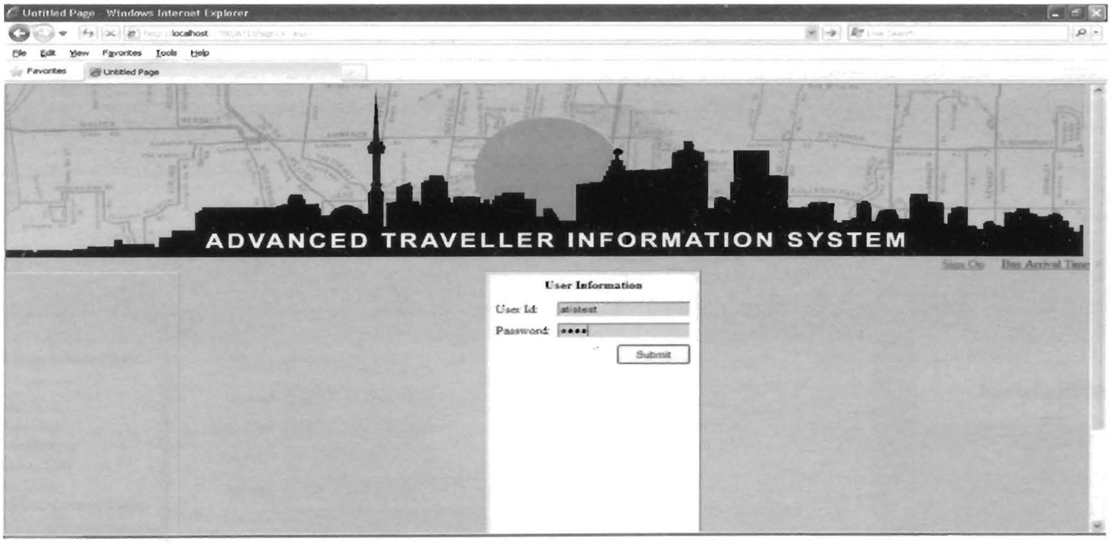

Untitled Page - Windows Internet Explorer
localhost
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Time

FIGURE 4-5 Welcome to ATIS

Administrators and dispatchers are asked to enter their credentials. Each user, dispatcher and administrator has own credentials as shown in FIGURE 4-6


Untitled Page - Windows Internet Explorer
localhost
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Time
Submit

User Information

User Id: atistest

Password: ••••

FIGURE 4-6 ATIS (sign on)

User credentials are verified against groups and roles. The system shows the options based on the group the user belongs to and the roles dedicated to the group. Administrators can perform various tasks like administrator, view online users, and communicate to the client by messaging, as shown in FIGURE 4-7.

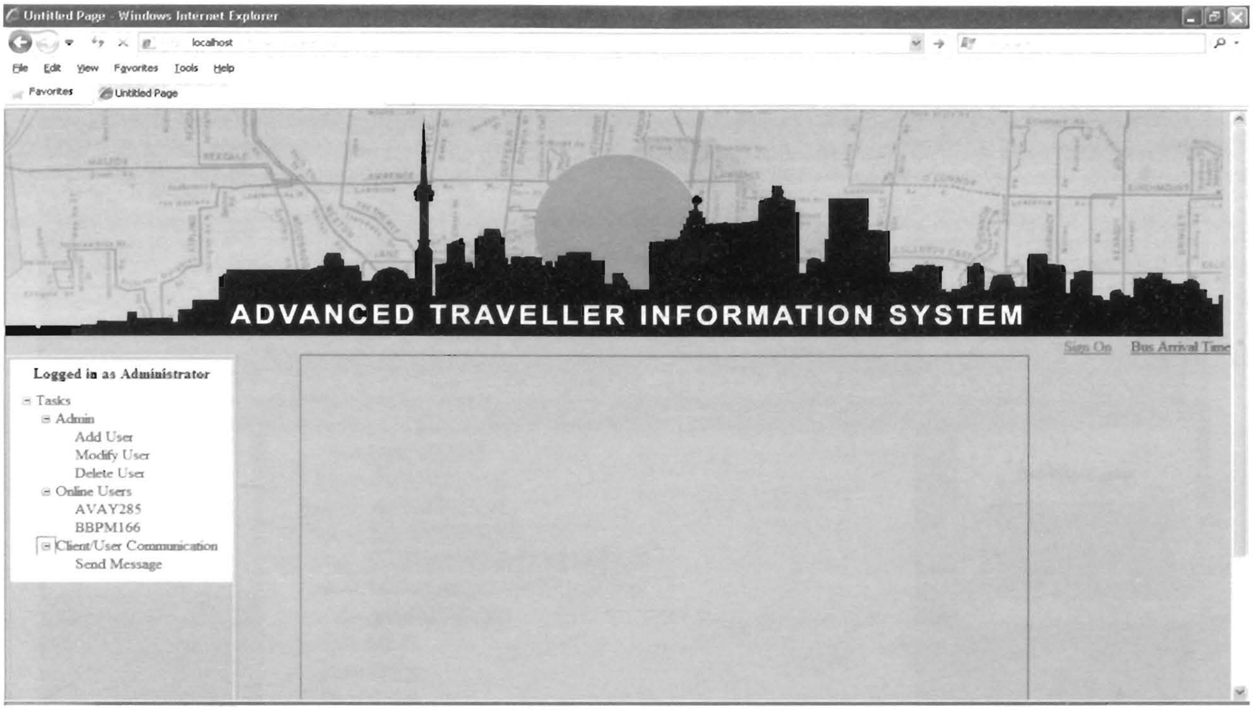

Untitled Page - Windows Internet Explorer
localhost
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Time
Logged in as Administrator
Tasks
Admin
Add User
Modify User
Delete User
Online Users
AVAY285
BBPM166
Client/User Communication
Send Message

FIGURE 4-7 ATIS (Administrator Login)

Administrator gives option to add a new user. This user can be a driver, dispatcher or any other administrator as shown in FIGURE 4-8.

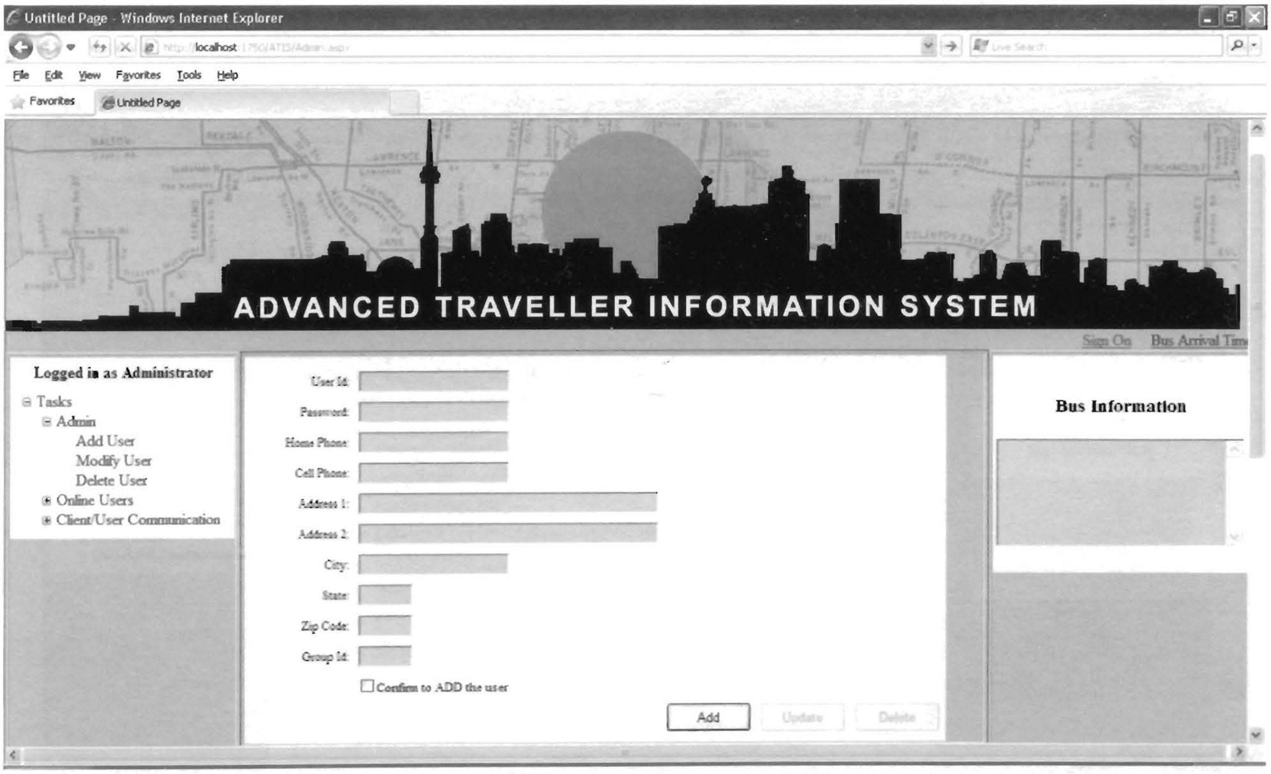
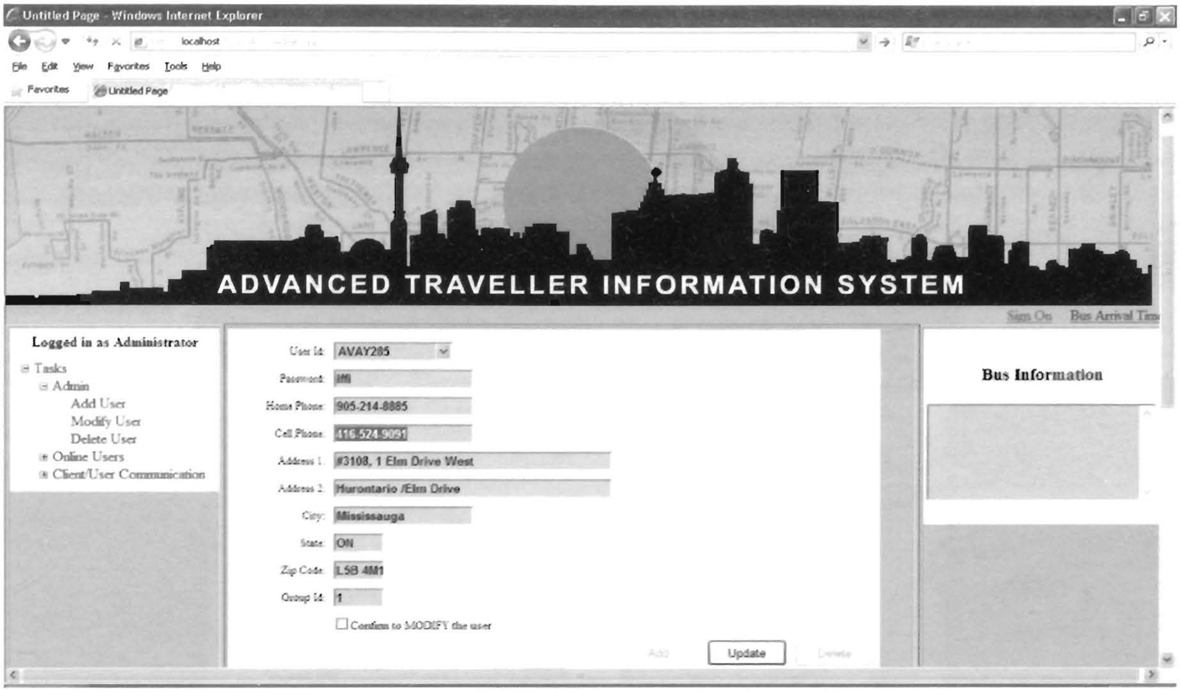

Untitled Page - Windows Internet Explorer
http://localhost:1750/ATIS/Admin.aspx
Live Search
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Time
Logged in as Administrator
Tasks
Admin
Add User
Modify User
Delete User
Online Users
Client/User Communication
User Id:
Password:
Home Phone:
Cell Phone:
Address 1:
Address 2:
City:
State:
Zip Code:
Group Id:
□ Confirm to ADD the user
Add
Update
Delete
Bus Information

FIGURE 4-8 ATIS (Add User)

Administrator allows modifying an existing user in case there is some change in address, phone, and password etc, as shown in FIGURE 4-9.


Untitled Page - Windows Internet Explorer
localhost
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Time
Logged in as Administrator
Tasks
Admin
Add User
Modify User
Delete User
Online Users
Client/User Communication
Bus Information
Add
Update
Delete

User Id: AVAY285

Password: iffi

Home Phone: 905-214-8885

Cell Phone: 416-524-9091

Address 1: #3108, 1 Elm Drive West

Address 2: Hurontario /Elm Drive

City: Mississauga

State: ON

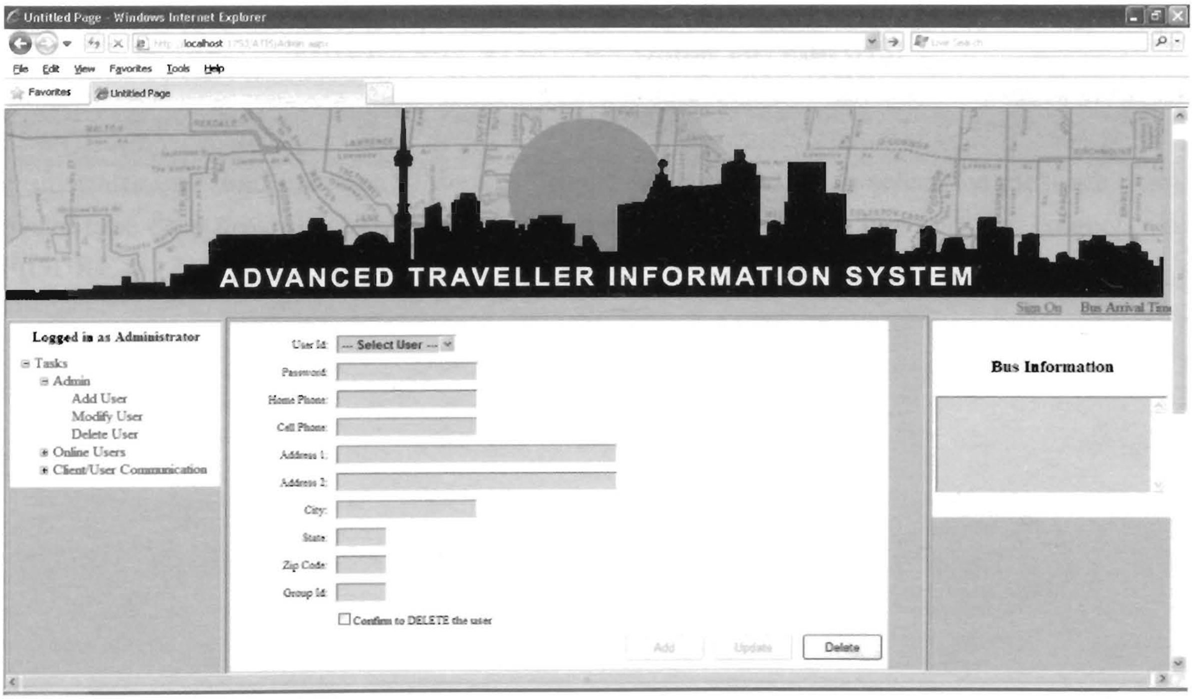
Zip Code: L5B 4M1

Group Id: 1

□ Confirm to MODIFY the user

FIGURE 4-9 ATIS Modify/Update user

Administrator allows deleting a user in case the user left the company or because of some other reasons as shown in FIGURE 4-10.


Untitled Page - Windows Internet Explorer
localhost
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Time
Logged in as Administrator
Tasks
Admin
Add User
Modify User
Delete User
Online Users
Client/User Communication
Bus Information
Add
Update
Delete

User Id: --- Select User ---

Password:

Home Phone:

Cell Phone:

Address 1:

Address 2:

City:

State:

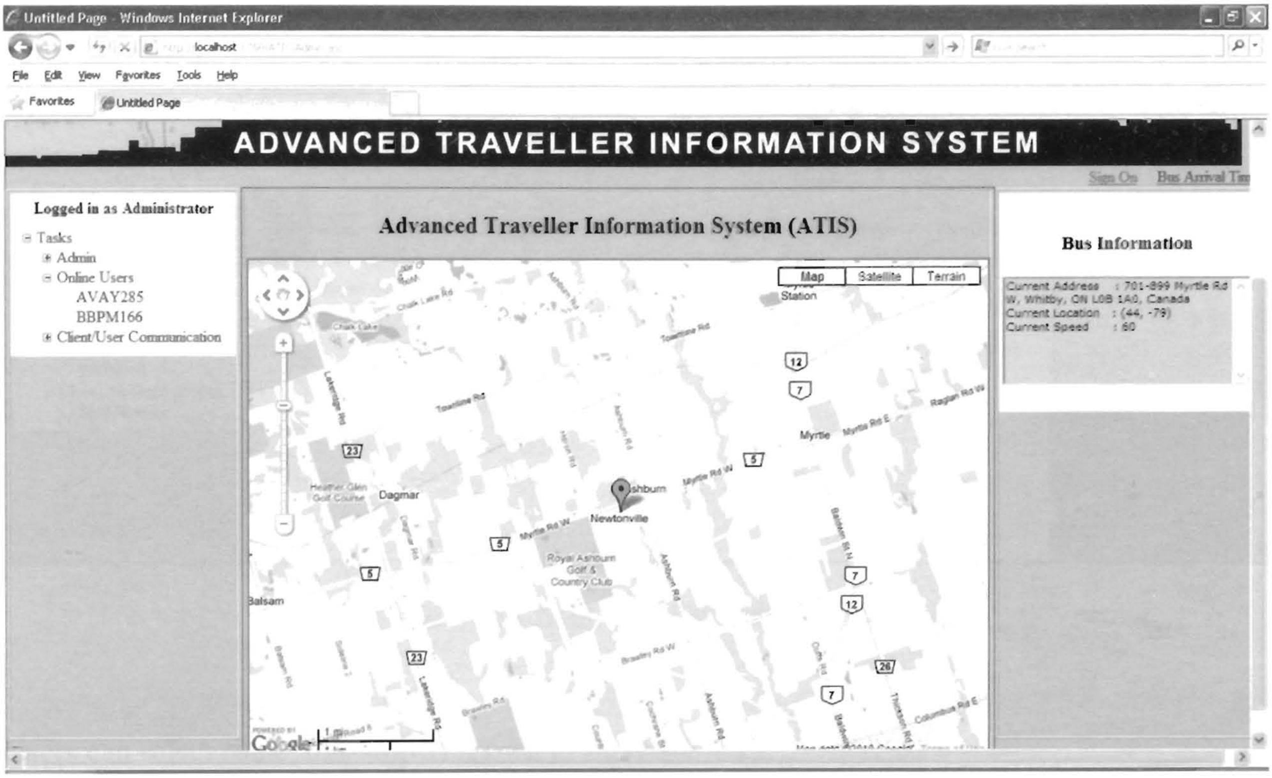
Zip Code:

Group Id:

□ Confirm to DELETE the user

FIGURE 4-10 ATIS (Delete User)

Administrator can view all clients currently logged in. The system allows the administrator to view the current address, geo-location and speed of the bus. The system uses GIS maps to display this data as shown in FIGURE 4-11.


Untitled Page - Windows Internet Explorer
localhost
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Tim
Map
Satellite
Terrain
Dagmar
Newtonville
Myrtle
Balsam
Google

Logged in as Administrator

- Tasks
  - Admin
  - Online Users
    - AVAY285
    - BBPM166
  - Client/User Communication

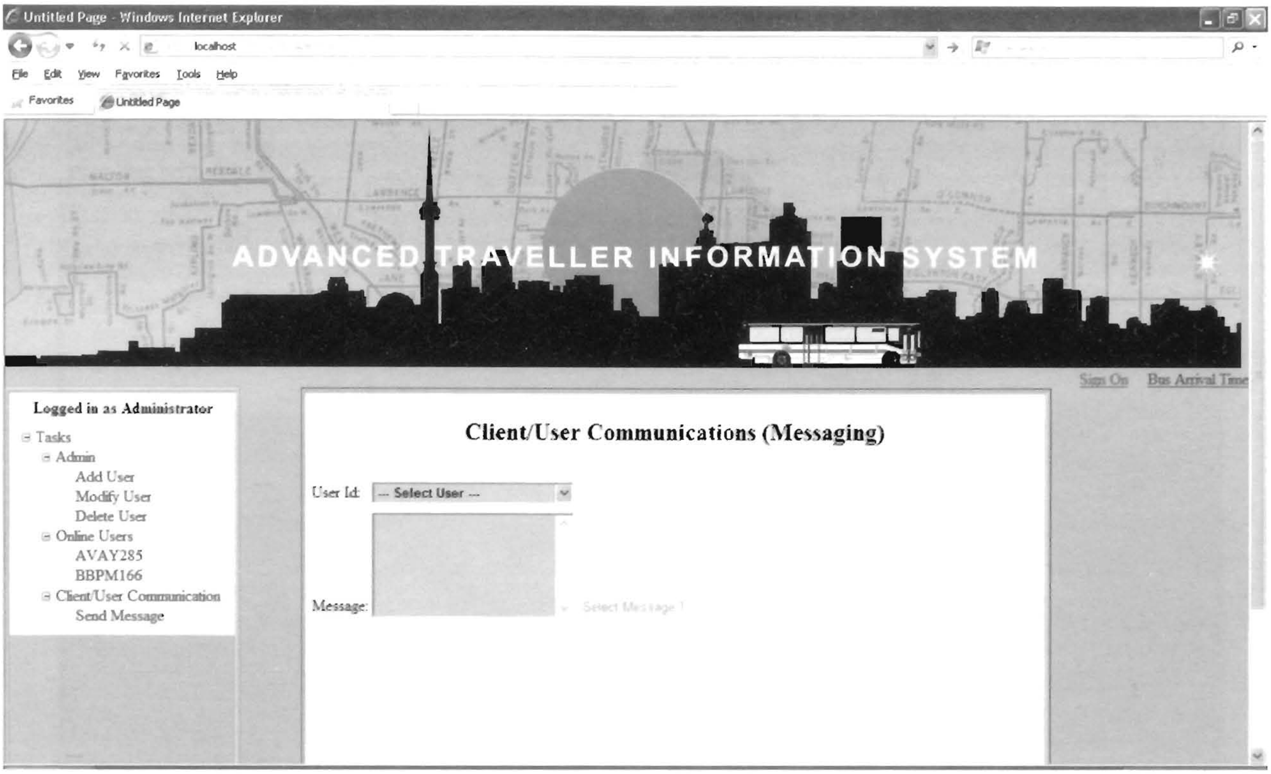
Advanced Traveller Information System (ATIS)

Bus Information

Current Address : 701-899 Myrtle Rd W, Whitby, ON L0B 1A0, Canada
Current Location : (44, -79)
Current Speed : 60

FIGURE 4-11 ATIS (View Online bus with GIS)

Administrator can send message to the bus driver. Administrator can select the message from the message type drop down list or can type in the message by selecting free form message as shown in FIGURE 4-12.


Untitled Page - Windows Internet Explorer
localhost
File Edit View Favorites Tools Help
Favorites
Untitled Page
ADVANCED TRAVELLER INFORMATION SYSTEM
Sign On
Bus Arrival Time

Logged in as Administrator

- Tasks
  - Admin
    - Add User
    - Modify User
    - Delete User
  - Online Users
    - AVAY285
    - BBPM166
  - Client/User Communication
    - Send Message

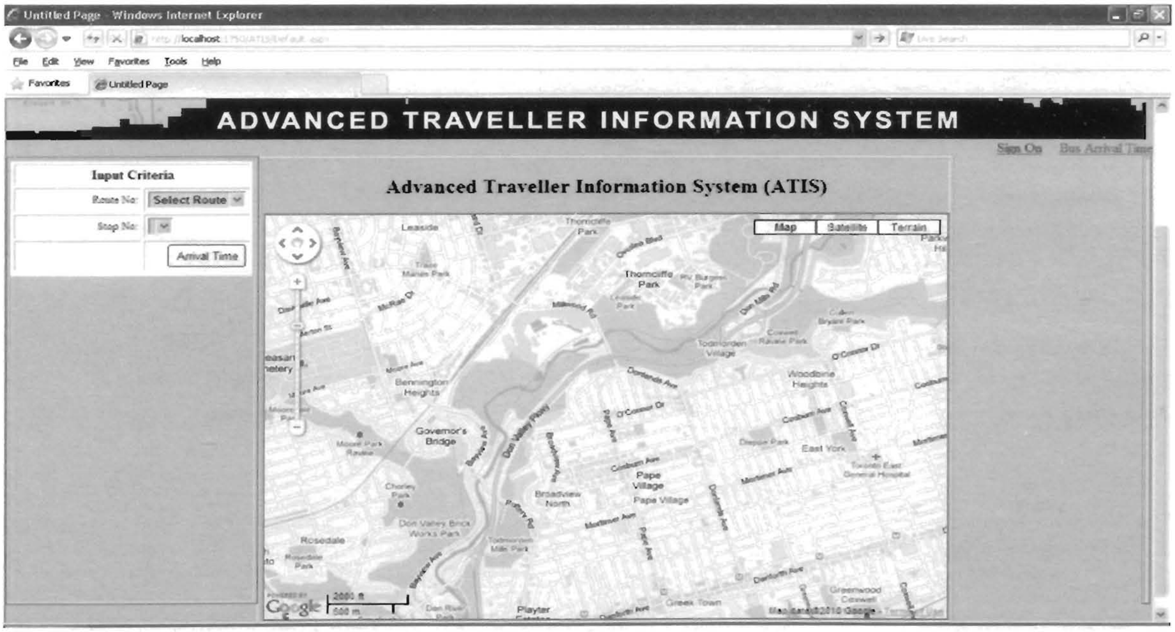
Client/User Communications (Messaging)

User Id: -- Select User --

Message:

FIGURE 4-12 Client/Administrator Messaging

Users can select route, bus-stop as shown in FIGURE 4-13.



Input Criteria

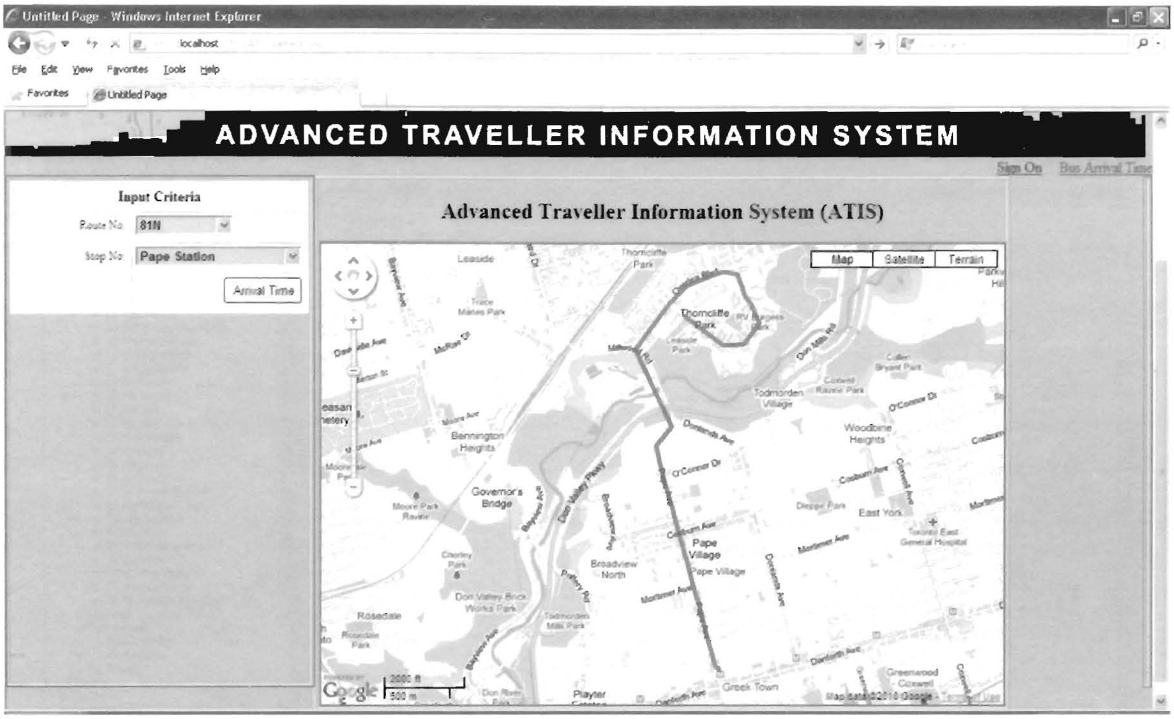
Route No: Select Route

Stop No:

Advanced Traveller Information System (ATIS)

FIGURE 4-13 ATIS (User route, bus-stop selection)

Once the user selected the bus-route, route will be highlighted on the map as shown in FIGURE 4-14.



ADVANCED TRAVELLER INFORMATION SYSTEM

Input Criteria

Route No: 81N

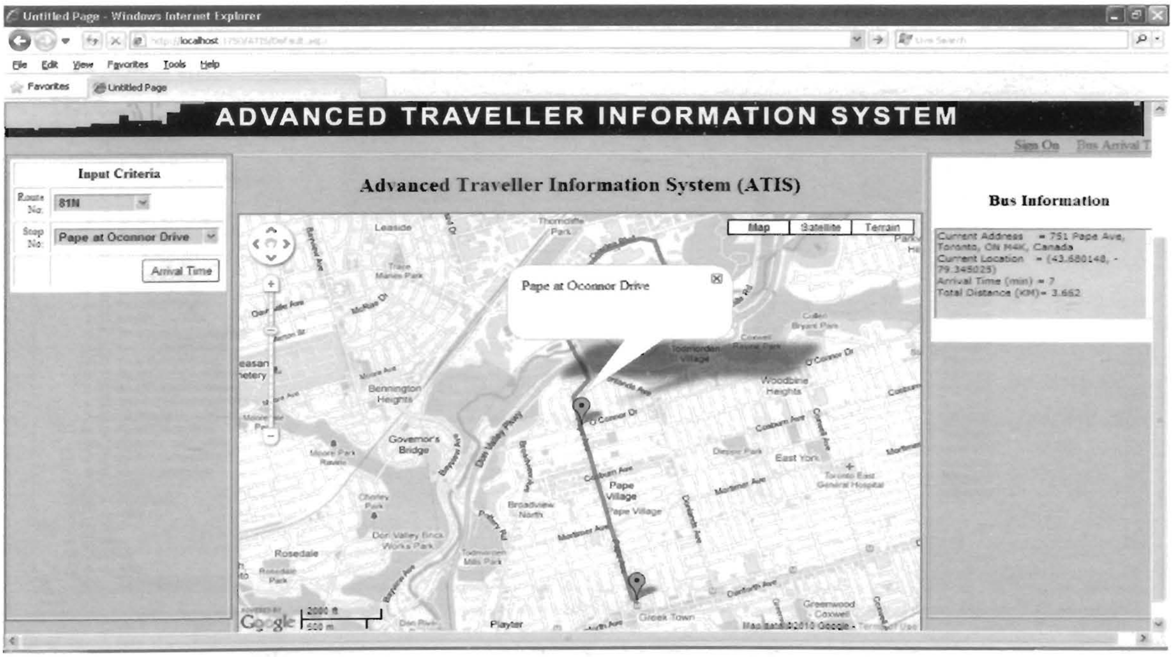
Stop No: Pape Station

Arrival Time

Advanced Traveller Information System (ATIS)

FIGURE 4-14 ATIS (Route-selected)

When a bus-stop is selected, system will display current bus location, the selected bus-stop, the estimated arrival time, and the bus distance, as shown in FIGURE 4-15.



ADVANCED TRAVELLER INFORMATION SYSTEM

Input Criteria

Route No: 81N

Stop No: Pape at Oconnor Drive

Arrival Time

Advanced Traveller Information System (ATIS)

Pape at Oconnor Drive

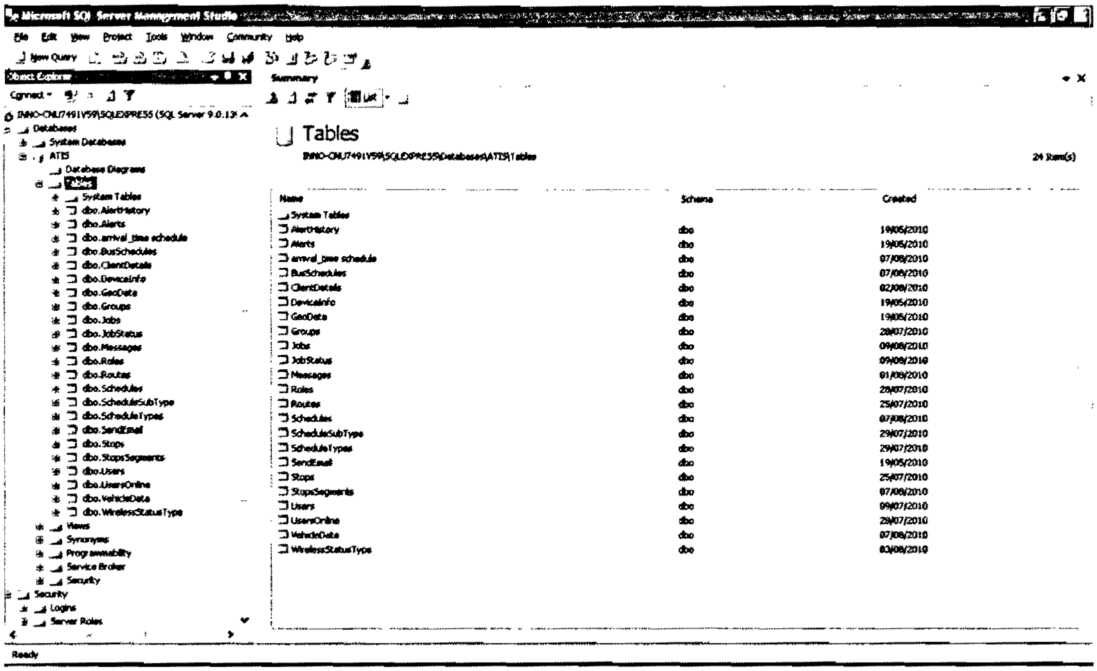
Bus Information

Current Address = 751 Pape Ave, Toronto, ON M4K, Canada
Current Location = (43.680148, -79.345025)
Arrival Time (min) = 7
Total Distance (KM)= 3.662

FIGURE 4-15 ATIS Bus-stop selected

4.3 DATABASE

Database is the source of data storage and retrieval. It consists of the tables where data is stored and the stored procedures. These are T-SQL scripts which web service calls to communicate to the data base, as shown in FIGURES 4-16 and 4-17.

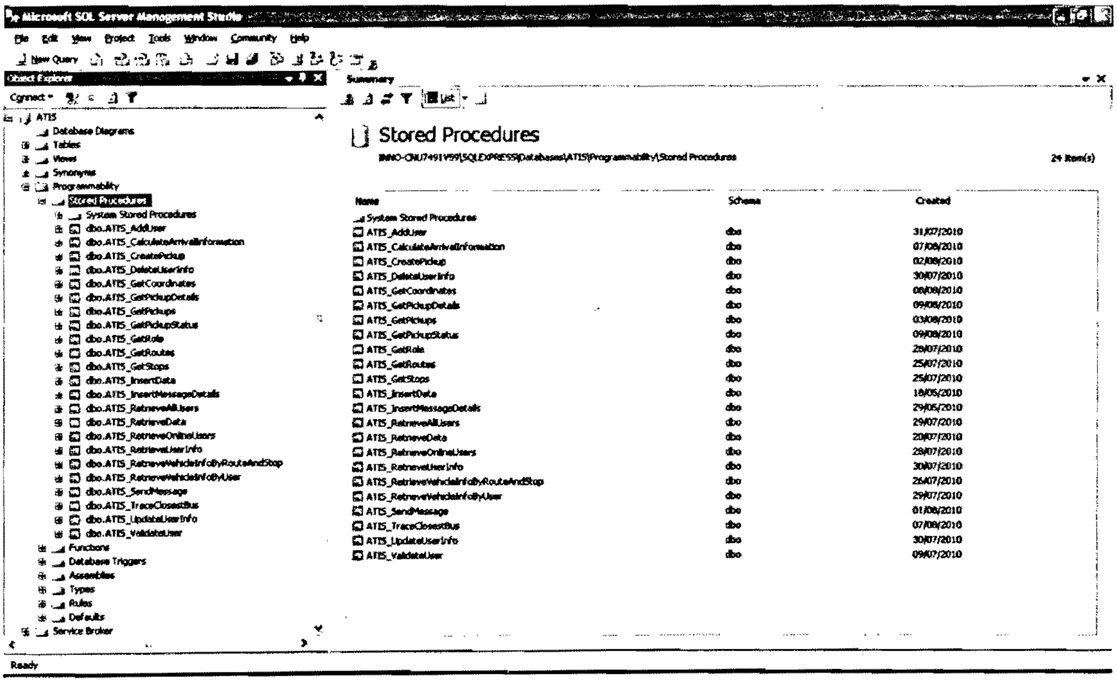

Microsoft SQL Server Management Studio
File Edit View Project Tools Window Community Help
New Query
Object Explorer
Summary
Ready

Tables

INNO-CNU7491V59\SQLEXPRESS\Databases\ATIS\Tables — 24 Item(s)

Name	Schema	Created
System Tables		
AlertHistory	dbo	19/05/2010
Alerts	dbo	19/05/2010
arrival_time schedule	dbo	07/08/2010
BusSchedules	dbo	07/08/2010
ClientDetails	dbo	02/08/2010
DeviceInfo	dbo	19/05/2010
GeoData	dbo	19/05/2010
Groups	dbo	28/07/2010
Jobs	dbo	09/08/2010
JobStatus	dbo	09/08/2010
Messages	dbo	01/08/2010
Roles	dbo	28/07/2010
Routes	dbo	25/07/2010
Schedules	dbo	07/08/2010
ScheduleSubType	dbo	29/07/2010
ScheduleTypes	dbo	29/07/2010
SendEmail	dbo	19/05/2010
Stops	dbo	25/07/2010
StopsSegments	dbo	07/08/2010
Users	dbo	09/07/2010
UsersOnline	dbo	28/07/2010
VehicleData	dbo	07/08/2010
WirelessStatusType	dbo	03/08/2010

FIGURE 4-16 Database Objects (tables)


Microsoft SQL Server Management Studio
File Edit View Project Tools Window Community Help
New Query
Object Explorer
Summary
Ready

Stored Procedures

INNO-CNU7491V59\SQLEXPRESS\Databases\ATIS\Programmability\Stored Procedures — 24 Item(s)

Name	Schema	Created
System Stored Procedures		
ATIS_AddUser	dbo	31/07/2010
ATIS_CalculateArrivalInformation	dbo	07/08/2010
ATIS_CreatePickup	dbo	02/08/2010
ATIS_DeleteUserInfo	dbo	30/07/2010
ATIS_GetCoordinates	dbo	08/08/2010
ATIS_GetPickupDetails	dbo	09/08/2010
ATIS_GetPickups	dbo	03/08/2010
ATIS_GetPickupStatus	dbo	09/08/2010
ATIS_GetRole	dbo	28/07/2010
ATIS_GetRoutes	dbo	25/07/2010
ATIS_GetStops	dbo	25/07/2010
ATIS_InsertData	dbo	18/05/2010
ATIS_InsertMessageDetails	dbo	29/05/2010
ATIS_RetrieveAllUsers	dbo	29/07/2010
ATIS_RetrieveData	dbo	20/07/2010
ATIS_RetrieveOnlineUsers	dbo	28/07/2010
ATIS_RetrieveUserInfo	dbo	30/07/2010
ATIS_RetrieveVehicleInfoByRouteAndStop	dbo	26/07/2010
ATIS_RetrieveVehicleInfoByUser	dbo	29/07/2010
ATIS_SendMessage	dbo	01/08/2010
ATIS_TraceClosestBus	dbo	07/08/2010
ATIS_UpdateUserInfo	dbo	30/07/2010
ATIS_ValidateUser	dbo	09/07/2010

FIGURE 4-17 Database Objects (Stored Procedures)

5 CONCLUSIONS AND FUTURE WORK

This project has presented a low cost tracking system using GPS and GPRS. This combination provides continuous and real time tracking. The cost of this proposal is much lower than the SMS based tracking system; also using free Google map in this project reduces the monthly fees.

The system is controlled by GIS-based software that supports any type of GPS/GPRS kit and wireless communication networks. Dispatch center can perform multiple actions by using this system and users can find the arrival time to their stop in a friendly interface. Map matching is used in this system to locate the bus-route on the Google map.

A mathematical model has been used to calculate the predicted arrival time. This model depends on the GPS data and the historical data. Bus speed estimation is developed to overcome the zero speed value that affects the arrival time prediction. The arrival time is calculated in minutes and rounded up when it is half a minute or above.

Future work includes:

- **Trip planner**: a system can give choices to customers who want information to plan their TTC trip in advance.
- **Display screens**: the system could include display screens at bus-stops and bus terminals. That will make TTC customers-friendly system, that the system provides real-time information to the costumers.
- **Cell phone service**: next bus information can be provides as a text message at bus-stops or on-board vehicle.

6 REFRENCES

1. R. F. Casey, L. N. Labell, L. Moniz, J. W. Royal, M. Sheehan, et al. " Advanced Public Transportation Systems: The State of the Art Update 2000". Publication FTA-MA-26-7007-00-1, *Federal Transit Administration, U.S. Department of Transportation,* December 2000.

2. R. H. Jeong. "The prediction of bus arrival time using automatic vehicle location systems data". *Doctoral dissertation, Texas A&M University*, December 2004.

3. " Automatic Vehicle Location Systems" http://www.vehiclelocationsystem.com/AVL%20Explained.htm. Accessed July 5, 2010.

4. "What is GPS and how does it work" http://gis.ednet.ns.ca/GPS/how_does_gps_work.htm . Accessed August 5, 2010.

5. B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, et al. "Global Positioning System: Theory and Practice" Springer: *New York*, NY, U.S.A., 1997.

6. "Glenn Baddeley - GPS - NMEA sentence information" http://home.pacific.net.au/~gnb/gps/nmea.html . Accessed August 5, 2010.

7. C. M. Johnson and E. L. Thomas. "Automatic Vehicle Location Successful Transit Applications: A Cross-Cutting Study", *Federal Transit Administration, U.S. Department of Transportation*, August 2000.

8. O. A. Bayari, and B. Sadoun, "New centralized automatic vehicle location communications software system under GIS environment", *International Journal of Communication Systems*, Vol 18, Issue 9, April 2005, pp. 833–846.

9. K. S. Hasan, M. Rahman, A. L. Haque, et al. "Cost Effective GPS-GPRS Based Object Tracking System", *Proceedings of the International Multi Conference of Engineers and Computer Scientists* 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong.

10. S. I. J. Chien, Y. Ding, and C. Wei. "Dynamic Bus Arrival Time Prediction with Artificial Neural Networks". *Journal of Transportation Engineering*, September/October 2002, pp. 429-438.

11. "Toronto Transit Commission Report". Aug. 26th, 2009. Available at www.ttc.ca Accessed August 10, 2010.

12. S. R. Ojili, "A Prototype of a Bus Arrival Time Notification System Using Automatic Vehicle Location Data". *Master's thesis, Texas A&M University*, College Station, June 1999.

13. W. Lin, and J. Zeng. "An Experimental Study on Real Time Bus Arrival Time Prediction with GPS Data. CD-ROM." *78th Annual Meeting of the Transportation Research Board, National Research Council*, Washington D.C., January 1999.

14. L. Weigang and W. Marlon, "Algorithms for Estimating Bus Arrival Times Using GPS Data". The IEE 5th *international Conference on Intelligent Transportation systems*, Proceedings 2002.

15. "TTC Routes and Schedules"
http://www.toronto.ca/open/catalogue.htm. Accessed August 4, 2010.

16. S. Christian Jensen and D. T. Syt'e. "GPS data management with applications in collective transport". *The IEE Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Dublin, Ireland 2008.

17. S. Christian Jensen and D. T. Syt'e. "A GIS-based Dynamic Shortest Path Determination in Emergency Vehicles". *World Applied Sciences Journal 3 (Supple 1):*pp. 88-94, 2008.

18. J. Greenfeld , "Automatic vehicle location (AVL) for transit operation", *IEEE Melecon '2000 10th Mediterranean Electro-technical Conference*, May 29-31, 2000 Cyprus (Peer)

19. O. Aloquili1, A. Elbanna, and A. Al-Azizi1: "Automatic vehicle location tracking system based on GIS environment" *IET Software*. 2009, Vol. 3, Issue. 4, pp. 255–263.

20. E. M. Adnan, M. Y. Arfat and M. Fraz: "Design and implementation of real time vehicle tracking system", *2nd International conference on Computer, Control and Communication,* Karachi, Feb. 17-18, 2009, pp. 1-5.

21. A. Karbassi and M. Barth: "Vehicle Route Prediction and Time of Arrival Estimation Techniques for Improved Transportation System Management", *IEEE Proceeding, Intelligent Vehicles Symposium*, July 28, 2003, pp.511-516.

22. Vehicle Tracking Systems – Overview (2008)
http://www.roseindia.net/technology/vehicletracking/Vehicle trackingSystems.shtml
Accessed August 4, 2010.

23. S. Ritier and J. MCCoy (1977). "Automatic Vehicle Location: An overview", *IEEE Transactions on* Vehicular Technology, Vol. 26 Issue 1. pp. 7-11

24. Automatic vehicle location (2008). Wikipedia http://en.wikipedia.org/wiki/Automatic_vehicle_location#Application_with_vehicles Accessed August 10, 2010.

25. Overview of GIS, http://www.gisdevelopment.net/tutorials/tuman006.htm. Accessed August 4, 2010.

26. GIS Applications, http://centrin.net.id/~agul/gis_apl.html. Accessed August 10, 2010.

27. Geographical Coordinates System, http://en.wikipedia.org/wiki/Geographic_coordinate_system . Accessed August 5, 2010.

28. Geo-code, http://en.wikipedia.org/wiki/Geocode . Accessed September 4, 2010.

29. Reverse Geo-coding, http://en.wikipedia.org/wiki/Reverse_geocoding .Accessed August 10, 2010.

30. J. B. Pick, "Geo-business GIS in the digital organization", John Wiley and Sons, 2008, pp. 46.

31. Shalaby, A., and A. Farhan "Bus Travel Time Prediction for Dynamic Operations Control and Passenger Information Systems". *CD-ROM. 82nd Annual Meeting of the Transportation Research Board,* National Research Council, Washington D.C., January 2003.

32. Farhan, A. "Bus Arrival Time Prediction for Dynamic Operations Control and Passenger Information System". *Master's thesis, Department of Civil Engineering, University of Toronto,* 2002.

7 APPENDIX

7.1 ABBREVIATION LIST

ACK	ACKNOWLEDGEMENT
APTS	ADVANCED PUBLIC TRANSPORTATION SYSTEM
ATIS	ADVANCED TRAVELLER INFORMATION SYSTEM
AVCS	ADVANCED VEHICLE CONTROL SYSTEM
API	APPLICATION PROGRAMMING INTERFACE
AVL	AUTOMATIC VEHICLE LOCATION
CCD	CALL CENTER DEPARTMENT
CVO	COMMERCIAL VEHICLE OPERATION
DMZ	DEMILITARIZED ZONE
GIS	GEOGRAPHICAL INFORMATION SYSTEM
GPRS	GENERAL PACKET RADIO SERVICE
GPS	GLOBAL POSITIONING SYSTEM
GSM	GLOBAL SYSTEM FOR MOBILE COMMUNICATION
GUI	GRAPHICAL USER INTERFACE
HTTP	HYPER TEXT TRANSFER PROTOCOL
ITS	INTELLIGENT TRANSPORTATION SYSTEM
LORAN	LONG RANGE AID TO NAVIGATION
MVC	MODEL VIEW CONTROLLER
NMEA	NATIONAL MARINE ELECTRONICS ASSOCIATION
SMS	SHORT MESSAGING SERVICE
SOAP	SIMPLE OBJECT ACCESS PROTOCOL
SQL	STANDARD QUERY LANGUAGE
SYN	SYNCHRONIZATION
TCP/IP	TRANSMISSION CONTROL PROTOCOL
T-SQL	TRANSACT- SQL
UDDI	UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION
UMTS	UNIVERSAL MOBILE TELECOMMUNICATIONS SYSTEM
VTS	VEHICLE TRACKING SYSTEM
WSDL	WEB SERVICE DESCRIPTION LANGUAGE
XML	EXTENSIBLE MARKUP LANGUAGE

7.2 DATABASE TABLES SCRIPTS

```
CREATE TABLE [dbo].[AlertHistory](
[group_id]          [INT]               NOT NULL,
[create_date]       [DATETIME]          NOT NULL DEFAULT (getdate()),
[user_id]           [INT]               NOT NULL,
[alert_id]          [INT]               NOT NULL,
[update_date]       [DATETIME]          NOT NULL,
CONSTRAINT [PK_avl_alert_history] PRIMARY KEY CLUSTERED

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


CREATE TABLE [dbo].[Alerts](
       [alert_id]   [INT]               NOT NULL,
       [alert_name] [VARCHAR](100)      COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
 CONSTRAINT [PK_avl_alerts] PRIMARY KEY CLUSTERED
(
       [alert_id] ASC
)

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



CREATE TABLE [dbo].[arrival_time schedule](
       [ScheduleType]  [VARCHAR](50)COLLATE Latin1_General_CI_AS NULL,
       [PeriodType]   [VARCHAR](50) COLLATE Latin1_General_CI_AS NULL,
       [StopId]       [VARCHAR](50) COLLATE Latin1_General_CI_AS NULL,
       [StopName]     [VARCHAR](50) COLLATE Latin1_General_CI_AS NULL,
       [ArrivalTime] [VARCHAR](50) COLLATE Latin1_General_CI_AS NULL,
       [StartingTime]  [VARCHAR](50)COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]

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


CREATE TABLE [dbo].[BusSchedules](
       [ScheduleType]       [FLOAT]                     NULL,
       [PeriodType]         [FLOAT]                     NULL,
       [StopId]             [FLOAT]                     NULL,
       [StopName]           [VARCHAR](255)  COLLATE Latin1_General_CI_AS NULL,
       [ArrivalTime]        [DATETIME]           NULL,
       [StartingTime]       [DATETIME]           NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[ClientDetails](
       [ConfirmationNo][VARCHAR](50)     COLLATE Latin1_General_CI_AS NOT NULL,
       [FirstName]          [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL,
       [LastName]           [VARCHAR](50) COLLATE Latin1_General_CI_AS NULL,
       [Address]            [VARCHAR](500)COLLATE Latin1_General_CI_AS NOT NULL,
       [City]               [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL,
       [State]              [VARCHAR](2)  COLLATE Latin1_General_CI_AS NOT NULL,
       [Zip]                [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL,
       [HomePhone]          [VARCHAR](15) COLLATE Latin1_General_CI_AS NOT NULL,
       [CellPhone]          [VARCHAR](15) COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]

--------------------------------------------------------------------------------
```

```
CREATE TABLE [dbo].[DeviceInfo](
       [DeviceId]   [INT]        NOT NULL,
       [DeviceName] [VARCHAR](10) COLLATE Latin1_General_CI_AS NOT NULL,
 CONSTRAINT [PK_DeviceInfo] PRIMARY KEY CLUSTERED
(
       [DeviceId] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

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

CREATE TABLE [dbo].[GeoData](
       [Geocode_date] [DATETIME]  NOT NULL,
       [DeviceID]     [VARCHAR](20)COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [Route]                          [VARCHAR](10)       COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [MessageID]               [VARCHAR](10)       COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [RSSI]                    [SMALLINT]                 NULL,
       [TowerID]                 [VARCHAR](10)       COLLATE
SQL_Latin1_General_CP1_CI_AS NULL,
       [ChannelID]               [VARCHAR](10)       COLLATE
SQL_Latin1_General_CP1_CI_AS NULL,
       [Coverage]                [TINYINT]                  NULL CONSTRAINT
[DF_GeoData_Coverage]  DEFAULT ((0)),
       [GPSStatus]               [VARCHAR](1)        COLLATE
SQL_Latin1_General_CP1_CI_AS NULL,
       [GPSTime]                 [DATETIME]                 NULL,
       [Longitude]               [FLOAT]                           NULL,
       [Latitude]                [FLOAT]                           NULL,
       [GPSModeIndicator]  [VARCHAR](1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
       [create_date] [DATETIME] NULL DEFAULT (getdate())
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Groups](
       [GroupId]    [INT] NOT NULL,
       [RoleId]     [INT] NOT NULL,
       [Details]    [VARCHAR](50) COLLATE Latin1_General_CI_AS NULL,
 CONSTRAINT [PK_UserGroup] PRIMARY KEY CLUSTERED
(
       [GroupId] ASC,
       [RoleId] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Jobs](
       [JobId]                          [INT] IDENTITY(1,1) NOT NULL,
       [WirelessStatusId]  [TINYINT]    NOT NULL CONSTRAINT
[DF__jobs__wireless_s__060DEAE8]  DEFAULT ((0)),
       [UserId]                  [VARCHAR](50) COLLATE Latin1_General_CI_AS NULL
CONSTRAINT [DF__jobs__user_id__08EA5793]  DEFAULT ((0)),
       [DispatcherId]            [VARCHAR](25) COLLATE Latin1_General_CI_AS NOT NULL
CONSTRAINT [DF__jobs__dispatcher__09DE7BCC]  DEFAULT ((0)),
       [JobStatusId]       [TINYINT]    NOT NULL CONSTRAINT
[DF__jobs__job_status__0AD2A005]  DEFAULT ((0)),
       [JobDate]                 [DATETIME]   NOT NULL CONSTRAINT
[DF__jobs__job_date__0BC6C43E]  DEFAULT (getdate()),
       [CreateDate]        [DATETIME]   NOT NULL CONSTRAINT
[DF__jobs__create_dat__0CBAE877]  DEFAULT (getdate()),
       [StatusChangeDate]  [DATETIME]   NOT NULL CONSTRAINT
[DF__jobs__status_cha__0DAF0CB0]  DEFAULT (getdate()),
       [AcceptedBy]        [VARCHAR](50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
CONSTRAINT [DF__jobs__accepted_b__0F975522]  DEFAULT (''),
       [Instructions]             [VARCHAR](500)COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL CONSTRAINT [DF__jobs__instructio__117F9D94]  DEFAULT (''),
       [ConfirmationNo]    [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL
CONSTRAINT [DF__jobs__confirm_nu__1367E606]  DEFAULT (''),
       [ScheduleTime]             [VARCHAR](20)  COLLATE Latin1_General_CI_AS NOT NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[JobStatus](
       [JobStatusId] [TINYINT]          NOT NULL,
       [Details]           [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Messages](
       [FromUser]          [VARCHAR](25) COLLATE Latin1_General_CI_AS NOT NULL,
       [ToUser]            [VARCHAR](25) COLLATE Latin1_General_CI_AS NOT NULL,
       [MessageRead] [TINYINT]          NULL,
       [CreateDate]  [smallDATETIME] NOT NULL CONSTRAINT [DF_Messages_CreateDate]
DEFAULT (getdate()),
       [ReadDate]          [smallDATETIME] NULL,
       [ProcessedDate] [smallDATETIME] NULL,
       [MessageDetails][VARCHAR](1000) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [MessageType] [TINYINT]          NOT NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Roles](
       [RoleId]      [INT]              NOT NULL,
       [Details]     [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL,
 CONSTRAINT [PK_Roles] PRIMARY KEY CLUSTERED
(
       [RoleId] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Routes](
       [RouteId]     [VARCHAR](20)  COLLATE Latin1_General_CI_AS NOT NULL,
 CONSTRAINT [PK_Routes] PRIMARY KEY CLUSTERED
(
       [RouteId] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Schedules](
       [RouteId]                  [VARCHAR](20)  COLLATE Latin1_General_CI_AS NOT NULL
CONSTRAINT [DF_Schedules_RouteId]  DEFAULT ('81N'),
       [StopId]                   [VARCHAR](20) COLLATE Latin1_General_CI_AS NOT NULL,
       [ScheduleType]             [INT] NOT NULL,
       [SchedulySubType]   [INT] NOT NULL,
       [ArrivalTime]       [VARCHAR](50)  COLLATE Latin1_General_CI_AS NOT NULL,
       [StartTime]                [VARCHAR](50)  COLLATE Latin1_General_CI_AS NOT NULL
```

```
) ON [PRIMARY]

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

CREATE TABLE [dbo].[ScheduleSubType](
       [ScheduleSubType]   [INT]              NOT NULL,
       [Description]       [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL
) ON [PRIMARY]


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

CREATE TABLE [dbo].[ScheduleTypes](
       [ScheduleType] [INT]                   NOT NULL,
       [Description]  [VARCHAR](50)     COLLATE Latin1_General_CI_AS NOT NULL,
 CONSTRAINT [PK_ScheduleTypes] PRIMARY KEY CLUSTERED
(
       [ScheduleType] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

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

CREATE TABLE [dbo].[SendEmail](
       [group_id]          [INT] NOT NULL,
       [send_email_id] [INT] IDENTITY(1,1) NOT NULL,
       [email_address] [VARCHAR](100) COLLATE Latin1_General_CI_AS NOT NULL,
 CONSTRAINT [PK_avl_send_email] PRIMARY KEY CLUSTERED
(
       [group_id] ASC,
       [email_address] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Stops](
       [RouteId]    [VARCHAR](20)       COLLATE Latin1_General_CI_AS NOT NULL,
       [StopId]     [VARCHAR](20)       COLLATE Latin1_General_CI_AS NOT NULL,
       [Details]    [VARCHAR](50)       COLLATE Latin1_General_CI_AS NOT NULL,
       [Latitude]   [DECIMAL](18, 10)   NOT NULL,
       [Longitude] [DECIMAL](18, 10)    NOT NULL,
       [Distance]   [DECIMAL](18, 0)    NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[StopsSegments](
       [RouteId]    [VARCHAR](20) COLLATE Latin1_General_CI_AS NOT NULL,
       [StopId]     [INT]               NOT NULL,
       [SegmentId] [INT]                NOT NULL,
       [Location]   [VARCHAR](200)      COLLATE Latin1_General_CI_AS NOT NULL,
       [Latitude]   [DECIMAL](18, 8) NOT NULL,
       [Longitude] [DECIMAL](18, 8) NOT NULL,
       [Distance]   [INT]                NOT NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[Users](
       [UserId]            [VARCHAR](25) COLLATE Latin1_General_CI_AS NOT NULL,
       [Password]          [VARCHAR](10) COLLATE Latin1_General_CI_AS NOT NULL,
       [Home_Phone]  [VARCHAR](15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [MobilePhone] [VARCHAR](15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [Address1]          [CHAR](50)          COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL,
       [Address2]          [CHAR](50)          COLLATE SQL_Latin1_General_CP1_CI_AS
NULL,
       [City]              [CHAR](20)          COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL,
       [State]                    [CHAR](2)           COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [ZipCode]           [VARCHAR](7) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
       [GroupId]           [INT]               NULL
) ON [PRIMARY]



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

CREATE TABLE [dbo].[UsersOnline](
       [UserId] [VARCHAR](25) COLLATE Latin1_General_CI_AS NOT NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[VehicleData](
       [VehicleId]                [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL,
       [Longitude]                [DECIMAL](18, 8) NOT NULL,
       [Latitude]                 [DECIMAL](18, 8) NOT NULL,
       [Speed]                           [INT]               NOT NULL,
       [Direction]                [CHAR](1)           COLLATE Latin1_General_CI_AS NOT
NULL,
       [CurrentAddress]    [VARCHAR](150)      COLLATE Latin1_General_CI_AS NULL,
       [CreateTime]        [DATETIME]          NOT NULL CONSTRAINT
[DF_VehicleData_CurrentTime]  DEFAULT (GetDate()),
       [DeviceId]                 [INT]               NOT NULL CONSTRAINT
[DF_VehicleData_DeviceId]  DEFAULT ((1)),
       [RouteId]                  [VARCHAR](20) COLLATE Latin1_General_CI_AS NULL,
       [CurrentStopId]            [INT]               NULL,
       [CurrentSegmentId]  [INT]               NULL,
       [TripStatus]        [TINYINT]           NULL
) ON [PRIMARY]

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

CREATE TABLE [dbo].[WirelessStatusType](
       [WirelessStatusId]  [TINYINT]           NOT NULL,
       [Description]       [VARCHAR](50) COLLATE Latin1_General_CI_AS NOT NULL
) ON [PRIMARY]
```

7.3 BUS SCHEDULE BY BUS STOP FOR THE STUDY PERIOD

TABLE 7-1 Bus Schedule by Bus Stop for the Study Period (Weekday)

Time Period	Stop 1	Stop 2	Stop 3	stop 4	stop 5	stop 6
Overnight	5:25	5:28	5:29	5:31	5:36	5:38
	5:39	5:42	5:43	5:45	5:50	5:52
	5:53	5:56	5:57	5:59	6:04	6:06
Morning peak period	6:07	6:10	6:11	6:13	6:18	6:20
	6:21	6:24	6:25	6:27	6:32	6:34
	6:26	6:29	6:30	6:32	6:37	6:39
	6:31	6:34	6:35	6:37	6:42	6:44
	6:35	6:38	6:39	6:41	6:46	6:48
	6:40	6:43	6:44	6:46	6:51	6:53
	6:45	6:48	6:49	6:51	6:56	6:58
	6:49	6:52	6:53	6:55	7:00	7:02
	6:54	6:57	6:58	7:00	7:05	7:07
	6:59	7:02	7:03	7:05	7:10	7:12
	7:03	7:06	7:07	7:09	7:14	7:16
	7:08	7:11	7:12	7:14	7:19	7:21
	7:13	7:16	7:17	7:19	7:24	7:26
	7:17	7:20	7:21	7:23	7:28	7:30
	7:22	7:25	7:26	7:28	7:33	7:35
	7:27	7:30	7:31	7:33	7:38	7:40
	7:31	7:34	7:35	7:37	7:42	7:44
	7:36	7:39	7:40	7:42	7:47	7:49
	7:41	7:44	7:45	7:47	7:52	7:54
	7:45	7:48	7:49	7:51	7:56	7:58
	7:50	7:53	7:54	7:56	8:01	8:03
	7:55	7:58	7:59	8:01	8:06	8:08
	7:59	8:02	8:03	8:05	8:10	8:12
	8:04	8:07	8:08	8:10	8:15	8:17
	8:09	8:12	8:13	8:15	8:20	8:22
	8:13	8:16	8:17	8:19	8:24	8:26
	8:18	8:21	8:22	8:24	8:29	8:31
	8:23	8:26	8:27	8:29	8:34	8:36
	8:27	8:30	8:31	8:33	8:38	8:40
	8:32	8:35	8:36	8:38	8:43	8:45
	8:37	8:40	8:41	8:43	8:48	8:50
	8:41	8:44	8:45	8:47	8:52	8:55
	8:46	8:49	8:50	8:52	8:57	8:59

	8:51	8:54	8:55	8:57	9:02	9:04
	8:55	8:58	8:59	9:01	9:06	9:05
Midday	9:00	9:03	9:05	9:07	9:13	9:15
	9:10	9:13	9:15	9:17	9:23	9:25
	9:20	9:23	9:25	9:27	9:33	9:35
	9:30	9:33	9:35	9:37	9:43	9:45
	9:40	9:43	9:45	9:47	9:53	9:55
	9:50	9:53	9:55	9:57	10:03	10:05
	10:00	10:03	10:05	10:07	10:13	10:15
	10:10	10:13	10:15	10:17	10:23	10:25
	10:20	10:23	10:25	10:27	10:33	10:35
	10:30	10:33	10:35	10:37	10:43	10:45
	10:40	10:43	10:45	10:47	10:53	10:55
	10:50	10:53	10:55	10:57	11:03	11:05
	11:00	11:03	11:05	11:07	11:13	11:15
	11:10	11:13	11:15	11:17	11:23	11:25
	11:20	11:23	11:25	11:27	11:33	11:35
	11:30	11:33	11:35	11:37	11:43	11:45
	11:40	11:43	11:45	11:47	11:53	11:55
	11:50	11:53	11:55	11:57	12:03	12:05
	12:00	12:03	12:05	12:07	12:13	12:15
	12:10	12:13	12:15	12:17	12:23	12:25
	12:20	12:23	12:25	12:27	12:33	12:35
	12:30	12:33	12:35	12:37	12:43	12:45
	12:40	12:43	12:45	12:47	12:53	12:55
	12:50	12:53	12:55	12:57	13:03	13:05
	13:00	13:03	13:05	13:07	13:13	13:15
	13:10	13:13	13:15	13:17	13:23	13:25
	13:20	13:23	13:25	13:27	13:33	13:35
	13:30	13:33	13:35	13:37	13:43	13:45
	13:40	13:43	13:45	13:47	13:53	13:55
	13:50	13:53	13:55	13:57	14:03	14:05
	14:00	14:03	14:05	14:07	14:13	14:15
	14:10	14:13	14:15	14:17	14:23	14:25
	14:20	14:23	14:25	14:27	14:33	14:35
	14:30	14:33	14:35	14:37	14:43	14:45
	14:40	14:43	14:45	14:47	14:53	14:55
	14:50	14:53	14:55	14:57	15:03	15:05
Afternoon peak period	15:00	15:03	15:05	15:07	15:13	15:15
	15:13	15:16	15:18	15:21	15:28	15:30
	15:20	15:23	15:25	15:28	15:33	15:35

	15:26	15:29	15:31	15:34	15:39	15:41
	15:32	15:35	15:37	15:40	15:45	14:47
	15:37	15:40	15:42	15:45	15:50	15:52
	15:43	15:46	15:48	15:51	15:56	15:58
	15:49	15:52	15:54	15:57	16:02	16:04
	15:54	15:57	15:59	16:02	16:07	16:09
	16:00	16:03	16:05	16:08	16:13	16:15
	16:05	16:08	16:10	16:13	16:18	16:20
	16:11	16:14	16:16	16:19	16:24	16:26
	16:17	16:20	16:22	16:25	16:30	16:32
	16:22	16:25	16:27	16:30	16:35	16:37
	16:28	16:31	16:33	16:36	16:41	16:43
	16:33	16:36	16:38	16:41	16:46	16:48
	16:39	16:42	16:44	16:47	16:52	16:54
	16:45	16:48	16:50	16:53	16:58	17:00
	16:50	16:53	16:55	16:58	17:03	17:05
	16:56	16:59	17:01	17:04	17:09	17:11
	17:01	17:04	17:06	17:09	17:14	17:16
Early Evening	17:07	17:10	17:12	17:15	17:20	17:22
	17:13	17:16	17:18	17:21	17:26	17:28
	17:18	17:21	17:23	17:26	17:31	17:33
	17:24	17:27	17:29	17:32	17:37	17:39
	17:29	17:32	17:34	17:37	17:42	17:44
	17:35	17:38	17:40	17:43	17:48	17:50
	17:41	17:44	17:46	17:49	17:54	17:56
	17:46	17:49	17:51	17:54	17:59	18:01
	17:52	17:55	17:57	18:00	18:05	18:07
	17:57	18:00	18:02	18:05	18:10	18:12
	18:03	18:06	18:08	18:11	18:16	18:18
	18:09	18:12	18:14	18:17	18:22	18:24
	18:15	18:18	18:20	18:22	18:27	18:29
	18:21	18:24	18:26	18:28	18:33	18:35
	18:28	18:31	18:33	18:35	18:40	18:42
	18:35	18:38	18:40	18:42	18:47	18:49
	18:40	18:43	18:45	18:47	18:52	18:54
	18:45	18:48	18:50	18:52	18:57	18:59
	18:55	18:58	19:00	19:02	19:07	19:09
	19:05	19:08	19:10	19:12	19:17	19:19
	19:15	19:18	19:20	19:22	19:27	19:29
	19:25	19:28	19:30	19:32	19:37	19:39
	19:35	19:38	19:40	19:42	19:47	19:49

	19:45	19:48	19:50	19:52	19:57	19:59
	19:55	19:58	20:00	20:02	20:07	20:09
	20:05	20:08	20:10	20:12	20:17	20:19
Late Evening	20:15	20:18	20:20	20:22	20:27	20:29
	20:25	20:28	20:30	20:32	20:37	20:39
	20:36	20:39	20:40	20:42	20:47	20:49
	20:48	20:51	20:52	20:54	20:58	21:00
	21:00	21:03	21:04	21:06	21:10	21:12
	21:12	21:15	21:16	21:18	21:22	21:24
	21:24	21:27	21:28	21:30	21:34	21:36
	21:36	21:39	21:40	21:42	21:46	21:48
	21:48	21:51	21:52	21:54	21:58	22:00
	22:00	22:03	22:04	22:06	22:10	22:12
	22:12	22:15	22:16	22:18	22:22	22:24
	22:24	22:27	22:28	22:30	22:34	22:36
	22:36	22:39	22:40	22:42	22:46	22:48
	22:48	22:51	22:52	22:54	22:58	23:00
	23:00	23:03	23:04	23:06	23:10	23:12
	23:12	23:15	23:16	23:18	23:22	23:24
	23:24	23:27	23:28	23:30	23:34	23:36
	23:36	23:39	23:40	23:42	23:46	23:48
	23:48	23:51	23:52	23:54	23:58	24:00:00
	24:00:00	24:02:00	24:03:00	24:05:00	24:09:00	24:11:00
	24:20:00	24:22:00	24:23:00	24:25:00	24:29:00	24:51:00
	24:40:00	24:42:00	24:43:00	24:45:00	24:49:00	24:51:00
	25:00:00	25:02:00	25:03:00	25:05:00	25:09:00	25:11:00
	25:20:00	25:22:00	25:23:00	25:25:00	25:29:00	25:31:00