SOFT GUARD

GSM/GPS Based Vehicle Tracking System

By

NC Areeb Ahmed Mir

NC HamdaBinteAjmal

PC SulemanNadeem

CaptainTehseen Nawaz

Submitted to the Faculty of Computer Science

National University of Sciences and Technology, Rawalpindi in partial fulfillment

for the requirements of a B.E Degree in Computer Software Engineering

AUGUST 2011

ABSTRACT

SOFT GUARD

GSM/GPS Based Vehicle Tracking System

Soft-Guard is a vehicle surveillance system that effectively employs technology of GPS (Global Positioning System) and GSM (Global System for Mobile Communication) to remotely track the vehicle and controls some of its functionality, even if the user is miles away from the vehicle. The system permits localization of the vehicle and transmitting the position to the owner on his mobile phone as a short message (SMS) at his request. The domain is not a new one there are a lot of vehicle-tracking systems available in the market which employ the GPS technology to track the vehicle. However those trackers are not user controlled, the owner of the car has to contact the company for finding out the location and has to request the company if the need arise (in case of theft) to shut down the car engine. This is problematic for the user as he has to first call the tracking company to find out the location of his car or to shut it down which is a hassle. Soft-Guard gives complete control over the tracking in the hands of the user; using GSM module in the tracker and his/her mobile handset the user will be able to communicate with and direct the tracker as required. The presented application is a low cost solution for tracking vehicle position and status in real-time, very useful in case of car theft situations and for monitoring adolescent drivers by their parents.

Soft-Guard has been developed on Microcontroller (PIC18F542) that is connected with a GSM module and a GPS receiver. It continuously takes data from GPS, extracts the location, updating the previous one every time. As soon as it receives an SMS, it breaks all its activities and handles the interrupt, i.e. it reads the SMS and responds of user commands. User commands are short and precise and have a definite format. The code is developed in MikroC (Compiler: MikroC PRO for PIC) and simulated in ISIS Proteus Professional. It is burnt into the Microcontroller that is connected on a training Kit for testing purposes.

Software programming techniques, code optimization and modular design have been kept in mind during development. The code is thoroughly tested from unit to deployment level in order to remove bugs, improve user interaction and achieve efficiency and reliability.

Thus, the main objective of this project is to develop a vehicle tracking system which can track and monitor the vehicle in the porch and on the roads alike. It also acts as an anti-theft system and prevents the car from getting stolen. The most significant objective of this project is to place absolute control of this system in the hands of the user whether they are inside the car or miles away from it; this will be done by adding the functionality of control using GSM technology and mobile phones.

This project is completed under supervision of MajDrAsifMasood [HoD IS Dept, MCS].

COPYRIGHT NOTICE

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of any other award or qualification either at National University of Sciences and Technology or any other institution.

DEDICATIONS

I dedicate this project to my parents

(Areeb Ahmed Mir)

I dedicate this project to my parents and husband

(HamdaBinteAjmal)

I dedicate this project to my parents

(SulemanNadeem)

I dedicate this project to my family and Pakistan

(CaptTehseen Nawaz)

ACKNOWLEDGMENTS

We are very grateful to Allah Almighty for giving us the confidence and strength for carrying out this project and for making us able to come up with an embedded system as well as a product that will not only be useful in the current era, but will shall be used in times to come in Pakistan as well as across the globe.

We also thank our instructors and friends for their moral and technical suggestions without which we might not have made it this far. Also we thank Mr. Aziz Ur Rehman [NASCOM] for his technical support and guidance.

Above all, we are deeply thankful to our beloved parents for their patience, support and prayers that helped us gain our goal. We simply cannot thank them in words for they sacrificed their own comfort in order to give us a life that we need and by sponsoring us throughout our life.

Table of Contents

Chapter 1

INTRODUCTION

1.1 Preface

Soft Guard is an embedded system designed to track, monitor and control the vehicle using mobile phone via SMS. The vehicle tracking system has a GPS receiver and GSM modem connected with the control board to communicate with the PIC18F452 microcontroller which directs and coordinates the operations of the tracker. This system is absolutely under the command of the vehicle owner, and can be configured using predefined commands sent through SMS. Most vehicle tracking systems commercially available are not user controlled instead the owner has to contact the tracking company to inquire the location of car or to cut the engine in case of theft. Soft Guard overcomes these issues it not only allows the owner to monitor the location of the car it also notifies the owner about the state of the car engine, engine temperature, oil pressure, speed of the car etc. Furthermore it allows the owner to control the car by disabling the ignition system if needed.

We have chosen GSM technology for communication between the tracker and the owner of the car because it is very common and SMS is an easy to use option for the vehicle owner and does not require learning difficult procedures for using the tracking system in addition SMS is very cheap and GSM and GPS is accessible globally. And no further cost is incurred for accessing GPS signals.

1.2 Project Vision

The present car tracking systems are very expensive and user does not have the full control and management of the system. So the current system will give the entire control and management of the vehicle to the user. And also it is very cost effective.

1.3 Proposed Solution

The solution for above mentioned problems is to track the vehicle's location using GPS and to inform the user about the location of the vehicle via GSM. Also, cater for two modes of operation, High Alert Mode and Normal Mode.Then to provide engine cut feature in case of theft of the vehicle and to generate automatically speed alerts.Especially, provide geo-fencing of the vehicle and generate door opening alerts and low battery alertsto the user. Lastly, make it a cost effective system to reduce the cost of the system.

1.4 Aim of the Project

The aim of the project SOFT-GUARD is to implement a GSM/GPS based car tracking system involving communication between the microcontroller, GSM, GPS modules and the owner's mobile. The implementation is mostly software based, no such hardware implementation is required in our project but the development of this system includes understanding of the associated hardware and how they can be operated using coding.

This could be achieved by developing an embedded system with following main modules.

1.4.1 GPS Tracking

GPS tracking is done using the GPS receiver which is connected with the microcontroller and is continuously sending the data to the microcontroller.

The microcontroller is responsible for extracting latitude, longitude and speed form the received data.

1.4.2 Messaging through SMS

User will be able to communicate with the tracking system via SMS which he can send through his cell phone. The tracking system contains a GSM modem which is connected to the microcontroller. The microcontroller is responsible for interpreting the message and taking the necessary action.

1.4.3 Alerts Notification

The sensors in the vehicle are connected with pins of the microcontroller to receive the input signal from the sensors. These signals denote the state of the sensors. The microcontroller is polling to check the states of these sensors and in case of an anomaly the user is notified via SMS.

1.5 Organization of Project Report

The project report has been drafted carefully deciding the sequence to be followed. After the introduction section, the report incorporates the Literature Review chapter summarizing the text studied and research aspects of our project before and during the project's execution. Subsequently, the System Analysis chapter comes which includes the major interface, functional and non-functional requirements of the system. Next is the System Design chapter comprising of the architectural diagram, high and low level design, data flow diagrams, events driven architecture and use case diagram. Following this the report includes the Implementation chapter identifying and elucidating the structural programming which is implemented. Then is the testing chapter incorporating the testing process employed to test the system and the results that were obtained. The next chapter then discusses the work that can be done in future to further enhance the system and ultimately this chapter wraps the report.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Few areas needed to be studied in order to get a better understanding of some basic knowledge required for developing Soft Guard. Topics which were studied are: Global Positioning System [GPS], how GPS works and its components, Global System for Mobiles [GSM], components of GSM, services and specifications of GSM, and how GSM works, MicroC Pro for PIC and Proteus software.

2.2 Global Positioning System

2.2.1 GPS

GPS employs orbiting navigational satellites for transmitting position and time data to the handheld receivers on earth. These handheld receivers then use the received data to calculate the latitude, longitude, altitude and velocity. GPS was developed by Department of Defense.[1]

2.2.2 History of the GPS

In 1969 Defense Navigation Satellite System (DNSS) was formed which was the foundation of GPS. Later in 1973 NAVSTAR Global Positioning System was brought into existence. 5 years later first 4 satellites were launched into the space known as Delta rocket. Over the next 15 years 20 more satellites were launched providing initial operational capability for GPS. After 2 years i.e. in 1995 full operational capability was achieved. It was finally in May 2000 that Military accuracy was made available to all users.[2]

2.2.3 Components of the System

Space segment; the space segment of GPS consists of 24 satellite vehicles and six orbital planes which are inclined 55^o with respect to equator and each orbit separated by 60^o. The orbit is at 20,200 km elevation above Earth. Satellite’s orbital period is of 11 hr 55 min precisely. At any time five to eight satellites can be seen from any point on Earth.

User Segment; the user segment of GPS consists of GPS antennas & receiver/processors. These receivers give accurate position, velocity, precise timing etc depending upon the data received from the satellite.

These GPS receivers are used in various applications such as aircraft, ground vehicles, and ships and by individuals (in mobile phones etc.).

2.2.4 How does GPS work?

Distance to a satellite is determined by measuring how long a radio signal takes to reach us from that satellite. To make the measurement we assume that both the satellite and our receiver are generating the same pseudo-random codes at exactly the same time. By comparing how late the satellite's pseudo-random code appears compared to our receiver's code, we determine how long it took to reach us. Multiply that travel time by the speed of light and you've got distance. Accurate timing is the key to measuring distance to satellites, and satellites are accurate because they have four atomic clocks ($100,000 each) on board. Receiver clocks don't have to be too accurate because an extra satellite range measurement can remove errors. To use the satellites as references for range measurements we need to know exactly where they are. GPS satellites are so high up therefore their orbits are very predictable. All GPS receivers have an almanac programmed into their computers that tells them where in the sky each satellite is, moment by moment. Minor variations in their orbits are measured by the Department of Defense. The error information is sent to the satellites, to be transmitted along with the timing signals.[3]

2.3 Global System for Mobile

Global System for Mobile (GSM) is a second generation cellular standard developed to cater voice services and data delivery using digital modulation.[4]

2.3.1 History of GSM

Developed by Group Spéciale Mobile (founded 1982), an initiative of CEPT (Conference of European Post and Telecommunication). The aim of developing GSM was to replace the incompatible analog system. Presently the responsibility of GSM standardization resides with special mobile group under ETSI (European telecommunication Standards Institute). Full set of specifications phase-I was made available in 1990.

Under ETSI, GSM is named as “**G**lobal **S**ystem for **M**obile communication“. Today many providers all over the world use GSM (more than 135 countries in Asia, Africa, Europe, Australia, America) with more than 1300 million subscribers worldwide.[5]

2.3.2 GSM Services

GSM provides three services:

Tele Services these are telecommunication services that enable voice communication via mobile phones. This service offers mobile telephony and emergency calling.

Bearer Services Includes various data services for information transfer between GSM and other networks like PSTN, ISDN etc at rates from 300 to 9600 bps. Short Message Service more commonly known as SMS is also provided by bearer service. SMS allows up to 160 character alphanumeric data transmission to/from the mobile terminal. It also provides Unified Messaging Services (UMS), group fax, voice mailbox and electronic mail.

Supplementary Services provides call waiting i.e. notification of an incoming call while on the handset; call hold - put a caller on hold to take another call; call barring- all calls, outgoing calls, or incoming calls; call forwarding- calls can be sent to various numbers defined by the user; multi party call conferencing - link multiple calls together; CLIP – Caller line identification presentation; CLIR – Caller line identification restriction and CUG – Closed user group.

2.3.3 GSM System Components

GSM system has various components. These components are Mobile Station (MS), Mobile Equipment (ME), Subscriber Identity Module (SIM), Base Station Subsystem (BSS), Base Transceiver Station (BTS), Base Station Controller (BSC), Network Switching Subsystem (NSS), Mobile Switching Center (MSC), Home Location Register (HLR), Visitor Location Register (VLR), Authentication Center (AUC) and Equipment Identity Register (EIR).

2.3.4 GSM Specifications

This section gives specifications of a GSM system. The frequency range specified for GSM is 1,850 to 1,990 MHz (mobile station to base station). The duplex distance is 80 MHz. Duplex distance is the distance between the uplink and downlink frequencies. A channel has two frequencies, 80 MHz apart.The separation between adjacent carrier frequencies. In GSM, this is 200 kHz. Modulation is the process of sending a signal by changing the characteristics of a carrier frequency. This is done in GSM via Gaussian minimum shift keying (GMSK). GSM is a digital system with an over-the-air bit rate of 270 kbps. GSM utilizes the time division multiple access (TDMA) concept. TDMA is a technique in which several different calls may share the same carrier. Each call is assigned a particular time slot. GSM uses linear predictive coding (LPC). The purpose of LPC is to reduce the bit rate. The LPC provides parameters for a filter that mimics the vocal tract. The signal passes through this filter, leaving behind a residual signal. Speech is encoded at 13 kbps. [6]

2.4 MikroC Pro for PIC

MikroC Pro for PIC is a Mikroe proprietary compiler for mikroC language, specifically designed for Microchip PIC microcontrollers. It provides the programmers with a list of libraries containing easy to use built in functions forwriting high level code for the microcontrollers which can be then compiled to generate the .hex file which is then burned into the microcontroller.

The syntax of the language is similar to C and C++, only the functions for the microcontrollers and handling the communication are specialized and unique.

2.5 Proteus

Proteus is software for microprocessor simulation, schematic capture, and printed circuit board (PCB) design. It is developed by Lab Center Electronics. It allows the creation of virtual circuits on the computer to test it, with ease, before creating or testing the actual hardware/circuit.

Chapter 3

SYSTEM ANALYSES

3.1 Introduction

This chapter covers the system analysis phase of the project. In this phase, first of all scope of the project is presented as it's clear definition and understanding is needed for the absolute comprehension of the system's requirements' specification phase, including major functional and non-functional requirements, is described. The requirements' specification phase is then followed by use case diagram and domain model of the system for the better understanding of the system analysis phase of the project.

3.2 Project Scope

The basic aim of our project is to implement a car tracking system using GPS and GSM module which would be controlled by PIC18F452 microcontroller. The project is mostly software based with major focus on programming of the microcontroller rather than developing the hardware. The main goal of this product is to create a car tracking system which is fully user controlled through mobile phones instead of being company dependent and provides anti-theft and alarm system as well as car monitoring features. The product is very cost effective as compared to the present systems available in the market. Implementation of the project includes design and development of hardware, microcontroller programming and demonstration of the final product.

3.3 Requirements Specification

3.3.1 Introduction

3.3.1.1 Purpose

This Software Requirements Specification document is written to record and analyze the requirements of Soft Guard. This document covers all the features and components of Soft Guard system. Also this document provides a complete description of all the functions and specifications of the GSM and GPS based car tracking system (Soft Guard), how and under what environment it operates, how is it controlled and how it responds to triggers.

3.3.1.2 Intended Audience

Project Team, so that they could see at any stage that which requirements have been met and which is left? And alsoso that they could complete the project within scope in all perspectives.

Project Evaluation Team, so that the evaluation team knows what scope of project is andwhat are the requirements of project?

Industry Experts, so that they are able to analyze the utilization of the product. And after seeing its utilization in the present market, can make out a decision to buy it and produce it on a large scale to be sold in the open market.

Users of System, so that they know what this is made to do? What are its functional and non-functional requirements? How to use it efficiently?

3.3.1.3 Project Scope

The basic aim of our project is to implement a car tracking system using GPS and GSM module which would be controlled by PIC18F452 microcontroller. The project is mostly software based with major focus on programming of the microcontroller rather than developing the hardware. The main goal of this product is to create a car tracking system which is fully user controlled through mobile phones instead of being company dependent and provides anti-theft and alarm system as well as car monitoring features. The product is very cost effective as compared to the present systems available in the market. Implementation of the project includes design and development of hardware, microcontroller programming and demonstration of the final product.

3.3.2 Overall Description

3.3.2.1 Product Perspective

Soft Guard is an unprecedented system. Although car tracking system itself is not a novel system, many car tracking systems are available in the market, which make use of the GPS to track the car. Soft Guard however uses the combo of GPS and GSM to provide full control of car tracking literally in the owner’s hand using his/her mobile phone. The software will be embedded in the microcontroller which will be at the heart of the tracking device, interfaced externally with car’s engine.

3.3.2.2 Product Features

The software will provide a list of features. These include tracking of the car’s location using GPS, informing of the location to the user via GSM (using SIM card). The system will provide engine cut feature in case of car theft so that it can be stopped.It will provide automatic speed alerts. The system will also incorporate geo fencing feature – the capability of cutting engine automatically if the car crosses the predefined boundary for e.g. if it crosses the boundary of Islamabad without shutting/altering geo fencing perimeter the engine will be cut.It will provide oil pressure level, temperature level and filter condition alerts.The system will provide door open alerts. An engine start alert will also be incorporated.It will provide 2 modes of operation, High alert mode and Normal mode. In **High alert Mode**, this is the more secure mode the system can be set to high alert mode when the user is away from the car so it will automate certain security tasks which would have to be done manually in normal mode. In **Normal Mode**, this is the normal mode of security in which automatic alerts will be minimal instead the alerts will be sent on user request.

3.3.2.3 User Classes and Characteristics

Users will be of two classes, Administrators and Normal users. **Administrators**are those users which are responsible and have privilege to install and deploy the system in the vehicle. This type of users should have proper training of the installation and deployment and basic features functioning of the system. **Normal Users** are those users who should only know how to use mobile. As they have to interact with the system via mobile to give commands and to respond to the alerts texted to them.

3.3.2.4 Operating Environment

Since the software is embedded in the microcontroller, controlling the entire car tracking system, it will be housed inside the car, interfaced through external hardware with the car engine. It will receive its power from the car battery and will be operating 24/7.

3.3.2.5 Design and Implementation Constraints

Understanding the working of car and its engine, and interfacing the system with the car might become an issue for us, due to limited know-how of this field. Another limitation is the memory of microcontroller, which is very scarce for flexible and efficient programming, thus developing an efficient code with the available memory resource is a major challenge.

A GSM jammer can make the system non-functional; however various security measures will be applied to keep the system up and running even in such condition.

3.3.2.6 User Documentation

The User Manual has been prepared containing all the necessary information about the Soft Guard system and troubleshooting procedure.

3.3.2.7 Assumptions and Dependencies

We assume that the persons who would be directly responsible for installing and deploying the system in vehicles are that much literate that they can not only understand the functioning of the system, but also the functioning of vehicle computer. And they can remove the errors and bugs if appear at any stage.

In the same way if we talk about the users of the system, the users are also believed to be the persons having knowledge of how to use a mobile phone. It is because they have to give commands and reply to system on the alerts generated by the system on their mobiles.

3.3.3 System Features

This project has a number of functional requirements which will be programmed during the development process. Following is a brief overview of these features, and our aims and vision of the final project put into words.

3.3.3.1 GPS car tracking

3.3.3.1.1 Description and Priority

This is the most fundamental feature of the product. This user or the owner of the vehicle, at any desired time can get the latitude, longitude and speed of the vehicle with an accuracy of 15m.

3.3.3.1.2 Stimulus/Response Sequences

The mobile number of the user or the owner of the vehicle is authorized to communicate with the GSM installed in the car. The user messages the car a specific code assigned to track the location and speed of the vehicle (let’s assume the code is “get location”.

The microcontroller decodes the message and invokes a function that receives data from the GPS module. It then parses the data, and extracts the latitude and longitude from NMEA $GPGGA header and speed from NMEA $GPVTG header. These values are sent back to the owner’s mobile phone number via a text message. The user now knows the exact location of his vehicle.

3.3.3.1.3 Functional Requirements

The software must be capable of receiving SMS message from user.It must decode the message (“get location”) and invoke the function of GPS tracking upon reception.It must parse the latitude, longitude at which the vehicle is located and its speed.It must store the values of latitude, longitude and speed in memory.It must send these values back to the cell phone number of the owner via AT commands.

3.3.3.2 Safety Modes

3.3.3.2.1 Description and Priority

The running software will have two modes: High alert Mode and Normal Mode. The system will be in either of these two modes at a time. High alert mode is for severe safety purposes. User can change the mode of the vehicle to high alert whenever he/she feels that the car is parked at a place where greater risk of theft or at night time. In this mode, there will be strict security measures.In normal mode, moderate safety measures will be taken. It will be the default mode.

3.3.3.2.2 Stimulus/Response Sequences

(Setting or changing the Mode)The owner of the vehicle sends aSMS to the vehicle with a specific code (let’s assume “high alert” or “normal mode”).The microcontroller receives the sms and decodes it upon reception.If the message is “high alert” (sent by an authorized telephone number), the microcontroller invokes a function which will turn on the high alert mode and activate all its functions (described in the later section).

If the message is “normal mode”, the microcontroller invokes a function which turns on the normal mode and activates all its function.

3.3.3.2.3 Functional Requirements

The software must be capable of receiving SMS message from user.It must decode the message (“high alert” and “normal mode”) and change the mode of the tracker.Normal mode must be the default mode.

3.3.3.3 Engine Cut

3.3.3.3.1 Description and Priority

For security measures, the owner of the car often needs to turn off the engine of the car, especially in conditions when the car has been stolen. When the engine stops, obviously the vehicle cannot move forwards and soon, after tracking the car through the GPS, the owner or the police can reach to the exact location where the car is.

Engine cut function will function in two different conditions. 1) When the car is in normal mode, engine will only be stopped when the user requests for it.2) When the car is in high alert mode, the engine will be stopped whenever some suspicious activity is sensed by the microcontroller.

3.3.3.3.2 Stimulus/Response Sequences

When Engine Cut in Normal Mode, the owner of the car (having an authorized telephone number) sends an SMS to the system. Let’s assume that the code of this function is “Engine Cut”. The microcontroller receives the message via GSM kit installed in the car. It decodes the message and invokes the corresponding function which cuts down the engine, stopping the vehicle wherever the car is. After this, the microcontroller invokes the GPS tracking function which gets the longitude and the latitude of the car location and the speed of the car and sends it back to the owner via an SMS. This will be the confirmation to the owner that the engine has been cut down at a specific location.

When Engine Cut in High Alert Mode and the door of the car opens, it sends an SMS (“Door Opens”) to the owner of the car. The microcontroller expects the owner to respond in two ways, Owner knows that the car is safe, and changes the mode of the car to Normal (For sequence of steps, please refer to section 3.2) and Owner fears a theft attempt and requests an Engine cut to stop the car where it is. (This will work same as the Engine Cut in Normal Mode)

In Automatic Shutdown, after 5 minutes (exact time still under consideration) have passed and the microcontroller does not receive a response from the owner via GSM, it automatically cut downs the engine. This is because is suspects the following cases: 1) Owner has not received the SMS yet (might be some network error or delay). 2) Owner has received the message but not read it yet.3)A GSM jammer has been used by the thieves to block the messaging between the car and the owner.So, after the timeout, the microcontroller automatically shuts down the engine.

3.3.3.3.3 Functional Requirements

The software must be capable of receiving sms message from user.It must decode the message (“engine cut” or “Normal Mode”).It must be interfaced with the car engine in a way that whenever it receives the “engine cut” message, it cuts down the connection of engine from car.It must have a timer function to calculate the time passed from the instant the “Door Opens” SMS has been sent to the owner.It must wait for 5 minutes for a response from the owner. Itmust decode the response message.If the response message is “Engine Cut”, it must cut the engine.If the response message is “Normal Mode”, change the mode of the car to Normal and do not cut the Engine.

3.3.3.4 Speed Alerts

3.3.3.4.1 Description and Priority

This feature is especially useful for parents who worry their children (or driver) over speed the car. The system will alert the owner via an SMS whenever the car exceeds the predefined speed.

3.3.3.4.2 Stimulus/Response Sequences

At the time of configuring the GPS tracker, the owner of the car stores a speed limit (let us assume 100Km/h).Whenever the car crosses the speed of 100km/h, the microcontroller sends an SMS containing text “Over speeding” to the owner of the car.

3.3.3.4.3 Functional Requirements

It must be able to detect the speed whenever it gets above 100km/hr and generate an interrupt.Whenever the interrupt is received by the microcontroller, it should send an SMS to the owner of the car via GSM.

3.3.3.5 Geo Fencing

3.3.3.5.1 Description and Priority

Reports each time the car moves in or out of a predetermined area from 2 meters to and infinite distance. These geo-fences are rectangular with reporting based upon whether the device is inside or outside the geo-fence area or when it has simply crossed the geo-fence boundary. When Geo-Fence is enabled the microcontroller will send an SMS message to advise the owner that the Geo-Fence boundary has been crossed.

3.3.3.5.2 Stimulus/Response Sequences

At the time of tracker configuration (at startup), the user inputs a geo fence (max and minimum values of longitude and latitude). Whenever the vehicle crosses the limit, it sends an SMS message to the owner containing text "Geo Fence Crossed". It then waits for a response from the user. There can be three cases: 1) The user sends a response message "OK". The microcontroller decodes this message and leaves the situation as it is. 2) The user sends a response message "Engine Cut". Engine cut function is activated (refer to section 3.2). 3) Owner does not respond at all. The microcontroller waits for 5 minutes (Time limit under consideration).If the user does not respond in these 5 minutes, there can be three cases; a) Owner has not received the SMS yet (might be some network error or delay), b) Owner has received the message but not read it yet, c) A GSM jammer has been used by the thieves to block the messaging between the car and the owner.So, after the timeout, the microcontroller automatically shuts down the engine.

3.3.3.5.3 Functional Requirements

It must be continuously taking data from GPS and parse the latitude and the longitude values, updating them every time it receives a $GPGGA header of NMEA protocol. Whenever the latitude gets less than or equal to the minimum latitude or gets more than or equal to maximum latitude (same for longitude), it generates an interrupt.When the interrupt is generated, the microcontroller should send an SMS message to owner of the car warning him at the Geo Fence has been crossed.

3.3.3.6 Door Open Alerts

3.3.3.6.1 Description and Priority

If the car is in "High- Alert Mode" and the door is opened, the tracker must send a SMS message to the owner warning him about this suspicious movement.

3.3.3.6.2 Stimulus/Response Sequences

When the door of the car is opened, micro controller sends an SMS message to the owner of the car via GSM containing text "DOOR OPENED". There can be three responses; 1) the user sends a response message "OK". The microcontroller decodes this message and leaves the situation as it is. 2) The user sends a response message "Engine Cut". Engine cut function is activated (refer to section 3.2)Owner does not respond at all. 3) The microcontroller waits for 5 minutes (Time limit under consideration).

If the user does not respond in these 5 minutes, there can be three cases;a) Owner has not received the SMS yet (might be some network error or delay). b) Owner has received the message but not read it yet. c) A GSM jammer has been used by the thieves to block the messaging between the car and the owner.So, after the timeout, the microcontroller automatically shuts down the engine.

3.3.3.6.3 Functional Requirements

It must continuously keep guard of the doors while in “High Alert Mode”.Whenever the door is opened, it generates an interrupt. When the interrupt is generated, the microcontroller should send an SMS message to owner of the car warning him that the door has been opened.Must be able to start a timer and wait for response from the owner. If the response isn’t received, it should cut off the engine, so that the car cannot start.

3.3.3.7 Engine Start Alert

3.3.3.7.1 Description and Priority

If the car is in “High- Alert Mode” and the engine is started, the tracker must send a SMS message to the owner warning him about this suspicious movement.

3.3.3.7.2 Stimulus/Response Sequences

When the engine starts, micro controller sends an SMS message to the owner of the car via GSM containing text “ENGINE START”. There can be three responses; 1) the user sends a response message “OK”. The microcontroller decodes this message and leaves the situation as it is. 2) The user sends a response message “Engine Cut”. Engine cut function is activated (refer to section 3.2). 3) Owner does not respond at all. The microcontroller waits for 5 minutes (Time limit under consideration).

If the user does not respond in these 5 minutes, there can be three cases; a) Owner has not received the SMS yet (might be some network error or delay). b) Owner has received the message but not read it yet. c) A GSM jammer has been used by the thieves to block the messaging between the car and the owner. So, after the timeout, the microcontroller automatically shuts down the engine.

3.3.3.7.3 Functional Requirements

It must continuously keep guard of the engine while in High Alert Mode. Whenever the engine starts, it must generate an interrupt. When the interrupt is generated, the microcontroller should send an SMS message to owner of the car warning him that the engine has been started.Must be able to start a timer and wait for response from the owner. If the response isn't received, it should cut off the engine, so that the car cannot start.

3.3.3.8 Installation, Customization and Configuration

3.3.3.8.1 Description and Priority

The user must be able to store or change his cell phone number as a valid authorized number where the messages from tracker will be received at any time. Default phone number is 0300-0000000. Maximum speed limit can be changed at any time. Default max speed if 120km/h. Geo Fence can be changed at any time. Default is Geo Fence of Pakistan. Password can be changed at any time.

3.3.3.8.2 Stimulus/Response Sequences

For setting up /changing the owner number, every tracker would come with a password programmed into it. The buyer of the tracker knows the password (let’s assume the password is “PASSWORD”).He sends an SMS message to the tracker containing text “change number; 03001234567; PASSWORD”.Microcontroller receives the message and decodes it. It checks for the correctness of the password.If the password is correct, it sets up the authorized telephone number of the owner.After setting up the telephone number, it sends an SMS message to the owner containing text “number changed”.If the password is not correct, it simply displays on its LCD “Wrong password”.

For Setting up the speed limit, owner sends an SMS to the vehicle containing text “speed limit; 130; PASSWORD”. Microcontroller decodes the message and checks whether the password is correct.If the password is correct, the microcontroller invokes the change speed limit function, and changes the speed limit to 130km/h.Else if the password is incorrect, the microcontroller sends an SMS message to the owner via GSM containing text “Invalid password”.

For setting up the Geo-Fence, the owner sends an SMS to the vehicle containing text “GeoFence; maxLatitude; MinLatitude; MaxLongitude; MinLongitude; PASSWORD”.Microcontroller decodes the message and checks whether the password is correct. If the password is correct, the microcontroller invokes the change Geo-Fence Limit function, and changes the values of the Geo-Fence.Else if the password is incorrect, the microcontroller sends an SMS message to the owner via GSM containing text “Invalid password”.

3.3.3.8.3 Functional Requirements

It must be able to check validate the password written in the received message. It must be able to check whether the message is received from the authorized telephone number. It must be able to decode the message correctly to invoke the corresponding configuration function.After the message has been decoded, and password has been validated, it must parse the data, and update the values provided in the message.If the password is incorrect, it should send a warning message to owner.After changing the value, it should send a confirmatory message to the owner.

3.3.3.9 Low Battery Alert

3.3.3.9.1 Description and Priority

The product must detect whenever the battery of the car is low (below a specified level) and send a message alert to the owner. This feature is dependent upon some special hardware designed for the purpose and will only be implemented if the hardware is available.

3.3.3.9.2 Stimulus/Response Sequences

Microcontroller generates an interrupt whenever the battery is low.When the interrupt is received, it sends an SMS message via GSM to the owner of the car containing text “Battery low”.

3.3.3.9.3 Functional Requirements

Microcontroller must be interfaced with the car battery, so that it knows when the battery is low. It must be able to send an SMS warning message to the owner.

3.3.4 External Interface Requirements

3.3.4.1 User Interfaces

User interfacing i.e. the mode of communication between user and system is mobile phone. User and the system interact with each other via SMS messaging through GSM (user will have a mobile phone and system will have a GSM kit installed in it).

3.3.4.2 Hardware Interfaces

GSM-Microcontroller interface:GSM module hardware needs to be interfaced with the microcontroller so that the microcontroller can receive, send and decode SMS messages.

GPS-Microcontroller interface: This interfacing is necessary so that the microcontroller can receive NMEA data from the GPS module all the time.

Microcontroller-Car Door interface: For detection of opening and closing of the door of a vehicle.

Microcontroller-Engine interface: For Engine cut function.

Microcontroller-Engine Filter interface: To detect the condition of engine oil filter.

Microcontroller-Oil Pressure interface: To detect low oil level.

Microcontroller-Temperature interface: To detect high temperature of the engine.

No hardware will be used explicitly by our system it will only require servers and client machines which are connected to a data network.

3.3.4.3 Software Interfaces

No software interfacing required.

3.3.4.4 Communications Interfaces

Communication between GPS and Microcontroller needs the mutual understanding of the NMEA protocol. GPS sends the data to microcontroller through NMEA protocol. Microcontroller serial port (the one which is interfaced with the GPS module) and GPS module both must be synchronized at the baud rate of 4800, sending/receiving 8 bits at a time, using 1 stop bit and no parity bits.Communication between GSM module and Microcontroller takes place through a serial cable. SMS message communication between user and GSM module takes place through AT commands. GSM supports TCP/IP protocol.

3.3.5 Nonfunctional Requirements

3.3.5.1 Performance Requirements

It is expected that the system would be able to perform with maximum efficiency and reliability. It must return the results of user queries within minimum possible time and execute commands messaged by user just as they are received (maximum priority). GPS is expected to return location and speed results with maximum accuracy. GSM module must also efficiently communicate with the microcontroller and the user via mobile telephone network. The system should be up and running in all conditions and weather.

3.3.5.2 Safety Requirements

As it is not a safety critical system, it is not required to implement high safety measures. However, it must be ensured that the system works even if the owner of the car doesn't respond or the car is in an area of no network coverage and outside the geo fence. By this, car cannot be stolen easily.

3.3.5.3 Security Requirements

Owner of the system must be verified by the combination of his telephone number and a password given to him, so that any other non - authorized person cannot change the car settings.

3.3.5.4 Software Quality Attributes

Adaptability: System must be adaptable to all vehicles by making a few changes in the hardware interfacing.

Correctness: Must return correct results and decode the messages properly.

Maintainability: Minimum efforts needed to make specified changes. Software must be analyzable, changeable, stable and testable.

Portability: System and its software shall be able to be installed in all types of vehicles, and must be designed in a way that it can be later incorporated in railway systems.

Reliability: System design should not be complex to raise reliability issue. It must be simple and efficient to deliver correct results within minimum time duration.

Reusability: Design of the system must be generic so that it can be later on installed in all types of moving object, e-g motorbikes, trains, all models of car.

Robustness: System must be fault tolerant. Techniques like defect detect, defect reduction and defect removal must be applied at the testing phase so that the system is robust to all errors, faults and failures.

Usability: It must be easy to use. Codes should be easy and memorable and system should warn the user every time he enters a wrong code.

3.3.6 Other Requirements

3.3.6.1 Hardware Requirements:

The hardware which are used in Soft Guard include; PIC18F452 microcontroller, GSM module, GPS module, LCD’s, switches and other hardware for circuitry, Programmer software for microcontroller. All these mentioned components will be integrated in a single PCB and installed in the vehicle. Lastly, user must have a mobile phone.

3.3.6.2 Software Required During Development

The softwares which are used includeMikroC, mikroElektronika C compiler for Microchip PIC microcontrollers, Version: 8.1.0.0, Proteus, ISIS professional for Simulation and Programmer Software to program the microcontroller.

3.4 Conclusion

The system analysis of the project has been covered in this chapter. The scope of the project has been revised for the clear understanding of the requirements, key functional and non-functional requirements have been enumerated, the use case diagram showing the major actors and their actions have been included. The sequence diagram identifying the sequence of actions taken has been incorporated as well. This chapter has been written comprehensively so that the fore coming design chapter becomes easy to comprehend.

Chapter 4

SYSTEM DESIGN

4.1 Introduction

System design is a very important phase in the software development process. The succeeding development phase is performed taking into consideration the design constraints. This chapter of System design provides the design details of the GSM/GPS based vehicle tracking system, Soft-Guard. It contains a complete description of the design of Soft-Guard. It describes the architecture of Soft-Guard which is based on event driven architecture, it also describes the use cases of the system, the data flow diagrams, state machine diagrams, data design and circuit design of the hardware that we are currently using to simulate the software for the tracker.

4.2 Overview

Next serial is the Architectural Design that specifies the design entities that collaborate to perform all the functions included in the system. It also shows the flow of the data that takes place in the system in the form of Data Flow Diagram. It further shows the state machine diagrams which show the transitions that take place in car’s hardware which trigger specific events such as door open alert.

After that, the serial concerns the Data Structure Design, followed by the Use Case diagram. And lastly, the serial discusses the Circuit Design currently being used to simulate the tracker and to test the software.

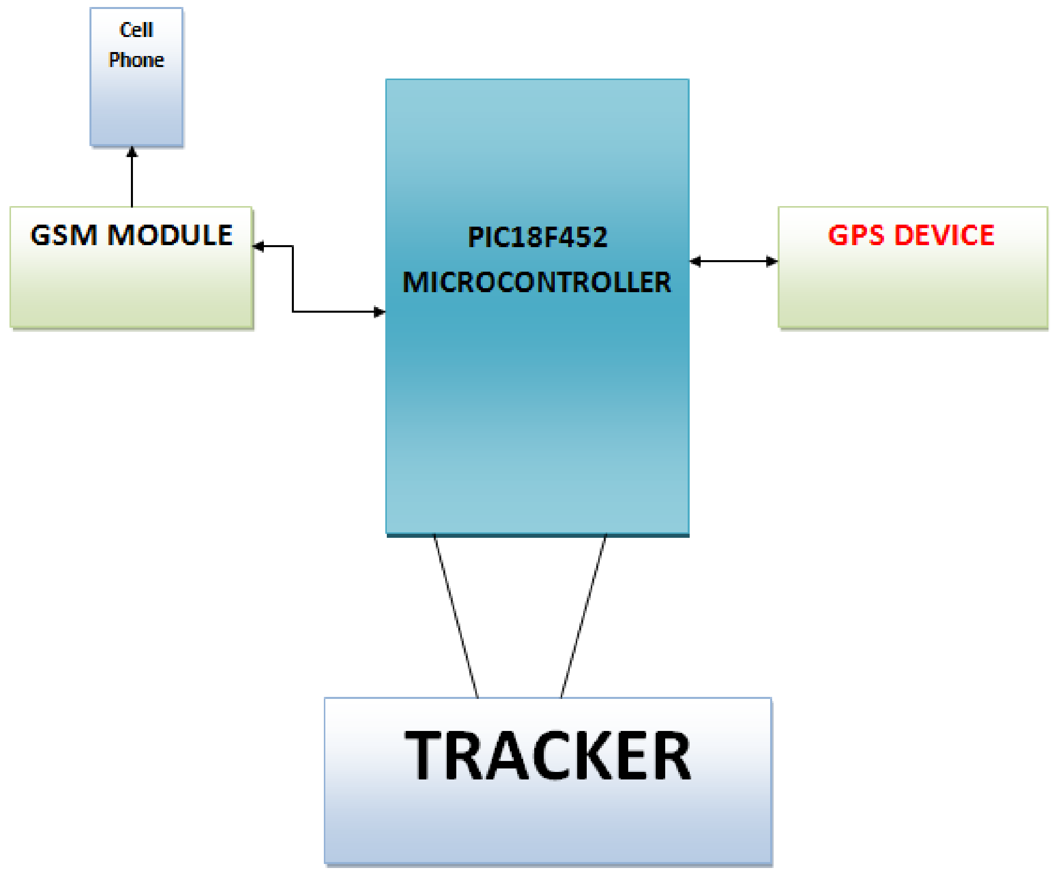
4.3 ARCHITECTURAL DESIGN

The architecture used for this system is an event triggered system. Since our tracking system is an embedded and a real-time system which responds to the actions/events triggered in the car such as door open, heating of the engine, dropping of engine oil level etc.; and it responds according to the user's SMS, that is why we are using Event Driven Architecture.

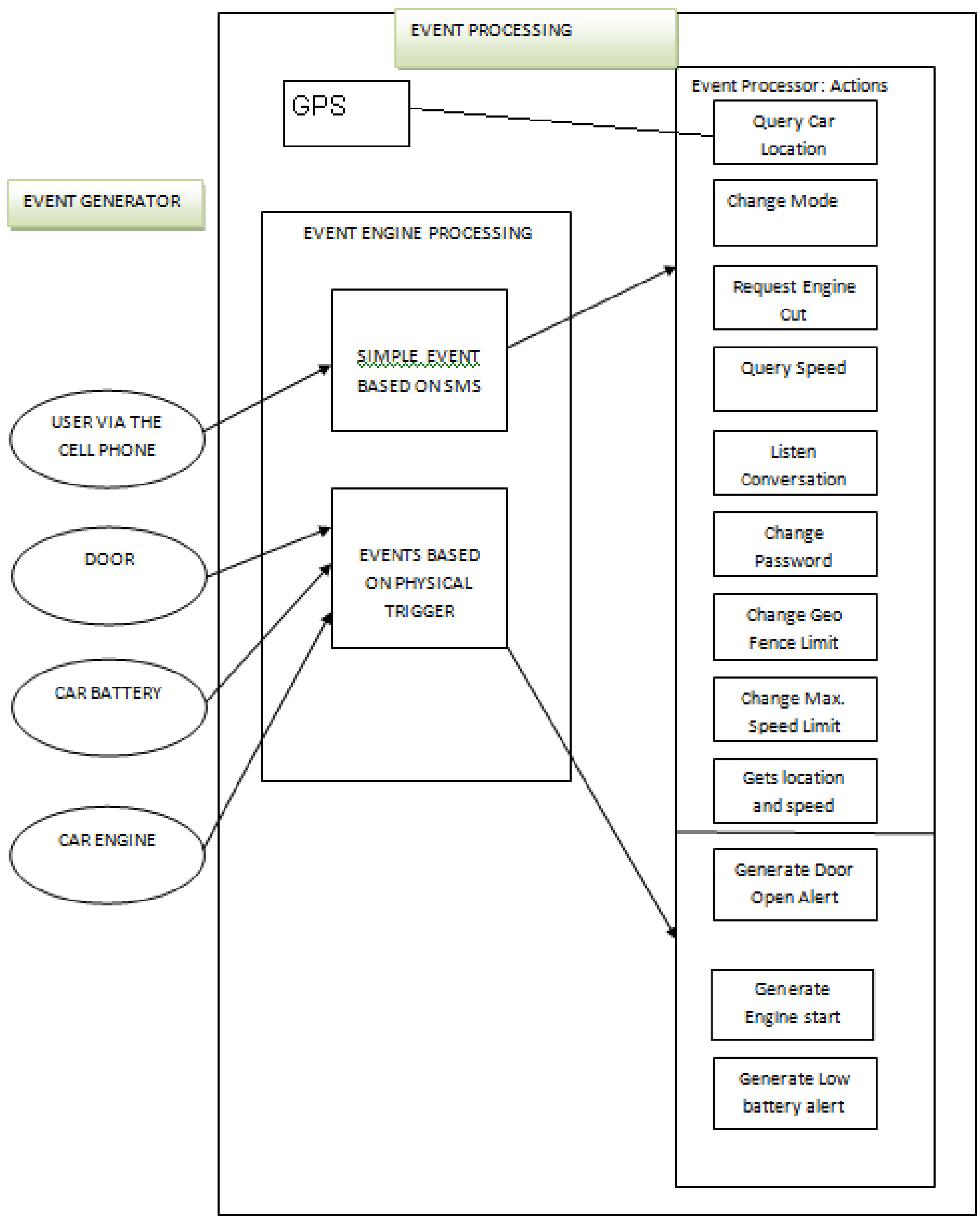
Event Driven Architecture:

The event driven architecture is a type of software architecture pattern that describes the generation, detection, processing and reaction to specific events. The term event can be defined as a significant change in state. These are the notable things that occur outside or inside our system which disseminates immediately to all interested parties (human or automated). The interested parties evaluate the event, and optionally take action. The event-driven action may include the invocation of a service, the triggering of a business process, and/or further information publication/syndication.

4.4 High Level Architectural Description:



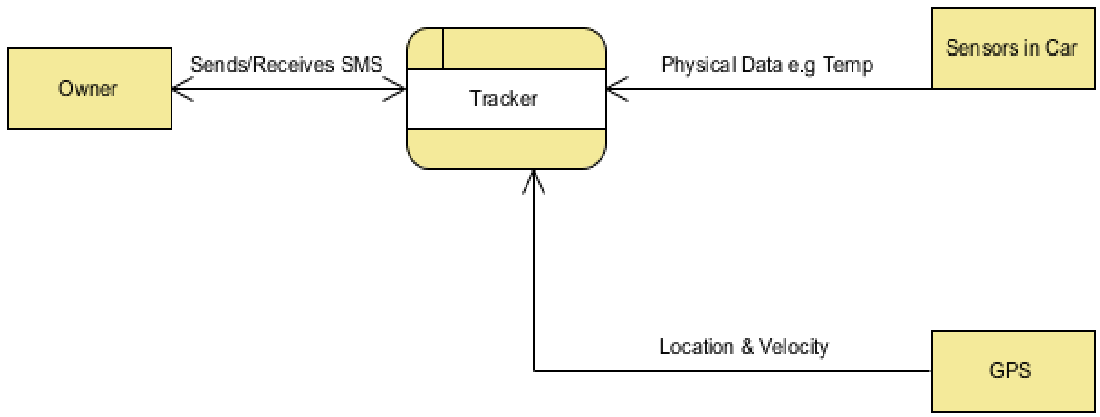
4.5 Event Driven Architecture:



EVENT PROCESSING
EVENT GENERATOR
GPS
Event Processor: Actions
Query Car Location
Change Mode
Request Engine Cut
Query Speed
Listen Conversation
Change Password
Change Geo Fence Limit
Change Max. Speed Limit
Gets location and speed
Generate Door Open Alert
Generate Engine start
Generate Low battery alert
EVENT ENGINE PROCESSING
SIMPLE EVENT BASED ON SMS
EVENTS BASED ON PHYSICAL TRIGGER
USER VIA THE CELL PHONE
DOOR
CAR BATTERY
CAR ENGINE

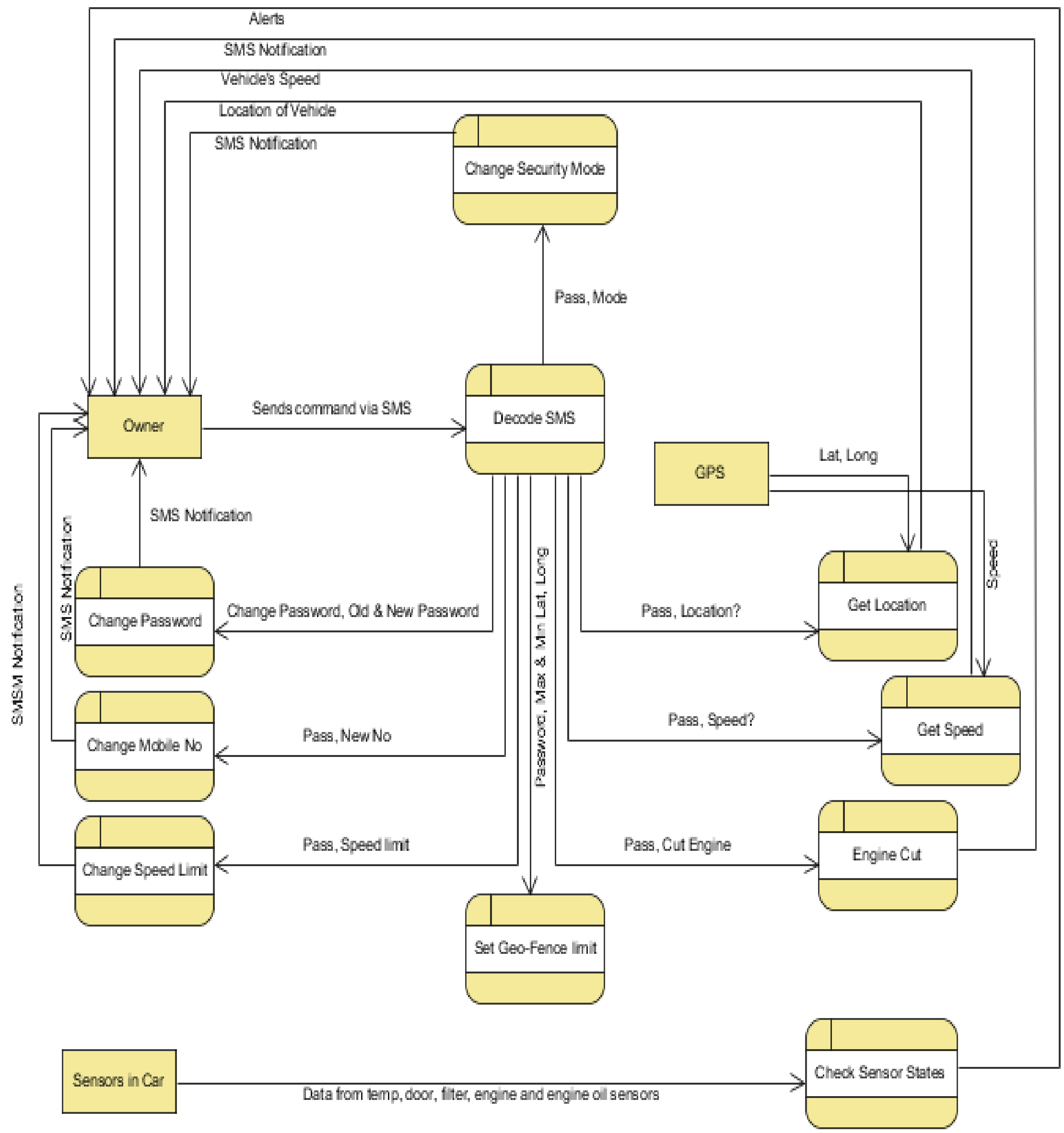
4.6 Component Decomposition Description:

4.6.1 Data Flow Diagram – Context Level:



The above diagram is the context level or Level 0 DFD. It shows the entities that cause the events to be triggered in the tracking system. The entities are the user (owner), sensors in the car and the GPS. The level 1 DFD is depicted below.

4.6.2 Level 1 Data Flow Diagram:

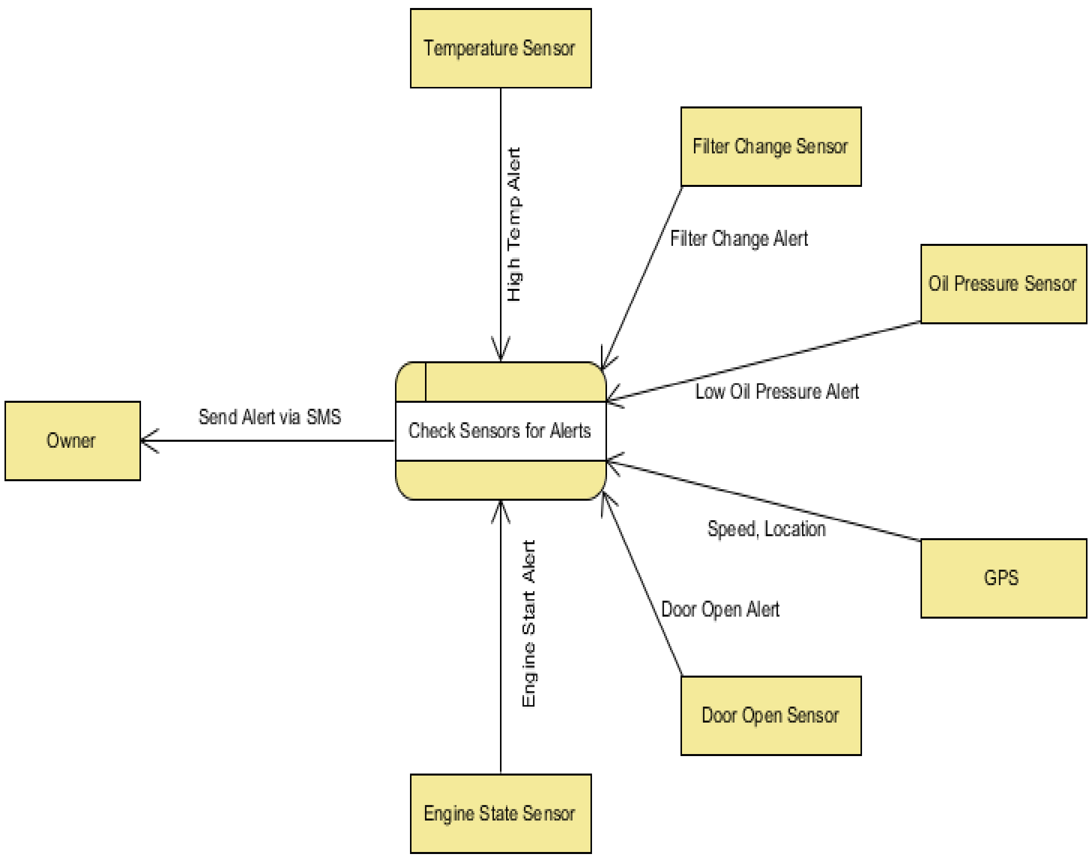


<Level 1 DFD of the system>

The level 1 DFD shows the flow of the data from the user to the tracker and then back to the user. The MCU in the tracker decodes the SMS performs the action that is commanded by the user in the SMS and then notifies the user. To check for the new SMS polling is used.

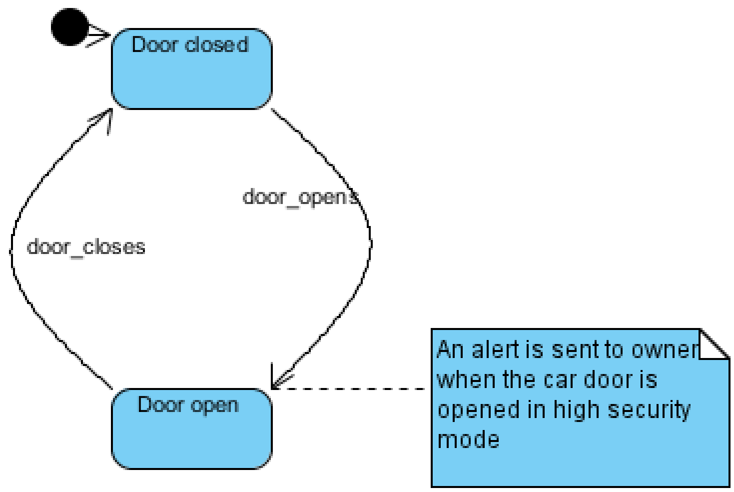
It also shows the generation of the alerts which may arise due to certain values from the sensors in the car for e.g. temperature sensor (alert is generated if engine heats up and the user is notified).

The process for generation of the alerts is further explained in Level 2 DFD.

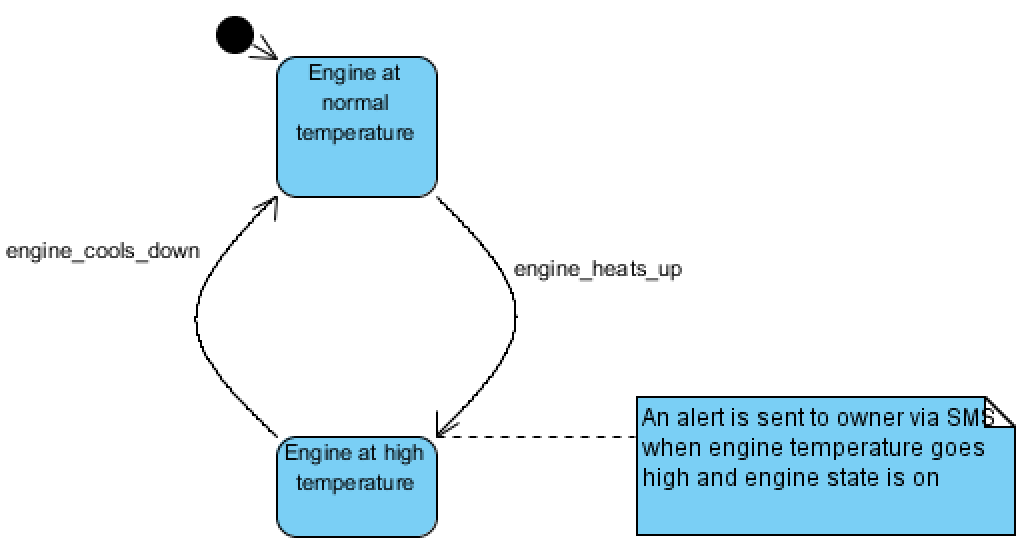


<Level 2 Data Flow Diagram of the system>

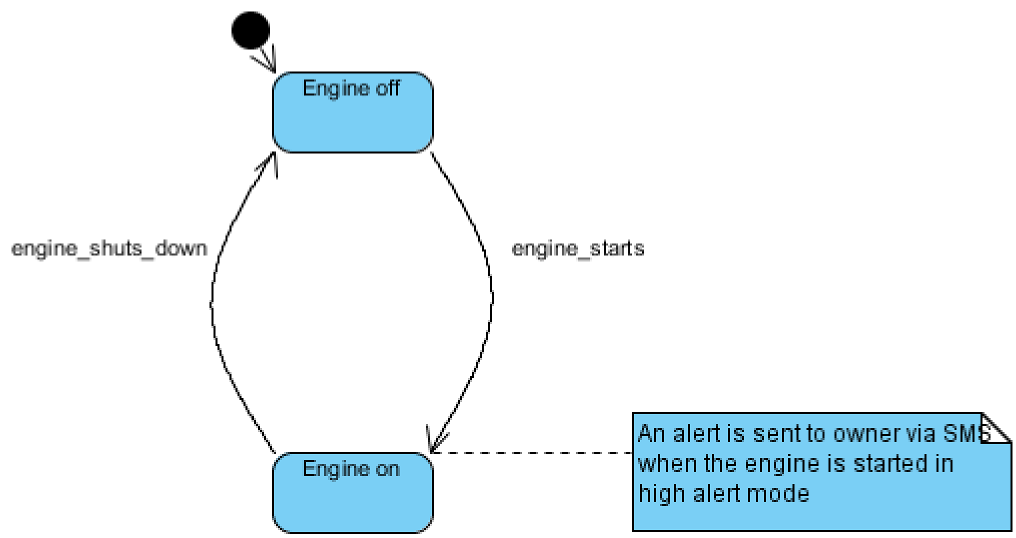
4.7 State Machine Diagrams:



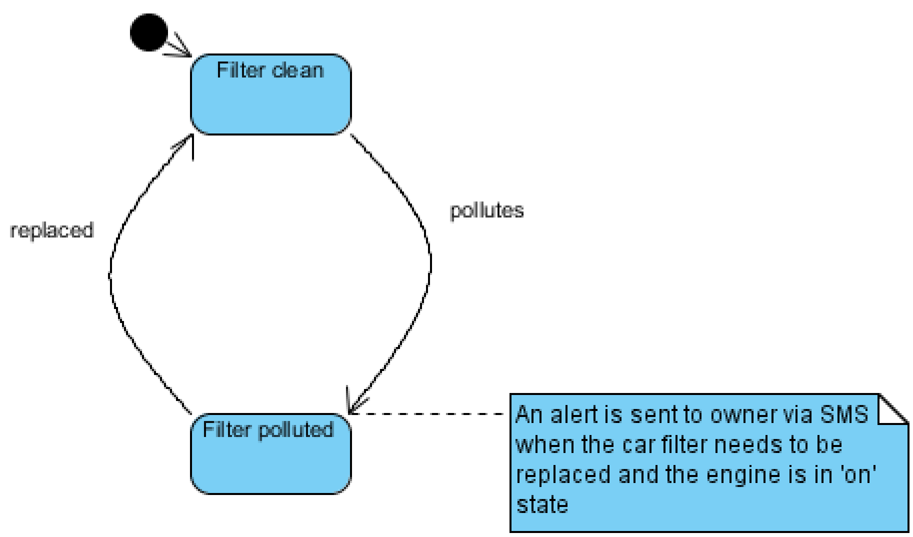
<Vehicle Door –State Machine Diagram>



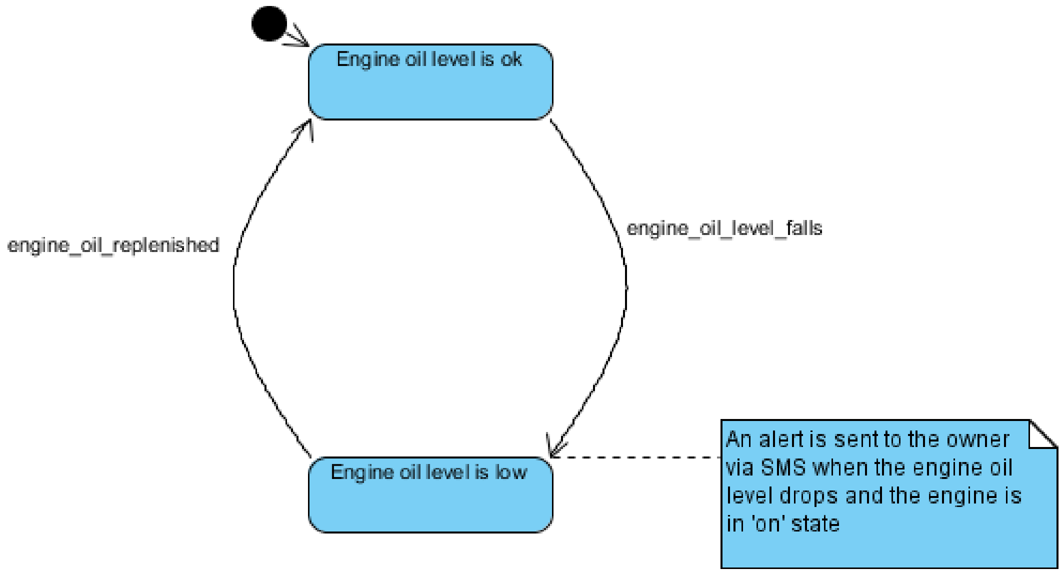
<Engine Temperature –State Machine Diagram>



<Engine –State Machine Diagram>

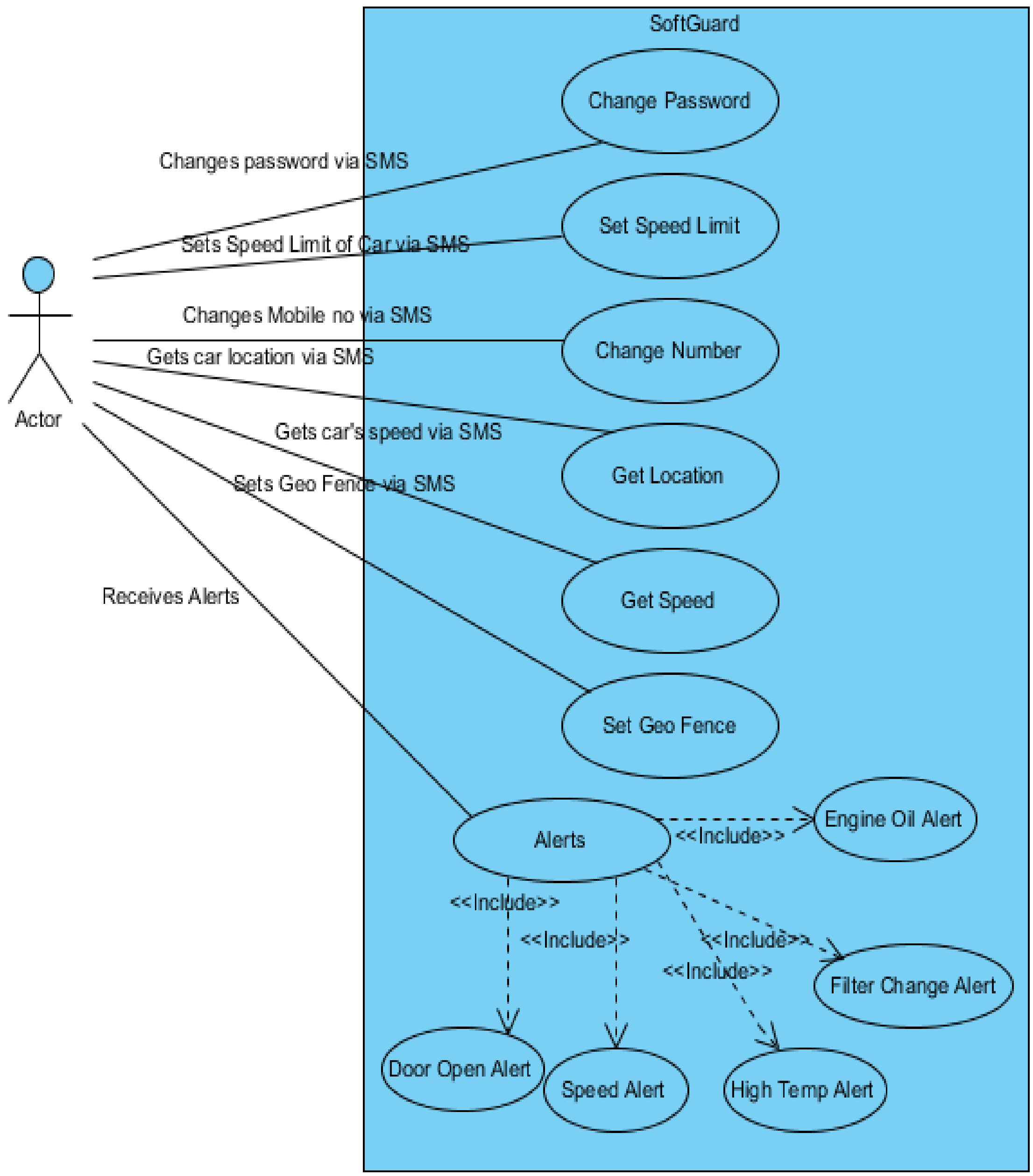


<Oil Filter –State Machine Diagram>



<Engine Oil Level – State Machine Diagram>

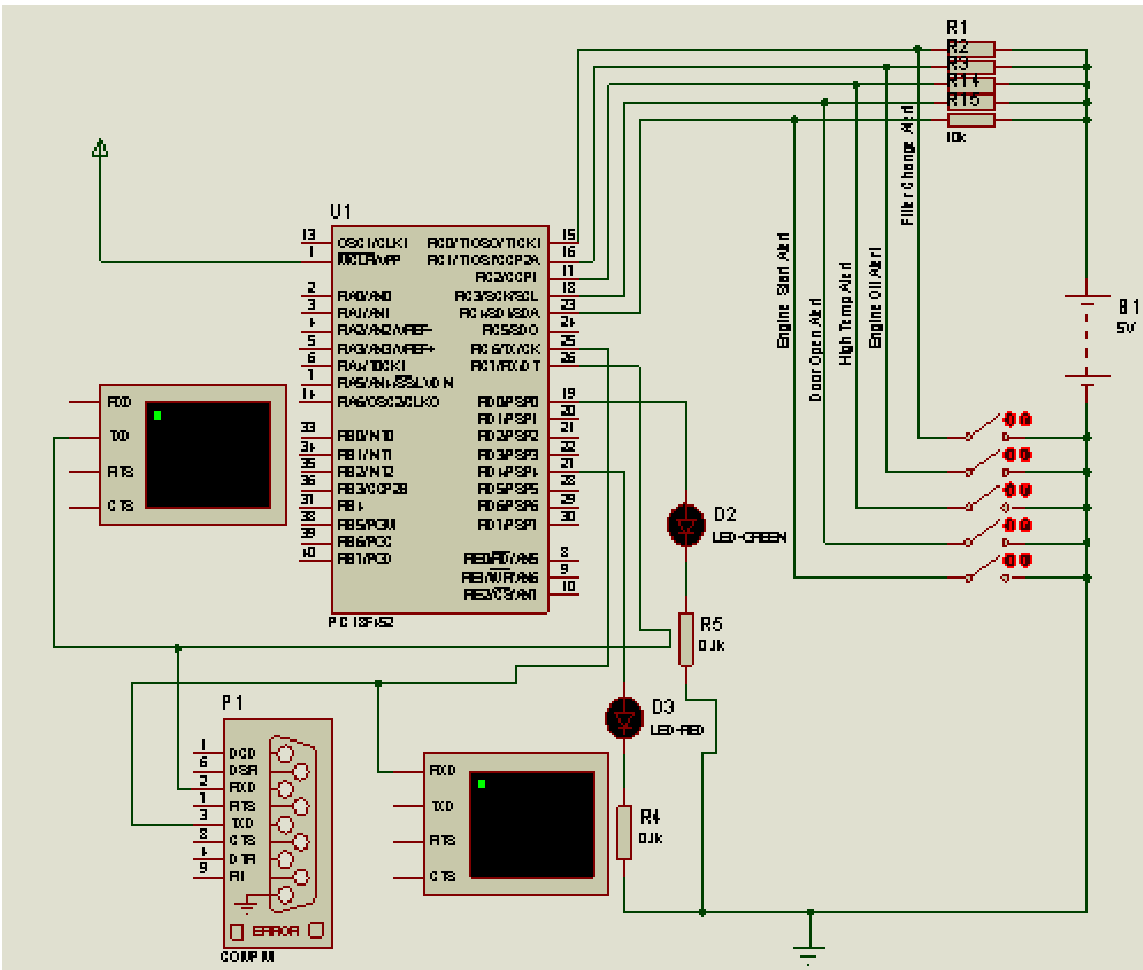
4.8 Use Case Diagram:



4.9 Circuit Diagram

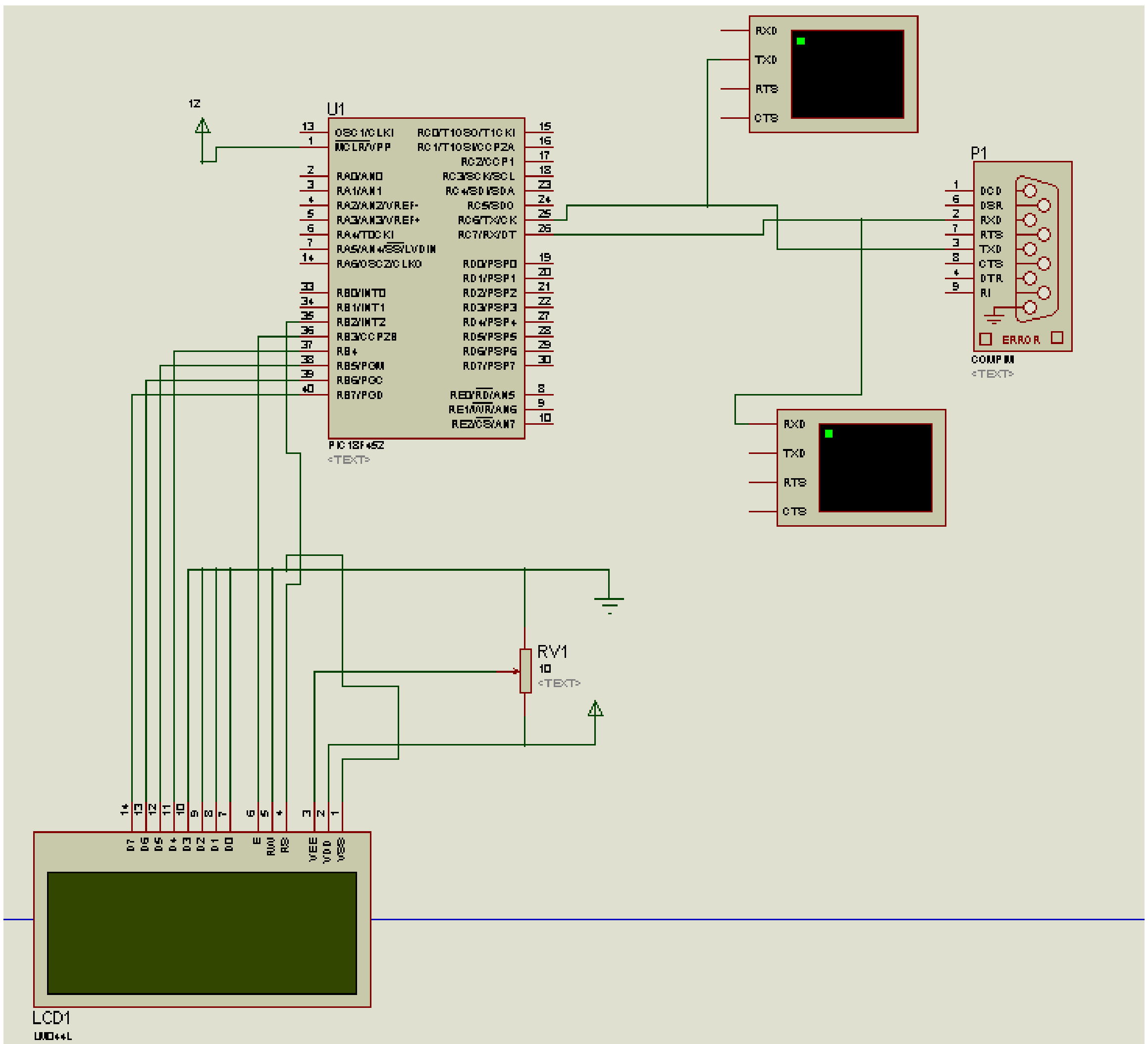
4.9.1 Circuit for Simulating Alerts:

This is the screenshot of the circuit that is used for simulating and testing the code for the alerts that would be generated in the car.



4.9.2 Circuit for Communicating with GPS/GSM and testing other functions:

This is the screenshot of the circuit that is used for simulating and testing the code for the sending/receiving SMS, extracting GPS data, customization functions of tracker and setting limits such as speed and geo-fence limits.



4.10 Conclusion

This chapter presented the architecture of Instant Messenger for Integrated Messaging System (IMIMS). It has incorporated the high level design, low level design, data flow diagram and class diagram of the system. Four main modules have been identified which are instant messaging, offline messaging, voice chat and file transfer.

Chapter 5

DEVELOPMENT

5.1 Introduction

This chapter presents the implementation details of the project. The coding is done in MikroC, the compiler used is MikroC PRO for PIC. The simulation software used for simulating the system is Proteus. We have used structured programming for developing the embedded system. The implementation chapter describes how the system functionalities are instilled in the system.

5.2 Implementation Language

The implementation language which has been chosen for the project is mikroC. It has been preferred over other languages for PIC such as PIC Basic, Assembly language and AVR because of the reasons that MikroC reduces the amount of code required to build applications.It has similar syntax as C++ provides familiarity and hence easy to use. It also provides efficient memory management. The MikroC PRO for PIC compiler provides intuitive tools for debugging the code.

5.3 Tool

We have used ISIS Proteus for simulating our software. Proteus is a microcontroller simulation software which provides you with loads of electronics components for simulation through which you can test your code written in any high level or assembly language (for the microcontroller). It avoids the hassle of testing the code over and over on the actual hardware and helps you fix the bugs easily.

5.4 Hardware

The hardware that we've used for this system is a microcontroller i.e. PIC18f452, a GSM modem, a GPS receiver, an LCD and control board on which the microcontroller is mounted and which connects all other components with the microcontroller. The communication between the controller and other devices is serial and therefore serial cables are used for communication purpose.

Functions Prototypes

```
void return_location(unsigned char*);
unsigned char verify_number(unsigned char*);
unsigned char verify_password(unsigned char*);
unsigned char interpret_command(unsigned char*);
void cut_engine();
void extract_number(unsigned char* );
void change_password(unsigned char*)      ;
void change_number(unsigned char*)      ;
void change_geofence(unsigned char* )     ;
void change_speed(unsigned char*);
void change_mode();
void write_password(unsigned char*);
void write_number(unsigned char*);
void write_speed(unsigned char* );
void get_speed();
void revoke_engine();
unsigned char compare_speed(unsigned char*,unsigned char);
void write_geofence(unsigned char* , unsigned char);
void send_sms(unsigned char* );
void doorOpenAlert();
void engineStartAlert();
void temperatureAlert();
void filterChangeAlert();
void engineOilAlert();
void resetAtShutDown();
unsigned char compare_location(unsigned char * , unsigned char*);
void StrConstCpy(unsigned char*, const unsigned char*);
```

The function prototypes of Soft Guard are shown above. Some of the important functions are described below along with their screenshots.

Interrupt Service Routine

```
void interrupt(){
       char i = 0;
       unsigned timeOut = 0;

         if (PIR1.RCIF )
         {
         LED6=1;
         while (timeOut < 30000)
               {
              if (UART1_Data_Ready()){
                 Rx_buff[i] = UART1_Read();
                 i++;
                 Rx_buff[i] = 0; // Add NULL
                 timeOut = 0;    // reset timeout
              }
              timeOut++;
              }
         New_Flag = 1;
         }
    }
```

The interrupt vector services the peripheral interrupt which occurs whenever there is a data_ready signal on Uart which is connected to GSM Modem via serial cable. It allows the SMS to be read and interpreted as soon as it reaches the SIM in GSM Modem. The ISR checks for peripheral interrupt as soon as it occurs the data from GSM modem is read through UART and stored in Rx_buff array and New_Flag is set to 1 to indicate that new message has been read.

Engine Start Alert

```
void engineStartAlert()
{
if(engineState==0 && PORTD.f2==0 && revoked==1)   //button pressed
    {
    LED1=1;
    engineState=1;
      if(mode==1)
      {
      alert=1;
      engineState=1;
      StrConstCpy(Global, engine_start);
      send_sms(global);  //waits for user response.
      }
    }
else if(PORTD.f2==1 && engineState==1)    //button released
    {
    engineState=0;
    LED1=0;
    }
}
```

This function checks for the change in car engine state. Whenever engine is started in high alert mode it notifies the owner and asks for necessary action. If the owner does not reply in 5 minutes the engine is cut using the relay which will be connected with car ignition system. The cut engine mechanism is implemented in cut_engine function. When PORTD.f2 goes low and the engine state is 0 and revoked is 1 (meaning ignition is enabled) the engine state is set to 1 indicating that the car is started and if the tracker is in high security mode the user is notified and alert is set to 1. When alert is set to 1 the counter is started in the main as shown in the snap below. If the user doesn’t reply in five minutes the ignition is disabled by calling the cut_engine function.

```
while(1)
{

if (alert==1) {
        count++;
        LCD_Chr(2,1,'c');
        }
        if(count>=50000 && alert==1)
        {
        LCD_Chr(2,1,'d');
        cut_engine();
        count=0;
        LED3=~LED3;
        Buzzer=~Buzzer;
        }
```

Chapter 6

TESTING & EVALUATION

6.1 Introduction

Testing is a very important phase in the software development process. Once the coding process is completed, then the software goes under the testing process which involves checking the codes for errors and bugs. It involves any activity aimed at evaluating n attribute or capability of a program or a system and determining that it meets it required results*. This chapter involves all the testing techniques which have been employed in the project and conclusions which have been deduced on the basis of the results of the testing procedures. Test cases for different units and components have been drafted illustrating their expected behaviors on the success and failure of each test. The output of each test is then compared with the one documented in the test case to make sure that the system behaves in the same way in which it is meant to behave.

6.2 Testing Process

The testing process has been carried out throughout the development process as an iterative approach has been used in the project for development. Each phase of development was visited several times making sure that the testing process goes in parallel with the development process. The testing was basically done at three levels, Unit testing, Integration and System testing.

6.2.1 Unit Testing

Unit testing has been done to determine that whether the individual units of the source program work in the same way in which they are expected to work. The units in the project include those methods which cannot be tested by simple inspection and those which blocks cannot be broken down into smaller units for testing. The identified units of the project along with the corresponding test cases are illustrated under the following headings. Following are the unit testing results:

S#	Unit	Expected functionality	Result	Further Testing Possible?
1	GSM Message sender	Sending of SMS from tracker to the owner	Successful	
2	GSM Message receiver	Tracker receives the SMS from the owner	Successful	
3	Message Authentication	Verifying that the message sent to tracker is from the original owner.	Successful	
4.	Message Interpretation	To parse the command sent by the owner of the vehicle.	Successful	
5.	GPS Data Parser	Parsing the data received from GPS system and get location etc	Successful	Yes
6.	GPS Data Comparer	Checking that vehicle is in the geo fence and speed limit.	Successful	
7.	SensorState Detector	Detecting the state of vehicle sensor for anomaly	Successful	
8.	Alerts Generator	In case of any violation the sensors should create an SMS	Successful	
9.	Ignition System Enable/Disable	To verify that Ignition system can be enabled and disabled via SMS by owner	Successful	

6.2.2 Component Testing

Different units together form a component. After unit testing, the components have been tested to make sure that they behave in the expected way. The test cases for different components of the system are elucidated and shown under the following headings. Following are the test cases for component testing:

S#	Component	Expected functionality	Result	Further Testing Possible?
1	GSM Controller	The message is sent, received, authenticated and interpreted.	Successful	Yes
2	GPS Controller	GPS data is parsed to extract location and speed of vehicle and necessary actions taken if speed or geo-fence limits crossed.	Successful	
3	Alert Generator	In case of any anomalous sensors state an alert should be sent via SMS.	Successful	
4.	Ignition Controller	To verify that Ignition system can be enabled and disabled by switching the relay state.	Successful	

6.2.3 Integration Testing

Integration testing means testing the functionality of the system stepwise while integration the components or modules. While amalgamating the components, tests are carried out each time the components are integrated. If the tests are successful, then further integration of the system takes place. Otherwise the components are debugged and integrated again and again until the tests are successful.

S#	Modules	Expected functionality	Result	Further Testing possible
1.	GSM Controller and Alert Generator	SMS are received and sent along with alerts notification.	Successful	Yes
2.	GSM Controller, Alert Generator and Ignition Controller	Controlling (enable/disable) the ignition of the vehicle.	Successful	
3.	GSM Controller, Alert Generator, Ignition Controller and GPS Controller.	Getting the location and speed data along with the control of ignition of the vehicle	Successful	

6.2.4 White Box Testing

White box testing or structural testing uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The white box testing of the system has been done at both unit testing and component testing stages.

6.2.5 Black Box Testing

Black Box Testing is testing without knowledge of the internal workings of the item being tested. It attempts to derive sets of inputs that will fully exercise all the functional requirements of a system. For each set of inputs, outputs are known and in black box testing, the inputs are fed in and if the output matches the predicted output it means that the system delivers the expected functionality.

6.2.6 Static Analysis of Code

Besides testing the code dynamically, static analysis of the code has been done as well to find defects, if any, in the blocks of code due to which it does not implement the exact requirement or to determine the ways by which the code can be optimized to make it full proof

The code has been statically analyzed in many ways which are briefly illustrated under following headings.

6.2.6.1 Control Flow Analysis

Control flow analysis has been carried out for the verification and validation of control blocks in the source code, for instance, the 'for', 'while' loops and the 'if' condition blocks. It has been observed that no unnecessary code has been included and all these blocks are optimized.

6.2.6.2 Data Analysis

Data analysis has been done to find and remove improper initializations, unnecessary assignments and those variables that are declared but never used. All such unnecessary lines have been eliminated thus giving a refine code.

6.3 Conclusion

This chapter illustrated the testing process of the system that has been carried out and the corresponding results obtained. The testing of a system has been done in complete detail. The test cases have been returned for the three main phases of the testing, unit testing, and component testing integration testing. Using these test cases, results of the test cases have been authenticated. Both white box and black box testing have been carried out to determine that whether the system delivers all the functional requirements that it should be delivering. Even static inspection of the code has been carried out as well so that it becomes optimized and does not become redundant. All the test results were very successful proving that system delivers all its functionalities in an efficient way.

Chapter 7

FUTURE WORK & CONCLUSION

This documentation describes the coming into being of the Soft Guard vehicle tracking system from its analysis phase to the implementation phase. It gives the detailed requirement specification of our system; how the system was designed to incorporate those requirements and how the design was mapped in the implementation phase of the project. The document also shows the testing that was carried out over the system to ensure the removal of the bugs from the system. Although we have covered all the requirements of the project, some extensions can be made in the future. The future work that can be carried out on the system is described below.

PCB Making And Deployment - The system can be made further efficient by doing more work on it. First of all a hardware module has to be designed and developed so as to reduce the size of the system so that it can be deployed in the car. A PCB will be made on which GPS and GSM will be integrated on it and the whole system will be interfaced with the system of a car with the help of an expert.

Android Software for Mapping - The system can be made more versatile by interfacing it with android software, like one in which there is a graphical map and user can see the movements of his car on it. Also it can have a Geo fencing function. Users will simply mark the boundaries of the allowed geo Fence on the map and system will raise alarm whenever the vehicle crosses the boundary. Many such softwares are already available in market and can be interfaced.

Using a More Efficient GSM Module - The GSM module we have used is slow and not very efficient. It takes almost a minute’s time to delete old SMS messages so as to allow next new message to pop up. To eliminate this delay we can change the GSM module with a newer and more efficient one and enjoy a system with no delays.

Using Multiple Microcontrollers - Speed of the system can be greatly improved by using multiple micro controllers (e.g. 1 for GPS processing and 1 for GSM processing).

GPRS Interfacing - The system can be interfaced with GPRS, so that in cases where GPS data is not available, it can take location data from GPRS.

APPENDIX A

<u>User Manual</u>

User Manual

This application software has two categories of users, Administrators and Users. We have discussed how to use the system easily and efficiently. Manual includes the detailed guideline for each function, command and stage of the system.

How to change the Password?

The procedure to be followed is; Send message, “CP,OLD_PASSWORD,NEW_PASSWORD;”. Example is: Cp,pswrd,admin;Password will be changed to admin; a confirmation message will be received.Default password is PSWRD.Password cannot be larger than 5 chars.Cannot contain special characters.Password is case-sensitive.Must end with semi colon.

Change Authoritative Number

The procedure to be followed is; Send message, “CN,PASSWORD,NEW_NUMBER;”. Example is: CN,pswrd,+923028516000; Number must contain international code.Length must be 13 numeric.Must end with semi colon.Only the authorized number holder has an authority to transfer the number.

Change Geo-Fence

The procedure is to be followed is; Send message, “Cg,pswrd,Max_latitude, Min_latitude, Max_longitude, Min_Longitude;”. Example is: Cg,pswrd, 33.786738, 33.488325, 073.196983, 072.866707;Latitude values are 9 chars long, first 2 chars before dot.Longitude values are 10 chars long, first 3 chars before dot.Must end with a semi colon.Default Geo-Fence of Rawalpindi /Islamabad to Rawat.

Change Maximum Speed Allowed

The procedure is to be followed is; Send message, “Cs,pswrd,new speed;”. Example is: Cs,pswrd,179.99;Speed must be 6 chars long, first 3 chars before dot.Max speed is 350 km/hr by default. Minimum is 70 km/hr , Less than 70 km/hr cannot be set. Semi colon is must.

Get Location

The procedure to be followed is; Send message, “Gl,”.A response message will consist of a link to a map that point to current location of the vehicle will be sent.Must end the message with a semi colon.

Get Speed

The procedure to be followed is; Send message, “Gs;”A response message will be sent by system that contains the speed of vehicle.Must end the message with a semi colon.

Cut Engine

The procedure to be followed is; Send message, “EC;”System cuts the engine, and a confirmatory message is sent.Must end the message with a semi colon.

Enable Engine Start

The procedure to be followed is; Send message, “RV;”System revokes the engine, i.e. enable it to start after the engine cut.A confirmatory respond will be send. It is a must end the message with a semi colon.

Negative Response Options

The following procedure is to be followed, when user doesn’t want to Cut the engine in case of crossing of geo-fence, opening of the door or engine is started in High Alert mode; Send message, “NO,”A confirmatory message, “Auto shut down of the vehicle has been cancelled”.Must end the message with a comma.

Mode of Operation

The following procedure is to be followed when the mode of the operation is required to be changed; Send message, “CM,”A confirmatory message regarding change of mode of operation will be send.Must end the message with a comma.

APPENDIX B

Bibliography

BIBLIOGRAPHY

[1] http://en.wikipedia.org/wiki/Global_Positioning_System

[2] www.gps.gov

[3] http://en.wikipedia.org/wiki/GPS_tracking_unit

[4] http://en.wikipedia.org/wiki/GSM

[5] http://www.gsmworld.com/

[6] http://en.wikipedia.org/wiki/Vehicle_tracking_system