

# Advanced Pickup & Delivery of Parcels Using

# **GPS/GIS** Technologies

# By

Arfan Muhammad

A project report 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

©Arfan Muhammad, 2010

# **Author's Declaration**

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

Arfan Muhammad

I further authorize Ryerson University to reproduce this report 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.

Arfan Muhammad

÷...

1

# **Borrower's Information**

٢

Ryerson University requires the signatures of all persons using or photocopying this report. Please sign below, and give address and date.

Name	Address	Date
	,	
•		
		- <b>·</b>
		3
	9 	
	ì	
		s. gan
	,	• •
		,

Ш

# Abstract

The main focus of this project addresses some of the key problems in the postal industry which are adhoc pickups, efficient and quick delivery, increase in vehicle maintenance, overtime and fuel cost. Companies involved in postal industry are spending millions of extra dollars to fulfill customer's need. There is a need of some reliable solution which can help improving the existing processes which are still using old technologies.

This project uses advanced and reliable state of the art technologies like Global Positioning System (GPS) and Geographical Information System (GIS) to improve the existing process of Pickup and Delivery (P&D) of parcels which are the two fundamental services provided by the postal industry. Today, there is a huge competition and the companies involved in postal industry are allocating a lot of budget to provide better, improved, reliable and satisfactory service to the customers.

This project presents an integrated approach to combine GPS and GIS to help improving the existing pickup/delivery process. A GPS enabled mobile device capable of wireless connectivity can be mounted in the courier's vehicle. The existing shipment and tracking application can be modified to communicate to GPS and transmit GPS data to the server wirelessly. GIS functionality can be embedded in the server side to calculate the shortest route between the courier's current position and the customer address. Similarly, digital maps can be used for visual display of the couriers and their online activities.

# Acknowledgement

During my graduate studies at Ryerson University, there are a number of people whom I would like to say my heartiest thanks because of their help, guideline, support and friendship due to which this project came to light.

I would like to thank the members of my project committee, Dr. Truman Yang and Dr. Yifeng He for their insights and guidance. In particular, Dr. Lian Zhao for her support, encouragement and guidance throughout the project. She has the greatest impact on my academic development during the project. I finished this project only because of her indepth knowledge and research in this field. She continuously gave very valuable suggestions, advice and recommendations. The completion of this project is credited to her. It has truly been an honor to be her student.

I owe a great many heartfelt thanks to Innovapost management, especially David Rea, Alnoor Ladak, Mike Liang, Radek Zelechowski and Gene Stec. Their continuous support, help and encouragement made it possible to finish this project on time.

I would also like to thank my classmate and good friend Bassim Ibrahim for his help and support. We studied and developed this project together and it was very good and pleasant experience.

My heartiest thanks to my best friend Azher Malik who helped me a lot in designing graphics for the website and who took care of me throughout M.Eng. Program.

My deepest gratitude and appreciation are reserved for my mother whose prayers and support helped throughout my life, my brother, sisters and niece. Without their constant love, support and encouragement, I would not have gone so far as to produce this report.

v

# **TABLE of Contents**

Abstract IV
AcknowledgementV
List of TABLESIX
List of FIGURESX
1 Introduction
1.1       Motivations       1         •       Fuel Consumption       2         •       Vehicle Maintenance       2         •       Courier Overtime       2
1.2       Current Technology
1.3       Limitations/Flaws       4         • Static Routing       5         • Courier Location       5         • Courier Guidance       5         1.4       Project Objectives
1.5       Project Modules & Contributions and Outline       6         1.5.1       Project Modules & Contributions       6         1.5.2       Report Outline       7
2Background82.1Automatic Vehicle Location (AVL) Systems82.1.1Benefits of AVL systems92.1.2Vehicle Location Technologies92.1.2.1Global Positioning System102.1.2.2Dead-Reckoning122.1.2.3Signpost /Odometer Systems122.1.2.4Radio Navigation /Location12
<ul> <li>2.2 Data Transmission</li></ul>
2.3Geographical Information System (GIS)142.3.1Philosophy of GIS152.3.2Advantages and Applications of GIS152.3.3GIS Location and Coordinates172.3.3.1Geo-coding and Reverse Geo-coding18•Geo-coding18

	• Rev	erse Geo-coding	. 19
	2.4 Dijkst	ra	. 20 <sup>-</sup>
	2.4.1 Bas	ic Flowchart	. 20
	2.4.2 Rou	iting Techniques	. 21
	2.4.2.1	Fixed /Deterministic	. 21
	2.4.2.2	Adaptive	. 21
	2.4.2.3	Flooding	. 22
	2.4.2.4	Multicasting	. 22
	2.4.2.5	Hierarchical	. 22
	2.4.2.6	Random	. 22
	2.4.3 Alg	orithm	. 22
	2.4.3.1	Problem Analysis	. 23
	2.4.3.2	Detailed Flowchart	. 24
2	Stratom Do	ni an	25
3	System De	sign	. 23
	3.1 Archit	ecture	. 25
	3.1.1 Mo	dules / Development Technologies	. 26
	3.2 Comp	onents / Modules	. 27
	3.2.1 Ser	ver	. 27
	3.2.1.1	TCP Connection Establishment Process	. 28
	3.2.1.2	Connection Establishment Functions	. 28
	3.2.1.3	Control Messages	. 28
	3.2.1.4	Pickup Job Assignment	. 29
	3.2.1.5	Package Delivery	. 30
	3.2.1.6	Startup Flow	. 31
	3.2.1.7	Processing Jobs/Messages Flow	. 32
	3.2.2 Clie	ent	. 32
	3.2.2.1	Mobile Device	. 34
	3.2.2.2	Login Flow	. 35
	3.2.2.3	Data Protocol	. 36
	3.2.3 We	b Service	. 36
	3.2.3.1	Process Flow	. 38
	3.2.4 We	D SITE	. 39
	3.2.4.1	User Kole Authentication	. 39
	3.2.5 Dat	abase	. 40
4	Developme	nt /Simulation	. 43
	4.1 Client	/ Server	. 43
	4.2 Web S	Site (User + Dispatch Application)	. 45
	4.3 Databa	ase	. 52
	1 A D::1+	ro Algorithm	52
		a Algorithill	53
	ч.ч.1 ГUП	schohanty / Example	
5	Conclusion	s and Future Directions	. 58

6	Bib	liography	60
7	Apj	pendix	63
	7.1	Abbreviation List	63
	7.2	Database Tables Scripts	64

# **List of TABLES**

TABLE 1-1: Limitations in the existing system	4
TABLE 2-1: RMC data format	11
TABLE 2-2: Advantages and disadvantages of different location technologies	13
TABLE 3-1: APDIS Models	26
TABLE 3-2: Client / Server data protocol	36
TABLE 3-3: Client / Server data protocol examples	36
TABLE 3-4: Database Objects (Tables)	40
TABLE 3-5: Database Objects (Stored Procedures)	42
TABLE 4-1: Distance between the nodes	53

# **List of FIGURES**

FIGURE 1-1: Current Architecture / Technology	3
FIGURE 2-1: Basic AVL System	8
FIGURE 2-2: GPS Satellites [3]	10
FIGURE 2-3: Four satellites are needed to calculate the position [3]	10
FIGURE 2-4: Latitude / Longitude Concepts [36]	17
FIGURE 2-5: Northern /Southern hemisphere with Longitude/Latitude	18
FIGURE 2-6: Dijkstra Algorithm Flowchart	20
FIGURE 2-7: Detailed Flowchart	24
FIGURE 3-1: System Architecture	25
FIGURE 3-2: Client / Server communication via wireless	27
FIGURE 3-3: Client/Server communications	30
FIGURE 3-4: Server Start-up Process	31
FIGURE 3-5: Processing Job/Message Flow Diagram	32
FIGURE 3-6: Wireless communication	33
FIGURE 3-7: Mobile Device used by a Courier	34
FIGURE 3-8: Login Flow Diagram	35
FIGURE 3-9: Data Exchange between tiers and database	37
FIGURE 3-10: Process Flow Diagram	38
FIGURE 3-11: User Authentication	39
FIGURE 4-1: Server Start-up	43
FIGURE 4-2: Client → Server Connection	44
FIGURE 4-3: Server Real-time Status	44
FIGURE 4-4: Client / Server Data Communications	45
FIGURE 4-5: Welcome to APDIS	46
FIGURE 4-6: APDIS (Sign On)	46
FIGURE 4-7: APDIS (Administrator Login)	47
FIGURE 4-8: APDIS (Administrator → Add User)	47
FIGURE 4-9: APDIS (Administrator → Modify/Update User)	48
FIGURE 4-10: APDIS (Administrator → Delete User)	48
FIGURE 4-11: APDIS (Administrator $\rightarrow$ View Online User with GIS)	49
FIGURE 4-12: APDIS (Administrator → View Pickup Jobs)	49
FIGURE 4-13: APDIS (Administrator $\rightarrow$ Client Communication via Messaging)	50
FIGURE 4-14: APDIS (Administrator → Create Pickup)	50
FIGURE 4-15: APDIS (Customer → Check Pickup Status)	51
FIGURE 4-16: APDIS (Customer → Trace Courier/Customer Location via GIS)	51
FIGURE 4-17: Database Objects (tables)	52
FIGURE 4-18 Database Objects (Stored Procedures)	52
FIGURE 4-19A: Dijkstra Execution (Step - 1)	54
FIGURE 4-20B: Dijkstra Execution (Step - 2)	54
FIGURE 4-21C: Dijkstra Execution (Step - 3)	55
FIGURE 4-22D: Dijkstra Execution (Step - 4)	55
FIGURE 4-23E: Dijkstra Execution (Step - 5)	56
FIGURE 4-24F: Dijkstra Execution (Step - 6)	56
FIGURE 4-25G: Dijkstra Execution (Step - 7)	56
FIGURE 4-26H: Dijkstra Execution (Step - 8)	57

# **1** Introduction

Package pickup and delivery are the most important services offered by the postal industry. A customer wants his/her package to be picked up and/or delivered in quick time. Because of this reason, companies involved in postal industry are focusing on improving the exiting process to fulfill customer's need by providing more reliable and satisfactory service.

Currently, a customer can visit any post office to send his/her package. Other way to send the package is either by calling a customer service or visiting the website to create a pickup. The first method is very straightforward. In the second method, the customer requests post office to have someone visit him/her and pickup the package. Such kind of pickup is called On Demand pickup because while creating pickup request (via phone or web), the customer mentions the scheduled time for a courier to come and pickup the package. Customer support staff asks various questions like name, address, time to pickup, payment mode, package dimensions (height/width/length and weight) etc. Call Center Department (CCD) representative enters all this information into the system using GUI based Dispatch application to create pickup request. Currently, second half of the day is reserved for the **PICKUPS**.

Package Delivery is always done by the courier. A courier serving the area goes to customer's address to deliver the package. Under special circumstances, customer is requested to go to a nearby post office to collect the package. Currently, first half of the day is reserved for **DELIVERIES**. There are 3 categories of the deliveries: 9:00 AM, 10:30 AM and regular. Packages under first two types must be delivered on or before this time otherwise company will have to pay to the customer a fixed amount as penalty. Regular delivery times are 10:30 AM onwards.

# **1.1 Motivations**

Current system allows couriers to serve in a specific region. Each courier is assigned a geographical area based on the postal code. He/she is responsible for pickup and delivery of all packages in this area. Recently, 50% of the pickups are on demand pickups in which case customer contacts the call center or browse website and requests someone to visit his/her place to pick up the package. Customer mentioned the pickup time and the courier reaches within the scheduled pickup time window to collect the package. A courier carries a mobile device capable of wireless communication. The centralized system automatically sends the pickup requests to the courier through wireless network and the courier goes to the destination address to pick up the package. It happens most of the time that a courier serving in the area is on one end of the region and the destination address is on the other end. He would go all the way down to pickup this package although another courier serving in next area may have been very close to the destination

address. It happens because the current system does not allow this job to send to the nearest courier allowing to pickup package in shortest possible time.

A bigger challenge in today's postal industry is how to improve the existing process which caters for customer's need and at the same time, how company itself can get maximum benefits.

### **G** Fuel Consumption

With the existing technology and pickup process, companies involved in postal industry are spending huge amount on fuel every year. Major portion of this loss is from the special pickup process because the system sends pickup requests continuously to the couriers and they have to plan their trip accordingly.

#### Vehicle Maintenance

Current system is based on postal codes. A courier login the system using route id (post code) and the system sends all pickup jobs for this route to the courier and he/she has to go to destination based on the pickup scheduled time. A courier has to drive from one end to another a number of times in a day which increases mileage on the vehicle.

### Courier Overtime

Service providers believe in "Customer satisfaction is guaranteed". Imagine, a courier is just ending his/her day and a pickup job is delivered by the system. The courier has to entertain this otherwise the company can lose the customer. To entertain such variances, courier would need to work extra hours and the company will have to pay overtime.

### **1.2 Current Technology**

Current technology uses GPRS as wireless communication media to communicate to the couriers. The system uses static method of assigning new jobs to the couriers because each courier serves a specific area (based on postal codes). When a customer requests a call center, or goes to website to create a pickup request, he/she gives detailed information like full name, address, postal code, number of packages, package(s) dimension, payment mode etc. The system creates the pickup job and saves into the database. There are few services that are scheduled to continuously monitor the system database for any new PICKUP job. As soon as a new record is created in these tables, one service gets this information, process it and the wireless service sends it to the courier who is serving that area. The wireless service sends this job wirelessly using wireless network (GPRS in current scenario). The wireless service knows which courier is serving

this specific route so the right courier gets the job request. The courier receives this job, views it and then accepts or declines. Received, viewed, accepted, rejected status are sent back to the system wirelessly and the system updates the job status accordingly. The basic system architecture is shown in FIGURE 1-1.



FIGURE 1-1: Current Architecture / Technology

FIGURE 1-1 shows 3 scenarios of data exchange between mobile device and the backend servers. The scanner or mobile device resides inside courier's vehicle and is capable of making wireless connection using GSM/GPRS. A courier needs to sign on before using this device. Upon successful sign on, the device is registered with the system. Courier keeps this device with him/her throughout the day. All scanned pins and customer's signatures are continuously stored in the local database. On returning back to the office, this scanner is handover to the supervisor who makes sure to connect it to the backend servers to download all information from the local database. The device is then kept on AC source to start charging. The remote management server is responsible to retrieve scanner's inventory and send any updated version of software application or windows patch.

#### 1.2.1 Job Status

The courier receives all kind of pickups/messages wirelessly from the server. Various changes happen in the job status.

#### Sending/Pending

Server received a job from the central database and is ready to send to the courier.

### Wireless Sending

Server has made decision to send this job to a particular courier and is trying to send wirelessly.



Server has successfully sent the job to the specific courier.

### **3** Viewed

As soon as the courier receives this job in the form of message, he is alerted through a new incoming message. The courier views it and the status of the job changes to VIEWED.

### Rejected

If the job is not relevant to the courier, he may reject it.

### Accepted

This indicates that the courier has accepted the job.

#### Completed

The courier has successfully completed the job.

### Failure

The courier is unable to complete the job because of various reasons (customer not available, office closed etc).

# 1.3 Limitations/Flaws

There are some limitations and flaws in the existing system. These issues need to be fixed to offer a more reliable, flexible and efficient system. TABLE 1-1 provides a list of issues in the current system.

#### **TABLE 1-1: Limitations in the existing system**

Issue ID	Issue	
1	Static Routing of the calls	
2	No Courier Information Available	
3	No Guidance available to the Courier	
4	Increase in Vehicle Maintenance	
5	Increase in Courier Overtime	
6	Increase in Fuel Consumption	

# Static Routing

System uses static method to send pickup request to the couriers. The job is sent to the courier even if he is far away from the destination. The job remains in pending queue if a courier is not logged in to the route that means the system will not entertain this job and the dispatcher has to call the courier and verbally give information about the customer address and other details.

## Courier Location

The system is not capable of detecting the courier's location.

# Courier Guidance

Most of the couriers are very experienced and they are very much familiar with the routes in their areas. Backup or new couriers do face some problem because they need assistance to reach the destination address to pickup or deliver a package. The system does not provide such assistance to the courier.

# **1.4 Project Objectives**

Given the problem description from the previous section, the objective of this research is to show how GPS and GIS provide a way to

- Oynamically allocate the pickups to the couriers.
- Trace a courier's current position based on geo-coordinates.
- Assist the courier in finding the customer address.
- Reduce vehicle maintenance cost annually.
- Reduce vehicle fuel charges annually.
- Reduce courier's overtime.

# 1.5 Project Modules & Contributions and Outline

### 1.5.1 Project Modules & Contributions

Following is the list of system modules and the brief introduction of their contribution to the system.

### Automatic Vehicle Location (AVL)

AVL is an important part of this project. This module consists of Client /Server application with database as storage media. Clients send geo-location to the server and the server continuously stores this data into the database. The main contribution of this module in the system is to store GPS data to trace the courier's truck. This module is a TCP/IP based multithreaded client/server application developed using Microsoft .Net (C#). Client side application continuously communicates to GPS on COM port to receive geo-location (Latitude, Longitude, Speed and Direction) and sends to the server side application. A GPS enabled mobile device capable of wireless connectivity was used to develop a standard AVL system (refer to Chapter 2).

#### Web Service

Like website, this module also hosts on web server. This service acts as the gateway between server and the database. It provides a communication channel between server and database as well as website and database. The main contribution of this module in the system is to provide a data layer for an application with n-tier architecture. It is behind the firewall to avoid any SQL injection from the hackers (refer to Chapter 3). The development was done using Microsoft .Net framework (C#, ADO.Net).

#### Website (Central Application)

There is always a need to have some centralized application which can provide functionality to the users based on users group and role assigned. Users like to see the status of their pickup/delivery; administrators want to perform administrations and the dispatchers like to do dispatch and view real-time activities. The main contribution of this module in the system is to provide a centralized platform to be used by the administrators, dispatchers and the customers to retrieve the desired information. Microsoft .Net was used to develop various parts of this application. ASP.Net and Java Script were used on client side while all objects, controllers and the models were developed in C#. This website communicates to the web service which in turn communicates to the database via ADO.Net. Communication between website and database happens through the web service (refer to Chapter 3).

### Oatabase

A storage media is required in any application to store and retrieve the data. The main contribution of this module is to provide a data storage and retrieval media. Geo-data, pickup jobs, messages and alerts are stored in the database. Microsoft SQL Server 2005 was used to create the database and Microsoft T-SQL language was used to create various tables and stored procedures. The web service which is a data layer calls stored procedures to send/receive data from/to the database (refer to Chapter 3).

### 1.5.2 Report Outline

The remaining of the report is organized as following:

- Chapter 2 covers background information on Automatic Vehicle Location Systems. The information involves a discussion of various 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. It also covers Dijkstra (a shortest path algorithm). GIS has big contribution in this project so it is also covered in this chapter.
- Chapter 3 presents system design. It highlights system architecture. 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 dependent on other objects. It also discusses the Google map and JavaScript APIs for GIS and Dijkstra algorithm to find the shortest route.
- Chapter 4 provides the simulation result with step by step processing and execution. It also shows how client application connects to the server and how TCP/IP handshake works. It shows how GIS can be used to display data on the digital maps.
- Chapter 5 concludes recent work and summarizes future directions and endeavors of this project.
- Chapter 6 focuses on the references used.

# 2 Background

# 2.1 Automatic Vehicle Location (AVL) Systems

AVL are computer-based vehicle tracking systems. They are used in tracking fleet, police, transit, ambulance, and military purposes. FIGURE 2-1 shows a schematic diagram of an AVL system used in transit agencies. It shows the standard AVL system with GPS signal arriving from the satellites to the truck. The client communicates to the server and sends vehicle location to the dispatch software.



FIGURE 2-1: Basic AVL System

AVL is an advanced method to track and monitor a vehicle remotely. Hardware based AVL system is installed in vehicle and this GPS enabled system receives signals from the communication satellites. The client software communicates to GPS receiver and determines its current location, speed and heading. This data can be stored locally or directly transmitted to the server. Currently, the server side application uses digital maps integrated with GIS (Geographical Information System) to display the current position of the vehicle. Most popular Automatic Vehicle Location is based on GPS, GIS and GSM (Global standard of mobile communication).

AVL system 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 truck or vehicle, a wireless communications network (GPRS or GSM) link between the vehicle and the dispatch center, and a web based tracking application for dispatch.

### 2.1.1 Benefits of AVL systems

The benefits of AVL systems are as follows:

- Provide real-time information to the public.
- Improve schedule reliability.
- Improve service efficiency.
- Reduce operating and maintenance costs.
- Enhance safety and security
- Transit agencies can respond more quickly to emergency situations

With advantages gained by using AVL systems, many transit agencies are planning to install AVL systems as the cost of AVL systems has dropped significantly.

### 2.1.2 Vehicle Location Technologies

AVL systems use one of the four types of navigation technology, or may combine two of these technologies to compensate for inevitable shortcomings of any one technology. The four principal technologies employed for AVL systems are:

- 1. Global Positioning System
- 2. Signpost and Odometer
- 3. Radio Navigation/Location
- 4. Dead-Reckoning

### 2.1.2.1 Global Positioning System

GPS is 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. GPS is a satellite-based navigation system which is funded and controlled by the US. GPS employs the signals transmitted from a network of satellites orbiting the earth. These signals are picked up by a receiver onboard the bus. 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 consists of 24 satellites as shown in FIGURE 2-2 [3]. To compute current position, speed and direction, signals from at least four satellites are used as shown in FIGURE 2-3 [3].



GPS Nominal Constellation

20,200 km Altitudes, 55 Degree Inclination

FIGURE 2-2: GPS Satellites [3]



FIGURE 2-3: Four satellites are needed to calculate the position [3]

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]. Vehicle position can be calculated using four satellites. 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 NMEA -0183. NMEA-0183 Standard input/output messages protocol could be in different types:

- GGA Global positioning system fixed data.
- GLL Geographic position latitude/longitude
- GSA GNSS DOP and active satellites
- GSV GNSS satellites in view
- RMC Recommended minimum specific GNSS data
- VTG Course over ground and ground speed

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

Name	Example	Units	Description
Message ID	\$GPRMC		RMC protocol header
UTC (time)	101841	hhmmss.ss	Universal Time Coordinates
Status	Α		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	Е		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></lf></cr>			End of message termination

 TABLE 2-1: RMC data format

### 2.1.2.2 Dead-Reckoning

Dead reckoning systems use odometer and compass to determine the vehicle position by measuring distance traveled from a known location and direction of travel. Dead reckoning sensors can measure distance and direction from a fixed point. 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 (tire circumference changes with wear).

### 2.1.2.3 Signpost /Odometer Systems

Before GPS, the signpost /odometer system was the most common and popular navigation system. System comprises of a receiver mounted on the bus, while transmitters are placed along the route. Utility poles and signposts are most commonly used as mounting locations for these transmitters. 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 is very costly as it requires installation of infrastructure.

### 2.1.2.4 Radio Navigation /Location

These systems also work in combination with other systems. Radio location systems use a low-frequency signal to cover the system, and the buses are located as they receive the signal. LORAN-C (Long Range Aid to Navigation) is the most common type of landbased radio location. A LORAN receiver on a vehicle makes time measurement of signals transmitted from at least three locations. 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 advantages and disadvantages of each technology.

Туре	Advantages	Disadvantages
GPS	1 - Accurate	1 - Signal attenuation by foliage and
	2 - Global coverage	tunnels.
	3 - Moderate cost per vehicle	2 - Subject to multi-path errors
Dead Reckoning	1 - Inexpensive	1 - Accuracy degrades with distance
	2 - Self-contained on vehicle (no	traveled.
	infrastructure costs)	2 - Requires direction indicator and
	3 - Only odometer needed (if on-	maybe map matching for off-route use.
	route is assumed)	3 - Corrupted by uneven road surfaces,
		steep hills, or magnetic interference.
Signpost and	Low in-vehicle cost	1 - Requires well-equipped infrastructure
Odometer	No blind spots or interference	2 - No data outside of deployed
	Repeatable accuracy	infrastructure.
		3 - Frequency of updates depends on
		density of signpost
Radio Navigation/	Moderately accurate	1 - Signals are susceptible to
Location	Costly	electromagnetic distortion by power
		lines, neon signs, railroad tracks, etc.

TABLE 2-2: Advantages and disadvantages of different location technologies.

### 2.2 Data Transmission

Position information, regardless of which location technology is adopted, needs to be transmitted to the dispatch center. Polling and Reporting are the two most common data transmission technologies [1, 7].

With polling technology, the server requests each bus for its location at regular intervals. With the reporting method, each bus reports its location after fixed interval of time.

# 2.2.1 Global System for Mobile Communication (GSM)

GSM is the most popular second generation (2g) digital cellular communication system in the world. In general, 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 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 mainly a computer based information system used to represent /analyse the geographic features digitally [31]. A Geographic Information System (GIS) captures, stores, analyzes, manages, and presents data that is linked to location. Technically, GIS is defined as 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 [31].

The power of a GIS comes from the ability to relate different information in a spatial context and to reach a conclusion about this relationship. 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 making 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" [31], 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).

Work on GIS began in late 1950s, but the first GIS software came only in late 1970s from the lab of the **ESRI**. Canada was the pioneer in the development of GIS as a result of innovations dating back to early 1960s.

Much of the credit for the early development of GIS goes to Roger Tomilson. Evolution of GIS has transformed and revolutionized the ways in which planners, engineers, managers etc. conduct the database management and analysis.

# 2.3.1 Philosophy of GIS

The proliferation of GIS is explained by its unique ability to assimilate data from widely divergent sources, to analyze trends over time, and to spatially evaluate impacts caused by development [31]. For an experienced analyst, GIS is an extension of a person's own analytical thinking. The system has no in-built solutions for any spatial problems; it depends upon the analyst. The importance of different factors of GIS in decreasing order is as below:

- Spatial Analysis
- Database
- Software
- Hardware

GIS involves complete understanding about patterns, space, and processes or methodology needed to approach a problem. It is a tool acting as a means to attain certain objective quickly and efficiently. Its applicability is realized when the user fully understands the overall spatial concept under which a particular GIS is established and analyses his specific application in the light of those established parameters. Before the GIS implementation is considered the objectives, both short term and long term, have to be considered. Since the effectiveness and efficiency (i.e. benefit against cost) of the GIS will depend largely on the quality of initial field data captured, organizational design has to be decided upon to maintain this data continuously.

# 2.3.2 Advantages and Applications of GIS

The Geographic Information System has been an effective tool for implementation and monitoring of municipal infrastructure. GIS is being used in various fields [32].

The use of GIS has been in vogue primarily due to the advantages mentioned below:

- Planning of project
- Make better decisions
- Visual Analysis
- Improve Organizational Integration

Following is the few example application of GIS [32].

GIS is used in a variety of agricultural applications such as managing crop yields, monitoring crop rotation techniques, and projecting soil loss for individual farms or entire agricultural regions.

A GIS is a software tool for managing business information of any kind according to where it is located. It can help in tracking of customers, site businesses, target marketing campaigns, optimize sales territories, and model retail spending patterns. It helps understanding and evaluating the data by creating graphic displays using information stored in the database.

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 geological features, analyze soils and strata, assess seismic information, or create 3-dimensional displays of geographic features.

Mapping is an essential function of a GIS. Professionals from variety of fields are using GIS to help others understand geographic data. A person doesn't have to be a skilled cartographer to make maps with a GIS.

It can help with risk management and analysis by showing which areas will be prone to natural or man-made disasters. Once identified, preventive measures can be developed that deal with the different scenarios.

It is being used to manage transportation infrastructure or logistical problems. GIS can be used in monitoring rail systems and road conditions or finding the best way to deliver goods or services, AVL (Vehicle Tracking System, Intelligent Transportation Systems) and transport arrival/departure schedules

# 2.3.3 GIS Location and Coordinates

Each location has some input information and GIS processes that information of the object. This is commonly done by identifying the X-Y earth coordinates of an object at a time and place [36]. The most common coordinate system is Longitude and Latitude as shown in the FIGURE 2-4.



FIGURE 2-4: Latitude / Longitude Concepts [36]

Longitude (abbreviation: Long. or  $(\lambda)$  pronounced lambda) is the angle east or west of a reference meridian between the two geographical poles to another meridian that passes through an arbitrary point. A meridian is a line on the surface of the earth extending from the North Pole to the South Pole, always at the same longitude. For instance, any point lying on north-south meridian that is at a 22.5° angle to the West of Greenwich, England,

lies on the meridian of -22.5 degrees longitude. All meridians are halves of great circles, and are not parallel and they converge at the north and south poles.

Latitude (abbreviation: Lat. or ( $\varphi$ ) pronounced phi) is the angle from a point on the Earth's surface and the equatorial plane, measured from the centre of the sphere [33]. Lines joining points of the same latitude are called parallels, which trace concentric circles on the surface of the Earth, parallel to the equator. The North Pole is 90° N; the South Pole is 90° S. The 0° parallel of latitude is designated the equator. The equator is the fundamental plane of all geographic coordinate systems. The equator divides the globe into Northern and Southern Hemispheres. FIGURE 2-5 shows the east and west angles for longitude and north and south angles for the latitude.



FIGURE 2-5: Northern /Southern hemisphere with Longitude/Latitude

A line passing to the rear of the Royal Observatory, Greenwich (near London in the UK) has been chosen as the international zero-longitude reference line, the Prime Meridian. Places to the east are in the eastern hemisphere, and places to the west are in the western hemisphere. The antipodal meridian of Greenwich is both 180°W and 180°E.

### 2.3.3.1 Geo-coding and Reverse Geo-coding

### • Geo-coding

Geo-coding is the process of finding geographic coordinates (often expressed as Latitude and Longitude) from physical address, such as house number, street address, or zip code (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.

A Geo-code representation format is a combination of some or all of the following geospatial attributes [34]:

- Geo-code Format Registry Number,
- Latitude
- Longitude
- Altitude
- Date
- Local time
- Global time in Universal Time Coordinates (UTC) format

- Coordinate Reference System (Coordinate Reference System Type Registry Number - ISO 19115,
- Geodetic Reference Ellipsoid, Registry Number, Geodetic Reference Datum Registry Number, Coordinate Format Registration Number - ISO 19145)
- Coordinate Sensor Accuracy for Latitude, Longitude and/or Altitude
- Internet Protocol version 4 (IPv4) address or Internet Protocol, version 6 (IPv6) address in Compressed, Uncompressed or Fully Uncompressed decimal format
- Other geospatial attributes

Mandatory attributes included in the Geo-code representation format are Geo-code Format Registry, Latitude, and Longitude. All other attributes are voluntary. When customer calls CCD and requests to create a pickup, the system performs geo-coding to convert customer's physical address to geo-location since digital maps can understand only geo-locations.

### • Reverse Geo-coding

Reverse geo-coding is the process of back (reverse) coding in which a point location (Latitude, Longitude) is converted to a readable address or place name [35]. This permits the identification of nearby street addresses, places, and/or aerial subdivisions such as neighborhoods, county, state, or country. Combined with geo-coding and routing services, reverse geo-coding is a critical component of mobile location-based services and Enhanced 911 to convert a coordinate obtained by GPS to a readable street address which is easier to understand by the end user.

Reverse geo-coding can be carried out systematically by services which process a coordinate similarly to the geo-coding process. For example, when a GPS coordinate is entered, the street address is interpolated from a range assigned to the road segment in a reference dataset that the point is nearest to. If the user provides a coordinate near the midpoint of a segment that starts with address 1 and ends with 100, the returned street address can be somewhere near 50.

This approach to reverse geo-coding does not return actual addresses, only estimates of what should be there based on the predetermined range. Alternatively, coordinates for reverse geo-coding can also be selected on an interactive map, or extracted from static maps by geo-referencing them in a GIS with predefined spatial layers to determine the coordinates of a displayed point. Many of the same limitations of geo-coding are similar with reverse geo-coding.

When customer browse website and wishes to see the status of his pickup request, the system performs reverse geo-coding to convert geo-location of the courier's truck to physical address since customers are interested in physical address only.

### 2.4 Dijkstra

In order to move from one location to another, the system must determine the path or the route links between source and destination. This can be achieved by using some algorithm which takes the dynamic decisions and directs the datagram from source to the destination. This is better known as Routing Algorithm. Routing Algorithms are classified depending upon the technique they employ to calculate the shortest path. Dijkstra Routing Algorithm is one of them.

Dijkstra's Algorithm is one of the most popular algorithms in computer science. It is also popular in operations research. It was conceived by Edsger Dijkstra, a Dutch scientist in 1956 [11]. The algorithm is generally used in routing and it solves the single source shortest path for a graph with non-negative edge costs and it produces a shortest path tree.

### 2.4.1 Basic Flowchart

FIGURE 2-6 shows basic flowchart of Dijkstra's Algorithm. It describes how a shortest path establishes between source and destination.





In route optimization, shortest path is the cost and this algorithm finds the path with lowest cost between a given source node and all other adjacent nodes till destination. It can also be used to find the costs of shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined.

Routing can be compared to the process of building graphs and directing the datagram from source to destination [30]. In order to make such dynamic routing decisions, router exchange information using a special algorithm known as Routing Algorithm [28].

### 2.4.2 Routing Techniques

The Routing Techniques can be sub-divided into following categories:

- Fixed / Deterministic
- Adaptive
- Flooding
- Multicasting
- Hierarchical
- Random

### 2.4.2.1 Fixed /Deterministic

Practical multi-computers use Deterministic Routing [20]. Fixed Routing has been widely used in practice due to its simplicity. However, message cannot take advantage of alternative paths to avoid congested network regions. Dimension-Ordered Routing [21], [22] is a typical example of deterministic routing where messages visit network dimensions in a predefined order.

### 2.4.2.2 Adaptive

Many fully Adaptive Algorithms, which overcome the performance limitations of deterministic routing, have been proposed in the past few years [23], [24], [25]. For instance, the concept of virtual channels has been generalized in [26] to develop deadlock-free fully adaptive routing algorithms.

### 2.4.2.3 Flooding

A static type of routing in which every incoming packet is sent out on every outgoing line except the one it arrived on [28]. Flooding is not practical in most applications, but it does have some uses. For example, in military applications, where large number of routers may be blown to bits at any instant, the tremendous robustness of flooding is highly desirable.

### 2.4.2.4 Multicasting

For some applications, widely-separated processes work together in groups. Sending a message to such a group is called Multicasting, and its routing algorithm is called Multicast Routing. To do multicast routing, each router computes a spanning tree covering all other routers in the subnet.

### 2.4.2.5 Hierarchical

As Networks grow in size, the router's routing tables grow proportionally. Not only router memory is consumed, but more CPU time is needed to scan them and more bandwidth is needed to send status reports. Such problems are solved by using Hierarchical Routing [28], where the routers are divided into regions, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the other region.

### 2.4.2.6 Random

In case of Random Routing, a node selects only one outgoing path for retransmission of an incoming packet. The outgoing link is chosen at random, excluding the link on which the packet arrived [30]. It is further sub-divided into two categories: Distance Vector Routing Algorithm and Link State Routing Algorithm [28], [30]. Bellman-Ford Routing Algorithm [28], [30] is an example of the former while Dijkstra Routing Algorithm [28], [30], [31] belongs to the latter.

### 2.4.3 Algorithm

Dijkstra Algorithm is one of the most common shortest path algorithms and can be applied to network routing. This network routing can either be a LAN/WAN etc. routing or geo-routing (roads network). The Algorithm is an iterative process that works through a graph or a set of vertices & paths to calculate the shortest path from any one source node to every other node in the set.

### 2.4.3.1 Problem Analysis

- 1. Read the number of nodes present in the network.
- 2. Read the activities between all the nodes.
- 3. Identify Source and Destination.
- 4. Find the distance of all the nodes from the source.
- 5. Select the node whose distance from the source is the least.
- 6. Lock the node that has passed i.e. source node in this case.
- 7. Find the distance of remaining nodes from the selected node and repeat steps 5 & 6 till the destination node is reached.
- 8. List all the selected nodes from source to destination in the sequence in which they are generated above to give the Optimum Path.

### 2.4.3.2 Detailed Flowchart

FIGURE 2-7 shows the detailed flowchart of Dijkstra's Algorithm. It describes step by step calculation of the shortest path between source and destination.





# 3 System Design

This chapter covers the design of the project in details. It provides detailed information about the architecture, various modules and how these modules are integrated to become a complete system.

### 3.1 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.

FIGURE 3-1 shows the system architecture. It describes various modules of the system and how they interact with each other.



FIGURE 3-1: System Architecture

# 3.1.1 Modules / Development Technologies

TABLE 3-1 shows various modules of the standard APDIS system and the technologies used to develop these modules.

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

### **TABLE 3-1: APDIS Models**
## 3.2 Components / Modules

### 3.2.1 Server

Server communicates to the clients on port number 8001. It controls handshake, broadcast and send/receive messages (tokens) to/from the clients. Server has the following two communication interfaces:

#### 1. Client

Client side application resides in the mobile device. Communication happens through TCP/IP protocol via wireless network. 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

Server communicates to the database through a web service. The web service is data layer and is hosted on the web server. It communicates to the database by calling stored procedures which avoids any chance of SQL injection. All read/write functionalities to/from the database are performed through the web service. This web service is behind the firewall and is in DMZ (demilitarized zone). It is necessary to secure the system so that database server is not exposed to the outer world. FIGURE 3-2 shows the client /server communication through wireless network.



FIGURE 3-2: 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 authenticate the user credentials and write this logging information to the database table(s).

## 3.2.1.1 TCP Connection Establishment Process

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

## 3.2.1.2 Connection Establishment Functions

The client /server connection establishment process 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 messages to each other. 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 (courier in this case) has own credentials which are sent upon connection request. Courier sends vehicle number plus user ID and password to the server and the server authenticates these credentials before accepting this connection.
- Parameter Exchange: Certain parameters may exchange that control the operation of the TCP.

## 3.2.1.3 Control Messages

Control messages controls the transmission. TCP uses these 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 carrying data. The first byte was introduced as data type to indicate the type of data contains in the packet.

First byte in the data transmission has the following values:

- 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 the message.

Second byte indicates the number of records contains in the packet while third byte contains the actual data delimited by semicolon.

## 3.2.1.4 Pickup Job Assignment

As soon as the server application starts, it starts an event handler which continuously polls the web service to check for the database jobs table for any new job. It performs the following actions upon receiving a new job (dispatcher creates a new job from the Dispatch application and this record is saved into the jobs table) from the database.

- Broadcast a message "SEND ME YOUR LOCATION" to all of the connected clients in that region.
- The clients in turns send their location in the form of 2 co-ordinates (LONGITUDE, LATITUDE).
- Performs reverse geo-coding to resolve customer's address to geo-location.
- Calls the web service to mark the status of this job as "SENDING".
- Calls the web service to update the geo-location of all of the connected clients.
- Executes Dijkstra's Algorithm to calculate the nearest client to the customer's location.
- Sends the job to the nearest client in the form of free test message.
- Calls the web service to mark the status of this job as "SENT".
- Receives the acknowledgement from the client.
- Calls the web service to mark the status of this job as "VIEWED".
- Client sends the "COMPLETE" status once the courier pickups the package.
- Calls the web service to mark the status of this job as "COMPLETED".

## 3.2.1.5 Package Delivery

Delivering a package to a customer starts after the pickup process. During pickup, a package gets picked up and delivery process helps delivering it to the destination. The process undergoes a series of actions before delivering the package to the customer.

- In Terminal
  - The package is in the terminal and ready to be sorted for specific region (postal code). The status of the package in this phase is called "IN TERMINAL".

Out For Delivery (OFD)

- The package is ready to go out for the delivery. In this phase, all the 0 packages in the terminal are scanned and sorted based on postal addresses. The status of the package in this phase is "OUT FOR DELIVERY".
- In Transit
  - The package is in transit (moving from one terminal to another). The 0 status of the package in this phase is changed to "IN TRANSIT".
- Delivery Failure
  - The courier is unable to deliver the package because of various reasons. It can be wrong address, office closed, locked etc.
- Delivered
  - The courier has successfully delivered the package to the customer. This is 0 the last phase of the life cycle of a package. The status of the package in this phase becomes "DELIVERED".

FIGURE 3-3 shows how client and server communicate to each other via wireless network.



FIGURE 3-3: Client/Server communications

## 3.2.1.6 Startup Flow

FIGURE 3-4 shows the flowchart of the server startup process.



FIGURE 3-4: Server Start-up Process

## 3.2.1.7 Processing Jobs/Messages Flow

FIGURE 3-5 shows the flowchart of processing a job or a message



FIGURE 3-5: Processing Job/Message Flow Diagram

## 3.2.2 Client

Each courier carries a mobile device (also called as scanner). Client side application resides in it. This application communicates to the server through TCP/IP communication protocol. The main functionality of this application is to communicate to GPS through COM port at regular interval of time and sends this information to the server. To avoid wireless network traffic, it is very easy to change the existing application so that the client may send geo-location only when requested by the server.

The application also called as P&D (Pickup & Delivery) takes care of all the activities performed by the couriers. A courier starts his day by Signing-On the application, which means the client side application makes connection to the server through TCP/IP.

FIGURE 3-6 shows high level communication between clients and server. It shows the on road courier trucks are connected to the server wirelessly.



**FIGURE 3-6: Wireless communication** 

## 3.2.2.1 Mobile Device

Motorola MC-70 is designed and manufactured by Motorola (formerly Symbol). This device is capable of GPS and wireless communication. It communicates to the GPS via COM port and NMEA protocol and connects to the wireless world using GSM/GPRS network. The client side application resides in this device, continuously reads COM port to read geo-information (geo-location, speed and direction) from the GPS and sends to the server in the predefined data format. CanadaPost and Purolator are currently using these devices.

FIGURE 3-7 shows Motorola MC-70 scanner which operates on windows mobile platform.





## 3.2.2.2 Login Flow

FIGURE 3-8 shows the flowchart of a client's login process. The process explains all the steps required for a client to login and send geo-data to the server.



FIGURE 3-8: Login Flow Diagram

## 3.2.2.3 Data Protocol

TABLE 3-2 shows how the data transmission is controlled using control bytes. Both client and server must agree on some common protocol otherwise the incoming and outgoing data can not be correctly interpreted. Client and server both follow the same protocol while sending /receiving data.

Character/Byte 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 records separated by comma. Entities within the records are separated by semicolons	

TABLE 3-2: Client / Server data protocol

Clients can send login, logout, data or message to the server. Similarly, server can send jobs and messages to the clients. TABLE 3-3 shows some examples of the data transmission. Client and server must strictly follow this data protocol.

TABLE 3-3: Client / Server data protocol examples

Туре	Example
LOGIN	0,1,AYAY285;****
LOGOUT	1,1, AYAY285
DATA	2,2,56.33;34.22;60;N,57.29;34.11;60;N
MESSAGE	3, 2, HI THIS IS TEST MESSAGE, CAN YOU PLEASE SEND SOME
	WITH A SCANNER AS I AM HAVING PROBLEM WITH THIS
	DEVICE. I AM AT BRITANIA/HURONTARIO INTERSECTION

## 3.2.3 Web Service

The most important part of the APDIS system which establishes a secured communication channel between the server/website and the database. This service is the data layer of the system and is deployed on the web server and it avoids any SQL injection from the hackers. 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 web based APDIS application and sends updated status back. The 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 process-able 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 Markup 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 in a manner similar to inter-process communication on a single computer.

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 various 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.

FIGURE 3-9 shows how the data exchange happens among various tiers and web service.



FIGURE 3-9: Data Exchange between tiers and database

## 3.2.3.1 Process Flow

FIGURE 3-10 shows the flowchart of the web service functionality. It describes how this module communicates to the database to insert/retrieve the data and returns to the server or website.



FIGURE 3-10: Process Flow Diagram

### 3.2.4 Web Site

The main function of this application is to provide a central mechanism for the users/customers, administrators and dispatchers to perform their tasks. This is a real time monitoring as well as administration tool. The administrators use this application to administrate the system, dispatchers use it to insert new job whenever any user calls the call center or sending messages to the online users, and customers use it to track their pickup requests. It continuously communicates to the web service for any new or updated status. As soon as the user log-in, the administration shows this user as logged-in and ready to accept any call. It continuously updates the status of the logged-in clients, jobs status, maps, and vehicle locations etc.

#### 3.2.4.1 User Role Authentication

FIGURE 3-11 shows flowchart of the user roles authentication.



FIGURE 3-11: User Authentication

## 3.2.5 Database

Oracle and SQL Server 2005 are the most popular Relational Database Management Systems (RDBMS) these days. Currently, most of the windows and web based applications are being developed using Microsoft technologies such as .Net (ASP.Net, C#, ADO.Net) so SQL Server is an ultimate choice for most of the companies. 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 for real-time and historical purposes. The dispatch application uses this data and displays the vehicles location in real-time mode while administrators and dispatchers can generate various history reports using archived data.

TABLE 3-4 shows a list of database tables created for data storage and retrieval.

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.
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] [DispatcherId] [JobStatusId] [JobDate] [CreateDate] [AcceptedBy] [Instructions] [ConfirmationNo]	The table is used to store job information. A newly created pickup job goes under various steps before completion

and the second second

**TABLE 3-4: Database Objects (Tables)** 

	[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.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.Users	[UserId] [Password] [Home_Phone] [MobilePhone] [Address1] [Address2] [City] [State] [ZipCode] [GroupId]	The table is used to store the users profiles.
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 3-5 shows a list of stored procedures. These procedures are invoked by the web service to communicate to the database.

Id	Stored Procedure Name	Brief Description
1	dbo.[APDIS_AddUser]	Called from data layer to add a new user
3	dbo.[APDIS_CreatePickup]	Called from data layer to create a pickup as
		requested by the customer. Administrator or
		dispatcher can perform this activity
4	dbo.[APDIS_DeleteUserInfo]	Called from data layer to delete an existing
		user from the system. Administrator or
		dispatcher can perform this activity
5	dbo.[APDIS_GetCoordinates]	Called from data layer to get geo-coordinate
		of the segments within each bus stop
6	dbo.[APDIS_GetPickupDetails]	Called from data layer to get pickup details
		(address, scheduled pickup time etc.).
7	dbo.[APDIS_GetPickups]	Called from data layer to get the pickup
		details. It can be in any status.
		Administrator or dispatcher can perform this
ļ		activity
8	dbo.[APDIS_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.[APDIS_GetRole]	Called from data layer to get the role of the
		user. This is called from website.
12	dbo.[APDIS_InsertData]	Called from data layer to insert GPS data
		coming from the client
13	dbo.[APDIS_InsertMessageDetail	Called from data layer to insert message
	s]	details. Administrator or dispatcher can
		perform this activity
14	dbo.[APDIS_RetrieveAllUsers]	Called from data layer to get a list of users
		that exist in the system. Administrator or
		dispatcher can perform this activity
16	dbo.[APDIS_RetrieveOnlineUsers	called from data layer to get a fist of
	1	online users. Administrator of dispatcher can
		Called from data lower to get the details of
17	dbo.[APDIS_RetrieveUserInio]	the year. Administrator or dispatcher can
		perform this activity
1.0	N. INDER Debuisme Validate Info	Called from data layer to get Vehicle
19	dbo.[APDIS_Retrievevenicienno	information based on user Id
	Byuser]	Information based on abor id
	When (ADDIG CondMaggaga)	Called from data layer to insert message
20	abo.[APDIS_SendMessage]	details for the server application to send to
		the online user
22	dha [ADDIG UndatelleerInfo]	Called from data layer to modify details of
22	apo.[APDI2_opdateoserinto]	an existing user from the system.
		Administrator or dispatcher can perform this
		activity
22	dha IADDIS Validatelleri	Called from data layer to validate user's
23	UDU. [AFDID_Validateobol]	credentials. It is called by the server upon
		receiving wireless connection from the user

La transformer and the second se

----

## TABLE 3-5: Database Objects (Stored Procedures)

# 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 continuously reads the COM for geo-location and TCP/IP interface to the server to send this data on 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

Advanced Information System (S	Server)	Advanced Information System (	(Server)
Connection Information IP Address: 127.0.0.1 Port No: 8001 Start/Stop Server:	Connected Clients	Connection Information IP Address: 27.001 Port No: 000 Start/Stop Server:	Connected Clients
Server Application Started	Ež	(it Server Started, Port = 800)	Exit

FIGURE 4-1: Server Start-up

Client can be started to establish a TCP/IP connection with the server. Each user has own credentials which he/she needs to pass before requesting for a connection. The application connects to the server on the port 8001 and sends login request as shown in FIGURE 4-2.

Advanced Information System (Client)	Advanced Information System (Client)	
User ID: AVAY285 Password: ****	User ID: AVAY285 Password:	
<u>C</u> onnect E <u>x</u> it	Disconnect Exit 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 list as shown in FIGURE 4-3.

📰 Advanced Information System (Se	erver)	-OX
Connection Information	□ Connected Clients AVAY285	1
Port No: 8001		
Start/Stop Server: 🔽		
		E <u>x</u> it

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 (currently 10 seconds) as shown in FIGURE 4-4.

📰 Connected to 127.0.0.1on port number 4242 💦 🔲 🗖 🔀
Client Server Data Communication:
20100907121319:0,1 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,-79.3480890;43.68749700;60;N 20100907121429:2,1,-79.34880960;43.68939130;60;N 20100907121439:2,1,-79.34880960;43.68939130;60;N 20100907121439:2,1,-79.34954320;43.69133720;60;N 20100907121449:2,1,-79.35095330;43.69535400;60;N 20100907121459:2,1,-79.34951070;43.69666680;60;N

FIGURE 4-4: Client / Server Data Communications

## 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 (C#) was used in data layer.

The application starts with the welcome page as shown in FIGURE 4-5. User is only interested in tracking his/her package while administrators and dispatchers can login with their credentials to perform their own tasks.



FIGURE 4-5: Welcome to APDIS

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



FIGURE 4-6: APDIS (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 add/remove/update users, view online users, view pickups and communicate to the client by messaging or creating pickups as shown in FIGURE 4-7



FIGURE 4-7: APDIS (Administrator Login)

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

C Untitled Page - Windows Internet Exp	rer	
G + 47 X # locabest.	121 PA 1112 T 12 K	<ul> <li>Businession</li> </ul>
Ele Edit Yew Figvorites Iools Help		
Pavortes 🙀 🗃	a he can . The she water	
Chttled Page		
1	-	
ADVA	CED PICKUP &	DELIVERY INFORMATION SYSTEM
Logged in as Administrator	0	
# Tasks	Contra	Courier Information
= Admin	Pausword	
Add User Model Live	House Phone	
Delete User	Cell Phone	
* Online Users * Pickum John	Address 1.	
* Client/User Communication	Addense 2	
NOT STREET STORE	City:	
	State.	
	Zip Code	
	Group 14:	
	Confirm to ADD the user	Add Upder- Drive

FIGURE 4-8: APDIS (Administrator → Add User)

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

Untitled Page - Windows Internet Exp	lórer	A CONTRACT OF A DESCRIPTION OF A DESCRIP	No. of Concession, Name	A DESCRIPTION OF THE OWNER OF	
G				2.3.17	
He Edit Yew Favorites Jools Help				1.7.14	
:: Favorites ::= 🔊					
Huntitled Page					
the star	ICED	RICKUR & DELI	VERY INFORMAT	TION SYS	
Logged in as Administrator					Sim On Trace Pac
- Tasks	User 1d.	atistest v		1000	
- Admin	Password	test		0.000	Courier Information
Add User Modify User	Home Phone.	414-111-1234		200	
Delete User	Call Phone	905-222-3456			
Online Users     Pickup Jobs	Address 1	960 AIRPORT ROAD			
* Client User Communication	Address 2	ORLANDO BUILDING			
Salar Charles Barries	City	MISSISSAUGA			
	State.	ON			
	Zip Code	L5R 2ME			
A SC STATISTICS	Group 1d:	2			
		$\hfill\square$ Confirm to MODIFY the user	Col Undate		
c			opuse		*

FIGURE 4-9: APDIS (Administrator → Modify/Update User)

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



FIGURE 4-10: APDIS (Administrator → Delete User)

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



FIGURE 4-11: APDIS (Administrator → View Online User with GIS)

Pickup Jobs help the administrator to view the current pickups and their status etc. as shown in FIGURE 4-12.



FIGURE 4-12: APDIS (Administrator → View Pickup Jobs)

Client Communication helps the administrator to send message to the courier. 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 4-13.



FIGURE 4-13: APDIS (Administrator → Client Communication via Messaging)

Administrator /Dispatcher can also create a pickup request in case a customer calls and requests to send courier to his/her address to pickup a package as shown in FIGURE 4-14.



FIGURE 4-14: APDIS (Administrator → Create Pickup)

The system helps/guides the customer to track his/her pickup as shown in FIGURE 4-15. Customer is asked to enter confirmation number. Customer must note down this confirmation number while calling the customer center or browsing the web to create a pickup request.



FIGURE 4-15: APDIS (Customer → Check Pickup Status)

System verifies the confirmation number and checks which courier has been assigned this job. It then displays the current location of the courier and the customer address on the map. The distance, current speed, estimated time taken by the courier to reach the customer and the directions are also displayed on the Google map as shown in FIGURE 4-16.



FIGURE 4-16: APDIS (Customer → Trace Courier/Customer Location via GIS)

## 4.3 Database

Database is the source of data storage and retrieval. It consists of various objects like tables which are used to store the data. FIGURE 4-17 shows the list of tables created for the entire system.

Nicrosoft SQL Server Management Stu	ıdio			C (2) (2)
Bie Edit View Project Loois Window	Community Help		and the second	
A New Casery of the Acade State of the	1.1 (4) 35 (2) 35 (3) (4)			
	a a p a p a p a a			
CODECT EXDERNE	Summary			- X
Connect · B/ - B F	A J J J T MUR - SI			- A
6 TANO-CNU7491V59(SQLEXPRESS (SQL Server 9.0	13.4			
Databases	Tables			
<ul> <li> System Databases</li> </ul>	Idules			
- J ATIS	IMNO-CNL/7491V59/SQLEXPRESSIDatabases1ATISITables			
ii Database Diagrams	the state of the s			24 Ren(s)
Tabler				
<ul> <li>System Tables</li> </ul>	Name			
<ul> <li>dbo AlertHistory</li> </ul>	Surface Tables	Schema	Created	
<ul> <li>dbo Alerts</li> </ul>	Alexandration			
<ul> <li>dbo.arrival_time.schedule</li> </ul>	- Ale theory	dbo	19/05/2010	
<ul> <li>dbo.BusSchedules</li> </ul>	LI AVERS	dbo	19/05/2010	
<ul> <li>I dbo ClientDetails</li> </ul>	] arrival_time schedule	doo	07/06/2010	
<ul> <li>dbo DeviceInfo</li> </ul>	BusSchedules	dbo	07/08/2010	
<ul> <li>dbo GeoData</li> </ul>		dbo	02/08/2010	
<ul> <li>dos Groups</li> </ul>	DeviceInfo	doo	19/05/2010	
<ul> <li>I dbo Jobs</li> </ul>	GeoData	da	19/05/2010	
<ul> <li>I dbo JobStatus</li> </ul>	Groups	the state	26/07/2010	
<ul> <li>I dbo Messages</li> </ul>	I Jobs	da	200012010	
<ul> <li>dbo Roles</li> </ul>	□ JobStatus	dec.	09908/2010	
<ul> <li>do Routes</li> </ul>	C Messages		09/08/2010	
<ul> <li>I dbo Schedules</li> </ul>	Roles	200 #10	01/08/2010	
<ul> <li>dbo ScheduleSubType</li> </ul>	Boutes	200	28/07/2010	
<ul> <li>dbo ScheduleTypes</li> </ul>	T Schedules	0.00	25/07/2010	
<ul> <li>I dbo SendEmal</li> </ul>	Televeletiet	dbo	07/08/2010	
<ul> <li>I dbo Stops</li> </ul>	T Sched delines	dbo	29/07/2010	
<ul> <li>I dbo StopsSegments</li> </ul>	The first first	dbo	29/07/2010	
<ul> <li>dbo Users</li> </ul>	- Server at	dbo	19/05/2010	
<ul> <li>do UsersOnline</li> </ul>	_ xops	dbo	25/07/2010	
<ul> <li>dbo VehideData</li> </ul>		dbo	07/08/2010	
<ul> <li>dbo WirelessStatusType</li> </ul>	Users	dbo	09/07/2010	
<ul> <li></li></ul>	UsersOnline	dbo	26/07/2010	
<ul> <li> Synonyms</li> </ul>	□ VehicleData	dbo	07/08/2010	
<ul> <li>Programmability</li> </ul>	WirelessStatusType	dbo	03/08/2010	
<ul> <li>Ja Service Broker</li> </ul>				
<ul> <li>A Security</li> </ul>				
<ul> <li>Lai Security</li> </ul>				
* Logns				
<ul> <li> Server Roles</li> </ul>	×.			
6 million Barris	8			
Ready				
			and the states and share have b	and the second se

FIGURE 4-17: Database Objects (tables)

Database contains another set of objects that are called stored procedures. These are T-SQL scripts which web service calls to communicate to the database. FIGURE 4-18 shows a list of stored procedures created for the web service.

Bit Die Weiter Dotte Date Barden Community Beiter         Die Lief Die Weiter Die Lief Die Barden Community Beiter         Die Lief Die Weiter Die Die Barden Community Beiter         Die Lief Die Weiter Die Die Barden Community Beiter         Die Die Die Barden Community Beiter         Die Die Die Barden Community Beiter         Die Die Barden Community         Die Die Die Barden Community         Die Die Die Barden Community         Die Die Die Die Barden Community         Die Die Die Die Die Barden Community         Die	Be Microsoft SQL Server Management Studio	the second s	the state of the second second		e x			
Description         Description           Outcome I         Description           International Control of Control o	File Edit View Protect Tools Window Community Help							
All       All       All         All       A	10 March 10 (10 (10 (10 (10 (10 (10 (10 (10 (10	to the seat -						
Numerative     Numerative       Interview     Interview       Interview		P D						
Cymer       Image:	Object Explorer	Summary			* X			
a Mitiliande Dargense         b Mitiliande Dargense         b Vision         b Vision <td>Connect. Bi = 1 T</td> <td>(法法学学)(10)・1000000000000000000000000000000000</td> <td></td> <td></td> <td>in the second</td>	Connect. Bi = 1 T	(法法学学)(10)・1000000000000000000000000000000000			in the second			
Autor Dayses     Autor Service     Autor Se	- I ATIS	8						
• Index       Second (Second (	Database Diagrams	Stored Procedures						
Image: Strams       Decode/Projection       24 Ban(s)         Image: Strams       Programmability (Strams Proceedings)       24 Ban(s)         Image: Strams       Strams Strams Strams Strams Strams       Strams Strams Strams Strams       Strams Strams Strams Strams Strams       Strams St	in Lib Tables	Stoled Flocedules						
Systems	e 🔄 Vers	INVO-CNL0491450(SQLE0PESS)(Databases)ATIS(Programmability(Reved Proteedures 24 Dees(s)						
Image: Programmedia:         Name         Stream         Orient             Protein Structures:         System Structures:	e 🛄 Synonyms							
New         New         Dealer           •         System Stared Procedures         •           •         -         -         System Stared Procedures         •           •         -         -         System Stared Procedures         •           •         -	<ul> <li>III Programmability</li> </ul>		and the second se	and a second sec				
System Stare Hoodwares         System Stare Hoodwares         doi:         11/17/2019           Columbia Alting Addition         Alting Addition         doi:         07/2019           Columbia Alting Columbia         Alting Addition         doi:         00/2019           Columbia Alting Columbia         Alting Columbia         doi:         00/2019           Columbia Alting Columbia         Alting Columbia         doi:         00/2019           Columbia Alting Columbia         Alting Columbia         doi:         20/2019           Columbia Alting Columbia <td< td=""><td><ul> <li>B Stored Procedurer</li> </ul></td><td>Name.</td><td>Schema</td><td>Created</td><td></td></td<>	<ul> <li>B Stored Procedurer</li> </ul>	Name.	Schema	Created				
Image: Market of the second of the	* 🛄 System Stored Procedures	System Stored Procedures						
m         dbs.A13_bandbeforeStryTendor         A15_bandbeforeStryTendor         A15_bandbeforeStryTendor </td <td>8 III db0.A113_Ad0.6er</td> <td>TIS_Add.iver</td> <td>dbo</td> <td>31/07/2010</td> <td></td>	8 III db0.A113_Ad0.6er	TIS_Add.iver	dbo	31/07/2010				
B         B<         B<<         B<         B<         B<	<ul> <li>B dbc.ATIS_CaculateArmainformation</li> </ul>	ATTS_CalculateArrivalInformation	doo	07/08/2010				
a         dbs.413         ATS_balactorie/on         dbs         AU(7)000           a         dbs.413         Sectorialis         dbs         AU(7)000           a         dbs.413         Sectorialis         dbs         OU(2000)           a         dbs.413         Sectorialis         dbs         OU(2000)           a         dbs.413         Sectorialis         dbs         OU(2000)           a         dbs.013         Sectorialis         dbs         Sectorialis           a         dbs.0133         Sectorialis <td><ul> <li>III doc.Alts_Georgeoup</li> <li>III doc.Alts_Georgeoup</li> </ul></td> <td>ATIS_CreatePickup</td> <td>000</td> <td>02/06/2010</td> <td></td>	<ul> <li>III doc.Alts_Georgeoup</li> <li>III doc.Alts_Georgeoup</li> </ul>	ATIS_CreatePickup	000	02/06/2010				
Image: Note: Sector S	<ul> <li>E do ATS_place.septro</li> <li>E do ATS_place.septro</li> </ul>	ATIS_DeleteUserInfo	dbo	30/07/2010				
Bits         ArtS_schebuschets         ArtS_schebuschets         ArtS_schebuschets           4         Abs.211_schebuschets         ArtS_schebuschets         Abs         00000000           4         Abs.211_schebuschets         Abs         00000000           4         Abs.211_schebuschets         Abs         0000000           4         Abs.211_schebuschets         Abs         2007000           4         Abs.211_schebuschets         ArtS_schebuschets         Abs         2007000           4         Abs.211_schebuschets         ArtS_schebuschets         Abs         2007000           5         Abs.211_schebuschets         ArtS_schebuschets         Abs         2007000           5         Abs.211_schebuschets         ArtS_schebuschets         Abs         2007000           6         Abs.211_schebuschebusches         ArtS_schebuschets <td< td=""><td><ul> <li>a an and a star catter and a start</li> </ul></td><td>ATIS_GetCoordinates</td><td>odbo</td><td>08/08/2010</td><td></td></td<>	<ul> <li>a an and a star catter and a start</li> </ul>	ATIS_GetCoordinates	odbo	08/08/2010				
Image: Construction of the second o	a III des ATTS Getildung	ATIS_GetPickupDetails	abo	09/08/2010				
*     do. A112_settide     A113_settidue     do.     0008/000       *     do. A113_settidue     do.     0008/000       *     do. A113_settidue     do.     0008/000       *     do. A113_settidue     do.     20/07/000	<ul> <li>An ATT GetRidun/Rahm</li> </ul>	ATIS_GetPictups	dbo	03/06/2010				
•     • <td># E do. ATIS GetRole</td> <td>ATIS_GetPickupStatus</td> <td>do</td> <td>09/08/2010</td> <td></td>	# E do. ATIS GetRole	ATIS_GetPickupStatus	do	09/08/2010				
a data Altij juerdingen     Altij deritatien     aba     26(07)000       8     data Altij juerdingen     Altij deritatien     aba     26(07)000       8     data Altij juerdingen     aba     10(07)000       8     data Altij juerdingen     aba     10(07)000       8     data Altij juerdingen     aba     10(07)000       8     data Altij juerdingen     aba     20(07)000       9     data Altij juerdingen     aba     20(07)000	+ III doi: ATIS GetRoutes	C ATIS_GetRole	dao	28/07/2010				
8     db. ATIS_instruction     ATIS deficies     db.     M0070300       8     db. ATIS_instruction     db.     M0070300       8     db. ATIS_instruction     db.     M0070300       8     db. ATIS_instruction     db.     M0070300       9     db. ATIS_instruction     db.     M0070300 <t< td=""><td>* 🖾 doc.ATIS GetStops</td><td>ATIS_GetRoutes</td><td>dbo</td><td>25/07/2010</td><td></td></t<>	* 🖾 doc.ATIS GetStops	ATIS_GetRoutes	dbo	25/07/2010				
a do. ATIS_incentificacy@initial     ATIS_suretificacy@initial     abs     140060009       a do. ATIS_incentificacy@initial.exes     ATIS_suretificacy@initial.exes     abs     2907000       a do. ATIS_interestation     ATIS_suretificacy@initial.exes     abs     2007000       a do. ATIS_inationerinitial.exes     ATIS_suretificacy@initial.exes     abs     2007000       a do. ATIS_instructure     ATIS_instructure     abs     20070000       a do. ATIS_instructure     ATIS_instructure     abs     20070000       a do. ATIS_instructure     ATIS_instructure     abs     20070000       a do. ATIS_instructure<	8 🖾 dbc.ATIS InsertDate	ATIS_GetStops	dbo	25/07/2010				
do.A132     Schoolskikers     A133_sourceskikers     A04     20050209       a     do.A132_shire-oldukters     A04     20070209       a     do.A132_shire-oldukters     A05     20070209       a     do.A132_shire-oldukters     A04     20070209       a     do.A132_shire-oldukters     A05     20070209       a     do.A132_shire-oldukters     A04     20070209       a     do.A132_shire-olduktershire-blockershire     A04     20070209       a     do.A132_shire-olduktershire-blockershire-blo	H 🖾 doc.ATIS InsertNessageDetais	ATIS_InsertData	dao	14/05/2010				
	# 🖾 dbo.ATIS_RetrieveAllUsers	C ATIS_InsertMessageDetals	diso	29/05/2010				
a         do. A132, juticeticity         A135, juticeticity         doi:         2800/2019           b         do. A132, juticeticity         A135, juticeticity         doi:         2800/2019           c         do: A132, juticeticity         A135, juticeticity         doi:         0708/2019           c         do: A132, juticeticity         A135, juticeticity         doi:         0708/2019           c         do: A132, juticeticity         A135, juticeticity         doi:         0708/2019           c         Doi:         A135, juticeticity         doi:         0708/2019           c         Doi:         A135, juticity         doi:	# 🖾 dbc.ATI5_RetrieveData	ATTS RetrieveAll.ters	disc	29/07/2010				
Image: Construction of the state o	# 🖾 dbo.A715_RetrieveColocitysers	ATIS RetrieveData	dao	20/07/2010				
Image: Construction of the co	# 🔟 dbo.ATIS_RetrieveUserInfo	ATIS RetrieveOnineUsers	dbo	20/07/2010				
a     do. AT32     performance     AT33     performance     doi:     At307000       b     a     doi:     AT33     performance     doi:     At307000       b     a     doi:     AT35     performance     doi:     At3070000       b     a     doi:     AT35     performance     doi:     At307000       b     a     doi:     AT35     performance     doi:     At307000       b     At35     machine     doi:     At307000     doi:     At307000       c     At35     performance     doi:     At307000     doi:     At307000       c     At35     performance     At33     performance     doi:     At307000       c     At35     performance     At33     performance     doi:     At3070000       c     At35     performance     At33     performan	🗄 🛄 dbc.ATIS_RetrievelieticietnfoliyRouteAndTitop	Artis Retrievel.ter/to/o	dea	30/07/2010				
	# 🖾 dzo.AT15_RetrieveWehicieInfoByUser	ATIS Retrieverielwielnfolk/Roxtaled/Rop	dbo	26/07/2010				
a     do. A12_instructionstitut     A13_instructionstitut     do.     01/062030       a     Control A13_instructionstitut     do.     07/062030       a     Control A13_instructionstitut     do.     07/072030       a     A15_instructionstitut     do.     00/07/0300       a     Distance Instructions     a.     00/07/0310       a     A15_instructionstitut     do.     00/07/0310	# 🔛 dbo.ATI5_SendMessage	ATTS Retrieververheiteltefollytiker	obs	29/07/2010				
ATIS_fores/basefile     A	🐵 🔛 dbc.A115_TraceClosestBus	III ATTS SendMessage	dbo	01/08/2010				
Control of ATTS Subdationer of a ATTS Subdationer of a and a appropriate a second and	😸 🛄 dbc.ATI5_JapdateUserInfo	III ATTS TraceClosestflux	dbo.	07/08/2010				
A rations     A rations	# Do.ATIS_ValidateUser	arts Undetellowinfo	dho	30/07/2010				
Duducer togens	+ 🔄 Punctions	ATTS ValidateLiser	dbo	09/07/2010				
* ADERCES * FORK * Defaults * Def	<ul> <li>Batabase Triggers</li> </ul>							
A g Tree a Ruler a Golden a Golden C C C C C C C C C C C C C C C C C C C	a 🛄 Assentcies							
a men a Detads a Server Bolar B C A	in the report							
* to verso * To ve	<ul> <li>US Policies</li> </ul>							
	<ul> <li>In terminal leaders</li> </ul>	Q.						
	X							
		and the second second second second second						

FIGURE 4-18 Database Objects (Stored Procedures)

## 4.4 Dijkstra Algorithm

Dijkstra Algorithm was developed in Microsoft .Net (C#). It starts form the source node and checks the adjacent nodes attach to it having least distance between them. It keeps on checking adjacent nodes and eventually reaches the destination node. This way, it ensures to calculate the most optimum path.

## 4.4.1 Functionality / Example

Suppose courier is at position 'O' and the customer location is at 'T'. Google APIs give intermediate nodes A, B, C, D, E and F. The APDIS system performs the following steps to calculate the shortest route between source (courier's truck) and destination (customer's location).

- 1. Calls web service to read pickup job from the database.
- 2. Knows the destination address since pickup job has customer's address and other details.
- 3. Performs reverse geo-coding to convert customer's physical address to geo-location. This is important because digital maps can display a point only based on geo-location.
- 4. Calls web service to get a list of all couriers logged in this region.
- 5. By now, it has geo-location of both source and destination. Calls GIS APIs to get all the nodes between source and destination. Source(s) are the couriers currently logged in the system and serving in this region.
- 6. Calls geocoder web service to get the distance between each node. TABLE 4-1 shows the nodes and their distance from the adjacent nodes.

Nodes	Distance in KM	Nodes	Distance in KM	Nodes	Distance in KM
$0 \rightarrow A$	2	$A \rightarrow F$	12	$D \rightarrow E$	1
$0 \rightarrow B$	5	$B \rightarrow C$	1	$D \rightarrow T$	5
$0 \rightarrow C$	4	$B \rightarrow D$	4	E → T	7
$A \rightarrow B$	2	$B \rightarrow E$	3	$F \rightarrow T$	3
$A \rightarrow D$	7	$C \rightarrow E$	4		

**TABLE 4-1: Distance between the nodes** 

7. Dijkstra's Algorithm executes to calculate the closest courier to the customer's location. It then calculates the shortest route between the courier and the customer.

FIGURES 4-19A - 4-19H show the step by step functionality of the algorithm. Nodes in blue color indicates the unprocessed nodes, green color nodes indicate the processed nodes while red nodes indicate the adjacent nodes to the current source node.

Highlight all the nodes. Start from source node (O). Adjacent nodes to the source (O) are now (A), (B), and (C) and highlighted in red color. Check the distance between source and adjacent nodes (A, B, C).



FIGURE 4-19A: Dijkstra Execution (Step - 1)

Selects the closest node (A) based on the distance. The closest node (A) becomes source node now. Adjacent nodes to the original source (O) and the current source (A) are now (B), (C), (D), (F) and they are highlighted in red color. Check the distance between the current source node (A) and adjacent nodes (B, D, and F). Check the distance between previous source node (O) and the current node (B) (O  $\rightarrow$  C  $\rightarrow$  B or O  $\rightarrow$  B). Selects the closest node to the current source A, verify the distance is less than the distance from previous source (O) to the node (B).



FIGURE 4-20B: Dijkstra Execution (Step - 2)

Closest node (B) becomes the current source now while nodes (O and A) are previous source nodes. Adjacent nodes to the previous source (A) and the current source (B) are now (C), (D), (E), and (F) and they are highlighted in red color Check the distance between the current source node (B) and adjacent nodes (C, D, E and F). Check the distance between previous source node (O and A) and the current node.



FIGURE 4-21C: Dijkstra Execution (Step - 3)

Check the distance between the latest sources to the adjacent nodes. Check the distance between the previous source (O, A) to current adjacent nodes. Compare and verify the closest distance. Select the distance which is closest to the current source (B) or the previous sources (O, A). Green nodes indicate the locked nodes and the system will not consider them again. Adjacent nodes now are (D, E, and F) and are highlighted in red color.



FIGURE 4-22D: Dijkstra Execution (Step - 4)

Check the distance between the current sources to the unsolved nodes. Check the distance between the previous source (O, A, B) to the unsolved nodes. Compare and verify the closest distance. Select the distance which is closest to the current source (E) or the previous sources (O, A, B).



FIGURE 4-23E: Dijkstra Execution (Step - 5)

Select the node which is closest to the current source (E) or the previous sources (O, A, B). Since node (D) is closest so it will be the next source. Now there are only two unsolved nodes (F, T).



FIGURE 4-24F: Dijkstra Execution (Step - 6)

Select the node which is closest to the current source (D) or the previous sources (O, A, B, E). Since node (T) is closest so it will be the next source.



FIGURE 4-25G: Dijkstra Execution (Step - 7)

System finds two optimal paths having the same distance of 13. First optimal path is shown in the FIGURE 4-19G and it follows the nodes  $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$ .

Second optimal path follows  $O \rightarrow A \rightarrow B \rightarrow E \rightarrow D \rightarrow T$  and is shown in the FIGURE 4-19H. The system now has two options to accept an optimal path between the courier and the customer. It will accept the first optimal path ( $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$ ) since it has less number of nodes and obviously will be quicker because of fewer intersections.



FIGURE 4-26H: Dijkstra Execution (Step - 8)

## 5 Conclusions and Future Directions

Integration of GPS and GIS technologies in the existing system can solve the current problems of adhoc pickups. Instead of static routing, the new process performs these steps to dynamically route a call to the courier.



Server broadcasts message to all the connected clients (couriers) serving the customer region and requests for geo-location.

- Client application sends current geo-location of the courier.
- Server performs reverse geo-coding on the customer's address and converts to geo-coordinates.
- Server executes the algorithm to calculate the current distance between customer and courier.
- Server selects the nearest courier and sends pickup job to him/her.

Overall advantages of the system are:

- 1. It provides improved, fast and reliable service to the customers.
- 2. It reduces the average mileage per courier which in turns reduces the fuel cost as well as vehicle maintenance.
- 3. The user can know the estimated time a courier comes to pickup/deliver a package.

Ц

- 4. It resolves the customer disputes related to arrival time, service duration and service location.
- 5. By knowing the exact location of a vehicle, pinpointed on a map, remote staff that is lost can be better helped. This helps the employee stay on schedule and not have to over speed or work over time to recover lost time.

#### Future work includes...

There is always a room to improve each and every process. This project is just a prototype which needs a lot of customization and enhancements. Some enhancements and/or modifications in the near future are:

#### 1. Courier's delivery Planner (Organizer)

A courier goes to deliver the package in the morning. A packages delivery organizer needs to be implemented to show the shortest routes among the addresses (normally 50-200).

#### 2. Text Messaging

Sometimes, a customer doesn't have internet access and he/she needs to know the time a courier will be at his/her doorstep. Text messaging service is required to send a text message to the customer as soon as the courier started driving to his/her address.

3. Adaptive /Real-time Algorithm

Sometimes, the streets/highways are closed or blocked because of maintenance/construction or traffic congestion. In this scenario, the algorithm should be smart enough to provide alternate route.

#### 4. Data Security

Microsoft .Net contains rich functionality of implementing security. Data encryption /decryption will be embedded using Cryptography classes.

## 6 Bibliography

- 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.
- "What is GPS and how does it work" <u>http://www.vehiclelocationsystem.com/AVL%20Explained.htm</u>. Accessed August 25, 2010.
- 3. "What is GPS and how does it work" http://gis.ednet.ns.ca/GPS/how\_does\_gps\_work.htm. Accessed August 10, 2010.
- 4. B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, et al. "Global Positioning System: Theory and Practice". Springer: *New York*, NY, U.S.A., 1997.
- 5. "Glenn Baddeley GPS NMEA sentence information" http://home.pacific.net.au/~gnb/gps/nmea.html. Accessed August 15, 2010.
- 6. 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
- 7. 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
- 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
- 9. J. S. Greenfeld, "Automatic vehicle location (AVL) for transit operation" 10th Mediterranean Electrotechnical Conference, May 29-31, 2000, Cyprus (Peer)
- O. Aloquili1, A. Elbanna, A. Al-Azizi1, "Automatic vehicle location tracking system based on GIS environment" *IET Softw*, 2009, Vol. 3, Issue. 4, pp. 255– 263.
- 11. M. A. Elahi, Y. A. Malkani, and M. Fraz, "Design and implementation of real time vehicle tracking system", 2<sup>nd</sup> International conference on Computer, Control and Communication, Karachi, Feb. 17-18, 2009, pp. 1-5.

- 12. 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.*
- 13. J. Zheng, Y. Wang, and N. L. Nihan, "*Tracking Vehicles with GPS*", <u>http://www.uwstarlab.org/STARLab\_Papers/2003\_GPSError-itsa.pdf</u>. Accessed August 10, 2010.
- 14. S. Christian Jensen and T. Dalia. "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.
- 15. Vehicle Tracking Systems Overview (2008) <u>http://www.roseindia.net/technology/vehicletracking/Vehicle trackingSystems.shtml</u>. Accessed August 22, 2010
- 16. Geographic Information System, <u>http://egsc.usgs.gov/isb/pubs/gis\_poster/</u>. Accessed August 30, 2010
- 17. <u>http://en.wikipedia.org</u>, <u>http://en.wikipedia.org/wiki/Automatic\_vehicle\_location#Application\_with\_vehicles</u>. Accessed August 30, 2010
- 18. B. M. Waxman, "Routing of Multipoint Connections", *IEEE Journal on Select. Areas Communication*, vol. 6, pp. 1617-1622, Dec.1988.
- 19. W. J. Dally, "Virtual Channel Flow Control", *IEEE Transcription on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194-205, 1992.
- 20. J. Duato, S. Yalamanchili, and L. Ni, "Interconnection Network: An Engineering Approach" *IEEE Computer Society Press*, 1997.
- 21. C. L. Seitz, "The Hypercube Communication Chip", Display File 5182: DF: 85, *Department of Computer Science, California Institute of Technology*, Mar. 1985, pp. 119-140.
- 22. Y. Boura, C. R. Das, and T. M. Jacob, "A Performance Model For Adaptive Routing in Hypercubes", *Proc. from Intel Workshop on Parallel Processing*, Dec. 1994, pp. 11-16.
- 23. J. Duato, "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Routing Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no.12, Dec. 1993, pp. 1320-1331.

- 24. X. Lin, P. K. Mckinley, and L. M. Lin, "The Message Flow Model for Routing in Wormhole-Routed Networks", *Proc. from Intel Conference on Parallel Processing*, 1993, pp. 294-297.
- 25. D. H. Linder and J. C. Harden, "An Adaptive and Fault Tolerant Wormhole Routing Strategy for K-ary n-cubes", *IEEE Transactions on Computers*, vol. 40, no. 1, Jan. 1991, pp. 2-12.
- 26. C. Y. Chou, D. J. Guan, and K. L. Wang, "A Dynamic Fault-Tolerant Message Routing Algorithm for Double-Loop Networks", *Information Processing Letters*, vol. 70, 1999, pp. 259-264.
- 27. S. Tanenbaum, "Computer Networking", Prentice Hall, 2002.
- 28. W. Stallings, "Data & Computer Communication", Prentice Hall, 2007
- 29. J. F. Kurose, and Keith W. Ross, "Computer Networking", Addison Wesley, 2<sup>nd</sup> Edition, 2002
- 30. R. T. Stevens, "Graphics Programming in C", BPB Publications, 1st Edition
- 31. Overview of GIS, <u>http://www.gisdevelopment.net/tutorials/tuman006.htm</u>. Accessed August 30, 2010
- 32. GIS Applications, <u>http://centrin.net.id/~agul/gis\_apl.html</u>. Accessed August 30, 2010
- 33. Geographical Coordinates System, <u>http://en.wikipedia.org/wiki/Geographic\_coordinate\_system</u>. Accessed August 30, 2010
- 34. Geocode, http://en.wikipedia.org/wiki/Geocode. Accessed August 30, 2010
- 35. Reverse Geocoding, <u>http://en.wikipedia.org/wiki/Reverse\_geocoding</u>. Accessed September 01, 2010
- 36. J. B. Pick, "Geo-business GIS in the digital organization", John Wiley and Sons, 2008, pp. 46
## 7 Appendix

## 7.1 Abbreviation List

ACK	ACKNOWLEDGEMENT			
APDIS	AUTOMATIC PICKUP & DELIVERY INFORMATION SYSTEM			
API	APPLICATION PROGRAMMING INTERFACE			
AVL	AUTOMATIC VEHICLE LOCATION			
CCD	CALL CENTER DEPARTMENT			
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			
P&D	PICKUP & DELIVERY			
RDBMS	RELATIONAL DATABASE MANAGEMENT SYSTEM			
SMS	SHORT MESSAGING SERVICE			
SOAP	SIMPLE OBJECT ACCESS PROTOCOL			
SQL	STANDARD QUERY LANGUAGE			
SYN	SYNCHRONIZATION			
TCP/IP	TRANSMISSION CONTROL PROTOCOL			
T-SQL	TRANSACT- SQL			
	UNIVERSAL DESCRIPTION, DISCOVERY AND			
UDDI	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] [create\_date] [DATETIME] NOT NULL, NOT NULL DEFAULT (getdate()), [user\_id] [INT] NOT NULL, [alert\_id] [INT] [update\_date] [DATETIME] NOT NULL, NOT NULL, CONSTRAINT [PK\_av1 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 ١ 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, [FLOAT] [FLOAT] [PeriodType] NULL, [StopId] [StopName] NULL, [Stophu][FIGH]NOLL,[StopName][VARCHAR] (255) COLLATE Latin1\_General\_CI\_AS NULL,[ArrivalTime][DATETIME][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 NOT NULL,

 [Address]
 [VARCHAR](50) COLLATE Latin1\_General\_CI\_AS NOT NULL,

 [Address]
 [VARCHAR](50) 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,

 [Address]
 [VARCHAR](50) COLLATE Latin1\_General\_CI\_AS NOT NULL,

 [City]
 [VARCHAR](50) COLLATE Latin1\_General\_CI\_AS NOT NULL,

 [Cip]
 [VARCHAR](15) COLLATE Latin1\_General\_CI\_AS NOT NULL,

 [CellPhone]
 [VARCHAR](15) COLLATE Latin1\_General\_CI\_AS NULL

 (CellPhone]
 [VARCHAR](15) COLLATE Latin1\_General\_CI\_AS NULL

) ON [PRIMARY] CREATE TABLE [dbo].[DeviceInfo]( [DeviceId] [INT] [DeviceName] [VARCHAR] (10) COLLATE Latin1\_General\_CI\_AS NOT NULL,

CONSTRAINT [PK DeviceInfo] PRIMARY KEY CLUSTERED 1 [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. [VARCHAR] (10) COLLATE [TowerID] 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)), [VARCHAR] (1) COLLATE [GPSStatus] SQL\_Latin1\_General\_CP1\_CI\_AS NULL, [GPSTime] [Longitude] [Latitude] [DATETIME] NULL, [FLOAT] NULL, [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)), [VARCHAR] (50) COLLATE Latin1 General CI AS [UserId] NULL CONSTRAINT [DF\_jobs\_user\_id\_08EA5793] DEFAULT ((0)), [DispatcherId] [VARCHAR](25) COLLATE Latin1 [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)), NOT NULL CONSTRAINT [DATETIME] [JobDate] [DF\_jobs\_job\_date\_OBC6C43E] DEFAULT (getdate()), [CreateDate] [DATETIME] NOT NULL CONSTRAINT [DF jobs create dat OCBAE877] DEFAULT (getdate()),

[StatusChangeDate] [DATETIME] NOT NULL CONSTRAINT [DF\_jobs\_status\_cha\_ODAF0CB0] 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 [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]( NOT NULL, [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]( [VARCHAR] (20) COLLATE Latin1 General CI AS NOT [RouteId] NULL CONSTRAINT [DF Schedules RouteId] DEFAULT ('81N'), [VARCHAR](20) COLLATE Latin1\_General CI AS NOT [StopId] NULL,

[ScheduleType] [SchedulySubType] [ArrivalTime] [StartTime]	[INT] [INT] NOT NUL [VARCHAR](50) [VARCH	NOT NULL, L, COLLATE Latin1_G AR](50) COLLATE L;	eneral_CI_AS atin1 Genera	NOT NULL, 1 CI AS NOT
) ON [PRIMARY]				
CREATE TABLE [dbo].[Sche [ScheduleSubType] [Description] ) ON [PRIMARY]	duleSubType]( [INT] [VARCHAR](50)	NOT NULL, COLLATE Latin1_Ge	neral_CI_AS	NOT NULL
CREATE TABLE [dbo].[Sched [ScheduleType] [IN [Description] [VA CONSTRAINT [PK_Schedule2 ( [ScheduleType] ASC	duleTypes](  T] RCHAR](50) [ypes] PRIMARY K	NOT NULL, COLLATE Latin1_Ge EY CLUSTERED	neral_CI_AS	NOT NULL,
)WITH (IGNORE_DUP_KEY = ( ) ON [PRIMARY]	OFF) ON [PRIMARY	1		
CREATE TABLE [dbo].[SendH [group_id] [send_email_id] [I [email_address] [V CONSTRAINT [PK_avl_send] ( [group_id] ASC, [email_address] AS )WITH (IGNORE_DUP_KEY = C ) ON [PRIMARY]	Cmail]( [INT] NOT NUL] NT] IDENTITY(1, ARCHAR](100) COI email] PRIMARY C C FFF) ON [PRIMARY	], NOT NULL, JLATE Latin1_Gener KEY CLUSTERED	al_CI_AS NOT	f NULL,
CREATE TABLE [dbo].[Stops [RouteId] [VARC [StopId] [VARC [Details] [VARC [Latitude] [DECI [Longitude] [DECIM [Distance] [DECI ) ON [PRIMARY]	]( HAR](20)( HAR](20)( HAR](50)( MAL](18, 10) AL](18, 10) MAL](18, 0)	COLLATE Latin1_Ger COLLATE Latin1_Ger COLLATE Latin1_Ger NOT NULL, NOT NULL, NULL	neral_CI_AS n neral_CI_AS n neral_CI_AS n	NOT NULL, NOT NULL, NOT NULL,
CREATE TABLE [dbo].[Stops [RouteId] [VARC [StopId] [INT] [SegmentId] [INT] [Location] [VARC [Latitude] [DECI [Longitude] [DECIM [Distance] [INT] ) ON [PRIMARY]	Segments]( HAR](20)COLLATE N HAR](200) MAL](18, 8)NOT AL](18, 8)NOT N	Latin1_General_C: NOT NULL, NOT NULL, COLLATE Latin1_Gen NULL, ULL, NOT NULL	:_AS NOT NUL] eral_CI_AS N	L, NOT NULL,

CREATE TABLE [dbo].[Users]( [UserId] [VARCHAR] (25) COLLATE Latin1\_General\_CI\_AS NOT NULL, [Password] [VARCHAR] (10) COLLATE Latin1 General CI AS NOT HULL, [Home\_Phone] [VARCHAR] (15) COLLATE SQL\_LAtin1 General CP1 CI AS NOT HULL, [MobilePhone] [VARCHAR] (15) COLLATE SQL\_LAtin1 General CP1 CI AS NOT HULL, [Address1] [CHARLAS (15) COLLATE SQL\_LATIN1 General CP1 CI AS NOT HULL, [Password] [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] ( [VARCHAR] (50) COLLATE Latin1\_General\_CI\_AS NOT [VehicleId] NULL, [Longitude] [DECIMAL] (18, 8) NOT NULL, [Latitude] [DECIMAL] (18, 8) NOT NULL, [Speed] NOT MULL, [INT] [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] がりした (TINYINT) ) ON [PRIMARY] CREATE TABLE [dbo]. [WirelessStatusType] ( NOT NULL, [WirelessStatusId] [TINYINT] [Description] [VARCHAR] (50) COLLATE Latin1\_General\_CI\_AS NOT NULL ) ON [PRIMARY]