



Palestine Polytechnic University

College of Engineering

Electrical Engineering Department

College of Information Technology and Computer Engineering

Computer Engineering Department

Digital Roads – Observation and Serving

Research submitted to fulfill the requirements of the seminar course in
Communications and Computer Engineering

Prepared by:

Yazeed Aljuneidi

Samer Alqadi

Ala'a Manasrah

Husam Sa'ed

Haneen Manasrah

Supervisors

Eng. Ayman Wazwaz

Dr. Amal Al-Dweik

2013-2014

Acknowledgment

The team advances great thanks to our great praised Allah who innovates our souls, lights our brains and illuminates the road of tomorrow ...

The team advances deep thanks to their dear supervisors Eng. Ayman Wazwaz and Dr. Amal Al-Dweik Who have granted their support, orientation, guidance and advices...

The team advances special thanks ...to our, families, dear teachers, lecturers, friends, and to all who contributed in the accomplishment of this project ...

We can only say for their gratitude... Thank You...!"

Dedication

To those who have dedicated their lives giving us the hope for tomorrow..., to those who have lighted our knowledge road... to our dear fathers...

To those who granted us the tenderness and taught us the patience..., to those whose hearts and souls have given us the protection and power to become what we are now..., to our beloved mothers...

To those who have contributed in the knowledge and science process..., to those whose thoughts enriched our knowledge towards a prosperous future ... to our dear instructors...

To those who sacrificed their lives and dignified the land with their bodies...to our model ...to the blessed Martyrs 'souls...

To our successor ... to the future generations...

To our soul mates who supported us ...To those who are always there when we need them ... to our friends and lovers... To all of these persons we would like to dedicate this project...

Abstract

Year by year, the street problems increase linearly with the growth of people. Accidents, speed violation, street crimes, time waste to reach your goal through streets ...etc. Our problem speak about the street status and how can be solved.

The solution to street problem come into two parts, either automatically or manually. Accidents and speed violations solved in automatic way, using sensors to measures the accurate value and compare it, then produce an action depends on the output value.

The system can notify policeman, driver and the ambulance when the violation occur and display the specific location on map. In addition, the shortest path between two points to minimize the time taken to reach the destination. The max time to notify the three parts is 2-3 seconds.

The system give us an accurate result for many cases. Cars locations, accidents, violations identified accurately with 1 – 10 meter error. Also the system work in the tunnels and other complicated street.

المخلص

يهدف هذا المشروع إلى بناء نظام مفيد ، والذي تم استخدامه لمراقبة ورصد حركة المرور في الشوارع، السيطرة على مشكلة سرقة السيارات، التخفيف من مشكلة حوادث السيارات و تسجيل معظم المخالفات المرورية التي يمكن أن تحدث من قبل السائقين على الطرق.

يتم الحصول على المعلومات من الشارع بطرق مختلفة منها ما يعمل بنظام آلي وغيرها اليدوي. من المخالفات التي يتم إكتشافها بشكل آلي ، تجاوز حد السرعة ، ومخالفة إشارة المرور ، وحوادث السير. أما بعض المخالفات فقد تكون مصدرها جهاز الشرطي ، بحيث يقوم بكتابة المخالفة وإرسالها إلى الخادم وبعدها تحول بشكل تلقائي إلى السائق.

في الخادم ، تم بناء قاعدة بيانات تتضمن معلومات هامة ، تقوم بحفظ جميع المعلومات عن السيارات و سائقها هناك ، وبعض المعلومات الأخرى حول الإسعاف ، الإطفاء و مواقع مراكز الشرطة وأرقام التواصل معهم.

تم برمجة النظام المركزي باستخدام لغة C# و قاعدة بيانات SQL تستخدم لرصد التجاوزات المرورية ، حساب حركة المرور ، رصد وقوع الحادث ، تتبع أي سيارة في نطاق تغطية شبكتنا والتحكم في إشارات المرور إذا لزم الأمر. بعض المعلومات تحول إلى مركز الشرطة أو إلى السائق أو رجال الشرطة على اجهزتهم النقالة من خلال شبكة GSM ، و يتم عرض هذه المعلومات كخريطة على جهاز الاندرويد مع سائق السيارة أو شرطي الدورية أو الإسعاف.

Table of contents

Acknowledgment	I
Dedication	II
Abstract	III
Table of contents	V
Table of figures.....	XIII
Table of tables.....	XVI
Chapter One	1
Introduction	
1.1 Overview	2
1.2 Project Idea Description	2
1.3 Motivations	3
1.4 Project Scope.....	4
1.4.1 Context	4
1.4.2 Information Objectives.....	4
1.4.3 Function and Performance.....	5
1.4.4 Assumptions and requirements	5
1.5 Summary	6
Chapter Two	7
Literature Review	
2.1 Overview	8
2.2 Theoretical background	8
2.2.1.1 Zigbee technology	8
2.2.1.1.1 Name of Zigbee	9
2.2.1.1.2 Zigbee Stack.....	9
2.2.1.1.3 Types of Zigbee devices:	11

2.2.1.2	GPS :	12
2.2.1.2.1	Space Segment	12
2.2.1.2.2	GPS control	13
2.2.1.2.3	User segment	14
2.2.1.2.4	How GPS works	14
2.2.1.2.5	Equations for location determination:	15
2.2.1.3	GSM	16
2.2.1.3.1	GSM system overview	17
2.2.1.3.2	GSM basic	17
2.2.1.4	GPRS service	19
2.2.1.4.1	Definition	19
2.2.1.4.2	GPRS and packet data in GSM	20
2.2.1.4.3	GPRS Network Elements	22
2.2.2	Software Theory	22
2.2.2.1	C# Programming Language	23
2.2.2.2	Microsoft SQL Database	23
2.2.2.3	Android (Operating System)	24
2.2.2.4	SQLite Database	25
2.3	Literature review	25
2.3.1	Traffic Message Channel (TMC)	26
2.3.1.1	Overview	26
2.3.1.2	Advantages:	27
2.3.1.3	Disadvantages:	27
2.3.2	Vehicle Information and Communication System(VICS)	27
2.3.2.1	Overview	27
2.3.2.2	Advantages:	28
2.3.2.3	Disadvantages:	28
2.3.3	Intelligent traffic with automatic violation system	28
2.3.3.1	Overview	28
2.3.3.2	Advantages:	29
2.3.3.3	Disadvantages:	29

2.3.4 Innovative congestion control system for ambulance using ZigBee.....	29
2.3.4.1 Overview	29
2.3.4.2 Advantages.....	30
2.3.4.3 Disadvantages:	30
2.4 Summary	30

Chapter Three..... 32

Project Management Plan

3.1 Overview	33
3.2 Project Management Plan.....	33
3.2.1 Project Tasks.....	33
3.2.2 Risks.....	36
3.2.3 Project Resources and Estimated Cost.....	39
3.2.4 Time Estimation.....	40
3.2.4.1 Timeline Table	40
3.2.4.2 Activity Network.....	41
3.2.4.3 Gantt Chart.....	42
3.3 Project Methodology	44
3.3.1 Options.....	44
3.3.2 Analysis of each Option.....	44
3.3.2.1 Java Programming Language.....	44
3.3.2.2 Difference between C# and Java.....	46
3.3.2.3 Oracle Database	47
3.3.2.3.1 Database Storage Structures.....	47
3.3.2.4 Difference between Oracle and MySQL.....	48
3.3.2.5 Microcontrollers.....	50
3.3.2.5.1 PIC.....	50
3.3.2.5.2 Arduino.....	50
3.3.2.6 Data Transfer.....	51
3.3.2.6.1 SMS (Short Message Service).....	51

3.3.2.6.2	GPRS (General packet radio service)	51
3.3.2.7	Wireless Communication	52
3.3.2.7.1	Wi-Fi	52
3.3.2.7.2	Bluetooth	53
3.3.2.7.3	Zigbee	54
3.4	Project Components	55
3.4.1	Software Component	55
3.4.1.1	C# Programming Language	55
3.4.1.1.1	Inheritance in C#	55
3.4.1.1.2	Interfaces	56
3.4.1.1.3	Garbage Collector	56
3.4.1.2	Microsoft SQL Database	57
3.4.1.2.1	Architecture	57
3.4.1.2.2	Data storage	57
3.4.1.3	Android (Operating System)	58
3.4.1.4	SQLite Database	59
3.4.2	Hardware Components	59
3.4.2.1	Zigbee	59
3.4.2.2	GPS	60
3.4.2.3	GPRS modem	61
3.4.2.4	Arduino	62
3.4.2.5	ZigBee and other competitors	63
3.4.2.6	Why we chose ZigBee:	63
3.5	Summary	64

Chapter Four 65

Software Requirement Specification

4.1	Overview	66
4.2	Requirement Description	66
4.2.1	Actors	66

4.2.1.1	Administrator	66
4.2.1.2	Policeman	66
4.2.1.3	Driver	66
4.2.2	Usage Scenario	67
4.2.2.1	Administrator	67
4.2.2.1.1	Logging on and access the system data	67
4.2.2.1.2	Maintain the system	68
4.2.2.1.3	Monitor the system	68
4.2.2.2	Policeman	69
4.2.2.2.1	Tracing stolen car on a map	69
4.2.2.2.2	Receive a messages	70
4.2.2.2.3	Check driving license	71
4.2.2.2.4	Write violation report for specific car	72
4.2.2.3	Driver	73
4.2.2.3.1	Monitor the street state	73
4.2.2.3.2	Receive a violation message	74
4.2.3	Use Case Diagram	75
4.3	CRC (Class Responsibilities Collaborator) Modeling	76
4.3.1	Server Classes	76
4.3.1.1	Admin Class	76
4.3.1.2	GSM/Serial Class	77
4.3.1.3	Violation Class	77
4.3.1.4	Car Class	78
4.3.1.5	Message Class	78
4.3.1.6	Driver Class	79
4.3.2	Driver Android Application Classes	79
4.3.2.1	Map Class	79
4.3.2.2	Socket Class	80
4.3.2.2	Driver Class	80
4.3.3	Policeman Android Application Classes	81
4.3.3.1	Map Class	81

4.3.3.2	Socket Class	82
4.3.3.3	Policeman Class	82
4.4	Class Hierarchies and Relationships	83
4.4.1	Server Side Class Hierarchies and Relation	83
4.4.2	Android Application Side Class Hierarchies and Relation	85
4.4.2.1	Policeman Application.....	85
4.4.2.2	Driver Application.....	86
4.5	Conclusion	87

Chapter Five

Project Design

5.1	General Overview.....	89
5.2	Hardware components.....	89
5.2.1	XBee and XBee Pro RF Modules.....	89
5.2.2	Arduino.....	90
5.2.3	GPS Receiver (20 Channel EM-406A SiRF III Receiver with Antenna)	91
5.2.4	GPRS Modems.....	92
5.2.4.1	GSM/GPRS Module - SM5100B.....	93
5.2.5	Zigbee-Arduino Shield	93
5.2.6	GPRS shield (SM5100B Evaluation Board)	94
5.2.7	XBee Explorer Regulated	95
5.3	Hardware sub-system design	96
5.3.1	Car sub-system	96
5.3.2	The general fixed point	97
5.3.3	Core fixed point.....	97
5.3.4	Traffic light sub-system.	98
5.4	Overall system	98
5.5	Objects-Relational Model	102
5.5.1	Server Side.....	102
5.5.2	Driver and Policeman Android Application.....	103

5.6	State Behavioral Modeling	104
5.6.1	Admin Class (State Diagram)	104
5.6.2	GSMSerial Class (State Diagram)	104
5.6.3	Violation Class (State Diagram)	105
5.6.4	Car Class (State Diagram)	105
5.6.5	Message Class (State Diagram)	106
5.6.6	Driver Class (State Diagram).....	106
5.6.7	Policeman Android Application.....	107
5.6.7.1	Policeman Class (State Diagram).....	107
5.6.7.2	Driver Android Application.....	107
5.6.7.3	Driver Class (State Diagram).....	107
5.7	Subsystem Design	108
5.7.1	Server Application	108
5.8	Class and Object Design	109
5.8.1	Server Application	109
5.8.2	Policeman Android Application.....	112
5.8.3	Driver Android Application.....	114
5.9	Software Interface Design.....	115
5.9.1	Server Interface	115
5.9.2	Policeman Interface	116
5.9.3	Driver Interface	118
5.10	Conclusion.....	119

Chapter Six..... 120

System Implementation

6.1	Overview	121
6.2	Hardware Implementation	121
6.2.1	Zigbee Configuration	122
6.2.3	Car sub-system	126
6.2.4	General fixed node sub-system:.....	127

6.2.5	Core fixed node sub-system.....	127
6.2.6	Traffic Light sub-system	128
6.3	Software implementation	130
6.3.1	Driver Android Application.....	130
6.3.2	Policeman Android Application.....	131
6.3.3	Server Application	132

Chapter Seven..... 135

System Testing

7.1	Overview	136
7.2	Hardware Sub-System Testing	136
7.2.1	Basic packets generation test.....	136
7.2.2	Range, Ad-hock and Air bag sensor tests.....	137
7.2.3	Speed test.....	139
7.2.4	TCP connection between GPRS modem and server test	141
7.3	Software Testing	142
7.3.1	Connection Setup: Server, Drivers, Policeman.....	142
7.3.2	Policeman Interact with Server	144
7.3.3	Driver Request Street State.....	147
7.3.4	Additional Functions	148

Chapter Eight..... 149

Conclusion and Recommendations

8.1	Conclusion.....	150
8.2	Recommendations	150

References..... 152

Table of figures

Figure 1.1: Digital Roads	2
Figure 2.1: Zigbee alliance platform.	9
Figure 2.2: Zigbee networking topologies.	10
Figure 2.3: Satellites orbit.....	12
Figure 2.4: The locations of facilities.	13
Figure 2.5: The location of the first satellite from the object.....	14
Figure 2.6: The location of the second satellite from the object.	15
Figure 2.7: The location of the third satellite from the object.	15
Figure 3.1: the task network.....	41
Figure 4.1: Use Case Diagram	75
Figure 4.2: Admin/GSMSerial	83
Figure 4.4: Admin/Driver Relation	83
Figure 4.5: Admin/Violation Relation	84
Figure 4.7: GSMSerial/Message Relation	84
Figure 4.8: Driver/Car Relation	84
Figure 4.9: Driver/License Relation.....	85
Figure 4.10: Driver/Violation Relation	85
Figure 4.11: Policeman/Map Relation	85
Figure 4.12: Policeman/Socket Relation.....	86
Figure 4.13: Driver/Map Relation	86
Figure 4.14: Driver/Socket Relation.....	86
Figure5.1: XBee RF Module	90
Figure 5.3: XBee and XBee Pro dimensions	90
Figure 5.4: XBee schematic diagram.....	90
Figure 5.5: Arduino mega	91
Figure 5.6: GPS receiver	91
Figure 5.7: GPS receiver pin description.....	92
Figure 5.8: GPRS module.	93
Figure 5.9: Zigbee Arduino shield	93
Figure 5.10: GPRS shield	94
Figure 5.11: XBee explorer regulated	95
Figure 5.12: Car sub-system block diagram.....	97
Figure5.13: General fixed node block diagram.....	97
Figure 5.14: Core fixed node block diagram	98
Figure 5.15: Traffic light sub-system block diagram	98
Figure 5.16: overall system block diagram	99
Figure 6.17: overall system.....	100
Figure 5.19: Server UML Diagram.....	102

Figure 5.20: Driver UML diagram.....	103
Figure 5.21: Policeman UML diagram.....	103
Figure 5.22: Admin State Diagram.....	104
Figure 5.23: GSMSerial State Diagram.....	104
Figure 5.24: Violation State Diagram.....	105
Figure 5.25: Car State Diagram	105
Figure 5.26: Message State Diagram	106
Figure 5.28: Policeman State Diagram.....	107
Figure 5.30: Elaboration of a design component 1(Server).....	108
Figure 5.31: Elaboration of a design component 2(Server).....	108
Figure 5.32: Function Implementation (Login In)	109
Figure 5.33: Function Implementation (Forget Password).....	109
Figure 5.34: Function Implementation (Monitor Data Flow)	109
Figure 5.35: Function Implementation (Display Tables).....	110
Figure 5.36: Function Implementation (Add Driver)	110
Figure 5.37: Function Implementation (Delete Driver)	110
Figure 5.38: Function Implementation (Update Driver).....	111
Figure 5.39: Function Implementation (Get Driver Information).....	111
Figure 5.40: Function Implementation (Send Data to Serial)	111
Figure 5.42: Function Implementation (Is Car Licensed).....	112
Figure 5.43: Function Implementation (Determine Violation Cost).....	112
Figure 5.44: Function Implementation (Login System)	112
Figure 5.45: Function Implementation (Display Map).....	113
Figure 5.46: Function Implementation (Write Violation Report)	113
Figure 5.47: Function Implementation (Check Driver License)	113
Figure 5.48: Function Implementation (Request Violations List).....	114
Figure 5.49: Function Implementation (Login System)	114
Figure 5.50: Function Implementation (Monitor Street Request).....	114
Figure 5.51: Server interface (login screen).....	115
Figure 5.53: Policeman interface (1).....	117
Figure 5.55: Policeman interface (3).....	117
Figure 5.57: Driver interface (1).....	118
Figure 5.59: Driver interface (3)	119
Figure 6.2: XBee connected to computer using dongle.....	122
Figure 6.3: PC settings.....	123
Figure 6.4: Setting the Modem Configuration.....	124
Figure 6.5: real car sub-system	127
Figure 6.6: General fixed node sub-system	127
Figure6.7: real circuit for core fixed node sub-system	128
Figure 6.8: real circuit for traffic light sub-system.....	129
Figure 6.9: Driver android application	130
Figure 6.10: policeman android application	131

Figure 6.11: Server Main Interface	132
Figure 6.12: Add New Car	133
Figure 6.13: Edit Car.....	133
Figure 6.14: Add a new driver to the system.....	134
Figure 6.15: Edit driver information	134
Figure 7.1: the packet from the car to the fixed nodes.....	136
Figure 7.2: Range test setup	137
Figure 7.3: The received data at Core fixed node from three cars.....	138
Figure 7.4: Air bag sensor byte at 66666666 car	139
Figure 7.5: Speed test setup	140
Figure 7.6: Speed test results	141
Figure 7.7: GPRS connection test.....	142
Figure 7.8 : Server Sart Listening	143
Figure 7.9: The Policeman and Driver Connecting To Server	143
Figure 7.10: Server Established The Connection With Policeman and Driver.....	144
Figure 7.11: Policeman Trace Car By Car ID.....	144
Figure 7.12: Server Receive Message and parse it.	145
Figure 7.13: Tracing Car application	146
Figure 7.15: Write Violation.....	146
Figure 7.16: Street State with It Is Type.....	147
Figure 7.18: Street state application	147
Figure 7.19: Shortest Path	148

Table of tables

Table 2.1: Data Rate for GPRS Software	21
Table 3.1: Components Cost and Budget	39
Table 3.2: Working Cost.....	40
Table 3.3: TimeLine Table	40
Table 3.4: Gantti Chart.....	42
Table 3.5: Wi-Fi Specification	53
Table 4.1: Logging on and access the system data	67
Table 4.2: Maintain the system	68
Table 4.3: Monitor the system	68
Table 4.4: Tracing stolen car on a map	69
Table 4.5: Receive a messages.....	70
Table 4.6: Check driving license.....	71
Table 4.7: Write a violation report	72
Table 4.8: Monitor the street state	73
Table 4.9: Receive a violation message	74

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the number '1'.

1

Chapter One

Introduction

1.1 Overview

1.2 Project Idea Description

1.3 Motivations

1.4 Project Scope

1.5 Summary

1.1 Overview

The world is witnessing a dramatic evolution in various fields of technology, engineers work everywhere on implementing real life technologies to increase human well-being and solve real daily problems to serve and protect the humans from threats and risks.

The world suffers from major problems like wars, natural disasters, global warming, pollution...etc. One of these problems that kill millions of people and cause huge economic losses per year is the traffic accidents. Our system aims to solve part of this problem through digitalizing the roads, and that is the idea our project.



Figure 1.1: Digital Roads

In this chapter, we will describe our project, explain our motivations, and show our project scope.

1.2 Project Idea Description

The project aims to build a system, which will be used for observing and monitoring the street traffic and records the violations could be done by the drivers

on the roads. The car location identified by a GPS receiver -implanted in the car- and the car speed will be transferred over the car's Zigbee chip to the street node-Zigbee pro chip- and then to the server using a GPRS modem.

The GPRS modem will be connected to the server. In the server, a database will be built, which will include important information such as all the details about the cars and their drivers. Other information about ambulance, Fire Brigade and police stations location and contact numbers will be included too.

The central system (API) will be programmed using C# language and SQL server database, and will be purposed to record the violations, calculate the traffic, monitor the accident, track any car in our network range and control the traffic lights if needed. Most of the information will be transferred to the police station, other to the subscribers, drivers or police android device through GSM network. Other information will be displayed on a map at the android device of the car driver or patrolman.

Our system will be built to make the scenario of three fixed nodes, 200 meters apart from each other. Three cars moves in the fixed node range. A central fixed node will have a GSM/GPRS modem to deliver the data to the server.

After processing the data, a subscript message and some information will be transferred to the drivers and the patrolman android device, also the police man can demand to track any car in our system coverage range.

1.3 Motivations

The system focuses on these motivations:

- Build an Integrated system to help our community and government by reducing the vehicles accidents resulted by the violations done by drivers.

- Reduce the traffic Congestion Problem.
- Auto theft issue will be controlled.
- The fastest accidents notifications and handling techniques will be supplied.

1.4 Project Scope

1.4.1 Context

The system defines functions and solutions for the community, people on roads and for the government. The advantage of the project is being an integrated project that will over pass other projects that worked on the roads problems and traffic violations. The functions of our system meets other projects requirements by implying them in our system and solve the unsolved problems in the previous semi-solutions to get highest performance in solving the streets problems.

1.4.2 Information Objectives

The different input Data goes through the system as analog and digital signals, the Zigbee chip will transmit them to the street's nodes (Zigbee chip) to be re-transmitted to the server by a GPRS modem. The server processes the data in order to register the violations done by a specific car, determining its location and make notifications and actions if it is happened. After that, output data will be sent to

the drivers and policeman android device which is displayed on a map showing cars locations and warning messages.

1.4.3 Function and Performance

Cars speed and location will be carried from the car module (Zigbee Chip and GPS receiver) to the server through the streets nodes through GSM/GPRS modems.

The server side aims to store, read, process and send the data. If a violation occurred, our system will receive it, record it and send it to the nearest policeman, GPS coordinate and car specifications will be displayed on the map with bookmarks. Some of these data will be transmitted to the driver to inform him about the violation.

Another ability in our system is that it gives the driver important information about the streets. Such information will be shown on map on his/her android device.

1.4.4 Assumptions and requirements

- Each car will contain a GPS receiver.
- Each car must contain a Zigbee Chip.
- Each car must contain an Arduino.
- Some sensors must be in the car (like airbag sensor).
- Zigbee chips must be located in roads (fixed nodes), they must be 200 meters apart (maximum).
- A GPRS modem must be implanted in some fixed nodes (every five or six nodes will need one).

1.5 Summary

In this chapter, we made a general overview about our project, discussed a brief description of our idea, talked about our motivations that stimulated us to choose this project and we made a project scope at the end of this chapter.

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the number 2.

2

Chapter Two

Literature Review

2.1 Overview

2.2 Theoretical background

2.3 Literature review

2.4 Summary

2.1 Overview

In this chapter, we will mention the basic theoretical information about some technologies to be used in our project, in order to use them in an appropriate way and take the desired advantages of them.

We will talk about the Zigbee, GPS and GSM/GPRS technologies. Also, present a brief overview about some programming languages to be used in the project, as C#, Android and MySQL Languages.

2.2 Theoretical background

2.2.1 Zigbee Technology

The past several years have shown a rapid development of wireless networking. However, up to now wireless networking has been mainly focused on high speed communications and long range applications, such as the IEEE 802.11 Wireless Local Area Network (WLAN) standards. And there are many wireless monitoring and control applications in industrial and home environments, which requires longer battery life, lower data rate and less complexity than the existing standards. So, the Zigbee Alliance and the IEEE decided to join forces, and Zigbee is the commercial name for this new technology.^[1]

Zigbee is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802 standard for personal area networks. Applications include wireless light switches, electrical meters with in-

home-displays, and other consumer and industrial equipment that require short-range wireless transfer of data at relatively low rates. The technology defined by the Zigbee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth. Zigbee is targeted at radio-frequency (RF) applications that requires a low data rate, long battery life and secure networking.

2.2.1.1 Name of Zigbee

The name Zigbee is said to come from the domestic honeybee which uses a zigzag type of dance to communicate important information to other hive members. This communication dance (the Zigbee principle) is what engineers are trying to emulate with this protocol-a bunch of separate and simple organism that join together to tackle complex tasks.

2.2.1.2 Zigbee Stack

Zigbee protocol stack is shown in figure (2.1). It extends the IEEE802.15.4 standard with two layers (Network layer and Application layer) to support additional features such as routing reliability and network security and many other features.

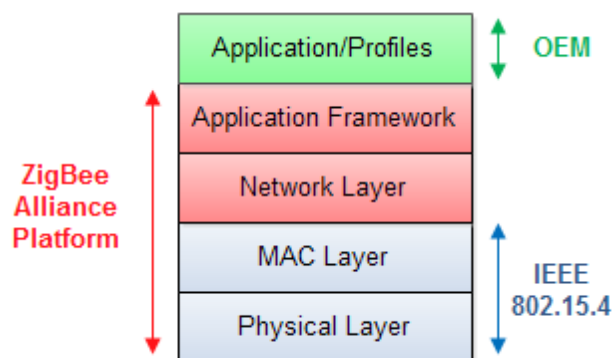


Figure 2.1: Zigbee alliance platform.

IEEE 802.15.4 standard defines two layers which are the physical layer (PHY) and the medium access control (MAC) layer. The PHY layer is concerned with the interface to the physical transmission medium (wireless channel) as well as exchanging data bits with the upper layer (MAC layer). The PHY layer specialises 27 channels distributed along three license-free frequency bands as follows: channels at 2.4 GHz with data rates of 250 kbps, 10 channels at 902 to 928 MHz with data rates of 40 kbps and one channel at 868 to 870 MHz with a data rate of 20 kbps.

The MAC layer is responsible for providing reliable communications between a node and its immediate neighbors, helping to avoid collisions and improve efficiency.

The way at which nodes are interconnected defines the network topology. Zigbee allows three different topologies which are star, tree and mesh topologies. In the star topology the network has a central node (coordinator), which is linked to all other nodes in the network and serve as their parent. All messages that are exchanged are traveling through this central node. Tree topology has a root that serve as a parent node for nodes in the lower level. Nodes in the second level can have children if they are fully functional nodes (ZC or ZRs). To reach its destination, a message travels up the tree from its source until it reach the first common parent with its destination and then goes down to the destination. Star and Tree networks defines single path between a source and its destination.

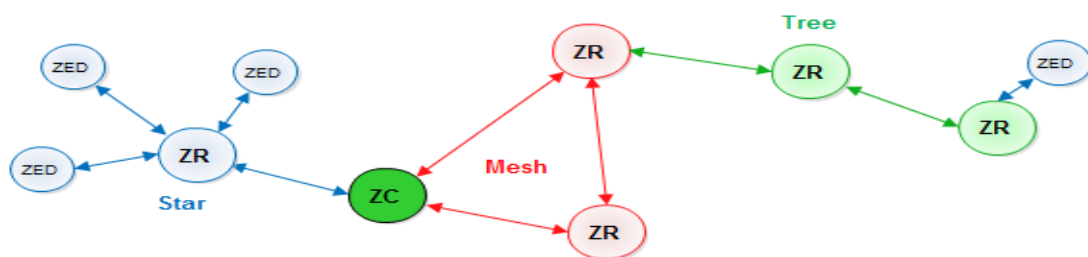


Figure 2.2: Zigbee Networking Topologies.

On the other hand, nodes that form mesh topology are interconnected with each other so that multiple pathways are available. Connections between nodes are dynamically updated and optimized through sophisticated, built-in mesh routing tables. Mesh networks are decentralized in nature; each node is capable of self-discovery on the network. If one node leaves the network, the mesh topology allows the others to reconfigure their routing paths based on the new network structure. The characteristics of mesh topology and ad-hoc routing provide greater stability in changing conditions or failure at single nodes. Figure 2.9 shows a typical hybrid network.

2.2.1.3 Types of Zigbee devices:

- a. Coordinator: coordinator organizes the network and maintains routing tables. Each network must be formed by a coordinator and never have more than one coordinator in the network regardless of the network topology.
- b. Router: router can talk to the coordinator and to other routers. It reduces function end devices. It can join existing networks, send information, receive information, and route information. A network may have multiple router radios.
- c. End devices: can join networks and send and receive information. They can talk to routers and the coordinator, but not to each other. End devices always need a router or the coordinator to be their parent device. Zigbee networks may have any number of end devices.

2.2.2 GPS

The Global Positioning System (GPS) is a U.S. owned utility that provides users with positioning, navigation, and timing (PNT) services, this system consists of three segments: the space segment, the control segment, and the user segment.^[2]

2.2.2.1 Space Segment

The GPS space segment consists of a constellation of satellites transmitting radio signals to users. The Air Force manages the constellation to ensure the availability of at least 24 GPS satellites, 95% of the time. For the past several years, the Air Force has been flying 31 operational GPS satellites, plus 3-4 decommissioned satellites ("residuals") that can be reactivated if needed.

GPS satellites fly in medium Earth orbit (MEO) at an altitude of approximately 20,200 km. Each satellite circles the earth twice a day. The satellites in the GPS constellation are arranged into six equally-spaced orbital planes surrounding the earth. Each plane contains four "slots" occupied by baseline satellites, this 24-slot arrangement ensures there are at least four satellites in view from virtually any point on the planet, Figure (2.3) shows the satellites orbit.

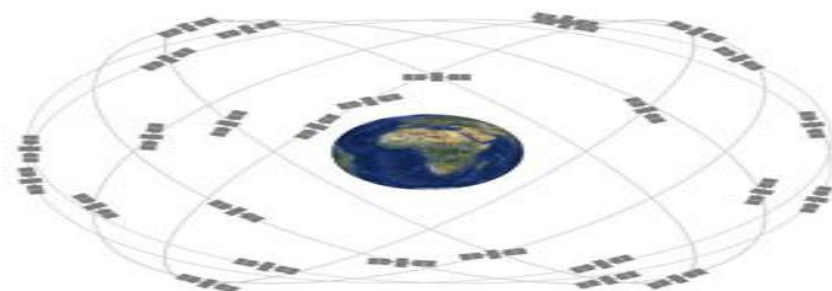


Figure 2.3: Satellites orbit.

The Air Force normally flies more than 24 GPS satellites to maintain coverage whenever the baseline satellites are serviced or decommissioned. The extra satellites may increase GPS performance but are not considered part of the core constellation.

2.2.2.2 GPS control

The GPS control segment consists of a global network of ground facilities that track the GPS satellites, monitor their transmissions, perform analyses, and send commands and data to the constellation.

The current operational control segment includes a master control station, an alternate master control station, 12 command and control antennas, and 16 monitoring sites. The locations of these facilities are shown in the figure (2.4).

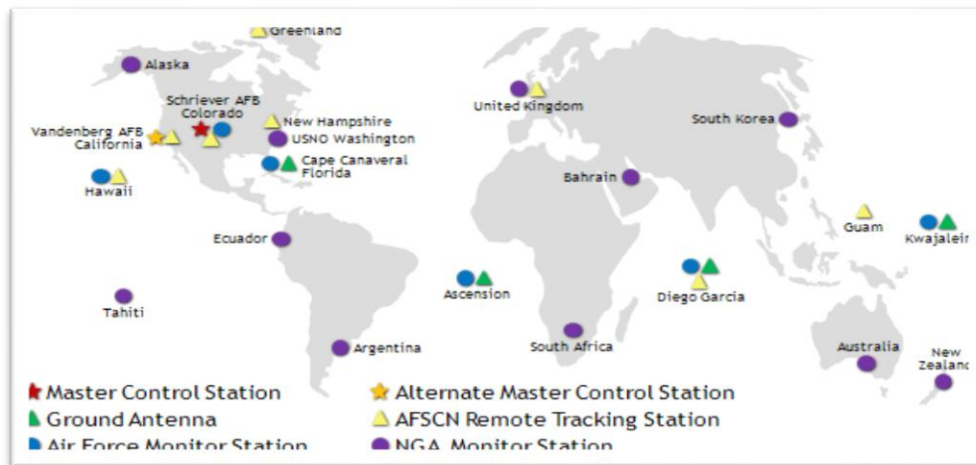


Figure 2.4: The locations of facilities.

2.2.2.3 User segment

The user segment consists of the GPS receiver equipment, which receives the signals from the GPS satellites and uses the transmitted information to calculate the user's three-dimensional position and time.

2.2.2.4 How GPS works

To know the GPS coordination of particle position (longitude, latitude and altitude) we must receive signals from four satellites at least.

The first satellite located at point S1 as shown in figure (2.5), transmits signals in speed of light that take T1 to reach the particle position ,after that the location of particle is a point in this circle, so we need another satellites to know the specific location.

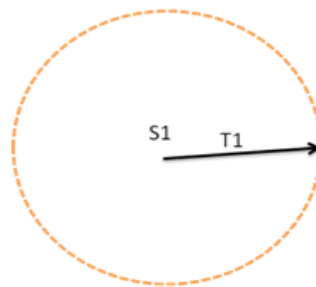


Figure 2.5: The location of the first satellite from the object

The second satellite located at point S2, transmits signals in speed of light that take T2 to reach the particle position , after that the location of particle maybe on L1 or L2, as shown in figure (2.6).

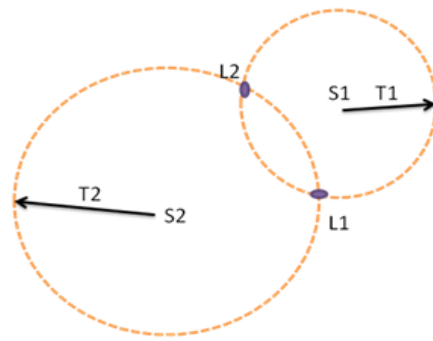


Figure 2.6: The location of the second satellite from the object.

The third satellite located at point S3, transmits signals in speed of light that take T_3 to reach the particle position, so the location of particle on L1, the fourth satellite used to know the altitude of particle above sea level as shown in Fig (2.7)

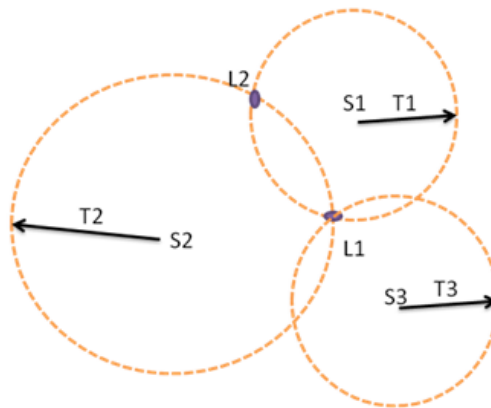


Figure 2.7: The location of the third satellite from the object.

2.2.2.5 Equations for location determination:

- The coordinates of position particle that we want to determine is X_0, Y_0, Z_0 (unknown).
- The coordinates of first satellite that located on S1 point is X_1, Y_1, Z_1 (known).
- The coordinates of second satellite that located on S2 point is X_2, Y_2, Z_2 (known).

- The coordinates of third satellite that located on S3 point is X3, Y3, Z3 (known).
- T1: time that the first satellite signal takes to reach the position of particle.
- T2: time that the second satellite signal takes to reach the position of particle.
- T3: time that the third satellite signal takes to reach the position of particle.
- T4: time that the fourth satellite signal takes to reach the position of particle.

$$(X1 - X0)^2 + (Y1 - Y0)^2 + (Z1 - Z0)^2 = [C(T1)]^2$$

$$(X2 - X0)^2 + (Y2 - Y0)^2 + (Z2 - Z0)^2 = [C(T2)]^2$$

$$(X3 - X0)^2 + (Y3 - Y0)^2 + (Z3 - Z0)^2 = [C(T3)]^2$$

$$(X4 - X0)^2 + (Y4 - Y0)^2 + (Z4 - Z0)^2 = [C(T4)]^2$$

After solving these four equations we will know X0, Y0, Z0 (the particle position coordinates).

2.2.3 GSM

GSM (Global System for Mobile communications) is a second-generation digital network, supporting voice and simple data services, including “dial-up” data and text messaging. Open, digital cellular technology used for transmitting mobile voice and data services. GSM supports voice calls and data transfer speeds of up to 9.6 kbps, together with the transmission of SMS (Short Message Service). It predates Code Division Multiple Access (CDMA), which is especially strong in Europe.

Enhanced Data rates for GSM Evolution (EDGE) (also known as Enhanced GPRS (EGPRS) is faster than GSM and was built upon GSM.^[3]

The system built based on cell phone technology. GSM is the dominant digital mobile phone standard for most of the world. It determines the way in which mobile phones communicate with the LAN-based network of towers.

2.2.3.1 GSM System Overview

“The GSM system was designed as a second generation (2G) cellular phone technology. One of the basic aims was to provide a system that would enable greater capacity to be achieved than the previous first generation analogue systems. GSM achieved this by using a digital TDMA (time division multiple access approach). By adopting this technique more users could be accommodated within the available bandwidth. In addition to this, ciphering of the digitally encoded speech was adopted to retain privacy. Using the earlier analogue cellular technologies it was possible for anyone with a scanner receiver to listen to calls and a number of famous personalities had been eavesdropped with embarrassing consequences. ”^[5]

2.2.3.2 GSM Basic

The GSM cellular technology had a number of design aims when the development started:

- It should offer good subjective speech quality
- It should have a low phone or terminal cost
- Terminals should be able to be handheld
- The system should support international roaming
- It should offer good spectral efficiency

- The system should offer ISDN compatibility

The resulting GSM cellular technology that was developed provided for all of these. The overall system definition for GSM describes not only the air interface but also the network or infrastructure technology. By adopting this approach it is possible to define the operation of the whole network to enable international roaming as well as enabling network elements from different manufacturers to operate alongside each other, although this last feature is not completely true, especially with older items.

GSM cellular technology uses 200 kHz RF channels. These are time division multiplexed to enable up to eight users to access each carrier. In this way it is a TDMA / FDMA system.

The base transceiver stations (BTS) are organized into small groups, controlled by a base station controller (BSC) which is typically co-located with one of the BTSs. The BSC with its associated BTSs is termed the base station subsystem (BSS).

Further into the core network is the main switching area. This is known as the mobile switching center (MSC). Associated with it is the location registers, namely the home location register (HLR) and the visitor location register (VLR) which track the location of mobiles and enable calls to be routed to them. Additionally there is the Authentication Centre (AuC), and the Equipment Identify Register (EIR) that are used in authenticating the mobile before it is allowed onto the network and for billing. The operation of these are explained in the following pages.

Last but not least is the mobile itself. Often termed the ME or mobile equipment, this is the item that the end user sees. One important feature that was first implemented on GSM was the use of a Subscriber Identity Module. This card carries with it the user identity and other information to allow the user to upgrade a phone

very easily, while retaining the same identity on the network. It was also used to store other information such as "phone book" and other items. This item alone has allowed people to change phones very easily, and this has fuelled the phone manufacturing industry and enabled new phones with additional features to be launched. This has allowed mobile operators to increase their average revenue per user (ARPU) by ensuring that users are able to access any new features that may be launched on the network requiring more sophisticated phones.

2.2.4 GPRS Service

2.2.4.1 Definition

General Packet Radio Services (GPRS), is a packet-based wireless communication service that provides data rates from 56 up to 114 Kbps, and continuous connection to the Internet for mobile phone and computer users. The higher data rates allow users to take part in video conferences, and interact with multimedia Web sites, and similar applications using mobile handheld devices as well as notebook computers. GPRS is based on Global System for Mobile communication (GSM) and complements existing services such circuit-switched cellular phone connections and the Short Message Service (SMS).

In theory, GPRS packet-based services cost users less than circuit-switched services, since communication channels are being used on a shared use, as packets are needed basis rather than dedicated to only one user at a time. It is also easier to make applications available to mobile users, because the faster data rate means that middleware currently needed to adapt applications to the slower speed of wireless systems are no longer be needed. As GPRS has become more widely available, along with other 2.5G and 3G services, mobile users of virtual private networks (VPNs)

have been able to access the private network continuously over wireless, rather than through a rooted dial-up connection.^[4]

GPRS also complements Bluetooth, a standard for replacing wired connections between devices with wireless radio connections. In addition to the Internet Protocol (IP), GPRS supports X.25, a packet-based protocol that is used mainly in Europe. GPRS is an evolutionary step toward Enhanced Data GSM Environment (EDGE) and Universal Mobile Telephone Service (UMTS).

2.2.4.2 GPRS and packet data in GSM

GPRS adds a packet-based data capability to the GSM system, and brings IP networking into the GSM world. As well as modifications within the handsets and the radio interface, GPRS also involves the addition of new packet routers into the GSM core network, producing a core network with separate circuit- and packet switched domains. Using GPRS, each user only requires radio resources when a packet of information is sent or arrives, it does not require the channel for the entire duration of the data transfer.

In this way, the single timeslot can be made available for more than one user, allowing radio channels to be shared, by up to seven users in GPRS.

The overall data rates in GPRS are specified for each timeslot, varying depending on what is known as the coding scheme. There are four coding schemes (CS) in GPRS (known as CS1, CS2, CS3, and CS4). The data rates for each coding scheme are shown in table 2.2. Note that each overall channel is actually designed to carry 22.8 kbps (as in standard GSM connections), and the extra capacity is actually

used for error protection. Where there is a good error protection for CS1, but not so good for CS4. In practice, only coding schemes 1 and 2 are widely used.

CS1 and CS2 offer good error detection and correction with low throughput; in the first step of GPRS software only these two techniques may be used. CS3 and CS4 provide higher throughputs but have little or no error correction capabilities.

Communication between the base station and any given mobile host is full-duplex but can be asymmetric; that is the down link and the uplink capacities need not be the same. Hence, the system can select the coding scheme based on the amount of data to be transferred through down link and uplink. Generally, the downlink has more frequency than the uplink. Both links can send user data as well as control data. There are some situations in which the amount of user data may be larger than the control data and vice versa. That is, the downlink may have more data to carry than the uplink. Thus, the selection of coding scheme may differ for downlink and uplink.

Table 2.1: Data Rate for GPRS Software

User Data Rate	CS1	CS2	CS3	CS4
1 Timeslot	9.05 kbps	13.4 kbps	15.6 kbps	21.4 kbps
8 Timeslot	72.4 kbps	107.2 kbps	124.8 kbps	171.2 kbps

2.2.4.3 GPRS Network Elements

Additional elements are required to provide the General Packet Radio Services (GPRS), including:

Core network GPRS support nodes (GSNs), comprising:

- Serving GSN (SGSN).
- Gateway GSN (GGSN).

These elements use IP (Internet Protocol) technology to route information to and from the mobile handset. They are effectively IP routers with modifications to allow for managing subscriber access, mobility, and IP sessions. A GPRS data session would be handled by the GSN rather than the MSC, which is used for voice. In addition, GPRS operation requires the BTS to be GPRS-capable, and a packet control unit must be available, usually at or within the BSC. These two base station subsystem (BSS) elements allow the data packets to transfer in the right format on the correct GPRS radio channels.

2.2.2 Software Theory

The computer cannot work independently. There is also need for the software to manage, read, write, save, clear data and do many operations. There are a lot of programming languages that are used to build out a computer program, choosing any language depends on the tasks of the program. The different programming languages used for different sides. In addition to the programming language and working on data. We need to create tables and a database to maintain with the system and the programming language.

2.2.2.1 C# Programming Language

C# (Pronounced "C sharp"), is as Microsoft simple, modern, object-oriented, and type-safe programming language derived from C and C++. Targeted at the web development community, where Java is the dominant language, C# is available as part of Microsoft's visual studio.NET. In addition to C#, visual studio .NET also supports Visual Basic, Visual C++, and Jscript.^[18]

C# is a new programming language from Microsoft designed specifically to target the .NET Framework. Microsoft's .NET Framework is a runtime environment and class library that dramatically simplifies the development and deployment of modern, component-based applications.

C# was designed by Anders Hejlsberg (creator of Turbo Pascal and architect of Delphi), Scott Wiltamuth and Peter Golde. Described in the C# Language Specification as a "simple, modern, object-oriented, and type-safe programming language derived from C and C++, C# bears many syntactic similarities to C++ and Java.

C# has no runtime library of its own. instead, C# relies on the Vast class library in the .NET Framework for all its needs, including console I/O, network and file handling , collection data structure, and many other facilities .

2.2.2.2 Microsoft SQL Database

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications. There are at least

different editions of Microsoft SQL Server aimed at different audiences and for different workloads.

Microsoft SQL Server is offered as a "Fully Managed" database. In the Fully Managed infrastructure, ITS owns, manages and supports the Microsoft SQL Server hardware and software resources. The service includes high availability features and regular backups.^[19]

2.2.2.3 Android (Operating System)

Android is a Linux-based operating system, designed primarily for touchscreen mobile devices such as smart phones and android device computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. The first Android-powered phone was sold in October 2008.^[20]

Android is open source and Google releases the code under the Apache License. This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications ("apps") that extend the functionality of devices, written primarily in a customized version of the Java programming language. In October 2012, there were approximately 700,000 apps available for Android, and the estimated number of applications downloaded from Google Play, Android's primary app store, was 25 billion.

Android powers devices from some of the best phone and android device makers in the world, like Samsung, HTC, LG, Motorola, Sony, Asus, and more. Android devices comes in all shapes and sizes, with vibrant high-resolution displays and cameras, giving you the flexibility to choose the one that's just right for you.. Android lets you choose what's right for you through a global partnership network of over 300 carriers, in over 169 countries.

2.2.2.4 SQLite Database

SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. The database requires limited memory at runtime (approx. 250 Kbytes) which makes it a good candidate from being embedded into other runtimes.

SQLite is embedded into every Android device. Using a SQLite database in Android does not require a setup procedure or administration of the database.

You only have to define the SQL statements for creating and updating the database. Afterwards the database is automatically managed for you by the Android platform. Access to a SQLite database involves accessing the file system. This can be slow. Therefore it is recommended to perform database operations asynchronously.^[21]

2.3 Literature review

A collection of projects and papers is presented and compared with our project. We collect four studies:

2.3.1 Traffic Message Channel (TMC)

2.3.1.1 Overview

TMC is a technology for delivering traffic and travel information to motor vehicle drivers. It is digitally coded, using the Radio Data System on conventional FM radio broadcasts. It can also be transmitted on Digital Audio Broadcasting or satellite radio. TMC allows silent delivery of dynamic information suitable for reproduction or display in the user's language without interrupting audio broadcast services. Both public and commercial services are operational in many countries. When data is integrated directly into a navigation system, traffic information can be used in the system's route calculation.^[5]

The Traffic Message Channel (TMC) is a digital data channel provided by the traffic news service that supplies continuously updated 'real time' traffic data. When your route has been selected, our navigation systems with dynamic route guidance make use of this data to check traffic conditions. If the TMC indicates a traffic jam or other hold-up, the system automatically calculates an alternative way to your destination. It takes into account the time required for the alternative route and if this is longer than driving through the traffic jam, your route is not changed. The Traffic Message Channel is now available in much of Europe with data transmitted digitally, decoded by your navigation system, and displayed in your own language.

2.3.1.2 Advantages:

- Continuous traffic roads monitoring.
- Directs the car drivers to the best way when congestion happened.

2.3.1.3 Disadvantages:

- used the radio data system to transmit the information (limited capacity for amount of data)
- its need to encoded and decoded tables to identify the locations
- Only takes over the congestion problem.
- It only gives notifications to the drivers about the roads status, does not take any actions.

2.3.2 Vehicle Information and Communication System(VICS)

2.3.2.1 Overview

This system uses roadside beacon unit, which is a transmitter device located on the speed limit sign .This unit transmits signals contains information regarding the road speed limit. The vehicle receive this information, and when it exceeds the speed

limit, the system makes warning to the driver .This system uses mobile communication according to the update in the speed limit of the road^[6]

2.3.2.2 Advantages:

- This system is simple, when the vehicle appears, the system works.
- The easy of updating the speed limit by using the mobile communication.

2.3.2.3 Disadvantages:

- The cost of this system, if we need to consider all roads in the city, and that we are handling only one road problem.
- The cost of maintenance.
- Only takes over the speed limits violation, and not every where
- It only gives notifications to the drivers about the violation, does not take any actions on signing the violation.

2.3.3 Intelligent traffic with automatic violation system

2.3.3.1 Overview

This project aims to plan and build a system, which will be used to reduce the amount of vehicle accidents, which occur every day because the illegal vehicle speed. We propose a model for the speed monitoring of vehicles, and create an

automatic violation system using GSM network-based on real-time localization system GPS. This model is based on sensing the location of the vehicle depends on the built in database. When the vehicle speed exceeds the speed limit, then the violation occurred will be recorded.

It Provides a new mechanism for stopping increasing of the speed more than the limit, and this dramatically reduces the traffic accidents, but if it happened; and this is the rate of a few, the system can help in speed up the rescue operation, and solve the delay problem in the ambulance arriving to the accident location by automatic alarm the emergency center about the location of accident once it occurred.^[7]

2.3.3.2 Advantages:

- Continuous cars speed monitoring.
- It records the speed violation if occurred.
- Takes some action by calling the police if an accident happened.

2.3.3.3 Disadvantages:

- Based only on GPS - does not work in the covered areas like tunnels.
- It only handles the speed and accidents issues.
- Does not make any control on the road.

2.3.4 Innovative congestion control system for ambulance using ZigBee

2.3.4.1 Overview

This project is designed to clear the traffic and provide a way to the ambulance so that the patient can be saved out of danger, and will reach the hospital safely and as quickly as possible.

To implement this application the project uses ZigBee technology, and here we will see how the system works:^[8]

- The ambulance will be attached with the ZigBee transmitter and the ZigBee receiver will be attached at the traffic signals.
- This transmitter transmits a unique code continuously into air.
- When the ambulance is near to the traffic signals, the unique code will be transmitted by the transmitter from the ambulance, will be received by the ZigBee receiver at the traffic signals.

2.3.4.2 Advantages

- The patient will reach the hospital safely and quickly.

2.3.4.3 Disadvantages:

- The traffic light control only.
- Our system faster than this system , since we track the ambulance continuously

2.4 Summary

In this chapter we mentioned the basic theoretical information about the technologies to be used in our project, discussed their operating principles and took a historical round in some of them.

In the second part of the chapter, we mentioned the most likely related works, we discussed them and mentioned their advantages and disadvantages.

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the number 3.

3

Chapter Three

Project Management Plan

- 3.1 Overview
- 3.2 Project Management Plan
- 3.3 Project Methodology
- 3.4 Project Components
- 3.5 Summary

3.1 Overview

In this chapter, we will explain software management plan in details, and explain project task and subtasks for everyone and draw Gantt chart, network activity and timeline chart. A discussion of project budget and risks will be introduced too.

3.2 Project Management Plan

3.2.1 Project Tasks

1) Project Idea (T1)

1. Identify needs and benefits of the idea.
2. Milestone product statements define.

2) Collecting Information (T2)

1. Interviews with computer and communication experts.
2. Observation of the streets violations and collect data.
3. Study related projects.

3) Studying of the concepts of communication (T3)

1. Study data transformation flow.
2. Specify the data type transmitted.
3. Define hardware limitations.

4) Software programming languages study (T4)

1. Study languages properties.
2. Do we need object oriented language and open source language.
3. Deal with serial port and I/O data.
4. Define libraries needed to support the project.

5) Feasibility Study (T5)

1. Software cost.
2. Hardware cost.
3. Operational cost.

6) Design Prototype (T6)

1. Simple design to define the requirements.
2. Simple code and hardware component interface.
3. Simple network interaction.
4. Simple data base.

7) Identify and design data base (T7)

1. Verifying data.
2. Start design tables.
3. Find relationships between tables.

8) Design interfaces (T8)

1. Driver interface.
2. Policeman interface.
3. Server interface.

9) Network and circuit design (T9)

1. Design speed and location component.
2. Design road/car circuits.
3. Design Road/Server Circuits.

10) Final collection of documents needed (T12)

1. Documents format.
2. Print documents.

11) Coding stage (T10)

12) Testing and maintenance (T11)

1. Testing.
2. Detecting errors.
3. Correcting errors.

3.2.2 Risks

In this section we will talk about the expected risks and limitation while design, implement and develop it. Then finds some of expected solutions to solve these problems. In addition to this, we will talk about critical risks in details by using Risk Information Sheet (RIS).

These risks can be classified as follows:

- **General risk**

Risks that exists in every project, like

- Lacking funding.
- Project requirements will be changed.
- Anybody from staff may absence.
- Staff inexperienced.
- Safety risks.
- Technology will not meet expectations.
- Lack of training on tools.
- Delivery deadline will be tightened.
- Difficulty in analyzing project requirement.

- **Specific risk:**

Risks that related to our project, like

- Risk in communication data.
- Lack of training on software.
- GPS connection may fail.
- Car model we do it maybe don't work.

- Database backup copy maybe damaged or lost.
- Failure in some part of project in the car, server or GPRS/GSM node uses it.
- Network connections may fail.
- Signals from car may fail, like airbag and speed signal.

Risks Solution:

Some of expected solutions to solve the problems that we face are:

❖ **General risk solution:**

- Search for source of fund like university (satisfied partially).
- Divide requirements and find details for every requirement.
- Study hard for technology and tools that we use it in our project.
- Trying to do tasks on time.

❖ **Specific Risk Solution:**

Risk information sheet			
Risk ID: 1	Date : 17/10/2013	Prob. : 70%	Impact : Medium
Description:			
Data safety and security			
Refinement / context :			
Sub condition 1:data may lost and stolen			
Sub condition 2: the security of the database none effectively.			

Sub condition 3: the design of the database may not confirm the security rules	
Mitigation / monitoring : Confirm the security rules. Using all security tools e.g. password, firewall.	
Management / contingency plan / trigger : Use backup of the program.	
Current status: Mitigation steps initiated 17/10/2013	
Originator : Software Team Group	Assigned : Eng. Husam Sa'ed

Risk information sheet			
Risk ID: 2	Date : 17/10/2013	Prob. : 70%	Impact : Medium
Description: Lack of funding, we face it when apply the project.			
Refinement / context : Sub condition 1: fund of software and hardware that transfer data like ZigBee and GPRS. Sub condition 2: some component we will import it from outside so that's expensive.			
Mitigation / monitoring : We will try to support our project by present budget from resources like our university			
Management / contingency plan / trigger : Decreasing cost by reducing number of component			
Current status: Mitigation steps initiated 17/10/2013			
Originator : Software Team Group		Assigned : Eng. Samer Alqadi	

Risk information sheet			
Risk ID: 3	Date : 17/10/2013	Prob. : 30%	Impact : low
Description: database backup copy may damage or lost			
Refinement / context : Sub condition 1: database that contain all data about cars may be damaged Sub condition 2: databasemaybelost or damaged.			
Mitigation / monitoring : We will make a copy and backup for database			
Management / contingency plan / trigger : If this risk happen we use backup of the program			
Current status: Mitigation steps initiated 17/10/2013			
Originator : Software Team Group		Assigned : Eng. Yazeed Aljuneidi	

3.2.3 Project Resources and Estimated Cost

Table (3.1, 3.2) below shows the hardware and the working cost needed in details.

Table 3.1: Components Cost and Budget

Project Component	Number Of Component	Price per piece
XBee	3	250 NIS
XBee Pro	3	300 NIS
Android device	2	500 NIS
BR- 355 GPS receiver	3	500 NIS
Arduino Microcontroller	5	350 NIS
GSM/GPRS modem	3	700 NIS

Software		500 NIS
XBee Explorer Dongle USB	1	200 NIS
XBee Explorer Regulated	2	70 NIS
Arduino-Zigbee shield	4	300 NIS
Arduino-GSM shield	2	300 NIS
GPRS Antenna	2	50 NIS
Other expenses		500 NIS
TOTAL		11240 NIS

We estimate the cost per user (GPS+Arduino+Zigbee) = 50+ 100+250 = 400 NIS.

And the software to free.

Every 128 m (see chapter 7) there is 1 Zigbee chip, so the cost per 128 m 300 NIS plus.

Table 3.2: Working Cost

Project Component	Developer	Salary/hour
Network designer	Eng.Ala'aManasrah	10 \$
Hardware expert	Eng. YazeedAljuneidi	10 \$
Server programmer	Eng. HosamSa'ed	10 \$
Android apps developer	Eng.SamerAlqadi	100 \$
Communication engineering	Eng. HaneenManasrah	10 \$

3.2.4 Time Estimation

3.2.4.1 TimeLine Table

Table 3.3: TimeLine Table

Task	Duration (weeks)	Dependencies
T1	2	

T2	3	T1
T3	3	T1,T2
T4	2	T3
T5	2	T3,T4
T6	4	T3,T4
T7	3	T4
T8	6	T7
T9	6	T6
T10	4	T1,T2
T11	4	T8,T10
T12	2	T8,T9,T11

3.2.4.2 Activity Network

Activity network, also called task network, is a graphical representation of the task flow for a project. Figure 3.1 show the task network.

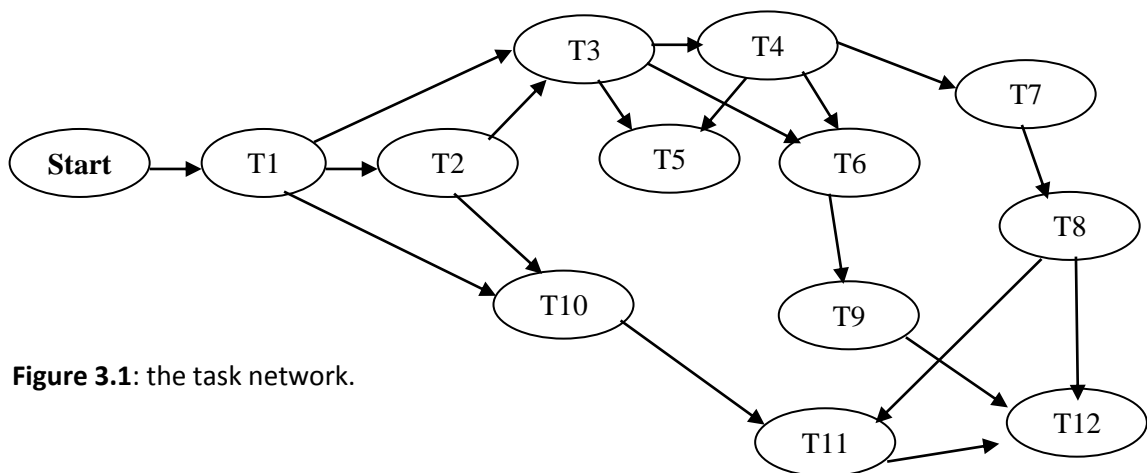
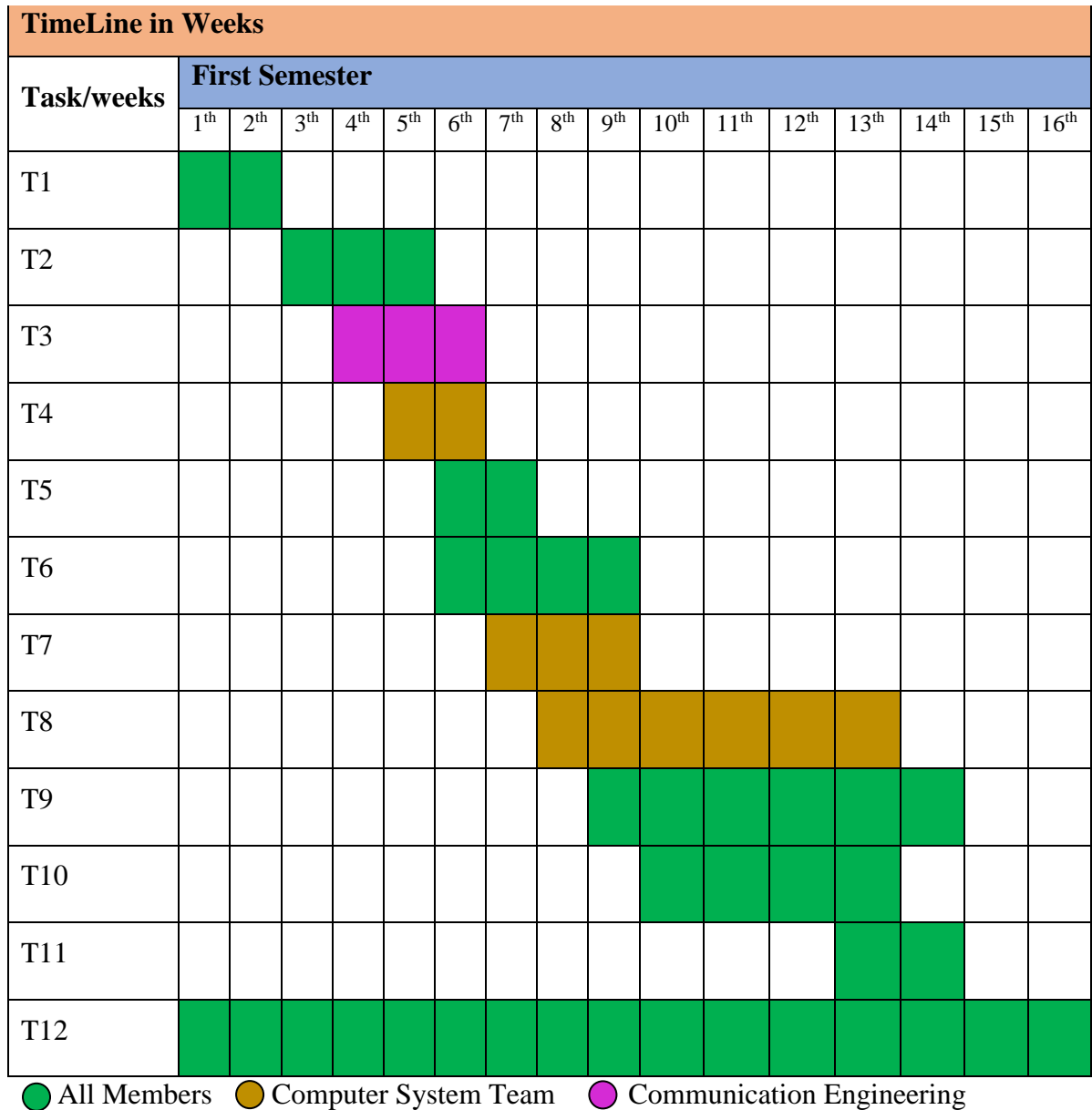


Figure 3.1: the task network.

3.2.4.3 Gantt Chart

Table 3.4: Gantt Chart



TimeLine in Weeks																
Task/weeks	Second Semester															
	1 th	2 th	3 th	4 th	5 th	6 th	7 th	8 th	9 th	10 th	11 th	12 th	13 th	14 th	15 th	16 th
T1	●	●	●	●												
T2	●	●	●	●	●											
T3			●	●	●	●	●	●	●	●	●	●	●	●	●	●
T4					●	●	●	●	●	●	●	●	●	●	●	●
T5								●	●	●	●	●	●	●	●	●
T6											●	●	●	●	●	●
T7													●	●	●	●

● All Members
● Computer System Team
● Communication Engineering

T1: Buy the hardware component

T2: Preparing software skills for the project

T3: Build up the software

T4: Testing the software

T5: Testing the hardware

T6: Testing the system

T7: Documentation writing

3.3 Project Methodology

3.3.1 Options

❖ Software Options

1. Java Programming Language.
2. C# Programming Language.
3. Microsoft SQL Database.
4. Oracle Database.
5. Android OS.
6. SQLite Database.

❖ Hardware Options

1. Microcontrollers.
2. Data Transfer.
3. GPS Receiver.
4. Wireless communication

3.3.2 Analysis of each Option

3.3.2.1 Java Programming Language

❖ Java is an object-oriented language

Java is a true object-oriented language. It uses the principles of abstraction, encapsulation and inheritance. It is largely based on the notion of classes and

requires the developer to structure data, something that is not necessarily true in the case of C++ development^[9]

❖ **Java and multithreading**

Java language makes it very easy to develop concurrent processes, which execute simultaneously within an application. This is called multithreading. This multithread mechanism provides rich functionality to applications. A user may, for example, want to fill in a form while printing a document. Multithreading also lets Java function on multi-processor machines without too many problems adapting. Other languages provide the multithreading mechanism, but they use non-standard external libraries to accomplish this. Java integrates this capability directly into the language, which provides greater portability with Java programs.

❖ **Java manages its runtime memory**

Java has a garbage collector, which is a mechanism that clears memory of all objects that are no longer being used. In C and C++ programs, developers must handle the destruction of created objects themselves. This method of managing memory in these programs requires many lines of code and a lot of fine-tuning. The pointer concept no longer exists in Java.

❖ **Java is secure**

All Java applications run in a secure environment. The virtual machine performs very strict verification of Java code before it is executed. Code cannot bypass the protection mechanisms imposed by the language. A class cannot for example access a field of another class that has been declared private. The code also cannot try to define pointers to directly access memory. Applications must not cause a stack to overflow.

3.3.2.2 Difference between C# and Java

Java is a programming language, which has been influenced by the C language. It derives much of its syntax from C and C++, however it has fewer low-level facilities than either. Java is a general-purpose programming language that is designed to have fewer implementation dependencies as compared to previous languages. It is concurrent, class-based, and an object-oriented language.

In the other hand, C#, also known as C sharp, is an object oriented programming language. C# is a multi-paradigm programming language. It includes strong typing, imperative, declarative, functional, procedural, generic, object-oriented (class-based), and component-oriented programming disciplines.^[10]

Main different between java and C#:

- C# contains more primitive data types than Java.
- C# supports operator overloading while Java does not.
- C# supports the struct type while Java does not and All data types in C# are struct.
- C# allows parameters to be passed by reference by using the ref keyword, whereas Java allows parameters to be passed by value.
- C# allows variable number of parameters using the param keyword.
- Declaration of arrays is different in C# as compared to Java.
- The convention for Java is to put one public class in each file and some compilers require this. C# allows any source file arrangement.
- Java does not support events and delegates.
- C# doesn't have checked exceptions.
- Java doesn't allow the creation of user-defined value types.
- Java doesn't have iterator blocks for simple implementation of iterators.
- Java doesn't have expression trees.

3.3.2.3 Oracle Database

A database management system (DBMS) is software that controls the storage, organization, and retrieval of data. Typically, a DBMS has the following elements:^[11]

- **Kernel code**

This code manages memory and storage for the DBMS.

- **Repository of metadata**

This repository is usually called a data dictionary.

- **Query language**

This language enables applications to access the data.

3.3.2.3.1 Database Storage Structures

- **Data blocks**

At the finest level of granularity, Oracle Database data is stored in data blocks. One data block corresponds to a specific number of bytes on disk.

- **Extents**

An extent is a specific number of logically contiguous data blocks, obtained in a single allocation, and used to store a specific type of information.

- **Segments**

A segment is a set of extents allocated for a user object (for example, a table or index), undo data, or temporary data.

- **Table spaces**

A database is divided into logical storage units called table spaces. A table space is the logical container for a segment. Each table space contains at least one data file.

3.3.2.4 Difference between Oracle and MySQL

The Oracle Database is an object-relational database management system (ORDBMS). It is commonly referred to as Oracle RDBMS or simply as Oracle. Software Development Laboratories (SDL) developed the original version of the Oracle software^[12]

MySQL is an open source relational database management system (RDBMS). MySQL is officially pronounced as "My S-Q-L", but it is also called "My Sequel". It is named after co-founder Michael Widenius' daughter, MySQL stands for Structured Query Language. MySQL is the world's most used RDBMS, and runs as a server providing multi-user access to a number of databases. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, which is now owned by Oracle Corporation MySQL is a popular choice of database for use in web applications.

Main different between Oracle and MySQL:

- Oracle supports the creation of programs that are embedded within the database by way of a procedural language and can be executed independently or triggered by certain events.
- Oracle is better for large scale deployments as it has extensive capabilities.
- Most editions of oracle have high licensing costs, which are needed in order to use the software. Oracle Express is freely available.
- MySQL is a free open source program; hence it is in easy reach of ordinary web publishers and small businesses.
- MySQL is provided under the GNU GPL license that basically means that anyone can use as long as any other work derived from it is shared under the same license.
- Oracle is often limited to huge companies.
- Oracle Support PL-SQL, in addition to SQL. MySQL support only SQL
- Oracle provides security features such as row lock while MySQL provides column lock.
- Oracle requires username, password, and profile validation at the time of logging while MySQL requires only username, password and host.
- Oracle9i Database supports the use of temporary tables for an individual session, or global to all users.
- MySQL is light, reliable and can connect to multi-Client Program.
- MySQL does not support data partitions, and requires a server for each set of data files. Scalability is limited to the size of a single server.
- Compared to Oracle, MySQL does not have Tablespace, Role management, snapshots, synonym and packages.

3.3.2.5 Microcontrollers

3.3.2.5.1 PIC

The PIC is one of the well-known microcontrollers. It features all the components which modern microcontrollers normally have. For its low price, wide range of application, high quality and easy availability, it is an ideal solution in applications such as: the control of different processes in industry, machine control devices, measurement of different values.^[13]

3.3.2.5.2 Arduino

Arduino is an open source board microcontroller, Arduino is designed to make electronic more accessible to artists, designer, hobbyists and anyone interested in creating interactive objects or environments.^[14]

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, and MaxMSP).

We want to use Arduino in our project because the PIC is a chip, while the Arduino is a complete circuit board with power supply, IO headers, easier to deal with, and easier in programming.

3.3.2.6 Data Transfer

3.3.2.6.1 SMS (Short Message Service)

SMS has limited message size, no more than about 160 bytes after headers, but this may not be a problem for most applications where the device simply sends location and a few other data items. Most service plans charge by the message, favoring infrequent (for example, once-per-day) reporting scenarios. But SMS messaging has more robust coverage and will continue to work on the fringe of rural coverage and other weak-signal scenarios when GPRS will not^[15]

3.3.2.6.2 GPRS (General packet radio service)

GPRS has many advantages: It supports standard IP protocols, so the same tools and infrastructure that have been built to support the Internet can be used in your GPRS-based solution. Also, lots of people are familiar with these technologies, so the pool of technical talent is large^[4]

Since most service plans charge “by the byte,” GPRS is probably a better choice for higher-frequency reporting scenarios (for example, hourly or whenever the device moves a certain distance).

Typically, developers have more options to optimize and reduce cost with GPRS than they do with SMS so we want to use it.

3.3.2.7 Wireless Communication

3.3.2.7.1 Wi-Fi

Wi-Fi is a direct replacement for a wired Ethernet cable and is used in the same situations to avoid running wires everywhere.

The benefit of Wi-Fi is that it can connect to an existing network hub or router, which means that a PC doesn't have to be left on to access a device using Wi-Fi. Remote access products like IP cameras use Wi-Fi so they can be connected to a router and accessed across the Internet. Wi-Fi is useful but not simple to implement unless you just want to connect a new device to your existing network^[16]

Specification:

Data rate	11 and 54 Mbit/sec
Range	50-100 meter
Networking topology	Point to hub
Complexity (device and application impact)	High
Power consumption (battery option and life)	High
Operating frequency	2.4 and 5 GHZ
Other information	device connection require up to 3-5 second
Typical application	Wireless LAN connectivity, broadband internet access

Table 3.5: Wi-Fi Specification

3.3.2.7.2 Bluetooth

Bluetooth is generally used for point to point communication, although Bluetooth networks can be established quite easily. Typical applications we are all familiar with allow data transfer from mobile phones to PCs. Bluetooth wireless is the best solution for these point to point links, as it has high data transfer rates and, with the right antenna, very long ranges of up to 1KM in ideal circumstances.

The commonest application we deal with is replacement of serial cables by using a serial to Bluetooth converter on one end e.g. solar panel array, and a USB to Bluetooth adapter to connect to a laptop or PC on the other end. These types of link are very easy to setup, often by just pressing a pairing button on the units to create a permanent Bluetooth link^[16]

Bluetooth can also be used to create small ad-hoc networks, often with one USB to Bluetooth convertor as the master and up to 4 serial to Bluetooth adapters as slaves.

Specification:

Data rate	1 Mbit/sec
Range	10 meter
-networking topology	Ad-hoc ,very small networks

-complexity (device and application impact)	High
- Power consumption (battery option and life)	medium
-operating frequency	2.4 GHZ
Other information	device connection require up to 10 second
- Typical application	Wireless connectivity, between devices such as phones ,PDA , laptops , headsets

Table 3.6: Bluetooth Specification

3.3.2.7.3 Zigbee

ZigBee is the only standards-based wireless technology designed to address the unique needs of low-cost, low-power wireless sensor and control networks in just about any market. Since ZigBee can be used almost anywhere, is easy to implement and needs little power to operate, the opportunity for growth into new markets, as well as innovation in existing markets, is limitless. Here are some facts about ZigBee:^[1]

- With hundreds of members around the globe, ZigBee uses the 2.4 GHz radio frequency to deliver a variety of reliable and easy-to-use standards anywhere in the world.
- Consumer, business, government and industrial users rely on a variety of smart and easy-to-use ZigBee standards to gain greater control of everyday activities.
- With reliable wireless performance and battery operation, ZigBee gives you the freedom and flexibility to do more.

- ZigBee offers a variety of innovative standards smartly designed to help you be green and save money.

3.4 Project Components

3.4.1 Software Component

3.4.1.1 C# Programming Language

The C# language is intended to be a simple, modern, general-purpose, object-oriented programming language. The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.^[17]

3.4.1.1.1 Inheritance in C#

Inheritance is the ability to create a class from another class, the "parent" class extending the functionality and state of the parent in the derived, or "child" class. It allows derived classes to overload methods from their parent class.

Inheritance is one of the pillars of object-orientation. It is the mechanism of designing one class from another and is one of the ideas for code reusability, supporting the concept of hierarchical classification. C# programs consist of classes,

where new classes can either be created from scratch or by using some or all properties of an existing class.

Another feature related to inheritance and reusability of code is polymorphism, which permits the same method name to be used for different operations on different data types.

3.4.1.1.2 Interfaces

An INTERFACE in C# is a type definition similar to a class, except that it purely represents a contract between an object and its user. It can neither be directly instantiated as an object, nor can data members be defined. So, an interface is nothing but a collection of method and property declarations.

3.4.1.1.3 Garbage Collector

The garbage collector is a background process running within your program. It is always present within all .NET applications. Its job is to look for objects (i.e. reference types) which are no longer being used by your program. If the object is assigned to null, or the object goes out of scope, the garbage collector will mark the object to be cleaned up at some point in the future, and not necessarily have its resources released immediately.

3.4.1.2 Microsoft SQL Database

3.4.1.2.1 Architecture

MS SQL database contain the protocol layer which implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc.

3.4.1.2.2 Data storage

Data storage is a database, which is a collection of tables with typed columns. SQL Server supports different data types, including primary types such as Integer, Float, Decimal, Char (including character strings), Varchar (variable length character strings), binary (for unstructured blobs of data), Text (for textual data) among others. The rounding of floats to integers uses either Symmetric Arithmetic Rounding or Symmetric Round Down (Fix) depending on arguments: `SELECT Round(2.5, 0)` gives 3.

SQL Server ensures that any change to the data is ACID-compliant, it uses transactions to ensure that the data base well revert to a known consistent state. SQL Server implement transactions using a write-ahead log.

SQL Server allows multiple clients to use the same database concurrently. As such, it needs to control concurrent access to shared data, to ensure data integrity-

when multiple clients update the same data, or clients attempt to read data that is in the process of being changed by another client.

3.4.1.3 Android (Operating System)

Android is a Linux-based operating system, designed primarily for touch screen mobile devices such as smart phones and tablets. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. The first Android-powered phone was sold in October 2008.

- **Android is characterized by:**

- * **Google maps:**

- It is a web mapping service application and technology provided by Google, that powers many map-based services, including the Google Maps website, Google Ride Finder, Google Transit, and maps embedded on third-party websites via the Google Maps API. It offers street maps, a route planner for traveling by foot, car, bike (beta), or with public transportation and a locator for urban businesses in numerous countries around the world.

- * **Google Play:**

- It is formerly known as Android Market, anybody can use it easily to publish and find applications directly accessible from the phone and the web.

- * **Open-source community.**

3.4.1.4 SQLite Database

SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. SQLite is available on every Android device. We can use it in Android does not require any database setup or administration and access to a SQLite database involves accessing the file system.

3.4.2 Hardware Components

3.4.2.1 Zigbee

ZigBee is used in applications that require a low data rate, long battery life, and secure networking. ZigBee has a defined rate of 250 Kbit/s, best suited for periodic or intermittent data or a single signal transmission from a sensor or input device. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range wireless transfer of data at relatively low rates. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth or Wi-Fi.

Because ZigBee nodes can go from sleep to active mode in 30ms or less, the latency can be low and devices can be responsive, particularly compared to Bluetooth wake-up delays, which are typically around three seconds. Because ZigBee nodes can sleep most of the time, average power consumption can be low, resulting in long battery life.^[1]

3.4.2.2 GPS

A GPS navigation device is a device that receives Global Positioning System (GPS) signals to determine the device's location on Earth. GPS devices provide latitude and longitude information, and some may also calculate altitude, although this is not considered sufficiently accurate or continuously available enough (due to the possibility of signal blockage and other factors) to rely on exclusively to pilot aircraft. GPS devices are used by the military, by aircraft pilots, by sailors, and for recreational purposes by the public.^[2]

GPS receivers use signals from satellites which orbit the earth at known positions. Each satellite has a unique identification code and sends a signal which the GPS receiver software can use to calculate the distance from the device to those satellites.

GPS receivers have many types:

❖ Type 1:

Not-self-contained receivers (without screen), also known as RS232 receivers or also GPS mice. This type needs a computer (often a Pocket-PC or Palm PDA) and according to the program in order to visualize the actual position of the GPS receiver. The link between the GPS and the computer can be wireless (Bluetooth), by means of a cable or via a card-slot or sleeve.

❖ Type 2:

Self-contained receivers (with screen). Here the computer is integrated in the GPS receiver. It is a mapping device which can be handheld or mounted in boat, car or plane.

❖ Type 3:

The more sophisticated receivers for professional use in agriculture, mapping and GIS, military, oil and gas, public safety, survey etc. The principle of these

receivers is the same as the above, but they often use extra receivers and special antennas for higher accuracy and can often store an enormous amount of in the field collected data, which can be treated later in the office.

❖ **Type 4:**

Android device, PDA's, and Phones with incorporated GPS receiver built in. these devices used for in-car navigation system.

3.4.2.3 GPRS modem

The word modem is derived from the words modulator demodulator. This device functions by enabling most computers to transmit data to different networks, such as mobile phones or other computers. A general packet radio service (GPRS) modem is another global system for mobile communications (GSM) modem that also supports wireless data transmission using GPRS technology. A GSM modem generally uses a circuit-switched type of technology in transmitting data, while the GPRS modem utilizes packet-switched technology to do the same task. This frequently results in faster transmission of data over a GPRS connection when using the GPRS modem.

The basic functions of a GPRS modem include wireless data communication and integration with several applications that require universal serial bus (USB) connections. Between a GPRS cell phone and a standalone GPRS modem, it is frequently better to use a GPRS modem in data transmission. In terms of sending and receiving SMS or MMS, there is negligible difference between the two devices. They both use the same rates and the same speed in transmitting messages.

3.4.2.4 Arduino

Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as shields. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus, allowing many shields to be stacked and used in parallel. Official Arduinos have used the megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560.^[14]

An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer.

There are many Arduino-compatible and Arduino-derived boards. Some are functionally equivalent to an Arduino and may be used interchangeably. Many are the basic Arduino with the addition of commonplace output drivers, often for use in school-level education to simplify the construction of buggies and small robots. Others are electrically equivalent but change the form factor, sometimes permitting the continued use of Shields, sometimes not. Some variants use completely different processors, with varying levels of compatibility.

3.4.2.5 ZigBee and other competitors

Wireless sensor networks are one of the most rapidly growing technologies and has a wide variety of applications. As a wireless technology, ZigBee becomes popular in the recent years due to its ultra-low power consumption. However, there are many wireless technologies that are competing with the ZigBee such as Wi-Fi and Bluetooth.

Most researchers prefer to exploit Wi-Fi technology in their projects due its availability in almost everywhere, and in every wireless device (such as laptops, smart phones, tablets and so on). A typical wireless router uses 802.11b or 802.11g with a stock antenna might have a communication range of 32 m indoors and 95 m outdoors. Wi-Fi transceivers are very suitable for applications that requires high data rate and/or an internet connection.

On the other hand, Bluetooth is primarily designed for low power consumption and short ranges communication (1m and 10m). Bluetooth is not targeted for a specific application, but supports a multitude of applications. Therefore the technology has been adopted by wide variety of devices, including computers, cell phones, headsets, PDAs and cars.

3.4.2.6 Why we chose ZigBee:

In our view ZigBee is the most appropriate wireless communication technologies for our project. There are many reasons why that is so, some of them are:

- ❖ **High Reliability:**

ZigBee is based on a solid global standard 'IEEE 802.15.4 standard' and it uses modulation technologies like O-QPSK, DSSS and CSMA/CA which are among the best in the respective domains. So our project tends to be more reliable and easy to demonstrate anywhere.

❖ **Easy to Use:**

Most ZigBee modules can directly be interfaced to microcontrollers, processors and even computers. And they available with a direct UART communication interface; this makes communication even easier.

❖ **Easy Availability:**

These days many ZigBee modules are available easily and made by various manufacturers so it is easily accessible and it's not worthless to us to use it.

❖ **Infinite Possibilities:**

ZigBee is the most flexible communication technology and it can be used for any type of wireless communication like simple point to point communication, Star network, Mesh Network, Tree Networks because ZigBee protocol has flexible architecture.

❖ **ZigBee has good data rate and low cost.**


3.5 Summary

In this chapter, the project management plan have been done, the project methodology have been discussed too. At the end of the chapter, we have discussed the project components and we made our choices of the components to be used in the project.

A dark blue vertical bar is on the left side of the page. A blue arrow points to the right from the bar, pointing towards the chapter title.

Chapter Four

Software Requirement Specification (SRS)

- 
- Several thin, curved lines in shades of blue and grey originate from the bottom left corner and extend upwards and to the right, crossing the table of contents items.
- 4.1 Overview
 - 4.2 Requirement Description
 - 4.3 Class Responsibilities Collaborator Modeling
 - 4.4 Class Hierarchies and Relationships
 - 4.5 Conclusion

4.1 Overview

In this Chapter, It is describe the requirements of our project, making simple scenarios to show how the project works. Also we are going to show in deeper details how can any user interact with the project. An exact explanation will be presented to show the exact interaction between the project components internally and externally.

4.2 Requirement Description

4.2.1 Actors

4.2.1.1 Administrator

1. Logging on and access the system data.
2. Maintain the system.
3. Monitor the system.

4.2.1.2 Policeman

1. Tracing stolen car on a map.
2. Receive message.
3. Check driving license.
4. Write violation report for specific car.

4.2.1.3 Driver

1. Monitor street state.
2. Receive a violation message.

4.2.2 Usage Scenario

4.2.2.1 Administrator

4.2.2.1.1 Logging on and access the system data

Table 4.1: Logging on and access the system data

Use case	Logging on and access the system data
Primary actor	Administrator
Goal In Context	Enable the admin to access the system and data
Preconditions	Logging on as administrator.
Trigger	The admin want to use data
Scenario	<ol style="list-style-type: none">1. The administrator enters his/her username and password (Logging on as administrator) on the server.2. The system displays the Administrator interface.3. The administrator chooses to display data on the screen.4. The system preview a list's of cars and drivers data.5. Then the admin can delete, add and update the data.
Exceptions	The admin username / password invalid

4.2.2.1.2 Maintain the system

Table 4.2: Maintain the system

Use case	Maintain the system
Primary actor	Administrator
Goal In Context	The admin able to maintain the system state
Preconditions	Logging on as administrator
Trigger	The admin want to maintain the system
Scenario	<ol style="list-style-type: none">1. The administrator enters his/her username and password (Logging on as administrator) on the server.2. The system displays the Administrator interface.3. The administrator checks system performance by adding and deleting data on the data base.4. The administrator also checks the GSM/GPRS transceiver performance by sends and receives messages.
Exceptions	<ul style="list-style-type: none">✓ Error in database access.✓ Fault in GSM/GPRS modem.

4.2.2.1.3 Monitor the system

Table 4.3: Monitor the system

Use case	Monitor the system
Primary actor	Administrator
Goal In Context	The admin able to monitor the system

Preconditions	Logging on as administrator
Trigger	The admin want to monitor data flow
Scenario	<ol style="list-style-type: none"> 1. The administrator enters his/her username and password (Logging on as administrator). 2. The system displays the administrator interface. 3. The administrator can display and see the data flow to and from the server. 4. The administrator monitors the street traffic and able to send messages policeman.
Exceptions	<ul style="list-style-type: none"> ✓ Error in database access. ✓ Fault in GSM/GPRS modem.

4.2.2.2 Policeman

4.2.2.2.1 Tracing stolen car on a map

Table 4.4: Tracing stolen car on a map

Use case	Tracing stolen car on a map
Primary actor	Policeman
Goal In Context	Enable the policeman to trace and monitor a specific car using android maps.
Preconditions	<ul style="list-style-type: none"> • The android application installed on the android phone. • Logging on the android application and enter car ID.
Trigger	The Policeman want to catch the car

Scenario	<ol style="list-style-type: none"> 1. Policeman enters his username and password (Logging on android application) on his phone. 2. The policeman chooses to trace a car. 3. The application require car ID to be traced and policeman approve car ID. 4. The android application requests the coordinate from the server. 5. The application starts to receive the coordinate data. 6. The car location displayed on the android map showing it is coordinate and location.
Exceptions	<ul style="list-style-type: none"> ✓ The policeman username or password invalid. ✓ The policeman enter wrong car ID. ✓ The car coordinate not stored on the server. ✓ The server not response.

4.2.2.2.2 Receive a messages

Table 4.5: Receive a messages

Use case	Receive a messages
Primary actor	Policeman
Goal In Context	The policeman able to receive a message about violation on the street.
Preconditions	<ul style="list-style-type: none"> • The android application installed on the android phone.

	<ul style="list-style-type: none"> • Logging on the android application.
Trigger	Violation on the street.
Scenario	<ol style="list-style-type: none"> 1. Policeman mobile phone turned on. 2. The android application receives the message and displays it on the screen. 3. The type of violation and the street location also contained. 4. The policeman can see the violation on the map screen.
Exceptions	Error in application will open the message.

4.2.2.2.3 Check driving license

Table 4.6: Check driving license

Use case	Check driving license
Primary actor	Policeman
Goal In Context	The policeman can check the driver license.
Preconditions	<ul style="list-style-type: none"> • The android application installed on the android phone. • Logging on the android application.
Trigger	The driver does not apply the rule
Scenario	<ol style="list-style-type: none"> 1. Policeman enters his username and password (Logging on android application) on his phone. 2. The server reply with a message about driver license expired date.

Exceptions	<ul style="list-style-type: none"> ✓ The policeman username or password invalid. ✓ The policeman enter wrong driver ID. ✓ The message lost will opening. ✓ The server not response.
------------	---

4.2.2.2.4 Write violation report for specific car

Table 4.7: Write a violation report

Use case	Report a specific car
Primary actor	Policeman
Goal In Context	Enable the policeman write a report and send it to the server.
Preconditions	<ul style="list-style-type: none"> • The android application installed on the android phone. • Logging on the android application.
Trigger	The driver does not apply the street rule or violation by driver.
Scenario	<ol style="list-style-type: none"> 1. Policeman enters his username and password (Logging on android application) on his phone. 2. The policeman chooses to write a report for the car. 3. The application displays the report forum. 4. The application requires the driver ID or car ID and violation type ...etc. 5. The android application sends the report to the server. 6. The server analyzes the message and stores the

	report in the database.
Exceptions	<ul style="list-style-type: none"> ✓ The policeman username or password invalid. ✓ The policeman enter wrong driver ID or car ID. ✓ The report message lost through the network. ✓ The server not response. ✓ The GSM/GPRS module not plug in server.

4.2.2.3 Driver

4.2.2.3.1 Monitor the street state

Table 4.8: Monitor the street state

Use case	Monitor the street state
Primary actor	Driver
Goal In Context	The driver can monitor the street state on a map.
Preconditions	<ul style="list-style-type: none"> • The android application installed on the android phone. • Logging on the android application.
Trigger	The driver wants to see the street state.
Scenario	<ol style="list-style-type: none"> 1. The driver enters his/her username and password (Logging on android application) on his phone. 2. The driver sends a request to the server to monitor the street. 3. The server sends the coordinate data to the driver device. 4. A map with bookmarks and traffic violation

	displayed on the application.
Exceptions	<ul style="list-style-type: none"> ✓ The driver username or password invalid. ✓ The server not response. ✓ The application does not receive the coordinate data from the server.

4.2.2.3.2 Receive a violation message

Table 4.9: Receive a violation message

Use case	Receive a violation message
Primary actor	Driver
Goal In Context	The driver can receive a violation message.
Preconditions	<ul style="list-style-type: none"> • The android application installed on the android phone. • Logging on the android application.
Trigger	Violation done by the driver.
Scenario	<ol style="list-style-type: none"> 1. Driver mobile phone turned on. 2. The android application receives the violation message and displays it on the screen. 3. The type of violation and the street location also contained.
Exceptions	Error in application will open the message.

4.2.3 Use Case Diagram

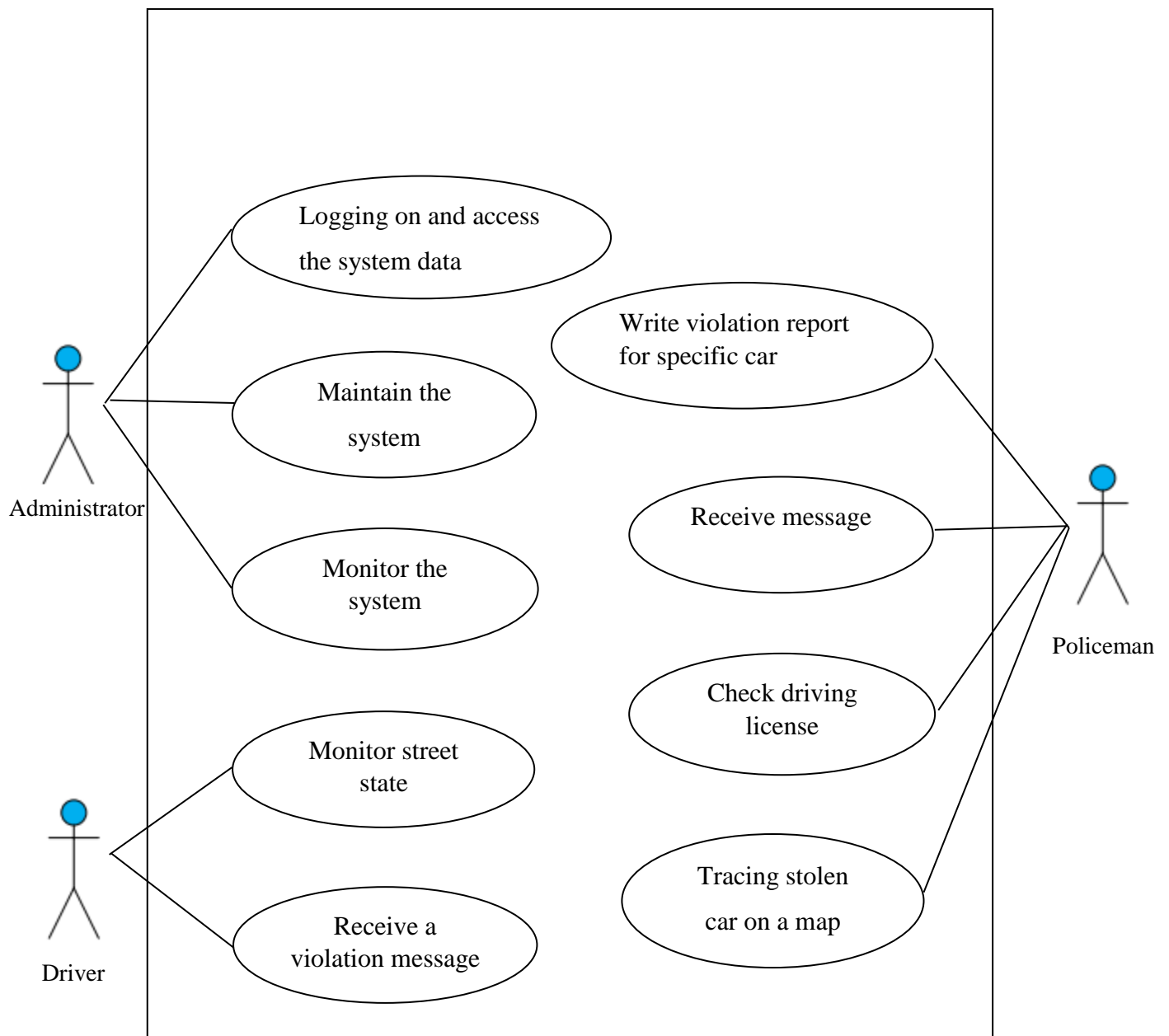


Figure 4.1: Use Case Diagram

4.3 CRC (Class Responsibilities Collaborator) Modeling

The classes divided into three categories: one related to sever and the other related to the driver and policeman android application.

4.3.1 Server Classes

4.3.1.1 Admin Class

Class: Admin	
Description: The main class in the server side, also responsible for doing the most important functions.	
Responsibility	Collaborator
Login The System	
Add/Modify/Delete Car	Car
Add/Modify/Delete Driver	Driver
Add/Modify/Delete Violation	Violation
Check Data Base Connection and data	
Monitor Data Flow	

4.3.1.2 GSM/Serial Class

Class: GSM/Serial	
Description: GSM/Serial class represents the relation between the server the GSM/GPRS module.	
Responsibility	Collaborator
Open Serial Port	
Send Data to SerialP	Message
Receive Data from SerialP	Message
Close Serial Port	

4.3.1.3 Violation Class

Class: Violation	
Description: Used to cluster all the violation of different type under main class called violation and provide a function that work on the violation data.	
Responsibility	Collaborator
Add Violation	
Edit Violation	
Delete Violation	

4.3.1.4 Car Class

Class: Car	
Description: The class used to store and retrieve cars information the server.	
Responsibility	Collaborator
Is Car licensed	
Get Car Information	
Add Car	
Edit Car	
Delete Car	

4.3.1.5 Message Class

Class: Message	
Description: The message class carries the message information to be transmitted or received.	
Responsibility	Collaborator
Initiate	
Send Message	
Receive Message	
Edit Message	

Delete Message	
----------------	--

4.3.1.6 Driver Class

Class: Driver	
Description: Used to interact with the driver information stored in the server database.	
Responsibility	Collaborator
Get Driver Information	
Check Driver License	
Add Driver	
Delete Driver	
Update Driver	

4.3.2 Driver Android Application Classes

4.3.2.1 Map Class

Class:Map	
Description: mainly the class used to initialize the map onon the driver android application screen.	
Responsibility	Collaborator

Initialize The Map	
Initialize Map Location	
Set Up My Location	
Get Route Directions	
Add/Clear Markers/Map	

4.3.2.2 Socket Class

Class: Socket Class	
Description: The class required by driver android application to interact with the server usually sends to and receivefrom server.	
Responsibility	Collaborator
Establish Connection	
Receive From Server	
Send To Server	
Close Connection	

4.3.2.2 Driver Class

Class: Driver Class
Description: The class required by driver android application to interact with the socket and map and interfaces.

Responsibility	Collaborator
Connection Management	
Controlling Maps	
Display Shortest Path	
Display Street State	
Update Street Status	
Display Violations	

4.3.3 Policeman Android Application Classes

4.3.3.1 Map Class

Class: Map	
Description: mainly the class used to initialize the map on the police android application screen.	
Responsibility	Collaborator
Initialize The Map	
Initialize Map Location	
Set Up My Location	
Get Route Directions	
Add/Clear Markers/Map	

4.3.3.2 Socket Class

Class: Socket Class	
Description: The class required by police android application to interact with the server usually sends to and receivefrom server.	
Responsibility	Collaborator
Establish Connection	
Receive From Server	
Send To Server	
Close Connection	

4.3.3.3 Policeman Class

Class: Driver Class	
Description: The class required by policeman android application to interact with the socket and map and interfaces.	
Responsibility	Collaborator
Connection Management	
Controlling Maps	
Display Shortest Path	
Display Black List Cars	

Display Street State	
Write Violation Report	
Trace Car	

4.4 Class Hierarchies and Relationships

4.4.1 Server Side Class Hierarchies and Relation

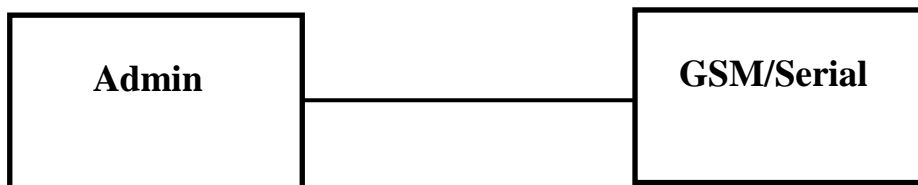


Figure 4.2: Admin/GSM Relation

The first relation between the admin and GSM/Serial classes (has knowledge of) as shown in Figure (4.2). The admin class use GSM/Serial as an interface between the server and the GSM/GPRS USB i.e. connected to the USB port. The admin can open, send, receive and close the serial port connection.

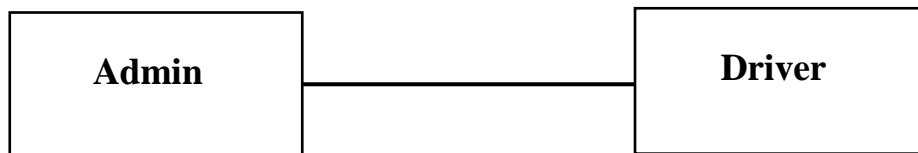


Figure 4.4: Admin/Driver Relation

The second relation between the admin and the driver classes (has knowledge of) as shown in Figure (4.4). The admin able to check driver license, add, delete, retrieve or update driver information.

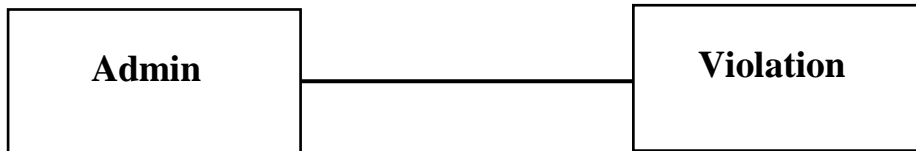


Figure 4.5: Admin/Violation Relation

The admin has a privilege to add, edit, delete violation, in addition to perform some operation like determine the cost, calculate the remains day for payment or number of violation. The relation described (has knowledge of) as shown in Figure (4.5).

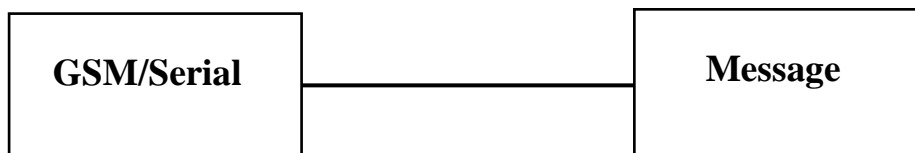


Figure 4.7: GSMSerial/Message Relation

The most important class GSM/Serial i.e. act as a link between the initiated message and the hardware component (GSM/GPRS USB). The message class used to initiate, send, receive or delete message. The relation described (has knowledge of) as shown in Figure (4.7).

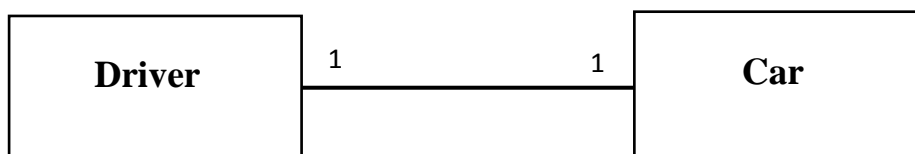


Figure 4.8:Driver/Car Relation

The relation in figure (4.8) is one to one i.e. the driver (person) is driving one car and the car driven by one person.

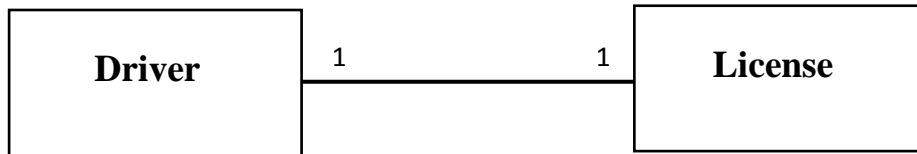


Figure 4.9: Driver/License Relation

In figure (4.9) the relation is one to one i.e. the driver has one license and the license owned by one driver.

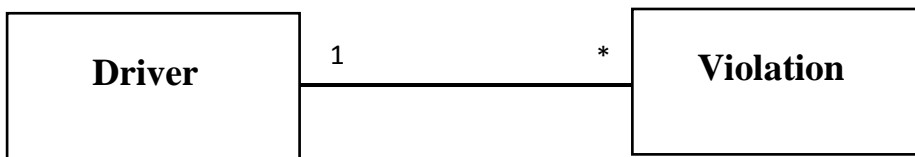


Figure 4.10: Driver/Violation Relation

In figure (4.10) the relation is one to many i.e. the driver has many violations and the violation for one driver.

4.4.2 Android Application Side Class Hierarchies and Relation

4.4.2.1 Policeman Application

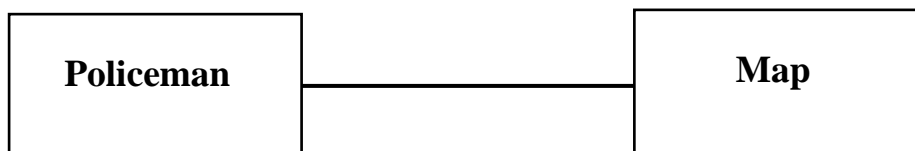


Figure 4.11: Policeman/Map Relation

The policeman use map class to initialize the map and put the markers on it and display the shortest path and trace a car i.e. the relation (has knowledge of) described in figure (4.11).

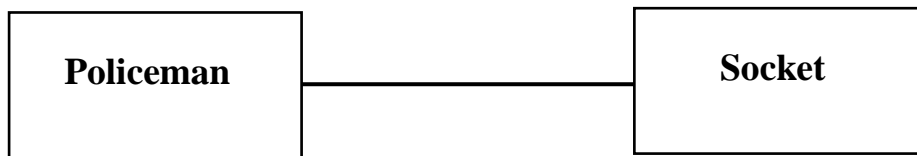


Figure 4.12: Policeman/Socket Relation

The policeman use class socket to connect to the server, send, receive request and terminate connection i.e. the relation (has knowledge of) described in figure (4.12).

4.4.2.2 Driver Application

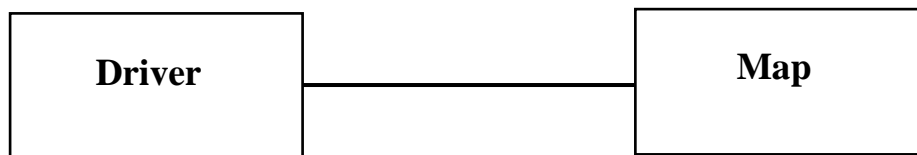


Figure 4.13:Driver/Map Relation

The driver use map class to initialize the map and put the markers on it and display the shortest path and trace a car i.e. the relation (has knowledge of) described in figure (4.13).

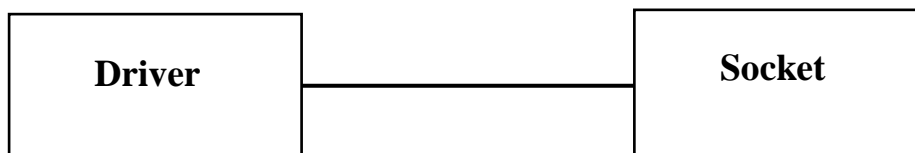


Figure 4.14: Driver/Socket Relation

The driver use class socket to connect to the server, send, receive request and terminate connection i.e. the relation (has knowledge of) described in figure (4.14).

4.5 Conclusion

In this chapter, we discussed the relationship between the project components and their interaction with the target users. A good explanation was shown in CRC (Class Responsibilities Collaborator) Modeling. Finally, class hierarchies and relationships have been listed.

Now, after we have discussed those things, and depending on the clear vision we had, we are able to go further and make a new step towards the design of our project that will be discussed in the next chapters.

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the number 5.

5

Chapter Five

Project Design

- 5.1 General Overview
- 5.2 Hardware components
- 5.3 Hardware sub-system design
- 5.4 Overall hardware system
- 5.5 Objects-Relational Model
- 5.6 State Behavioral Modeling
- 5.7 Software Subsystem Design
- 5.8 Class and Object Design
- 5.9 Software Interface Design
- 5.10 Conclusion

5.1 General Overview

In this chapter, we will describe the hardware and software design of our project, in hardware design we will describe our choices of the components we are going to use and make a deeper view on the chosen ones, make a good description of how to connect the devices with each other and show the overall system design.

Also we describe the software components and relations between them and the state diagram that show control and data flow. In addition to the explanation of each subsystem attributes, operations and relations with other subsystems. we show object interfacing and user interface design that's appear to the system user.

5.2 Hardware components

5.2.1 XBee and XBee Pro RF Modules.

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The modules operate within the ISM 2.4 GHz frequency band.



Figure 5.1: XBee RF Module

Figure 5.2: XBee Pro RF Module

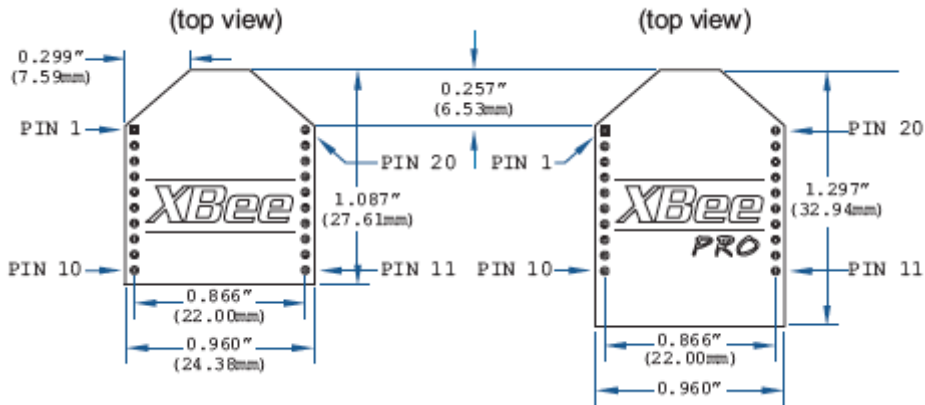


Figure 5.3: XBee and XBee Pro dimensions

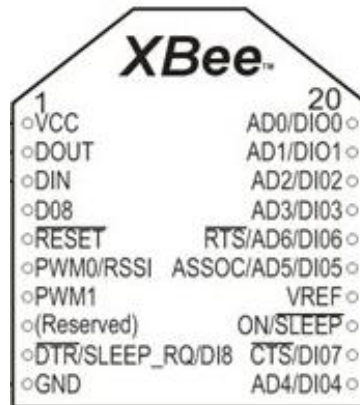


Figure 5.4: XBee schematic diagram

5.2.2 Arduino

The Arduino Mega is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB

cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila

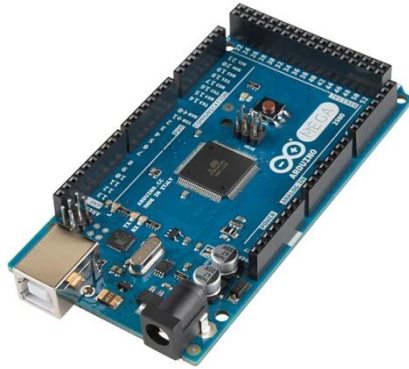


Figure 5.5: Arduino mega

5.2.3 GPS Receiver (20 Channel EM-406A SiRF III Receiver with Antenna)

This complete module is built upon the same technology as the ET-301, but includes on-board voltage regulation, LED status indicator, battery backed RAM, and a built-in patch antenna, 6-pin interface cable included.



Figure 5.6: GPS receiver

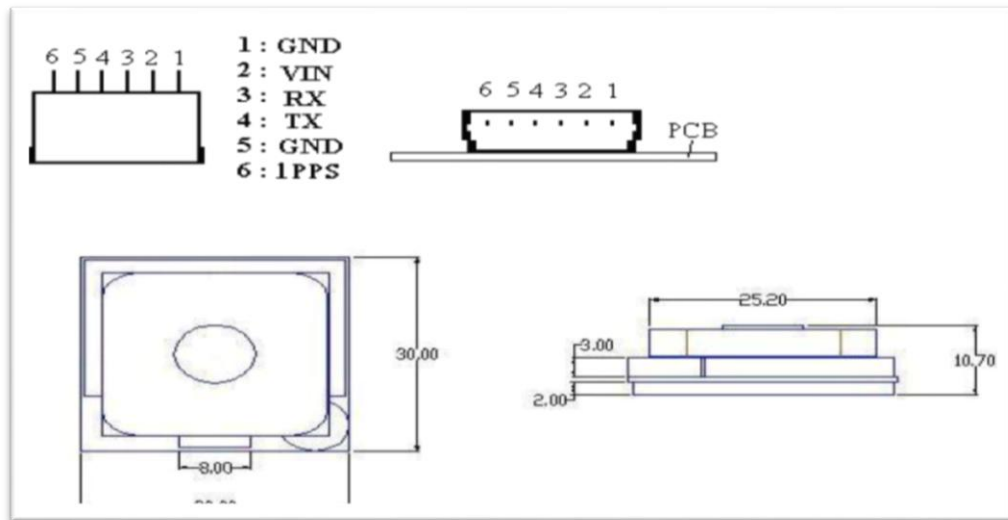


Figure 5.7: GPS receiver pin description

Pin description:

- VIN (DC power input): This is the main DC supply for a 4.5V ~6.5 DC input power
- TX: This is the main transmits channel for outputting navigation and measurement data to user's navigation software or user written software.
- RX: This is the main receive channel for receiving software commands to the engine board from SiRF demo software or from user written software.
- GND: Provides the ground for the engine board. Connect all grounds .
- 1PPS: This pin provides one pulse-per-second output from the engine board that issynchronized to GPS time.

5.2.4 GPRS Modems

5.2.4.1 GSM/GPRS Module - SM5100B



Figure 5.8: GPRS module .

The SM5100B is a miniature, quad-band GSM 850/EGSM 900/DCS 1800/PCS 1900 module, which can be integrated into a great number of wireless projects. You can use this module to accomplish cell phone can SMS text messages, GSM/GPRS connections and TCP/IP.

This module features two UARTS, an SPI interface, and two 10-bit ADCs. It also supports Li-ion battery charging, a 4x6 keypad, and an LCD interface. Inputs/outputs are available for a speaker and microphone. Power supplied to the module should be regulated between 3.3-4.2VDC (3.6V nominal).

5.2.5 Zigbee-Arduino Shield

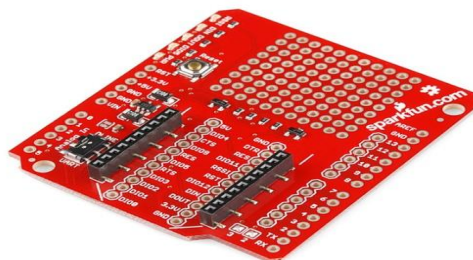


Figure 5.9: Zigbee Arduino shield

The shield form-factor mates directly with any board that has an Arduino standard footprint and equips it with wireless communication capabilities using the popular XBee module. This unit works with all XBee modules including the Series 1 and 2, standard and Pro versions.

The serial pins (DIN and DOUT) of the XBee are connected through an SPDT switch, which allows you to select a connection to either the UART pins (D0, D1) or any digital pins on the Arduino (D2 and D3 default). Power is taken from the 5V pin of the Arduino and regulated on-board to 3.3VDC before being supplied to the XBee. The shield also takes care of level shifting on the DIN and DOUT pins of the XBee. In the latest revision the diode level shifter is replaced with a more robust MOSFET level shifter.

The board also includes LEDs to indicate power and activity on DIN, DOUT, RSSI, and DIO5 pins of the XBee. The Arduino's reset button is brought out on the shield, and a 9x11 grid of 0.1" holes are available for prototyping.

5.2.6 GPRS shield (SM5100B Evaluation Board)



Figure 5.10: GPRS shield

This is an evaluation board for the SM5100B GSM/GPRS module. With FTDI's FT232RL connected up to UART0 of the SM5100B you can send AT commands from a PC terminal straight to the module.

This board comes as shown in the picture with the 60-pin SM5100B mating connector, SIM card socket, SPX29302 3.8VDC high-current voltage regulator, FT232RL USB-to-UART, Mini-B USB connector, and more supporting circuitry.

Features:

- 60-pin SM5100B mating connector populated on board
- Power from barrel jack regulated to 3.8VDC
- SM5100B UART0 connected to FT232RL UART-to-USB chip
- Mini-B USB connector integrated
- All pins of the SM5100B broken out to 0.1" pitch headers
- SIM socket integrated with SM5100B and populated on-board
- ON/OFF and Boot mode SPDT switches
- POWER ON momentary push-button switch
- Red power indicator LED
- RX, TX LEDs

5.2.7 XBee Explorer Regulated

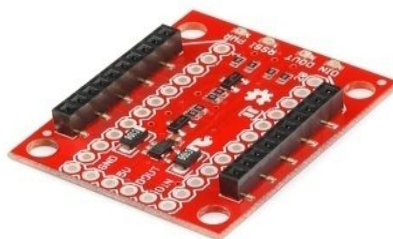


Figure 5.11: XBee explorer regulated

The XBee Explorer Regulated takes care of the 3.3V regulation, signal conditioning, and basic activity indicators (Power, RSSI and DIN/DOOUT activity LEDs). It translates the 5V serial signals to 3.3V so that you can connect a 5V (down to 3.3V) system to any XBee module.

This unit works with all XBee modules including the Series 1 and 2, standard and Pro versions. Plug an XBee into this breakout and you will have direct access to the serial and programming pins on the XBee unit and will be able to power the XBee with 5V.

5.3 Hardware sub-system design

In this section, we are going to divide the project into four sub-systems, describe each one individually, show the components in each one and how to construct it, additional block diagram will support the explanation of each sub-system.

5.3.1 Car sub-system

The car module will be considered as the first sub-system, this module will include a GPS receiver, XBee module, Arduino and its proper shield to make the connections. The main functionality is to take the GPS data, Air bag signal and any important signals from the car, make some processing in the Arduino, the Arduino output will be a packet including the calculated speed, the location information and the air bag sensor condition and some bytes will be reserved for future development if needed.



Figure 5.12: Car sub-system block diagram

5.3.2 The general fixed point

The general fixed point is the second sub-system in our project, it will include a Zigbee Pro module, an XBee explorer regulated and a 9-V battery. Its main functionality is to gather the packets from the on-road cars and deliver them to the core fixed point.



Figure5.13: General fixed node block diagram

5.3.3 Core fixed point

This is the third sub system in the project, it is developed general fixed point with an Arduino and a GPRS modem. Its main functionality is to gather the packets from the general fixed points and deliver them to the server.

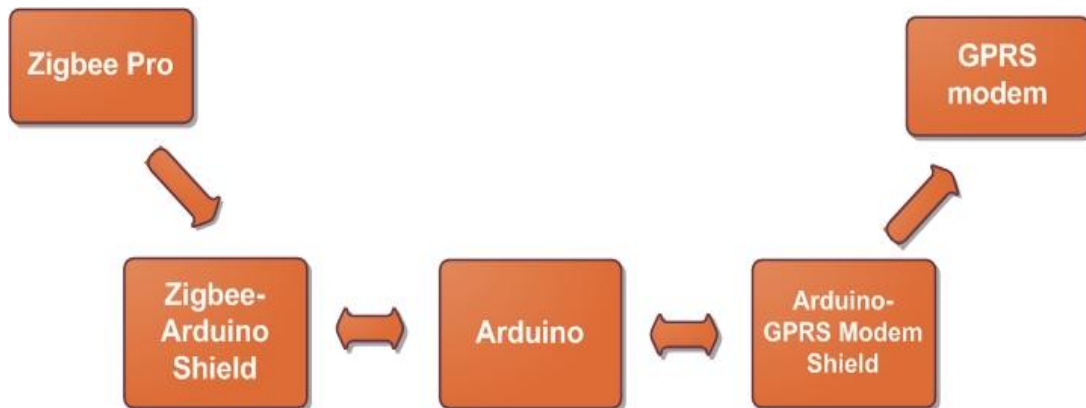


Figure 5.14: Core fixed node block diagram

5.3.4 Traffic light sub-system.

It is the fourth sub-system in our project, it will be a compensation of an Arduino and a GPRS modem, connected with a proper shield. This sub-system functionality will be controlling the traffic light, it will take its orders from the server, and these orders is to be delivered to the Arduino throw the GPRS modem.



Figure 5.15: Traffic light sub-system block diagram

5.4 Overall system

The overall system will be the interaction between the sub-systems described previously, Figures (6.21), (6.22), (6.23) describes the system throw a block diagram, an illustrative diagram and flowchart.

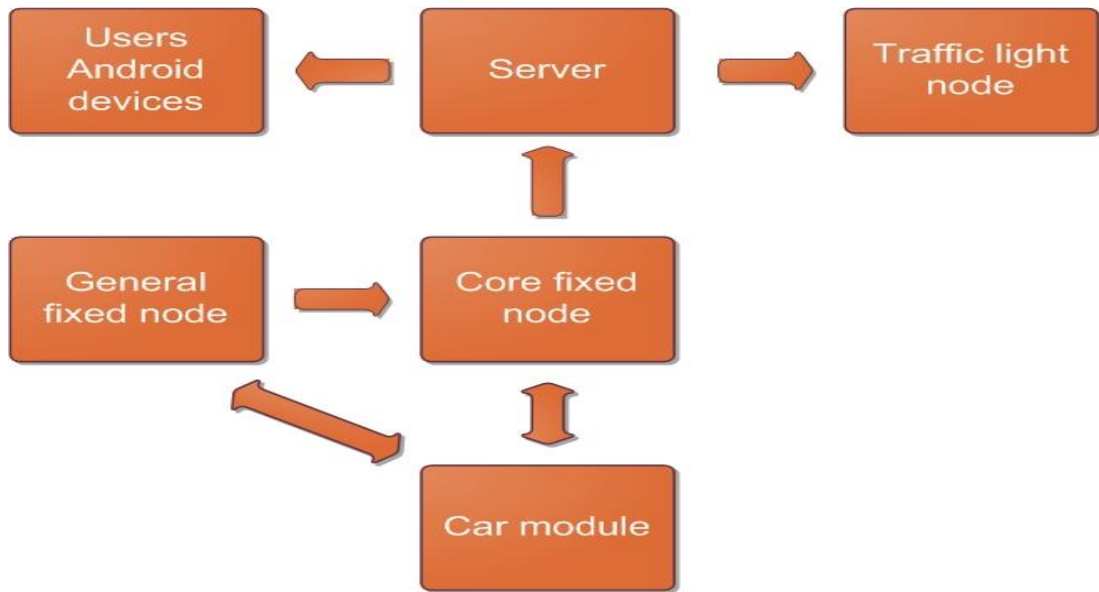
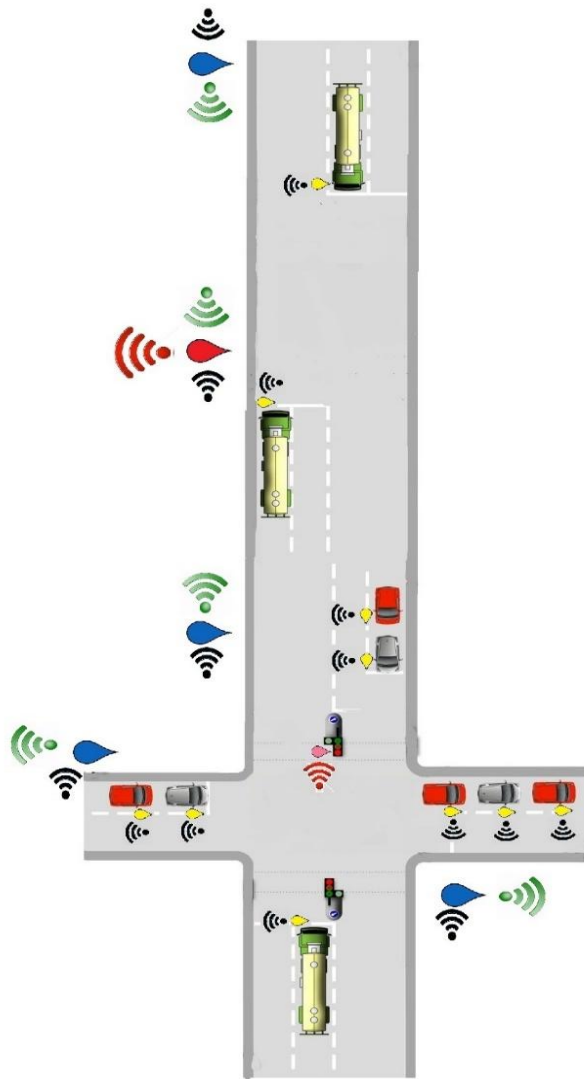
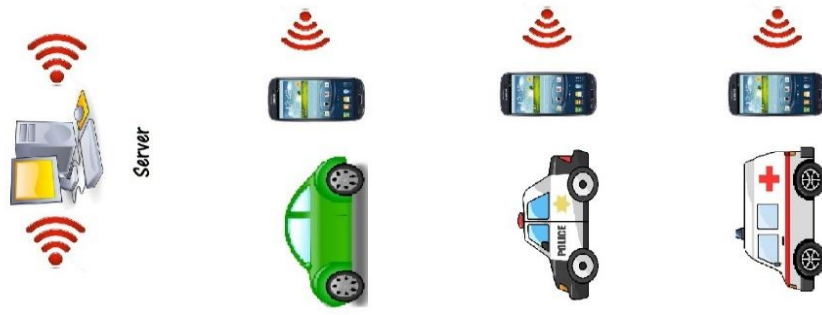









Figure 5.16: overall system block diagram



-  General fixed node
-  Core fixed node
-  Car module
-  Traffic light module

-  GPRS connection
-  Zigbee connection (fixed node to fixed node)
-  Zigbee connection (car module to fixed node)

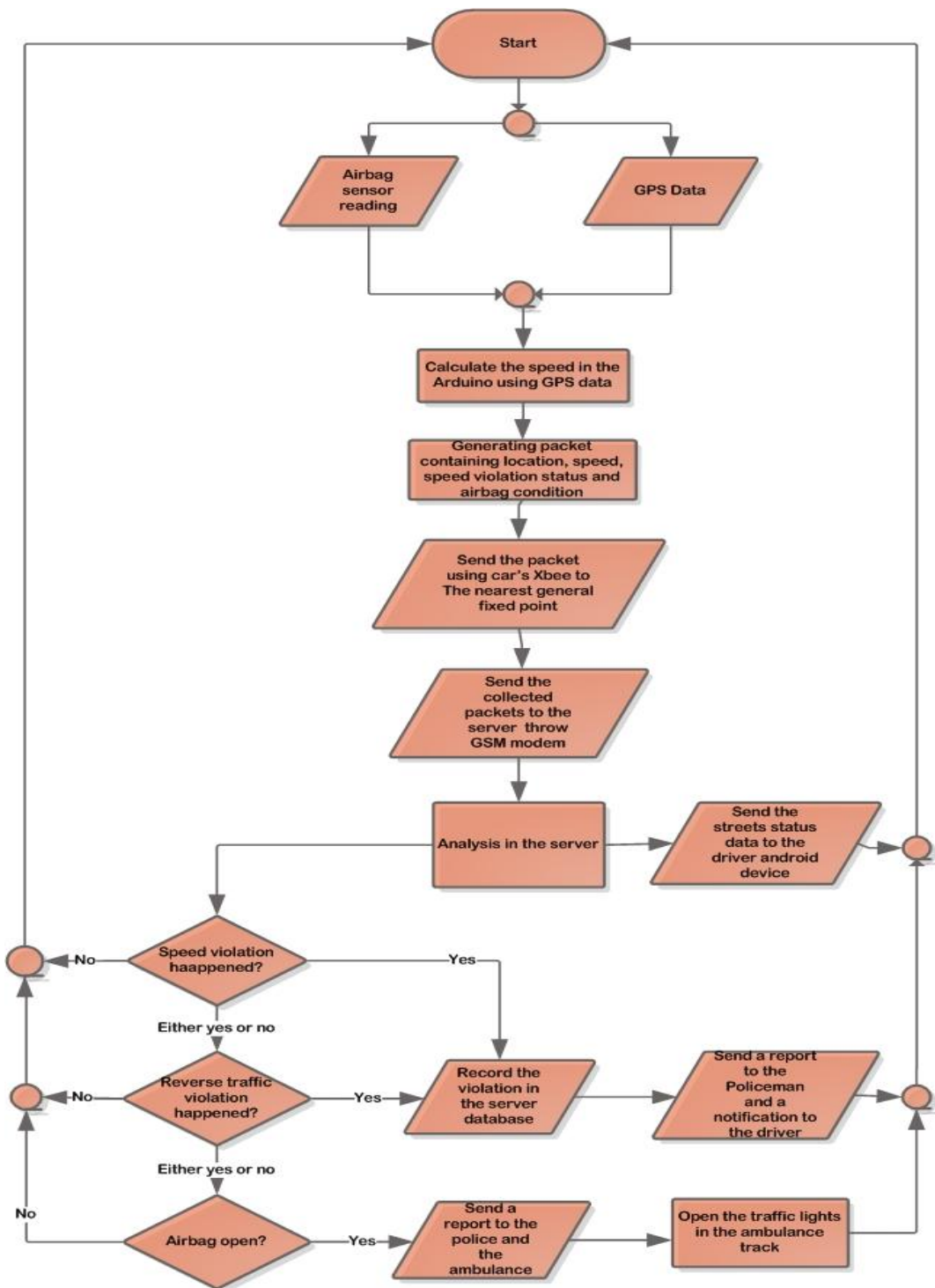


Figure 5.18: overall system flowchart

5.5 Objects-Relational Model

5.5.1 Server Side

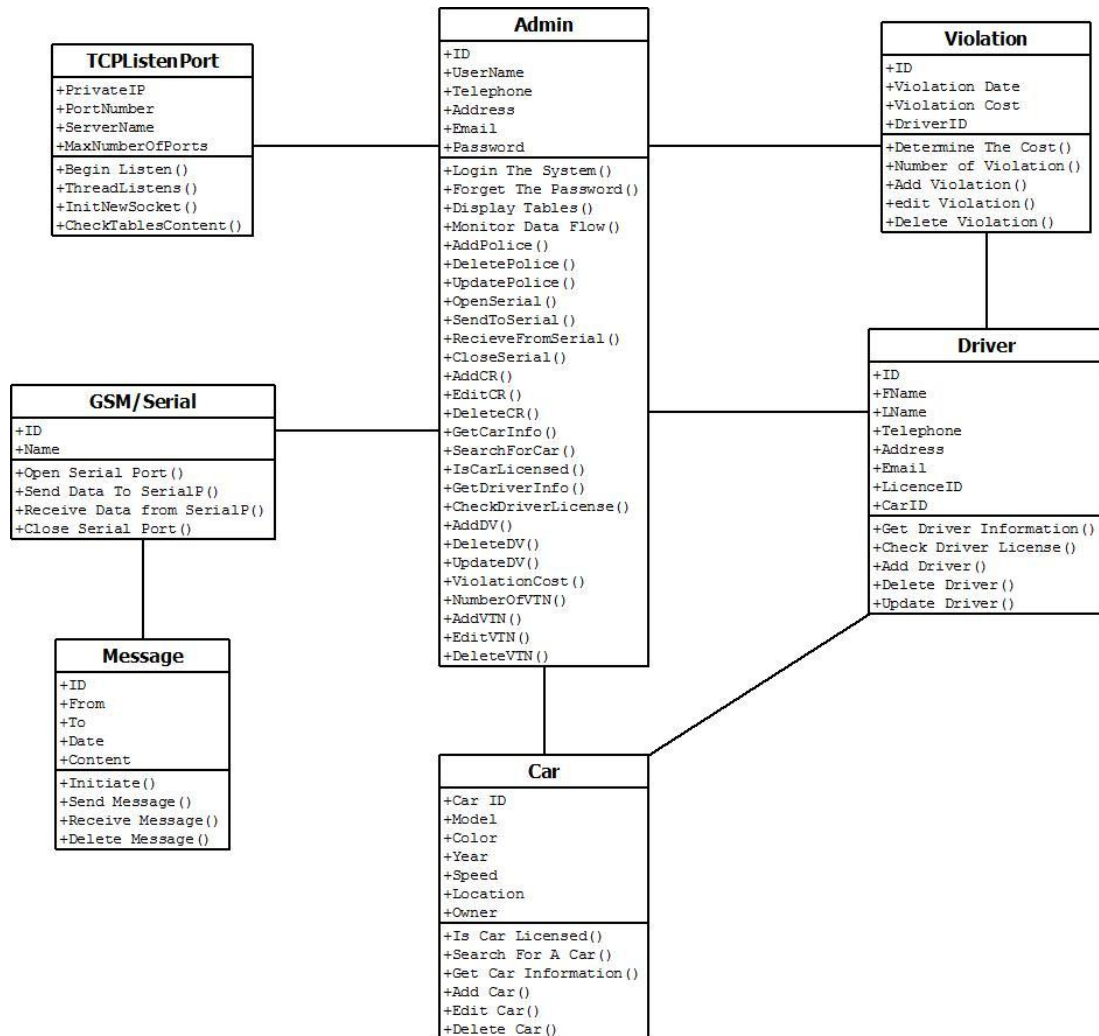


Figure 5.19: Server UML Diagram

5.5.2 Driver and Policeman Android Application

Figure (5.20 , 5.21) show the derived classes in the driver and policeman android application. And the attributes, operations and the relations between the classes.

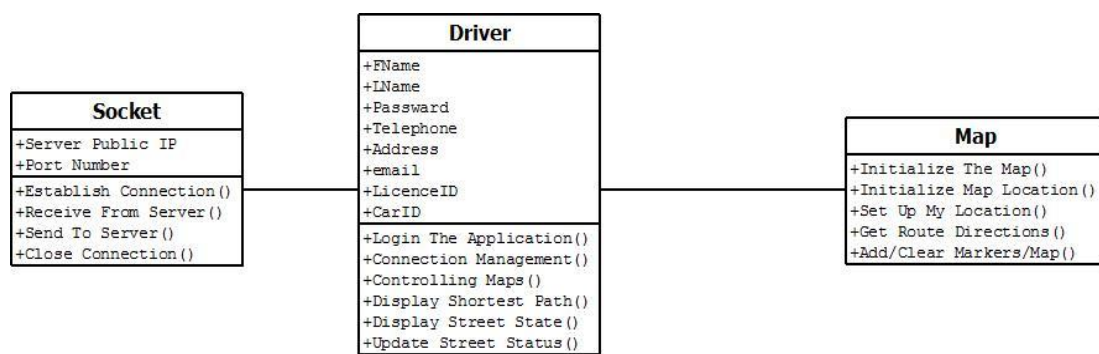


Figure 5.20: Driver UML diagram

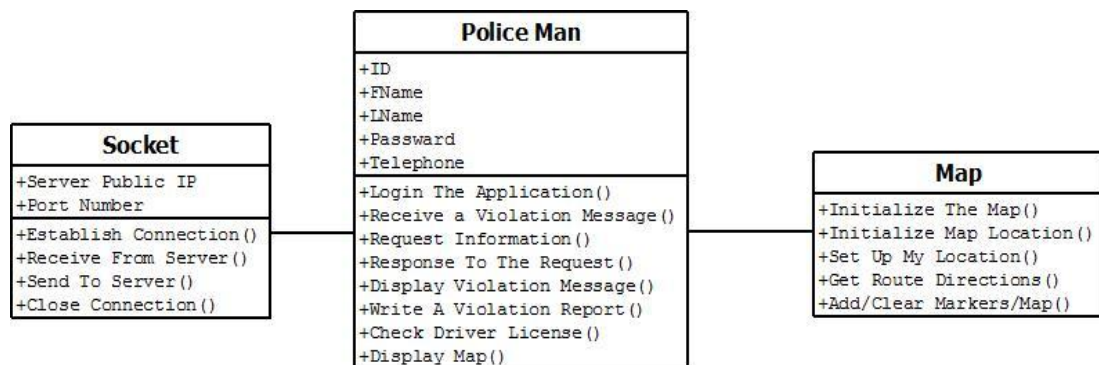


Figure 5.21: Policeman UML diagram

5.6 State Behavioral Modeling

5.6.1 Admin Class (State Diagram)

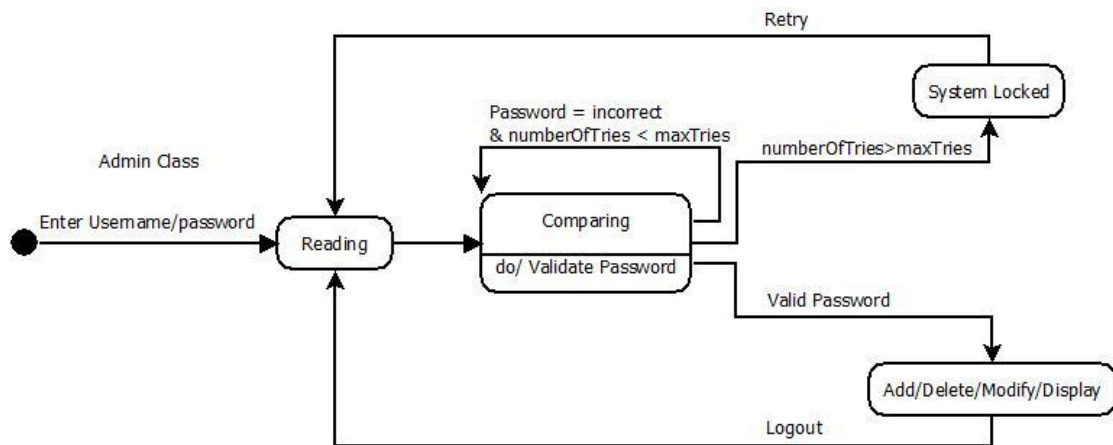


Figure 5.22: Admin State Diagram

5.6.2 GSMSerial Class (State Diagram)

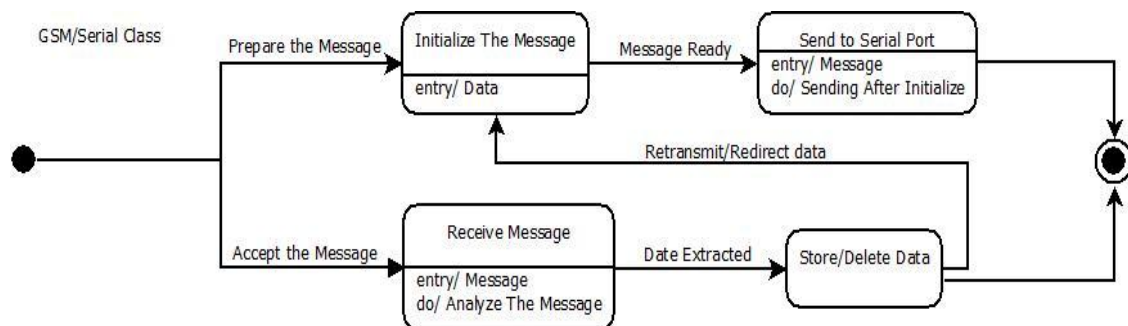


Figure 5.23: GSMSerial State Diagram

5.6.3 Violation Class (State Diagram)

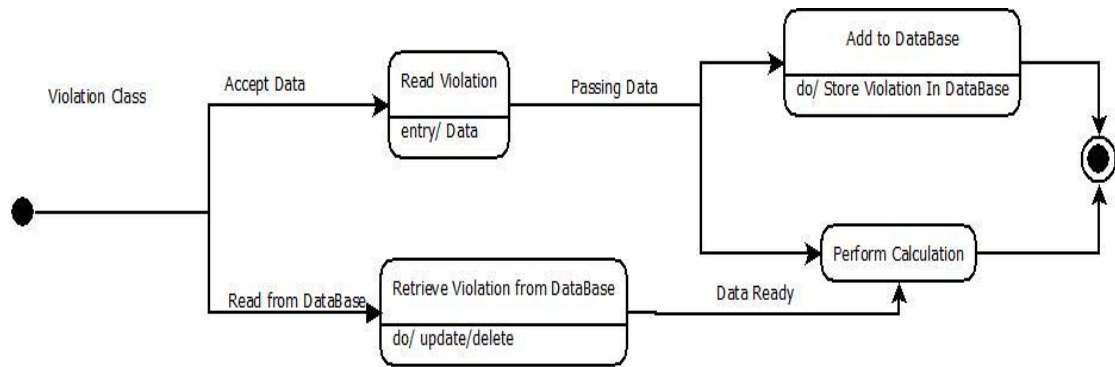


Figure 5.24: Violation State Diagram

5.6.4 Car Class (State Diagram)

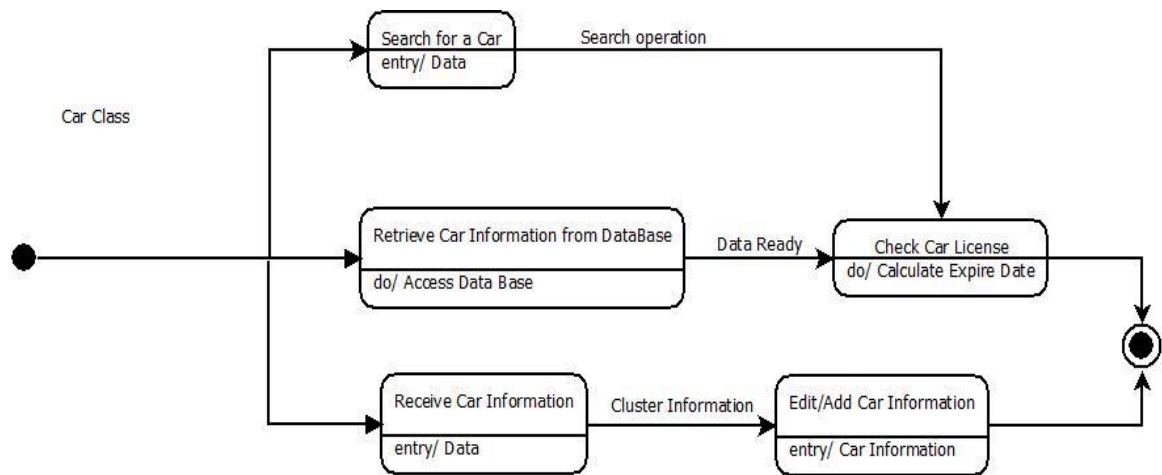


Figure 5.25: Car State Diagram

5.6.5 Message Class (State Diagram)

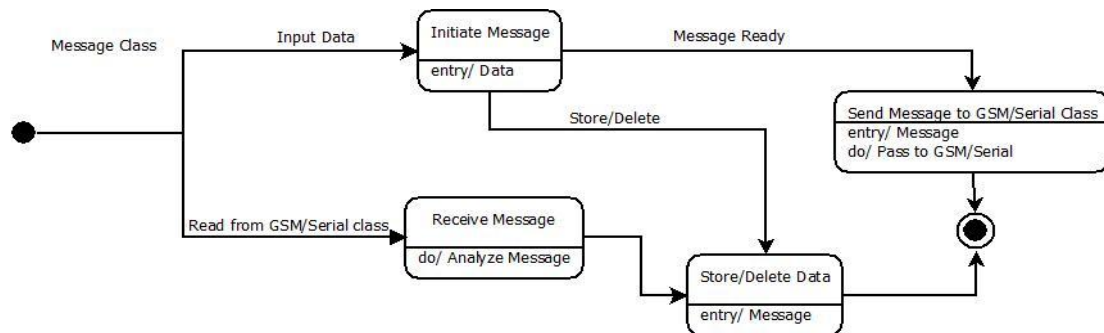


Figure 5.26: Message State Diagram

5.6.6 Driver Class (State Diagram)

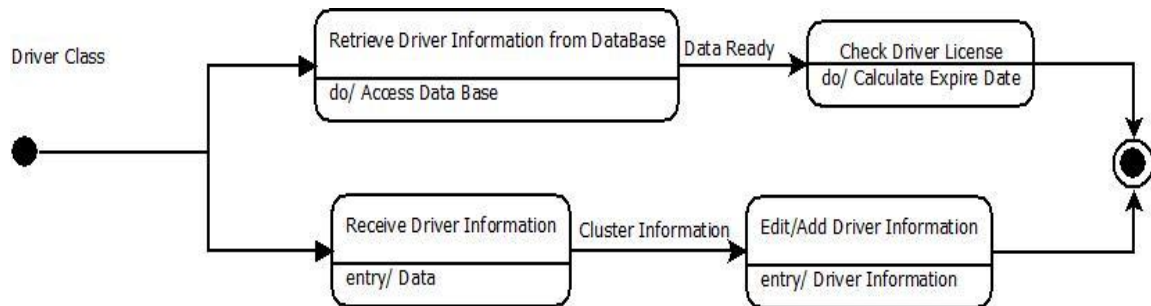


Figure 5.27: Driver State Diagram

5.6.7 Policeman Android Application

5.6.7.1 Policeman Class (State Diagram)

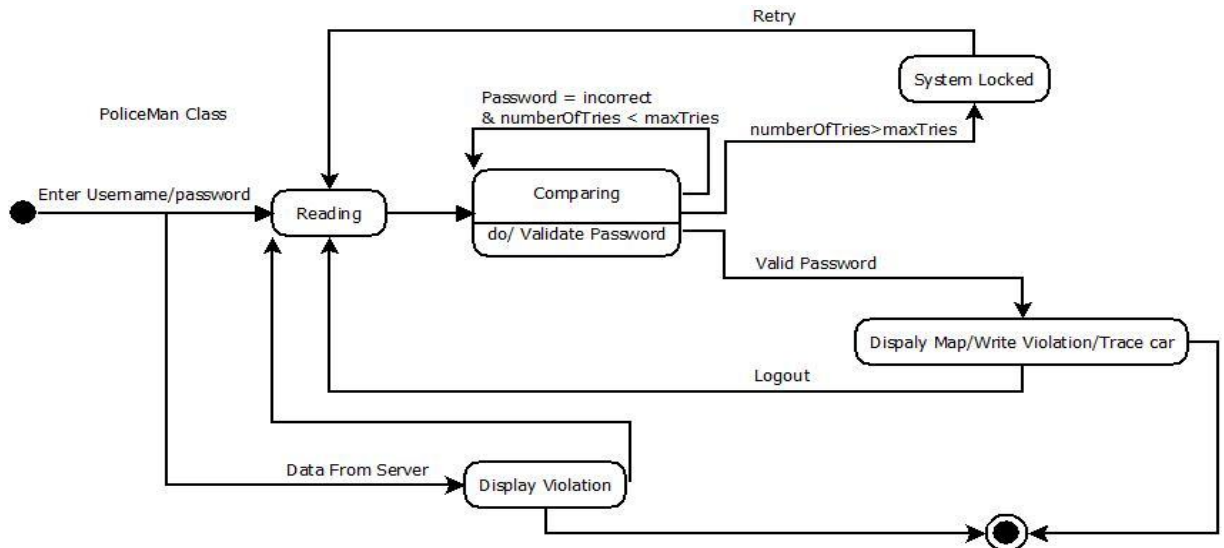


Figure 5.28: Policeman State Diagram

5.6.7.2 Driver Android Application

5.6.7.3 Driver Class (State Diagram)

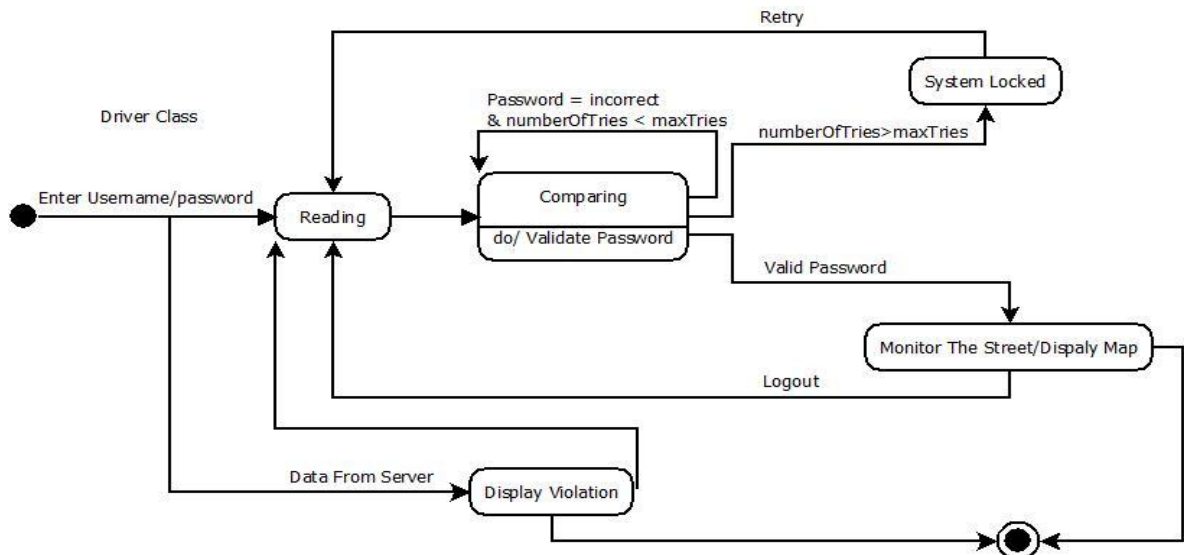


Figure 5.29: Driver State Diagram

5.7 Subsystem Design

5.7.1 Server Application

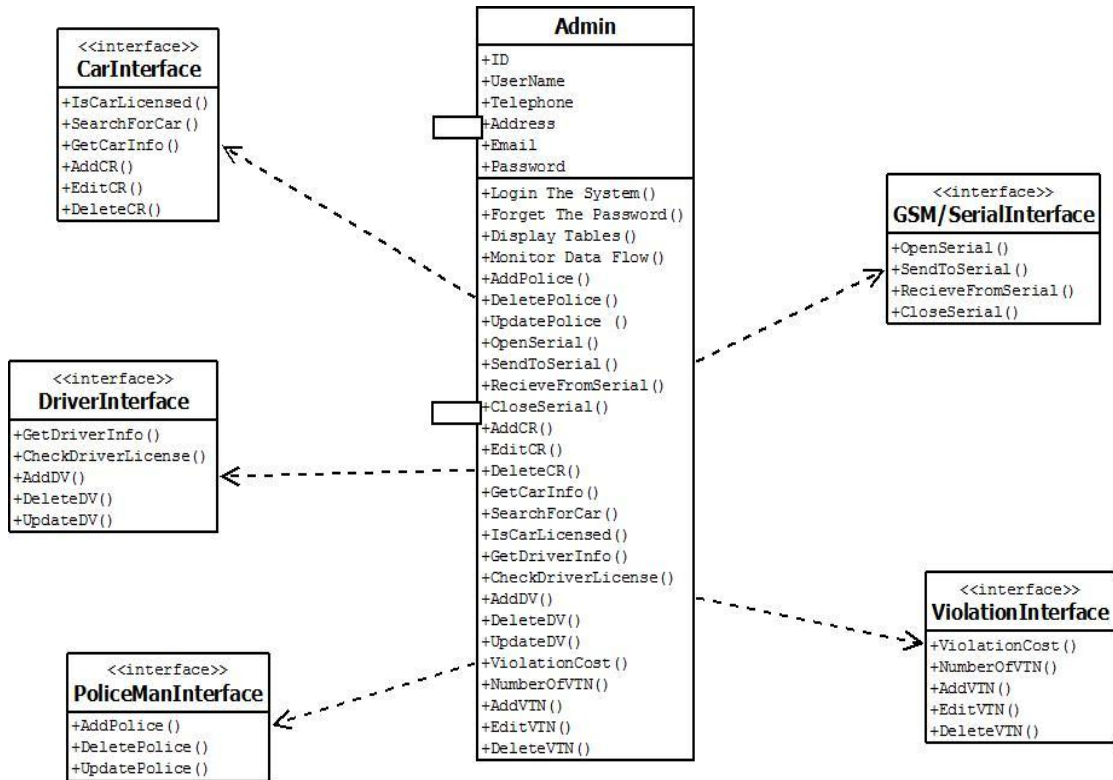


Figure 5.30: Elaboration of a design component 1(Server)

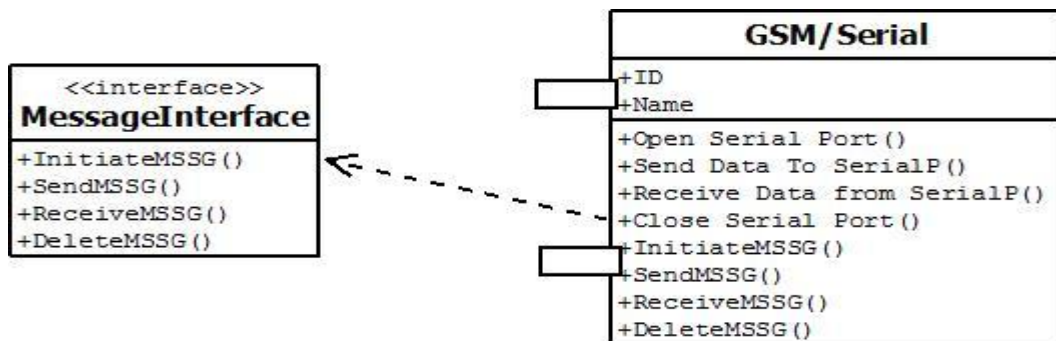


Figure 5.31: Elaboration of a design component 2(Server)

5.8 Class and Object Design

5.8.1 Server Application



Figure 5.32: Function Implementation (Login In)

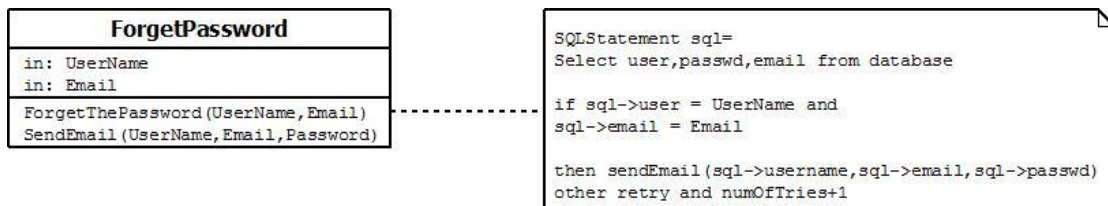


Figure 5.33: Function Implementation (Forget Password)

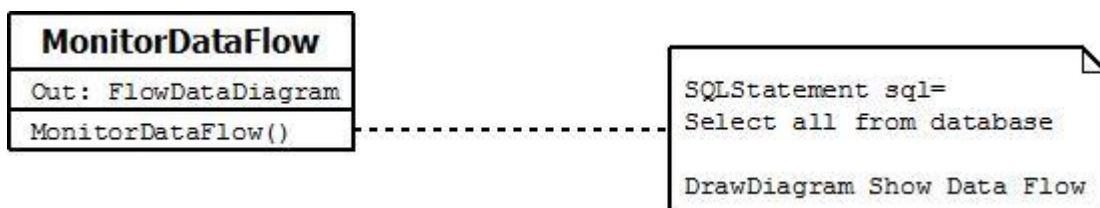


Figure 5.34: Function Implementation (Monitor Data Flow)



Figure 5.35: Function Implementation (Display Tables)

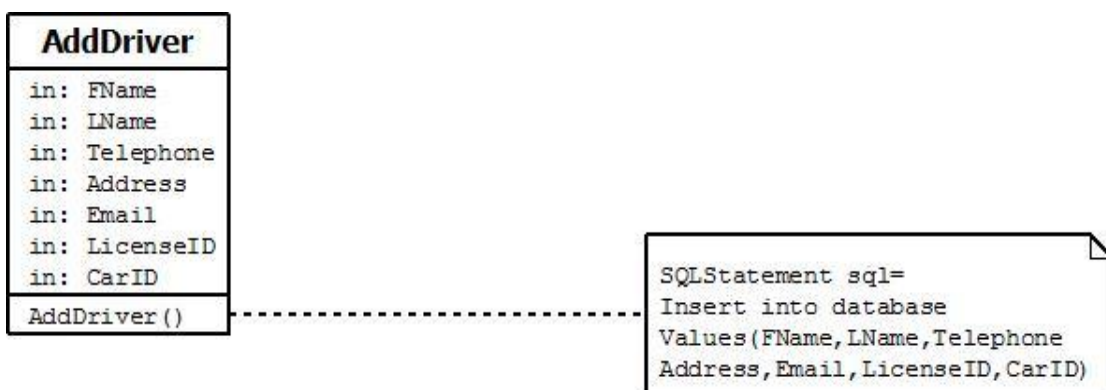


Figure 5.36: Function Implementation (Add Driver)

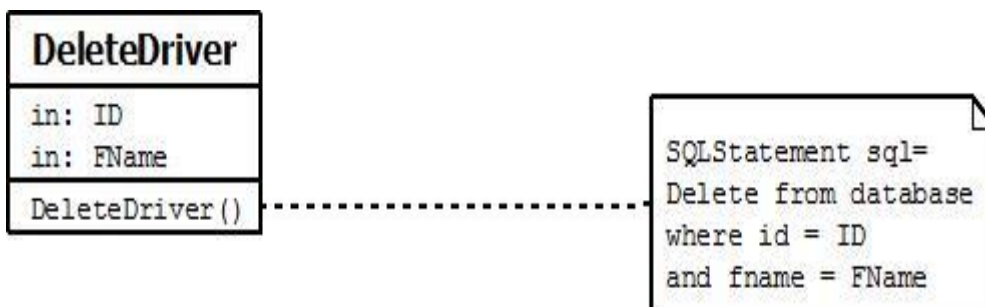


Figure 5.37: Function Implementation (Delete Driver)

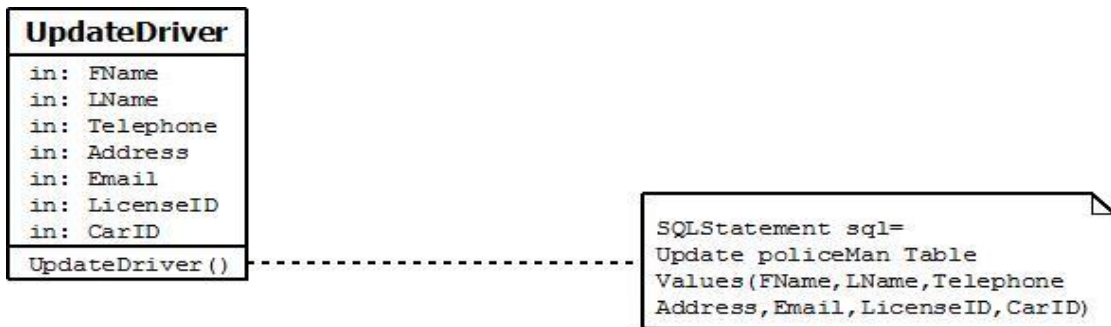


Figure 5.38: Function Implementation (Update Driver)

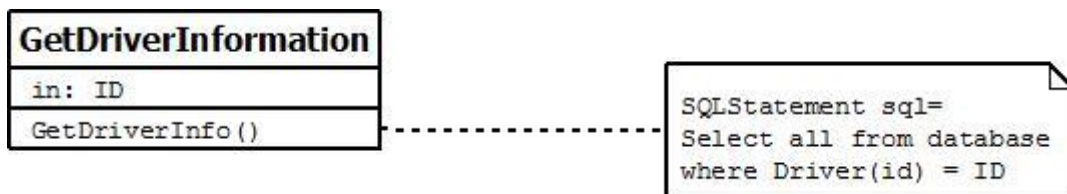


Figure 5.39: Function Implementation (Get Driver Information)

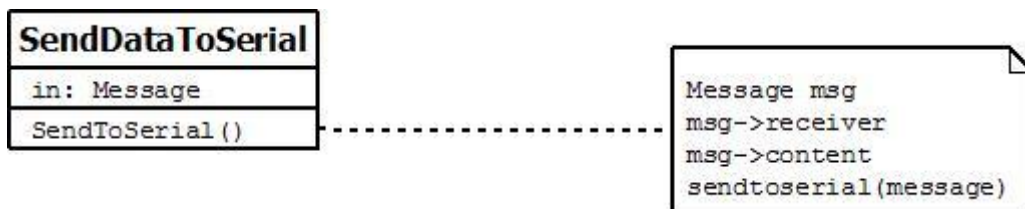


Figure 5.40: Function Implementation (Send Data to Serial)

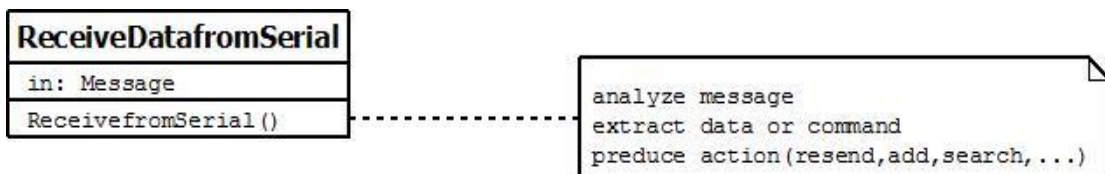


Figure 5.41: Function Implementation (Receive Data from Serial)



Figure 5.42: Function Implementation (Is Car Licensed)

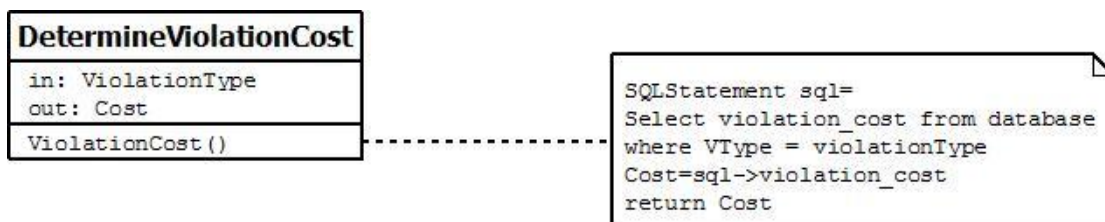


Figure 5.43: Function Implementation (Determine Violation Cost)

5.8.2 Policeman Android Application

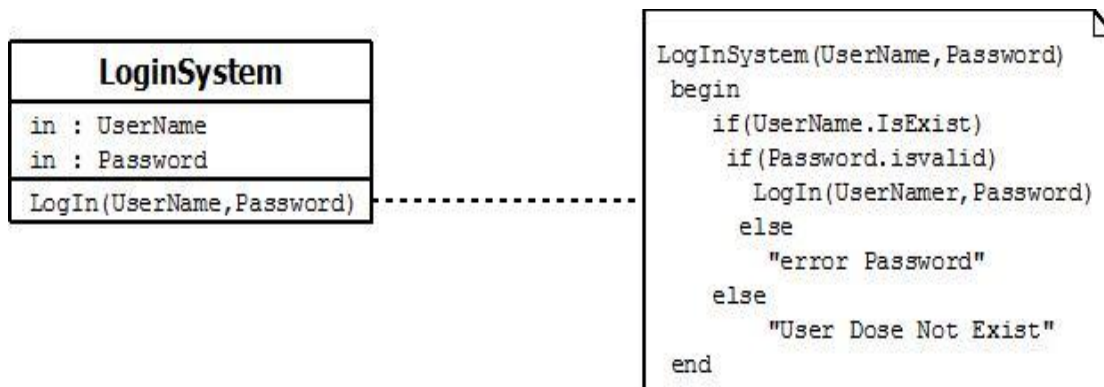


Figure 5.44: Function Implementation (Login System)

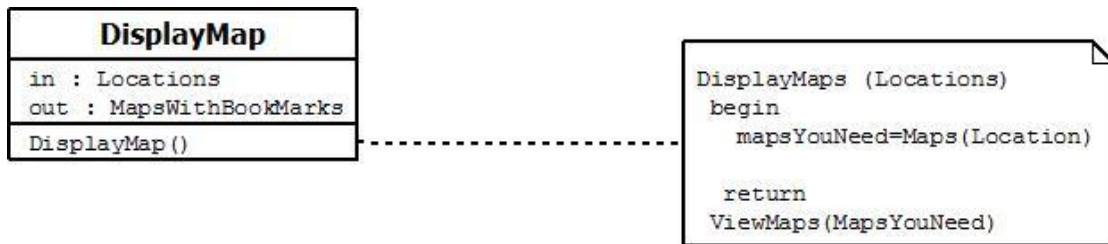


Figure 5.45: Function Implementation (Display Map)

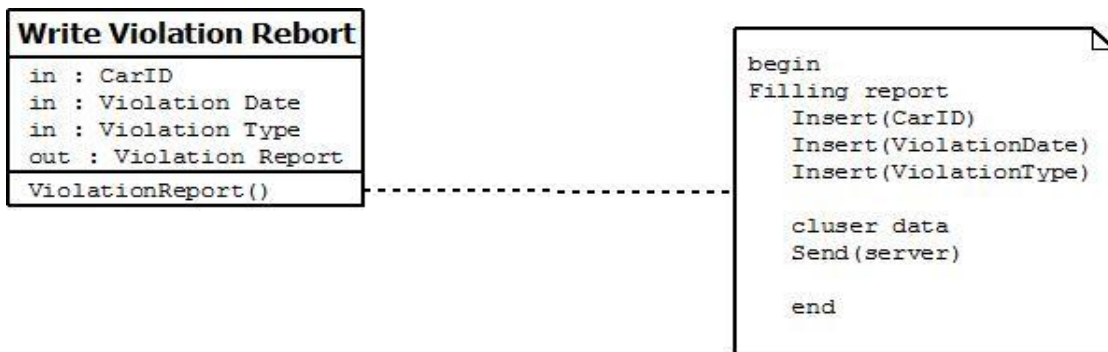


Figure 5.46: Function Implementation (Write Violation Report)

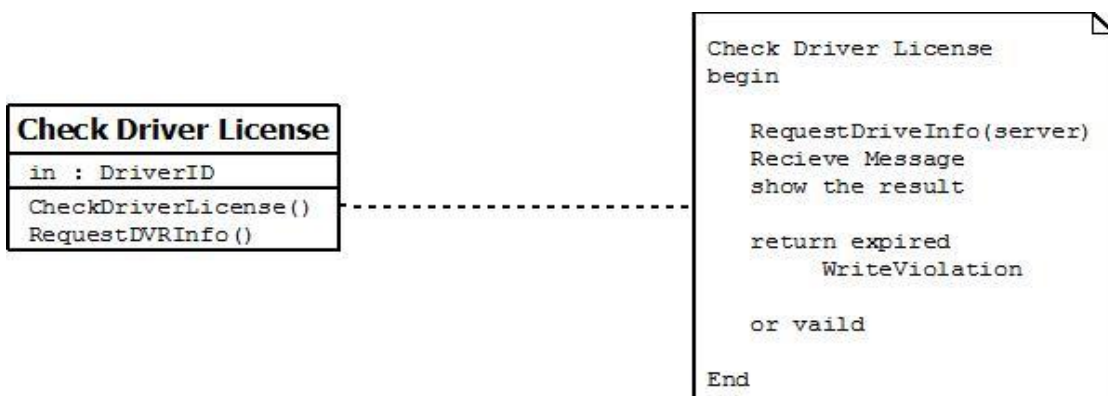


Figure 5.47: Function Implementation (Check Driver License)

5.8.3 Driver Android Application

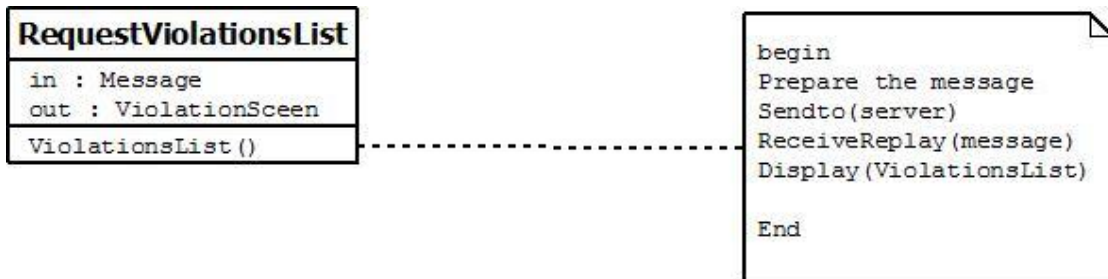


Figure 5.48: Function Implementation (Request Violations List)

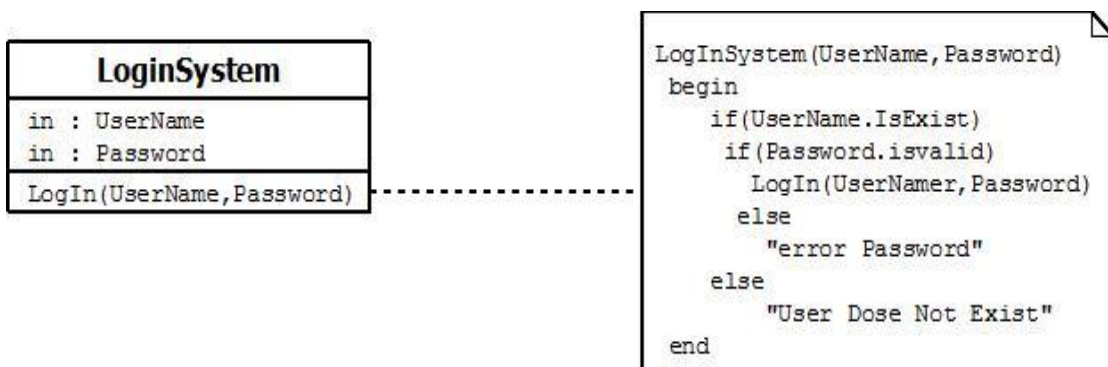


Figure 5.49: Function Implementation (Login System)

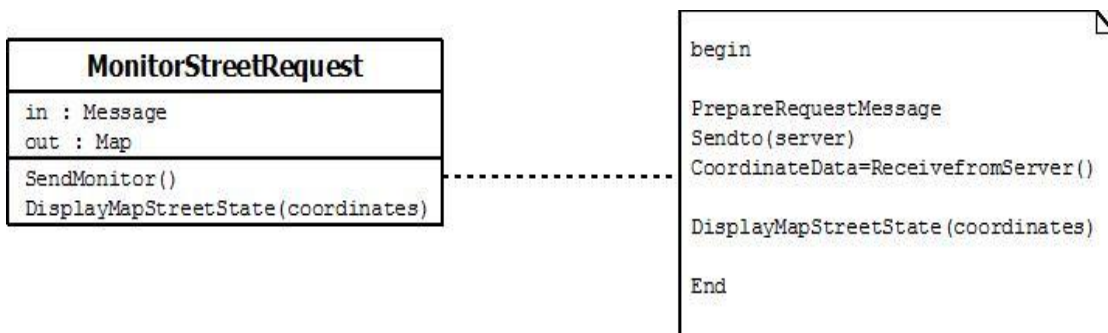


Figure 5.50: Function Implementation (Monitor Street Request)

5.9 Software Interface Design

In this section we will show and explain the expected software interfaces .

5.9.1 Server Interface

When any user runthe server application, login form face to Authentication user will be done by USER NAME and PASSWORD. Figure (5.41) show the expected login screen.

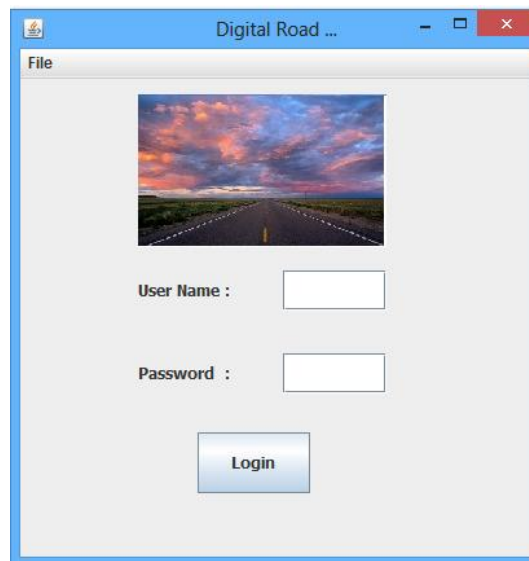


Figure 5.51: Server interface (login screen)

User can control server Contents (Cars, drivers, policemen and violations information) throw server main screen, Figure (5.42) shows the expected shape of the server's main screen.



Figure 5.52: Server interface (main screen)

5.9.2 Policeman Interface

When the Policeman application is run, login interface exhibit. The policeman can login to his android application by entering Name and Password as shown in figure (5.43), after this, a menu of operation will appear, and Figure (5.44) shows the expected Policeman interface that allows him to access to other applications screens. Figures (4.45) and (4.46) shows some interfacing screens that the cup would be allowed to use due to the tasks he is allowed to do.

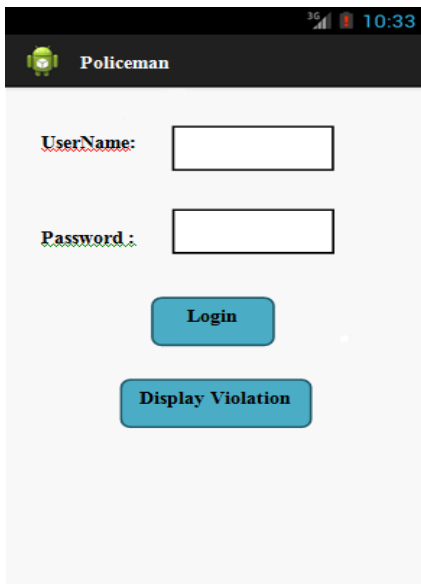


Figure 5.53: Policeman interface (1)



Figure 5.54: Policeman interface (2)

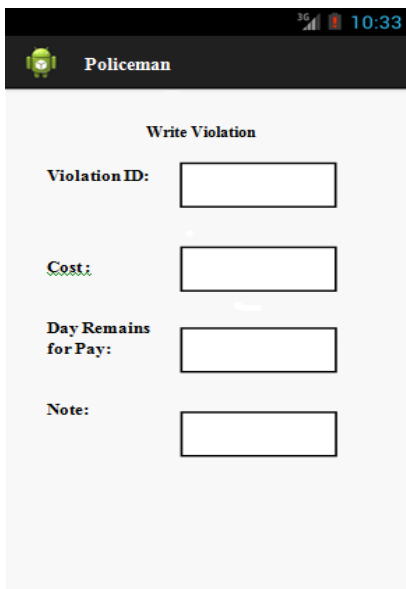


Figure 5.55: Policeman interface (3)

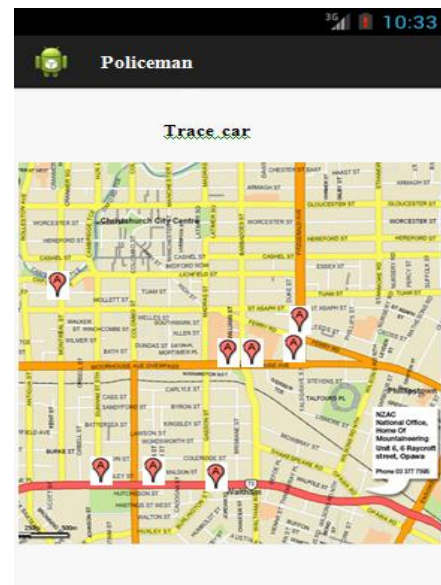


Figure 5.56: Policeman interface (4)

5.9.3 Driver Interface

Just as Policeman application, driver's application begins with login interface to Authentication user, then a menu will appear to allow the driver to choose one of the operations he could do, Figures (5.47), (5.48),(5.49) shows some of the driver interface suggested interfaces.

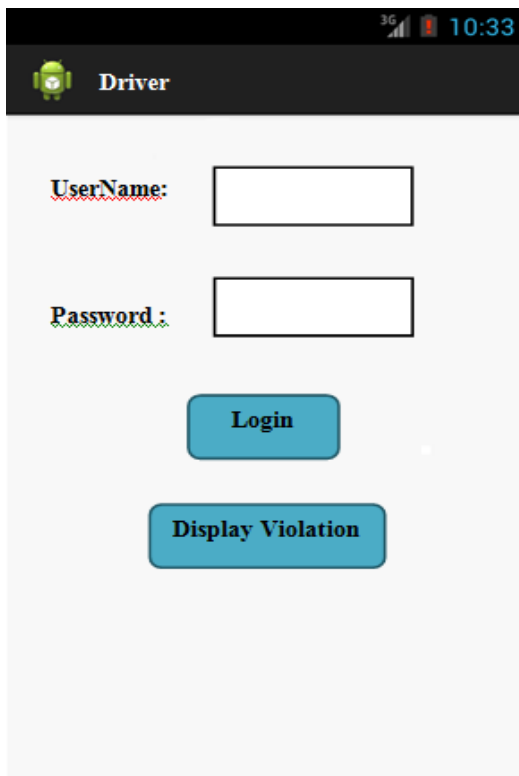


Figure 5.57: Driver interface (1)



Figure 5.58: Driver interface (2)

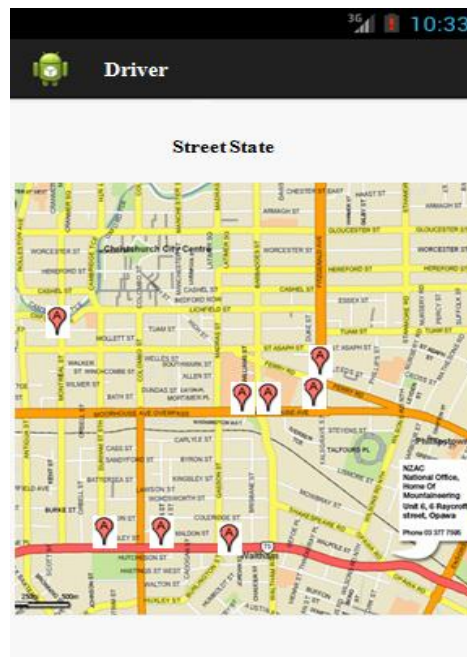


Figure 5.59:Driver interface (3)

5.10 Conclusion

In this chapter, we discussed the design of system. The classes attributes, operations and the relations. Data, control and state flow for each class explained. Also the pseudo code or functional implementation designed for the operation. The interface design for the server and the police/driver android application explained

Chapter Six

System Implementation

6.1 Overview

6.2 Hardware system Implementation

6.3 Software System Implementation

6.1 Overview

This chapter describes the hardware and software implementation of the system. It contains all real circuits of the sub-systems with real connections and configuration of all hardware components, for software part, we will discuss the real methods, objects, and interfaces.

6.2 Hardware Implementation

As we described earlier in chapter five, we divided our system into some sub-systems, here we are going to discuss them and describe how we implemented them in real, also we will describe the configurations and programming of the hardware component we used to build our system.

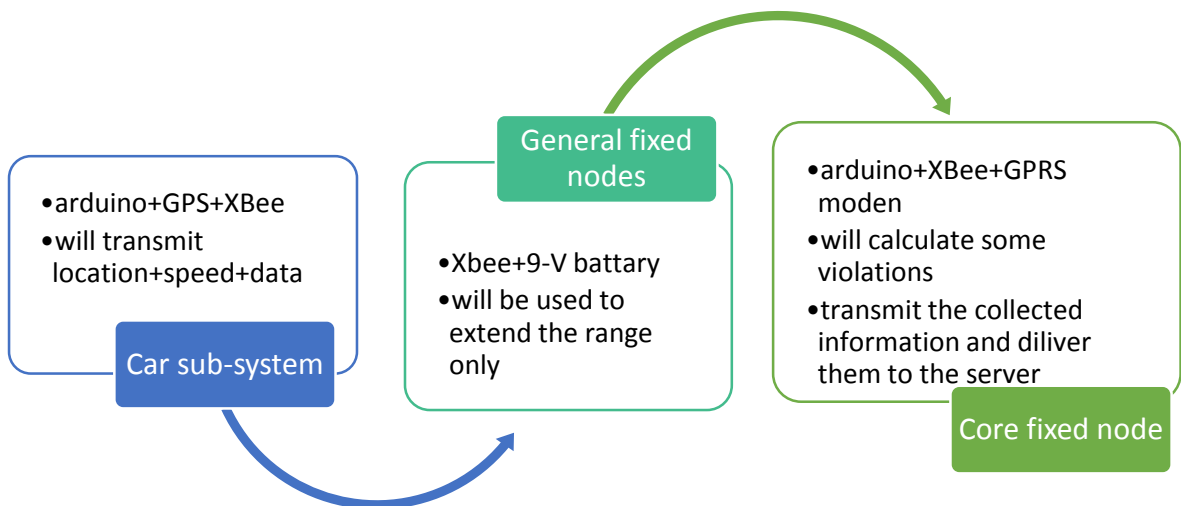


Figure 6.1: General block diagram

6.2.1 Zigbee Configuration

Zigbee is the main component of our project, the wireless sensor network is the core of our system where the Zigbee is the main component, so we are going to tell more about it.

XCTU is a free multi-platform application designed to enable developers to interact with Digital RF modules through a simple-to-use graphical interface. It includes tools that make it easy to set-up, configure and test XBee modules .

As for configuring our XBee chips, we used XBee explorer dongle, this dongle is to be inserted into the USB port in the computer with the chip on it, then the X-CTU program will be able to get into the configurations.



Figure 6.2:XBee connected to computer using dongle.

Here is a brief explanation of how to configure XBeechip using X-CTU

1- PC Setting window

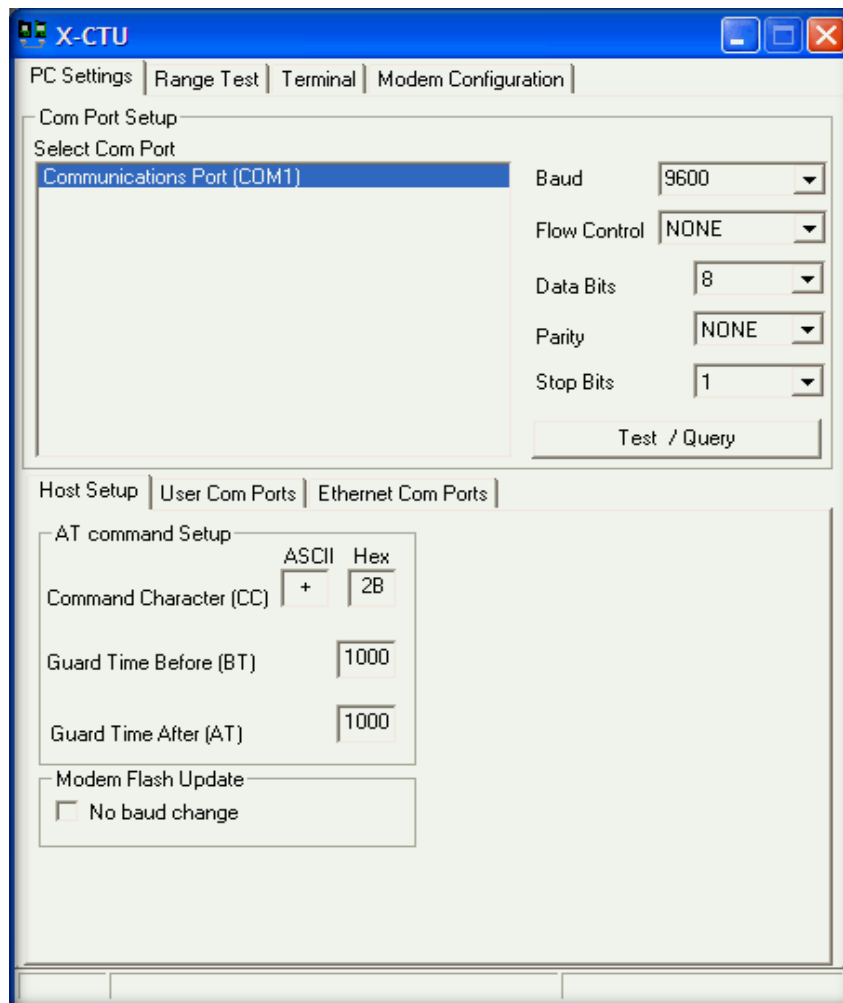


Figure 6.3: PC settings.

1. Choose the COM port which the dongle uses.
2. Choose the proper baud rate.
3. Click Test / Query to ensure that the X-CTU is communication with the XBee.

2- Modem Configuration window

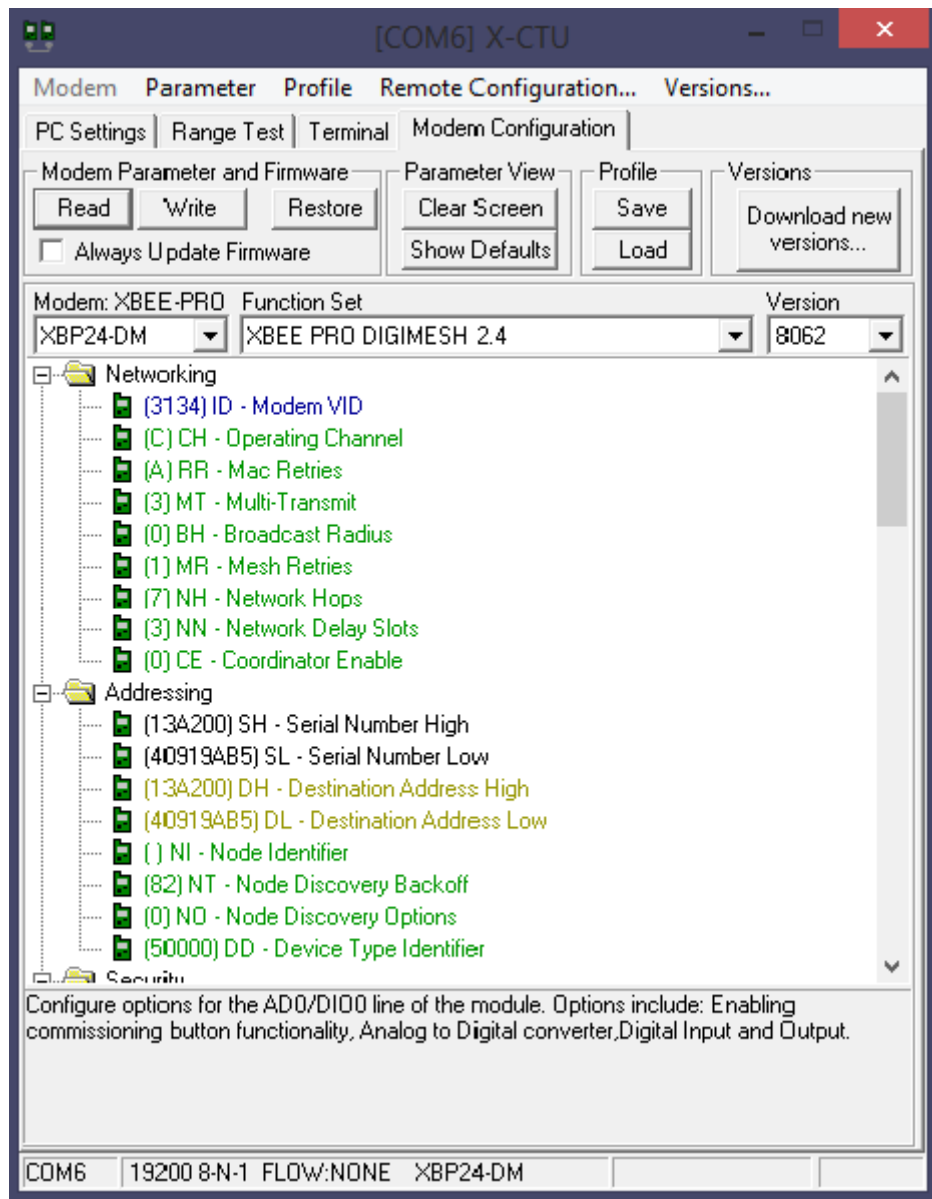


Figure 6.4: Setting the Modem Configuration.

In this window, we choose the proper settings to mate our application, and as shown in the figure, the important configurations have been set to make a wireless sensor network with an ID of 3134 works on channel C and sends the data to a specific address which is 13A20040919AB5, which is the core fixed node XBee

address.

6.2.2 Packet format analysis

The packet format which is going to be transferred between our sub-systems could be distributed as follows, it's important to mention that most of the fields is going to be filled in the car sub-system, the violations fields is to be filled later in the core fixed node.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44					

Header bytes : 0 - 1

Longitude with direction bytes :2 – 12

Latitude with direction bytes : 13 – 24

Speed Km/h bytes : 25 – 28

Airbag state byte : 29

Speed violation state byte : 30

Drives on opposite direction violation state byte: 31

Unusedbytes: 32 – 36

Car number bytes : 37 – 43

Terminatingbyte: 44

Example:

@\$3130.3320,N03505.3997,E0120010000007777777#

@\$: Header bytes .

3130.3320,N: Longitude

03505.3997,E:Latitude

0120: Speed Km/h bytes

0: Airbag state byte, becomes 1 if an accident happened

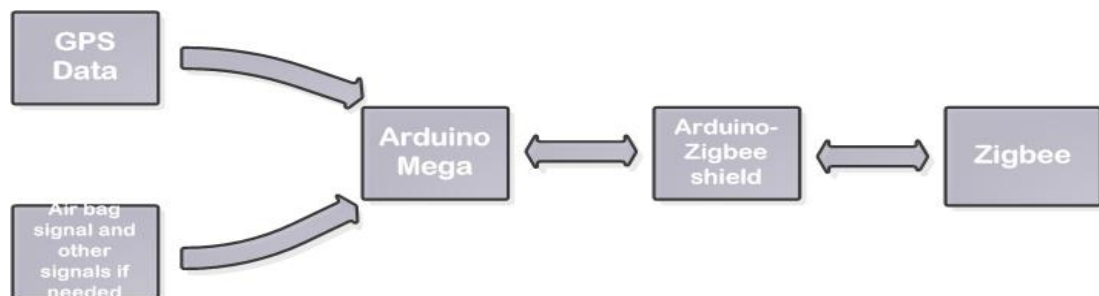
1: Speed violation state byte, this 1 shows that the driver broke the speed limit

0: Driving on opposite direction violation state byte, becomes 1 when the violation happens

000000: Other –unused yet- bytes

7777777: Car number

#: Terminating character byte



6.2.3 Car sub-system

This sub-system have been constructed of an Arduino mega, GPS receiver, Zigbee chip and a Zigbee-Arduino shield. The GPS receiver is connected to the Arduino mega through the serial port, the GPS gives us the location, the speed then is calculated, after that the packet is created and then to be sent over the Zigbee chip which is connected to the Arduino using the shield.

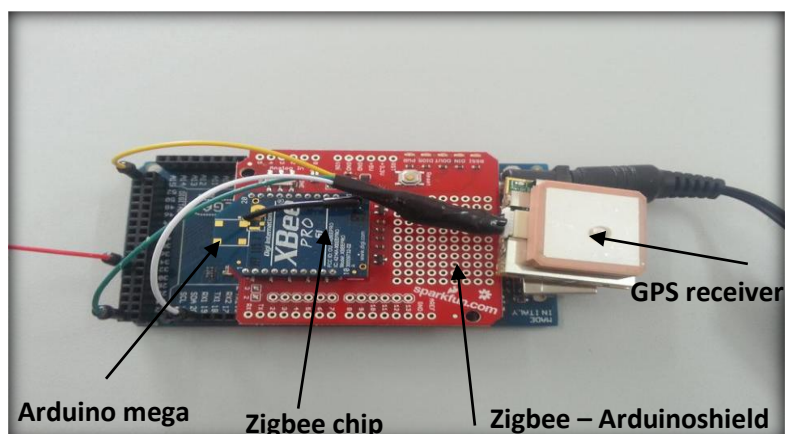


Figure 6.5: real car sub-system

6.2.4 General fixed node sub-system:



This is the simplest sub-system, it is a combination of a zigbee Pro chip, a regulation shield and a 9-V battery. The idea of this sub-system is to extend the range of the core fixed nodes, it will just help to deliver the data to the core nodes from far cars, tis helps to reduce the cost of the system.

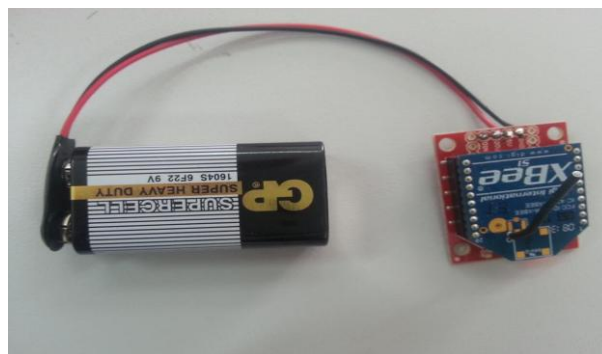
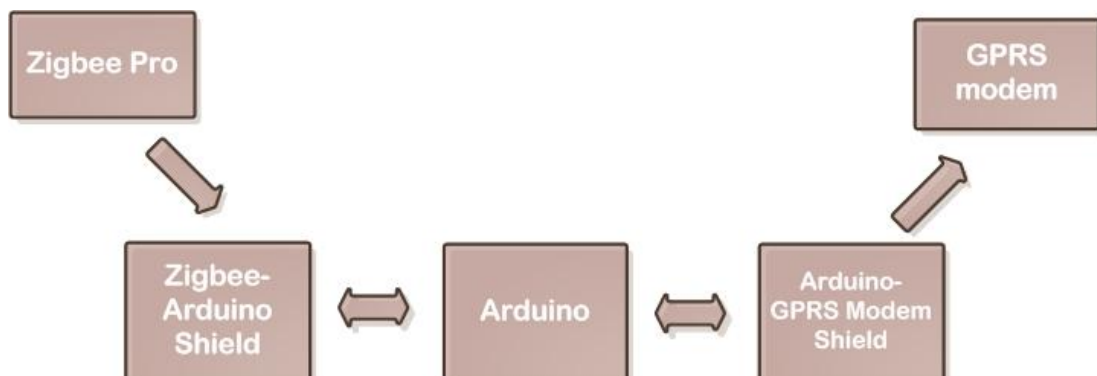


Figure 6.6: General fixed node sub-system

6.2.5 Core fixed node sub-system



This is the central node in the streets, its main function is to collect the packets generated in the cars and deliver them to the server through the GPRS modem, the modem opens a TCP socket with the server and directly sends the collected packets after checking the validity of them – checks the length and the heading bytes -.

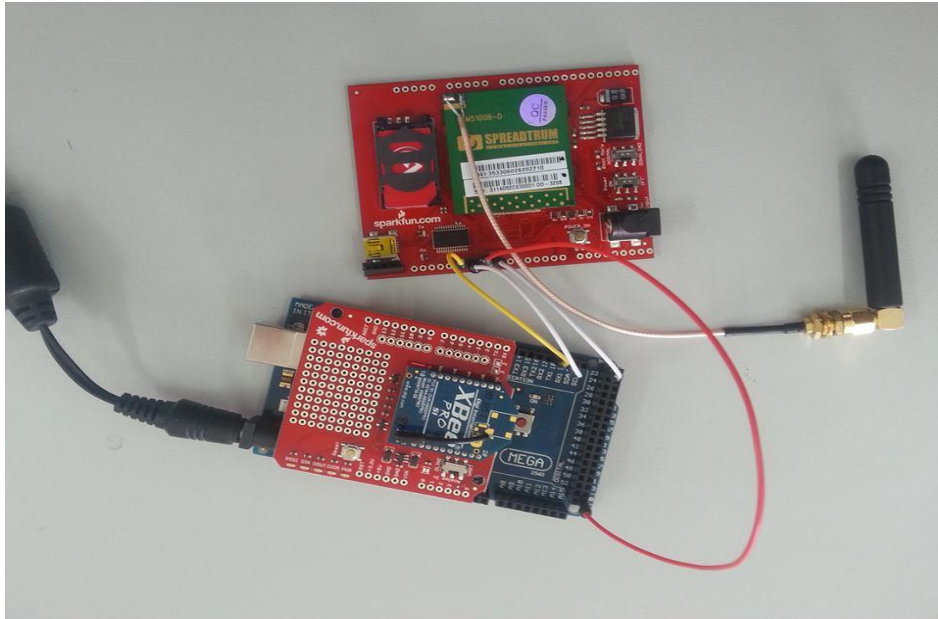


Figure 6.7: real circuit for core fixed node sub-system

6.2.6 Traffic Light sub-system

This sub-system have been designed to achieve two purposes, the first is to be automatically controlled to ensure the maximum speed in ambulance arriving in accidents' cases, the other is to make the traffic lights controllable wirelessly by the decision makers. Well, both options are supported in our system. Simply, we have a GPRS modem to get the data – GPRS or SMS -, the modem is connected to an Arduino which to be used to control the real traffic light controllers or the pic one in our test model shown in Fig(6.8).

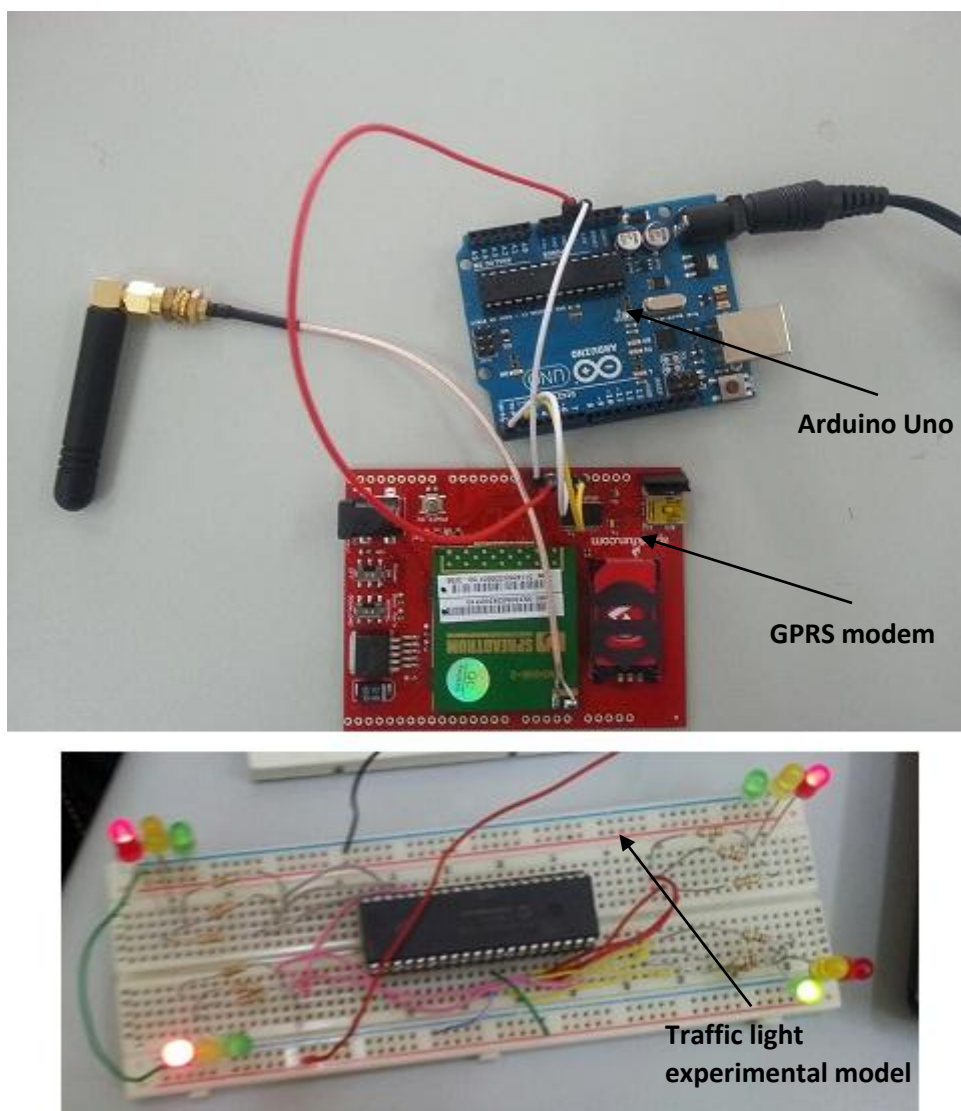


Figure 6.8: real circuit for traffic light sub-system

6.3 Software implementation

In this section, we are going to explain the real interfaces of our applications, and to make a brief explanation about those applications.

6.3.1 Driver Android Application

This contains:

1. Shortest Path icon: Direct the driver to the shortest path between two points with the specific time and distance and the reaching mode.
2. Street State icon: Feedback from the server to the driver about the street state.
3. Update State icon: Information feeds from the driver to the server about the street state.
4. Display Violation icon: Display the driver violations.

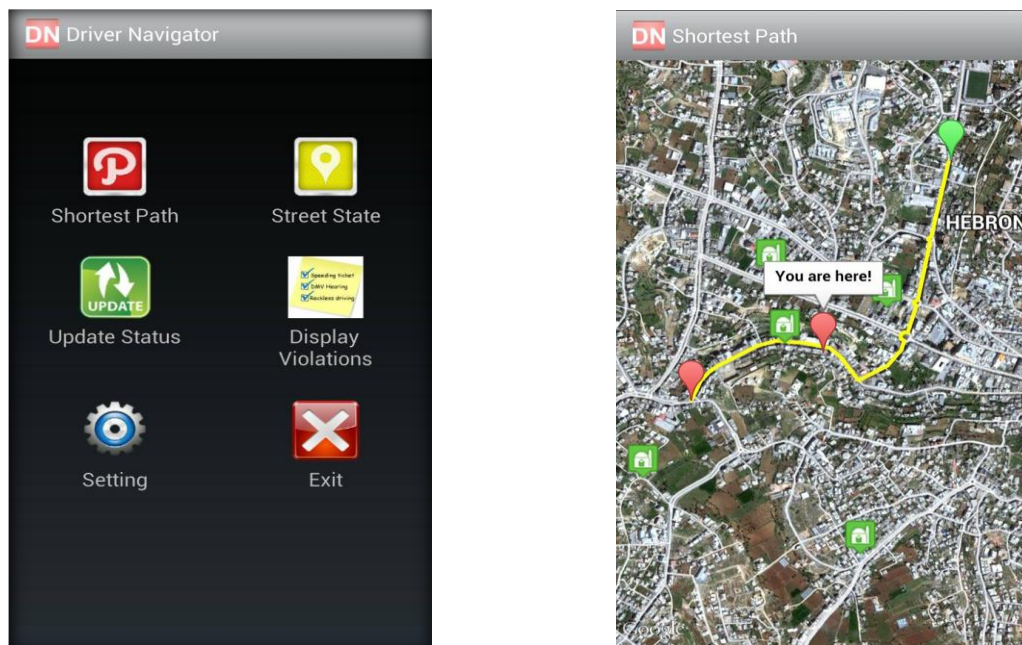


Figure6.9: Driver android application

Fig (6.9) show the Main screen for driver application with it is function, the second shows the shortest path between any two selected points.

6.3.2 Policeman Android Application

This contains:

1. Shortest Path: Direct the policeman to the shortest path between two points with the specific time and distance and the reaching mode.
2. Street State: Feedback from the server to the policeman about the street state.
3. Tracing Car: Allow the police man to tracks a specific car.
4. Black List Car: Contain all cars that added to the black list.
5. Write Report: Enable the Policeman to write violation report and submit it.
6. Check License: Check the license of any car.

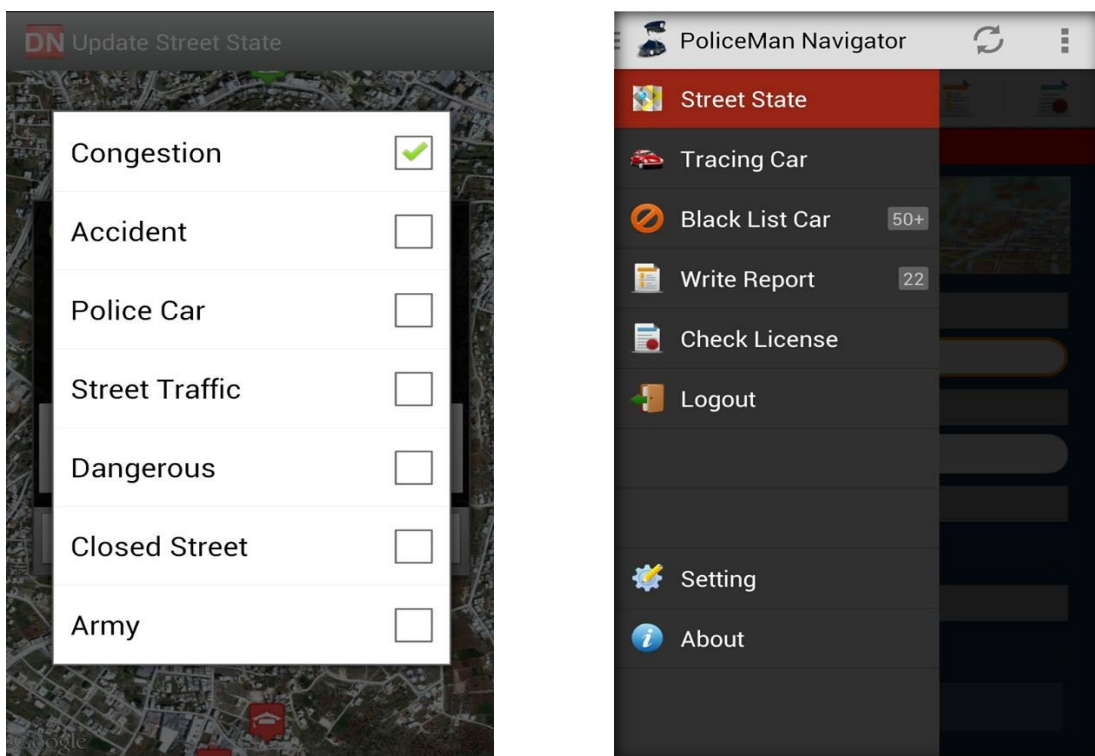


Figure6.10: policeman android application

6.3.3 Server Application

This picture shows the main interface of the server, which contains a lot of applications our system supports. Here we will list some pictures of the server interfaces and let them talk about themselves.

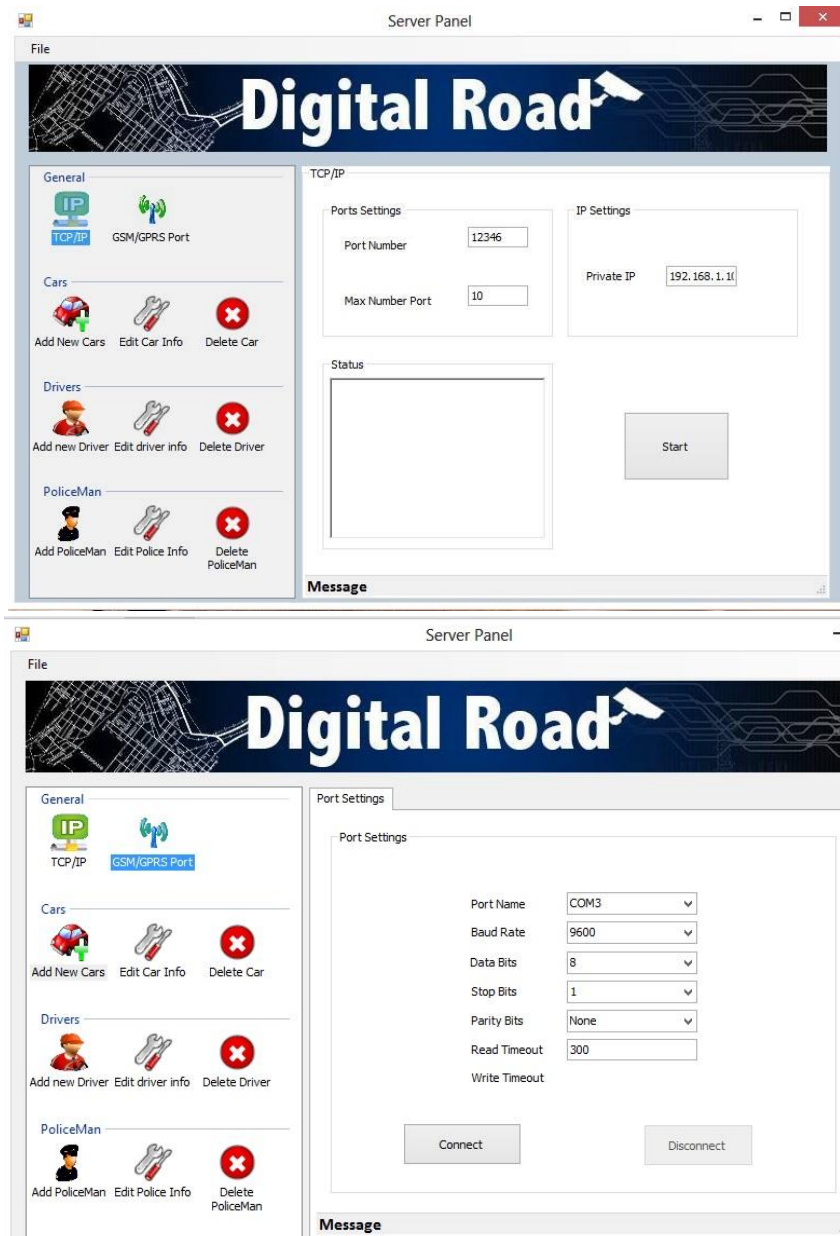


Figure 6.11: Server Main Interface

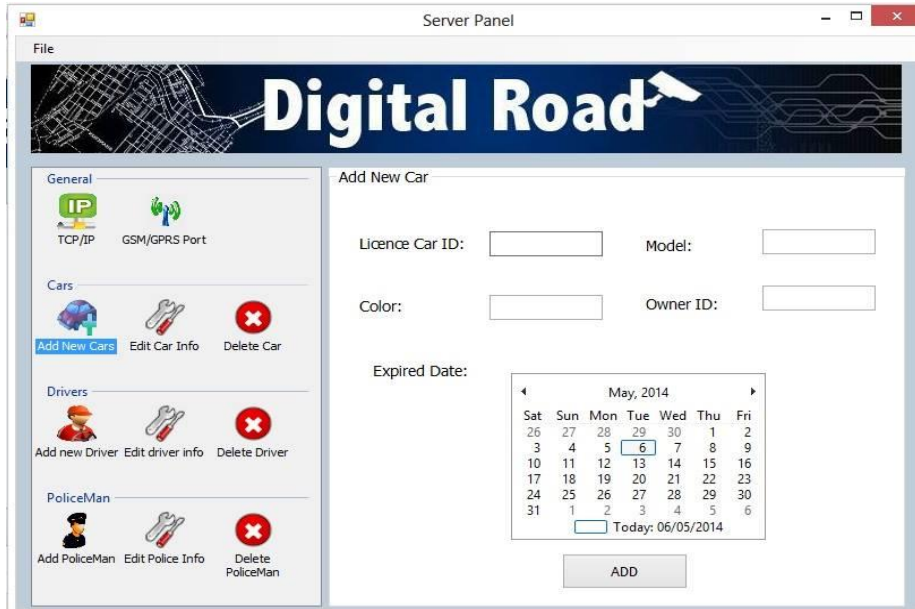


Figure 6.12: Add New Car

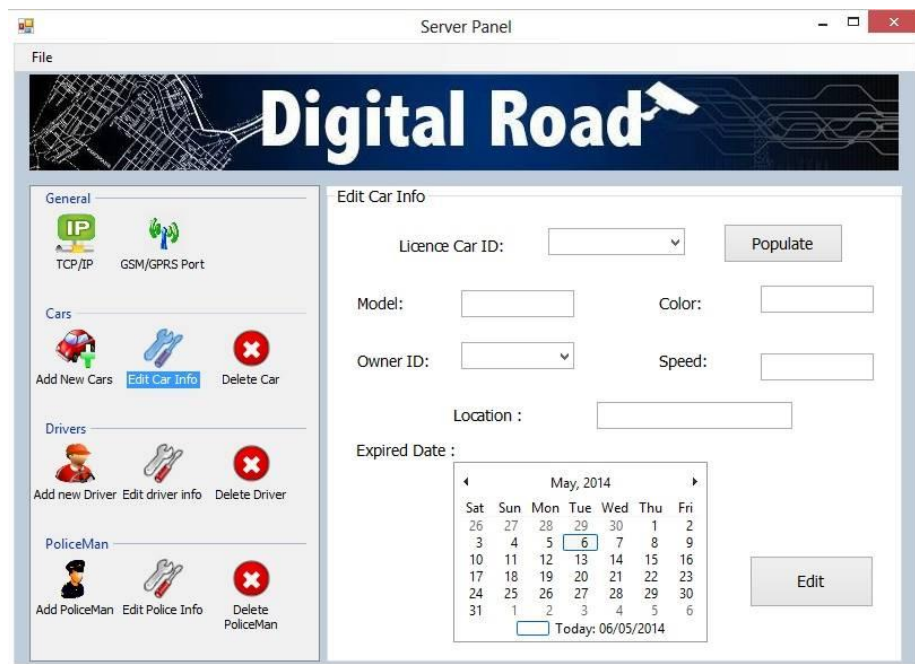


Figure 6.13: Edit Car

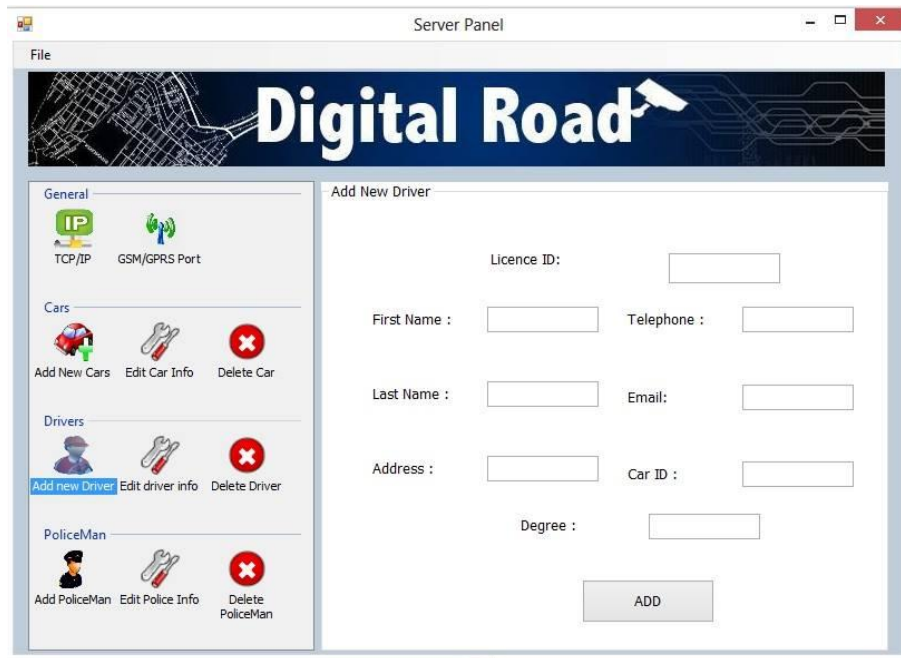


Figure 6.14: Add a new driver to the system

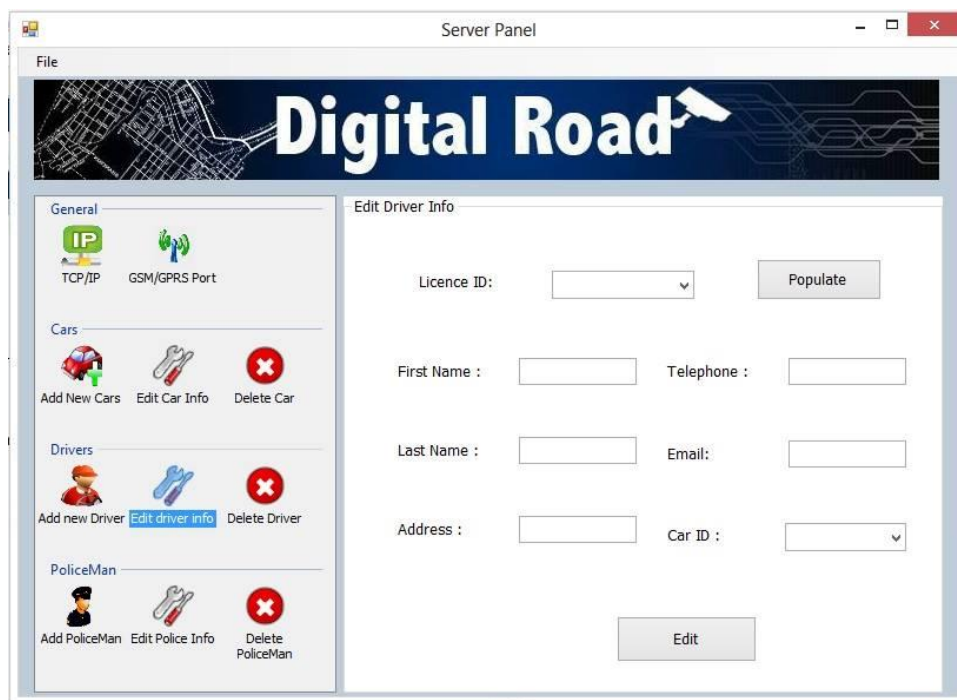


Figure 6.15: Edit driver information

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the number 7.

7

Chapter Seven

System Testing

7.1 Overview

7.2 Hardware sub-systems testing

7.3 Software Testing

7.1 Overview

In this chapter, the whole testing stages will be described, including a test of the entire interconnected set of components and software for the purpose of determining proper functions and achieving the desired goals of the system. We will talk about testing of each part of the system, and describe its scenarios.

7.2 Hardware Sub-System Testing

In this section we will discuss the results we got by making some tests on the hardware devices we used in the project, also the sub-systems have been tested and we are going to show the results in this section.

7.2.1 Basic packets generation test

In this test, we connected the car sub-system, which is a compensation of an Arduino, GPS, Zigbee and its shield, the system have been previously explained under the name of Car sub-system. The Arduino have been powered using the 5-V USB cable, we used the Zigbee to receive the packet, the zigbee was connected to the explorer dongle, and its serial have been monitored using X-CTU program, and the results is shown in Figure 7.1.

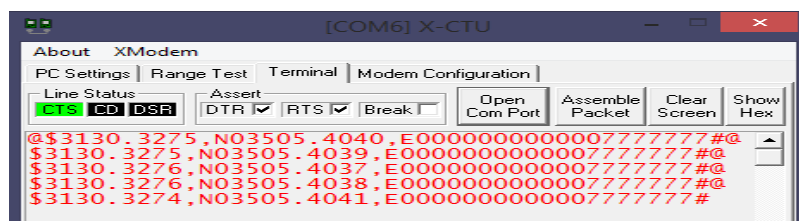


Figure 7.1:the packet from the car to the fixed nodes.

Figure 7.1 shows the received packet at the receiver XBee on X-CTU Serial monitor, the packet is 44 bytes and our packet format rules applies on it. The speed is 0 Km/h because the test was made on a non-moving car model, the air bag status is 0 also, and no violations were assigned.

7.2.2 Range, Ad-hock and Air bag sensor tests

Figure 7.2 shows the test setup...

All the cars was transmitting 1 packet per second

The spaces here have been maximized

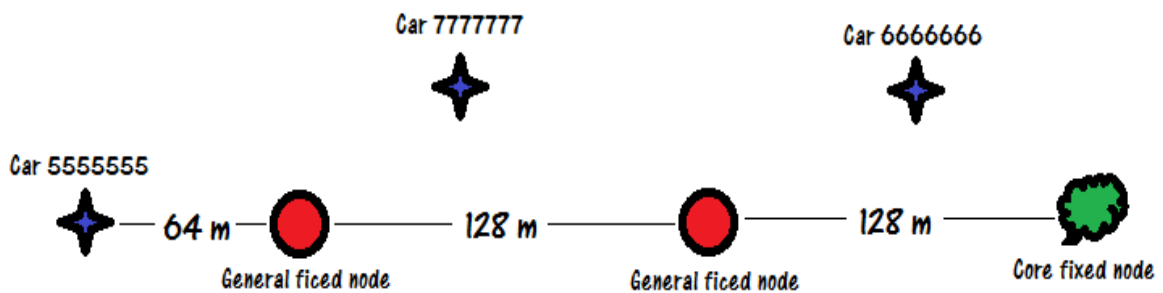


Figure 7.2: Range test setup

As shown in Figure 7.2, the maximum distance we could have was 64 m between the cars and any fixed node, we could also get the packets from car 5 and 7 in the core fixed node using the ad-hock property, and so, using two general fixed nodes extended the range of the core fixed node by 256 m. Figure 7.3 shows the packets received in the Core fixed node in this test.

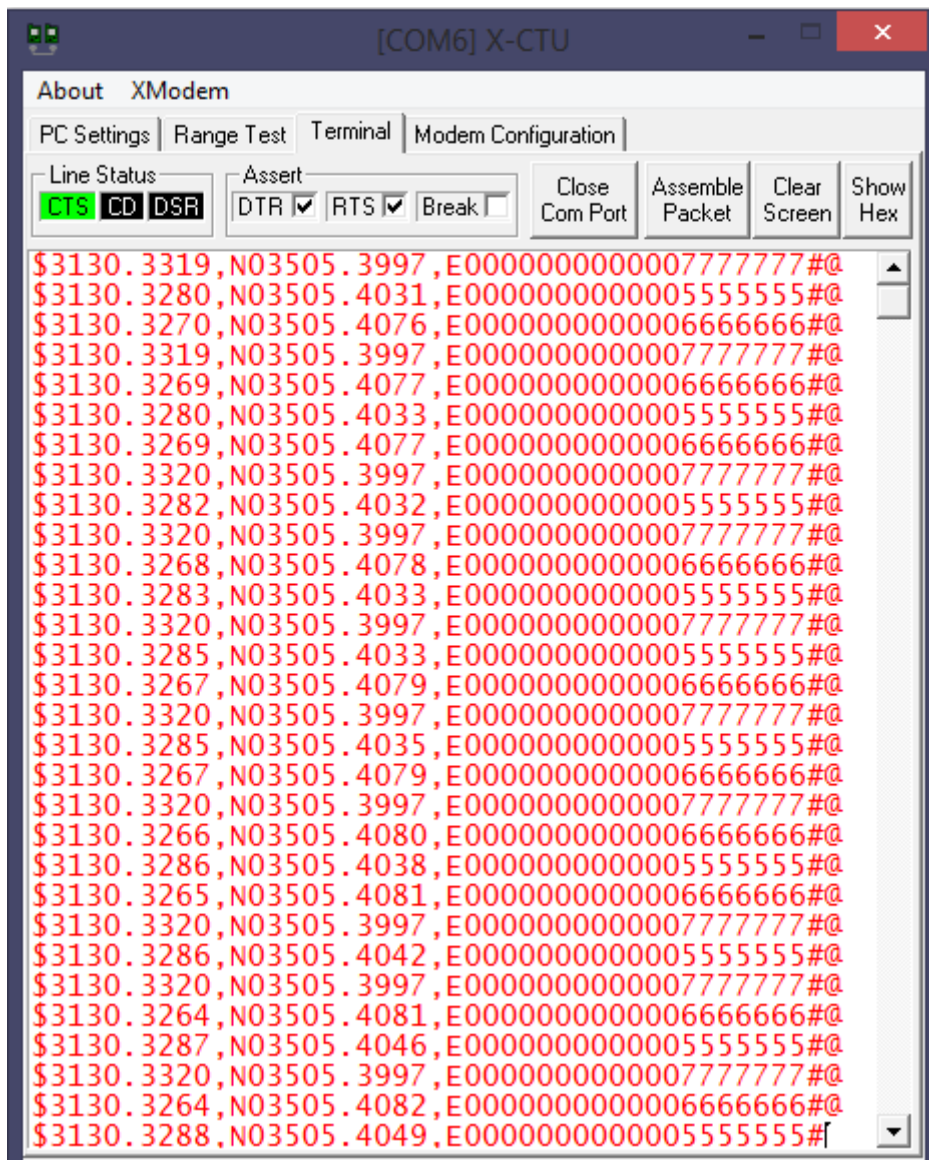


Figure 7.3: The received data at Corefixed node from three cars.

After that, we use activated the air bag sensor in 6666666 car, then we noticed that the air bag byte –byte number 30- turned into 1, which means that the sensor is active and so we can know that an accident happened in the location mentioned in the packet as shown in Figure 7.4.

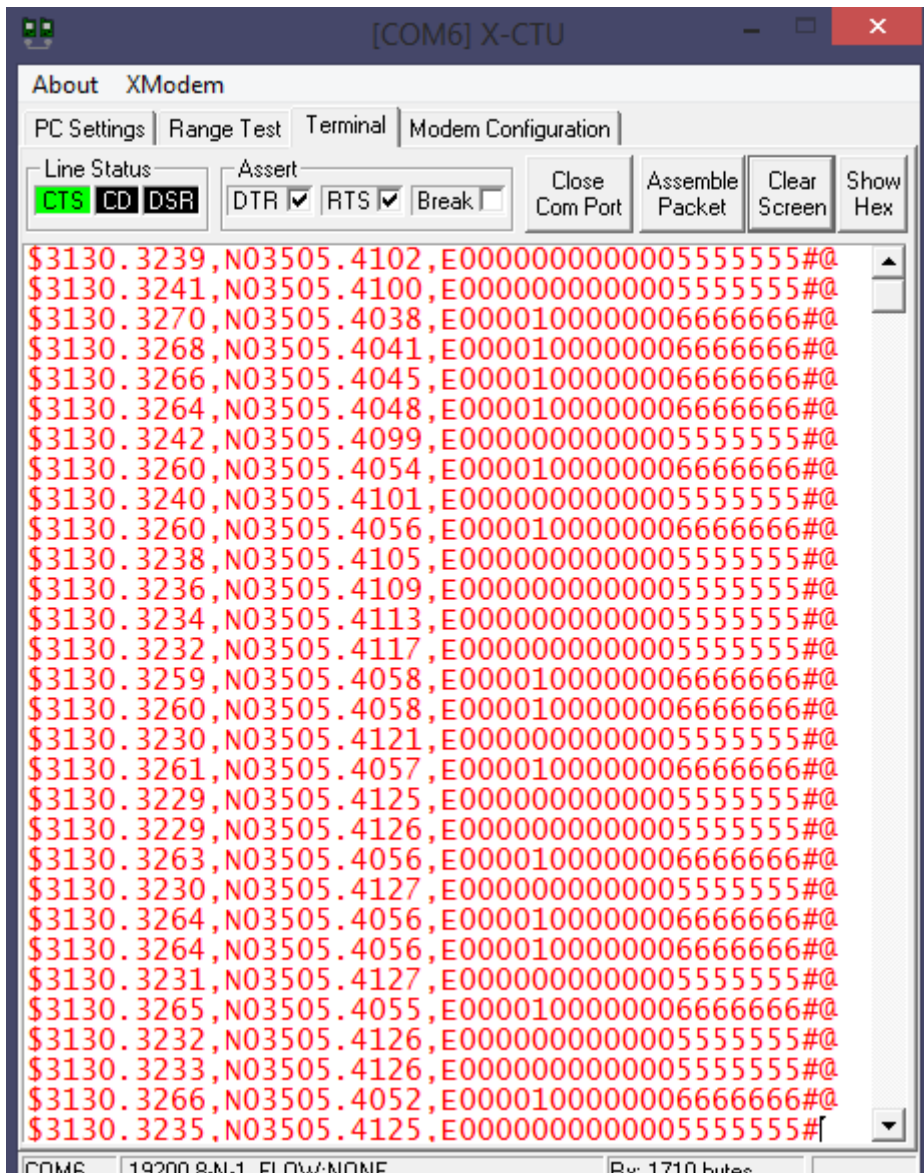


Figure 7.4: Air bag sensor byte at 6666666 car

7.2.3 Speed test

In this test, we are going to get in a real car, the car sub-system will be in the car and to be powered from the 12 V power jack in the car, we designed a simple 12 V to 6 V regulator to power the system, the following figures shows what we did.

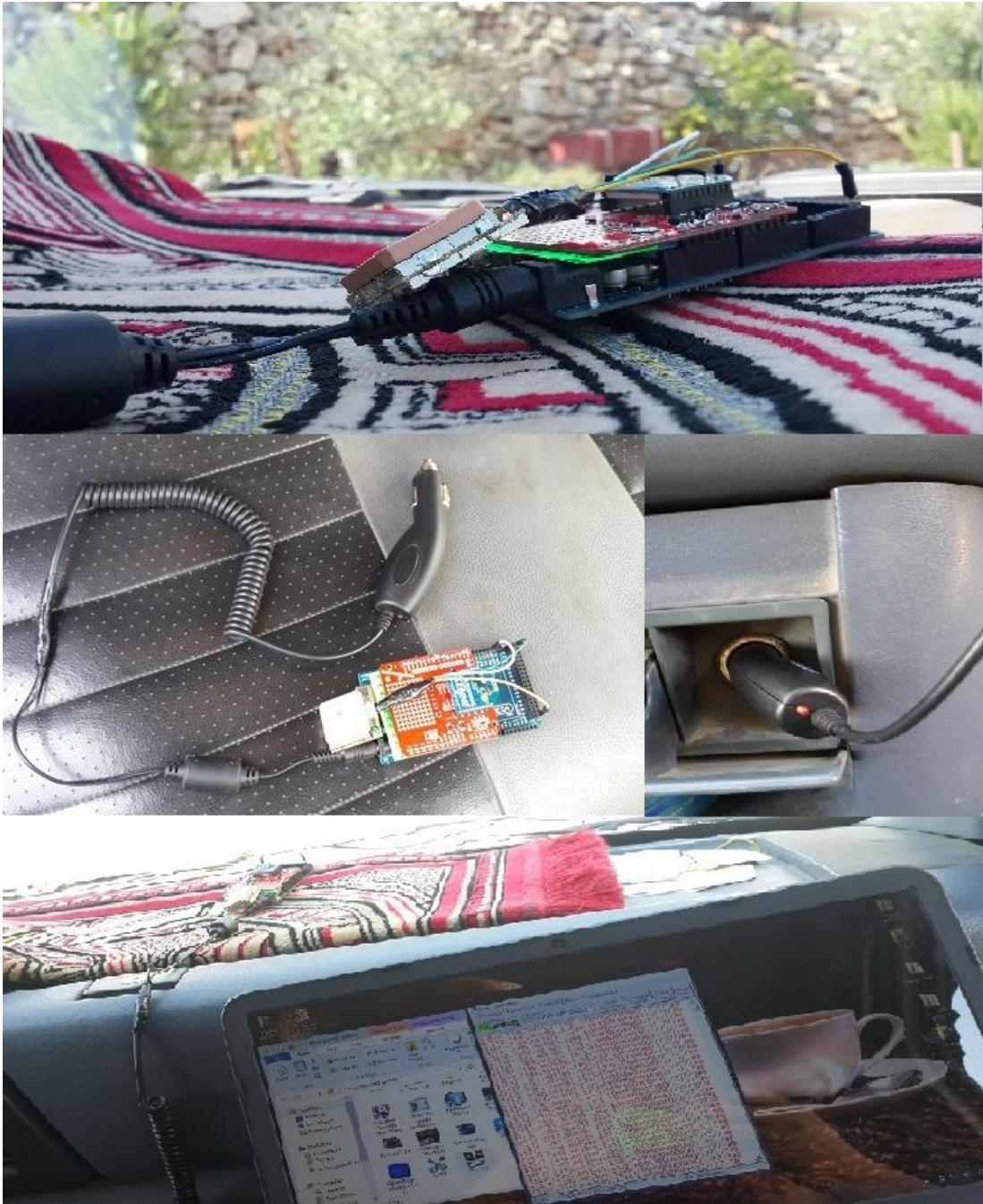


Figure 7.5: Speed test setup

The figure shows how we connected the car sub-system and how we powered it, the laptop was used to receive the readings using the Zigbee and the dongle. Figure 7.6 shows the results we got, the speed is presented in bytes 25, 26, 27 and 28.

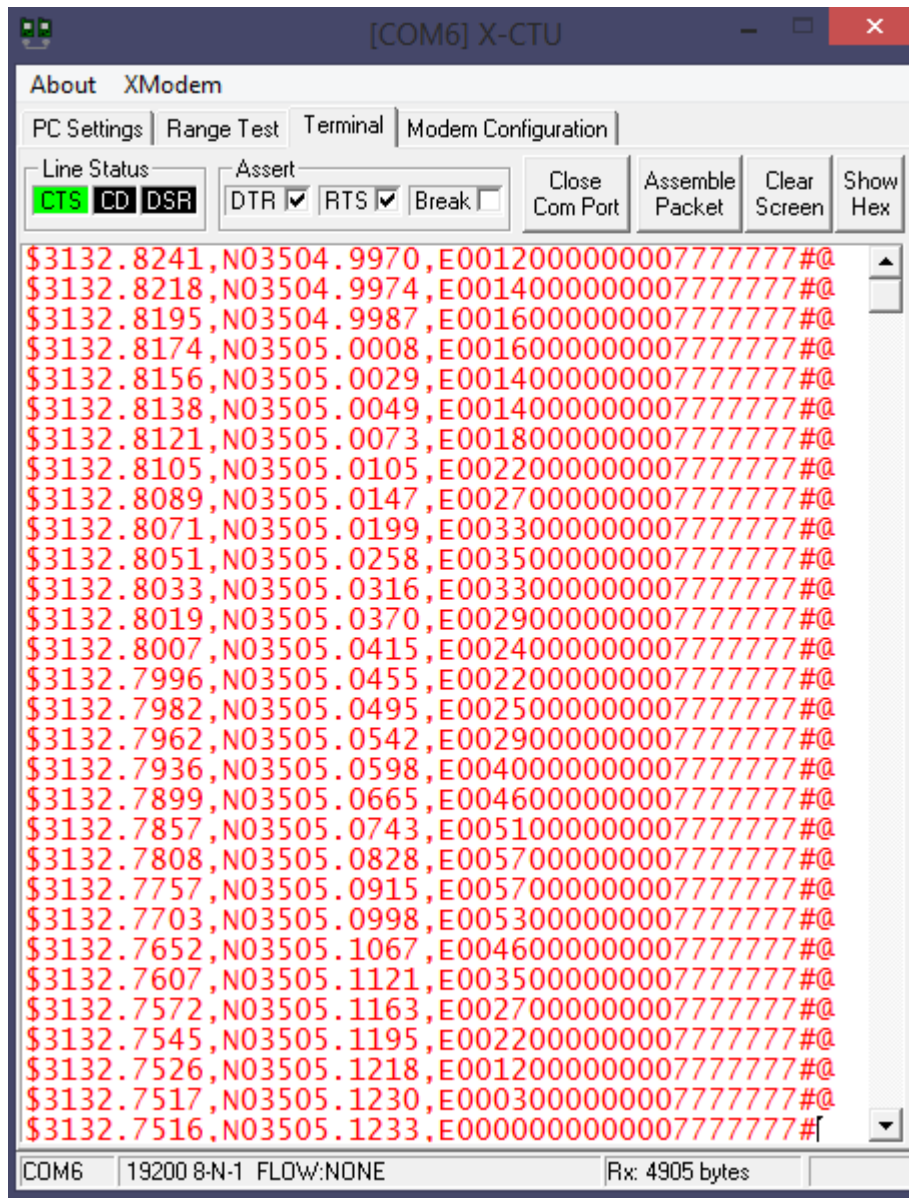


Figure 7.6: Speed Test Results

7.2.4 TCP connection between GPRS modem and server test

In this test, we dialed directly with GPRS modem serial using its AT commands through an FTDI cable, we opened a socket, sent a test data and then closed the socket, the AT command set that we used was (AT+CGACT ... AT+CGATT ... AT+SDATACONFIG ... AT+SDATASTART ... AT+SSTRSEND).

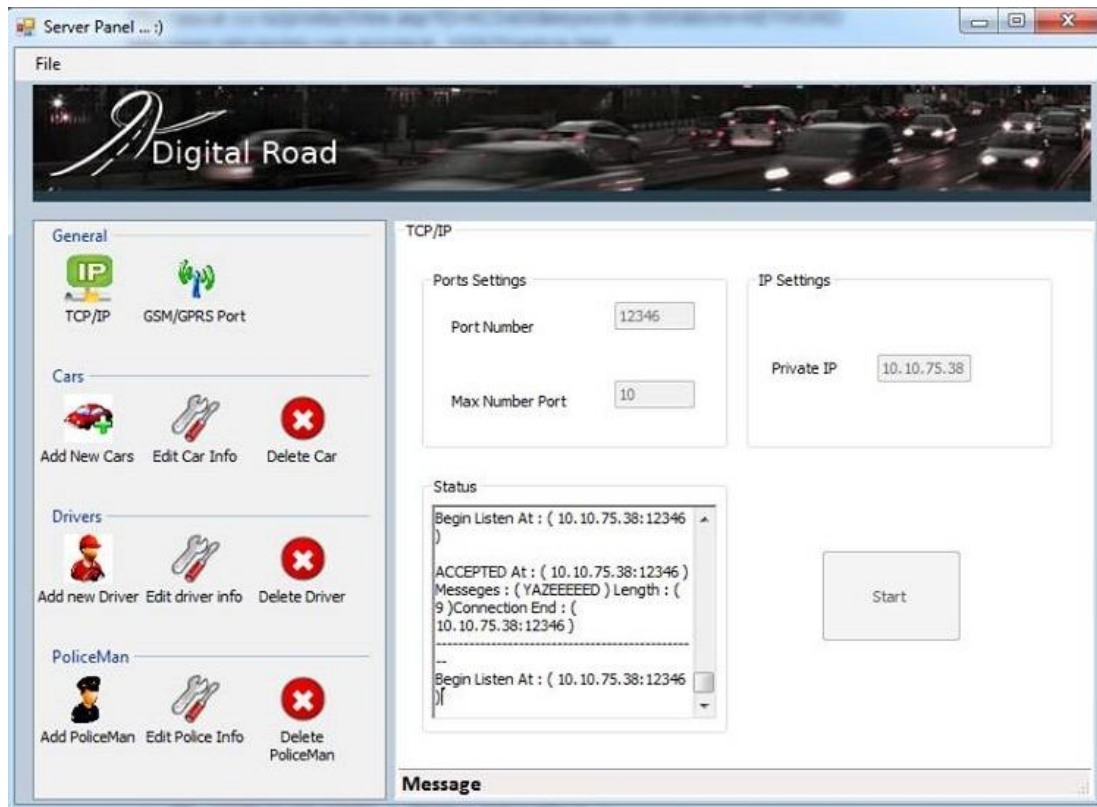


Figure 7.7: GPRS Connection Test

7.3 Software Testing

7.3.1 Connection Setup: Server, Drivers, Policeman

Server Start Listening test

Fig (7.4) shows the main interface for the server which represents the intermediate layer between the drivers and the police men, the first procedure is listening on the port available by the admin. Which can be used at any time. In Figure 7.4, the server started listening on ports 12345, 12347, 12348.

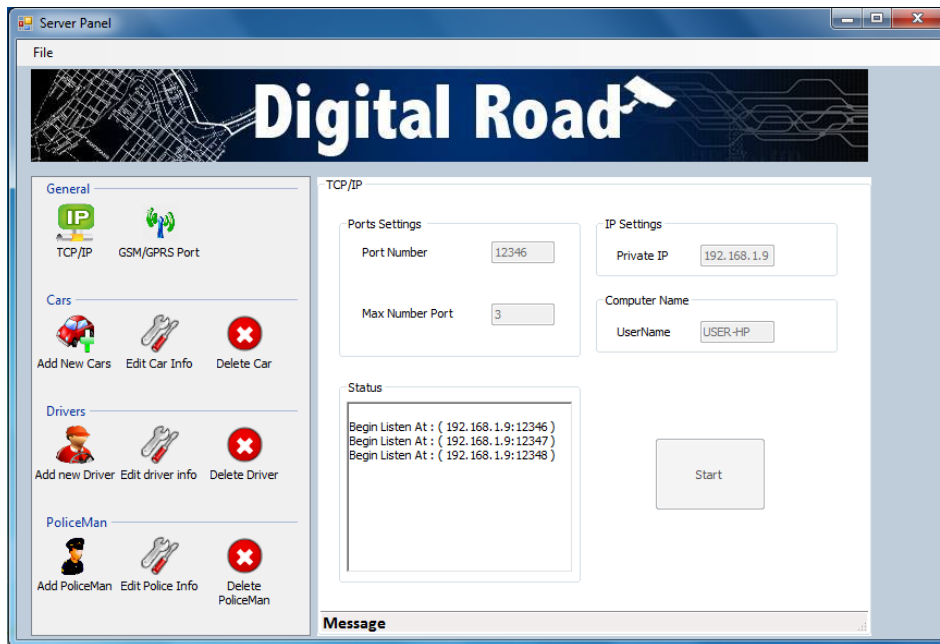


Figure 7.8 : Server Sart Listening

Policeman and Drivers Starts Connecting to Server

Fig (7.5, 7.6) shows the policeman and driver application starts connecting to the server with specified IP and PORT Number.

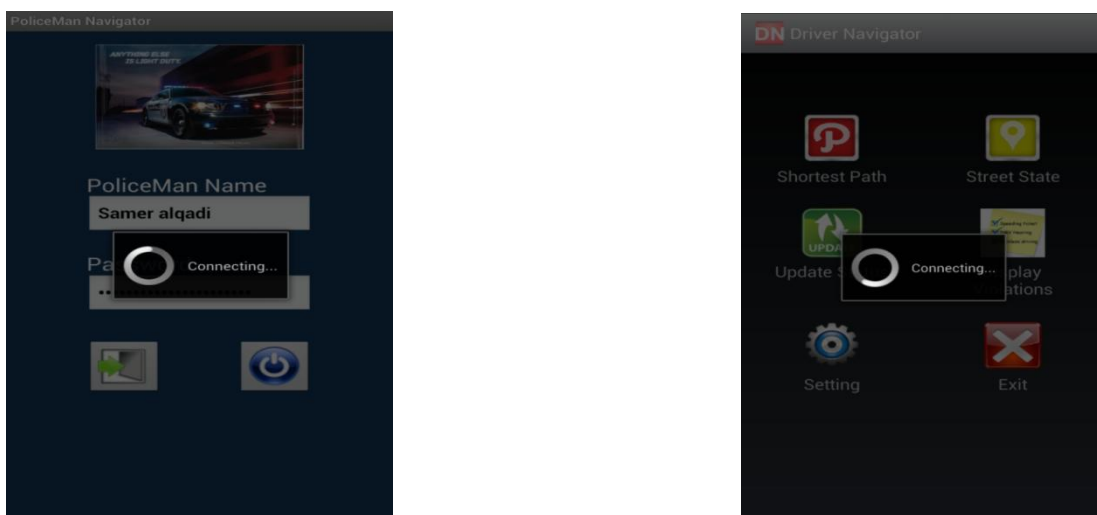


Figure 7.9: The Policeman and Driver Connecting To Server

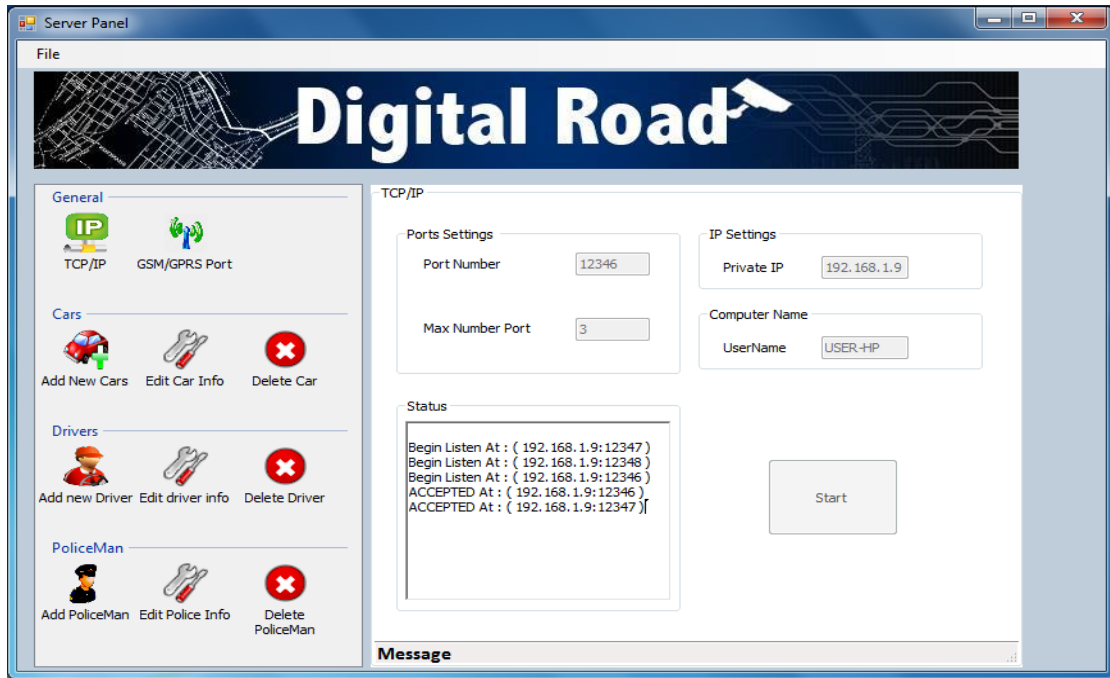


Figure 7.10: Server Established The Connection With Policeman and Driver.

7.3.2 Policeman Interact with Server

Send Trace Request:

In Fig. 7.11, the policeman send trace request to the server. The sever response by the car latitude, longitude which is displayed on map.

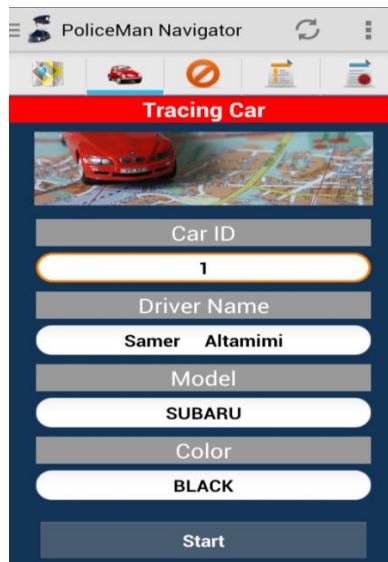


Figure 7.11: Policeman Trace Car By Car ID.

Fig. 7.12 shows the server received messages and displays the message and the length after the connection is established with the police man, then it parse the message and perform query on the database and return the result immediately.

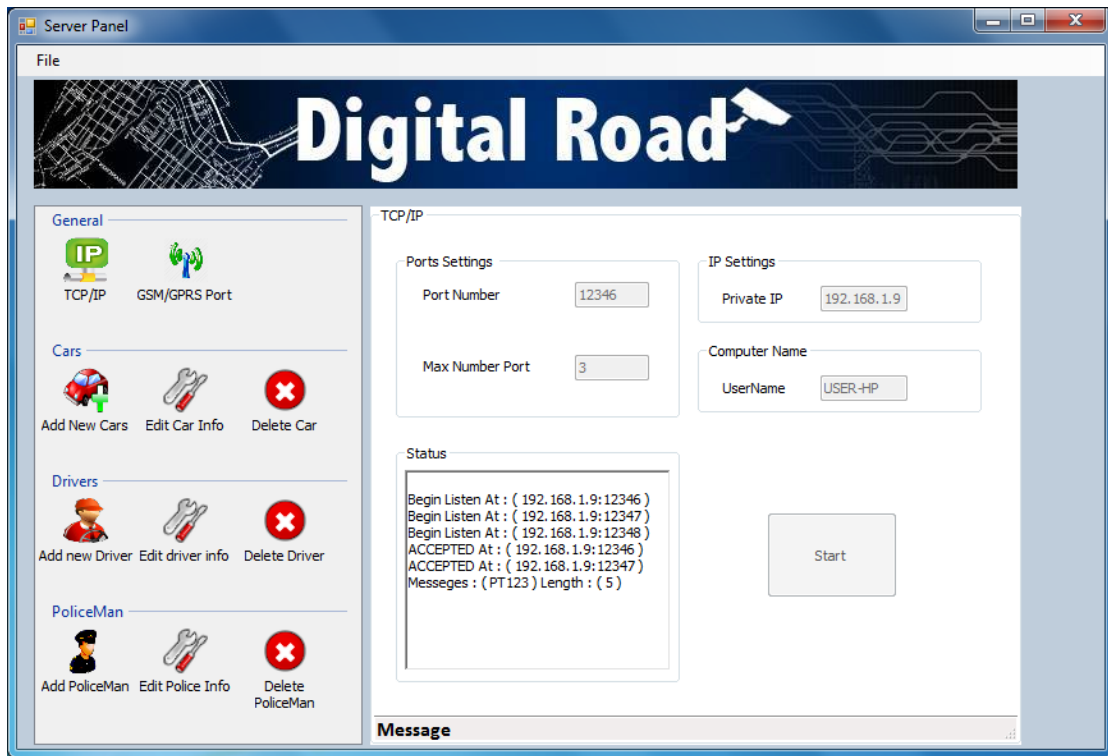


Figure 7.12: Server Receive Message and parse it.

Policeman application receive the response from the server and parsing data, then display last known locations about the car on the map.



Figure 7.13: Tracing Car application

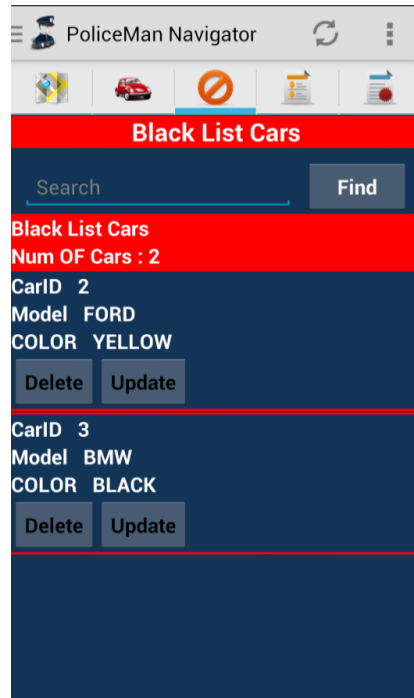


Figure 7.14: Black list application

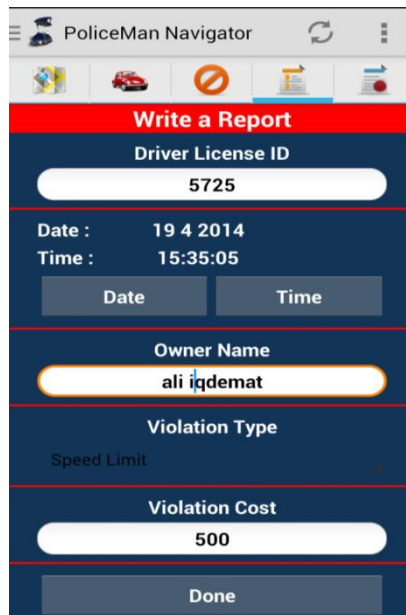


Figure 7.15: Write Violation

7.3.3 Driver Request Street State

Drivers can Read Response, Display Street Status on Map and Update Street Status

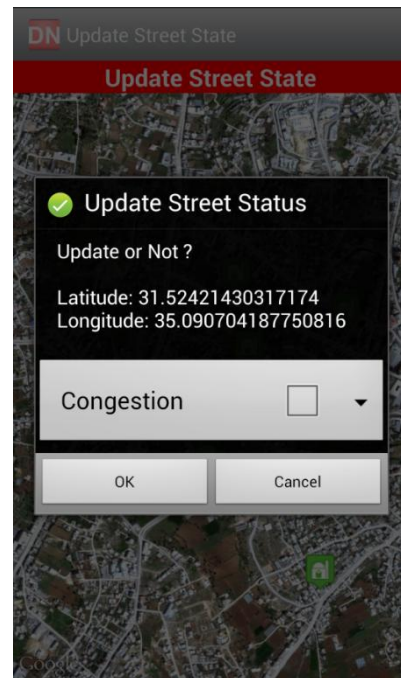
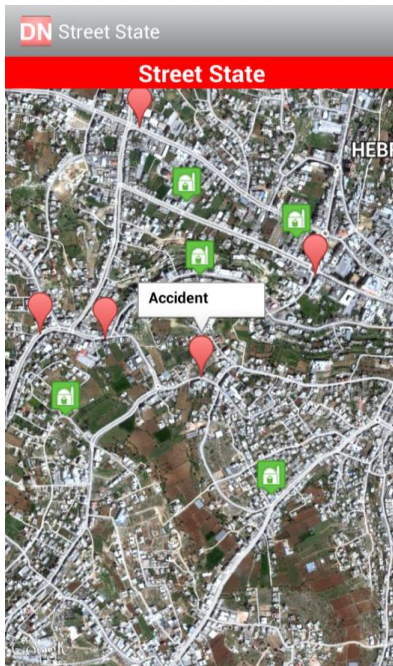


Figure 7.16: Street State with It Is Type **Figure 7.17:** Driver Update the Street Status

Figures (7.16, 7.17 and 7.18) on first figure street state update automatically each 5 second which is known (Keep a Life Technique), on the second the driver can update it in his location or by choosing other location, the state of the street could be one of the shown in Fig. 7.18.

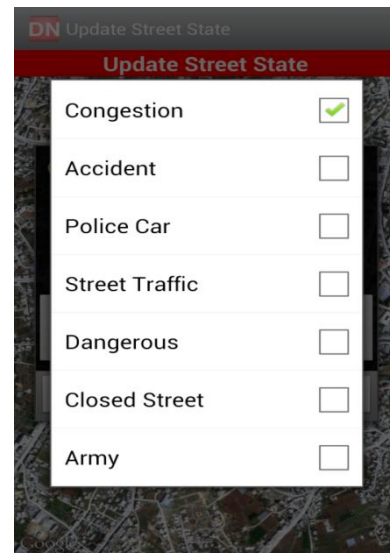


Figure 7.18: Street state application

7.3.4 Additional Functions

Shortest Path between Two Points

Figure 7.19 shows the result of booking two points on the map. After choosing the points, a request from the police man to Google server will be sent, it contains the first point (longitude and latitude) and the second point (longitude and latitude) and the mode of reaching the second point like driving or walking. Google server response with a JSON Object which contains the routing data as polygon and asset of information to reach the destination.

Fig. 7.19 shows the source and destination points and the distance (1.9 km), in addition to the duration (3 min) using driving mode.

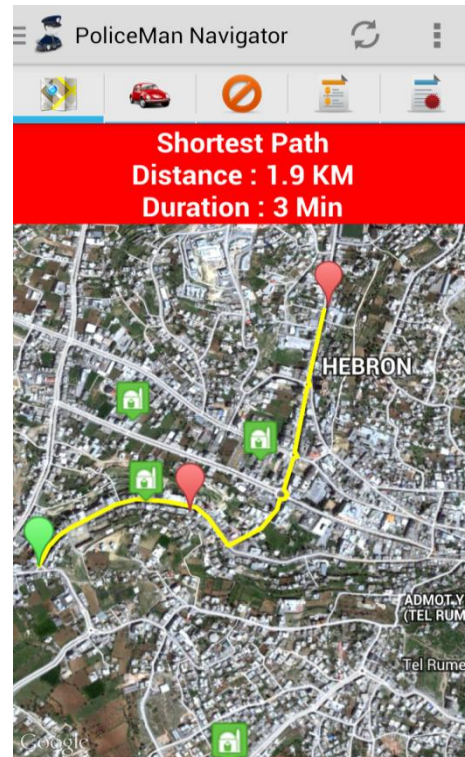


Figure 7.19: Shortest Path

A dark blue vertical bar runs down the left side of the page. A blue arrow-shaped graphic points to the right from the bar, containing the number 8.

8

Chapter Eight

Conclusion and Recommendations

8.1 Conclusion

8.2 Recommendations

8.1 Conclusion

The System Designed to be the fastest in response for street violations, congestion controlling, accident notification and detection and reduce the wasted time to reach your destination.

Also Our System was implemented to notify the Ambulance, policeman and the driver about the (violations, street state ...etc.) done on the roads. The time when the action done to the time when the driver, policeman or ambulance receive the notification examined (approximately) 2-3 seconds.

The covered area by the Zigbee chip give us 64 meter approximately, so every 128 meter we have a Zigbee chip in order to make the car always in the covered area range.

The system give us an accurate result for many cases. Cars locations, accidents, violations identified accurately with 1 – 10 meter error. Also the system work in the tunnels and other complicated street.

8.2 Recommendations

For the future works, we recommend the following:

- Try to use extended range XBee modules to reduce the cost even more.
- More applications could be simply added to the project with no change on the hardware structure, just like all kinds of traffic violations.
- Connect the car sub-system to the cars microcontroller, to ensure that no car can run if the system is down.
- Increase the security in the wirelessly transmitted data, find more ways for encryption.

References

- [1]: Ahamed, S. S. Riaz, The role of ZigBee technology in future data communication system, Journal of Theoretical & Applied Information Technology; 2009, Vol. 5 Issue 2, p129, February 2009
- [2]: Ahamed EL-Rabbany, Introduction to GPS: The Global Positioning System, Atrech house, Poston, London, 2002.
- [3]: William Lee, "Mobile Cellular Telecommunications", McGraw Hill, 2000.
- [4]: GPRS General packet Radio service, white paper,Usha Communication Technology, 26 june 2000.
- [5]: Modern Transport Telematics 11th international conference on transport systems Telematics, TST 2011,Katouice-Ustron,Poland , October 2011 , selected papers ..
- [6] [Online]: Japanese site:Vehicle Information and Communication System Center <http://www.vics.or.jp/english/vics/index.html> .
- [7]: graduation project, " Intelligent traffic with automatic violation system", Khalid I. Baradia, Abdullah Z. Ja'abary, Hassan A. Ja'afreh, Mustafa F. Zaitown, Palestine Polytechnic university 2012.
- [8]: Venkata Ramesh Gompa , "Innovative Congestion Control System for Ambulance Using ZigbeeWireless",Jul 03, 2012
- [9]: Tech Metrix," Java Application Servers Report", 1999
- [10]: [Online] Difference between java and C# <http://www.differencebetween.info/difference-between-java-and-c-sharp>.
- [11]: [Online] Oracle Database Concepts http://docs.oracle.com/cd/E11882_01/server.112/e25789/intro.htm#CNCPT88782.

- [12]: [Online] Difference between Oracle and MySQL
<http://www.differencebetween.info/difference-between-oracle-and-mysql>.
- [13]: *Milan Verle*, "PIC Microcontrollers", mikroElektronika; 1st edition (2008).
- [14]: [Online] arduino microcontroller product specification, feature, software
<http://arduino.cc/en/Guide/Introduction>.
- [15]: [Online] The 3rd Generation Partnership Project [3GPP]:
<http://www.3gpp.org/DynaReport/0340.html>.
- [16]: [Online] Encyclopaedia Britannica:
<http://www.britannica.com/EBchecked/topic/1473553/Wi-Fi>.
- [17]: Beginner Developer Learning Center
<http://msdn.microsoft.com/en-us/beginner/bb308730.aspx>.
- [18]: [Online] C# Programming Language Future Features
<http://msdn.microsoft.com/en-us/library/aa289180>
- [19]: [Online]: MS SQL capable and used
<http://www.host-shopper.com/what-is-ms-sql.html>.
- [20]: [Online] The latest news and product updates <http://www.android.com/about/>.
- [21]: [Online] Tutorials about Eclipse, Java, Android and Web development
<http://www.vogella.com/articles/AndroidSQLite/article.html>.