

**AJMAN UNIVERSITY**

College of Engineering and Information Technology

Electrical and Computer Engineering Department

COE431

Vehicle Speed and Location Monitoring System

**By**

Abdulrahman Hisham

Niel Carlrich Maritz

Hassan Ali Jarko

**Supervisor**

Dr. Khalid Ammar

***Submitted in Partial Fulfillment of the Requirement for the B.Sc. Degree in Computer Engineering,***2019

# PROJECT DECLARATIONS

1. It is declared that this project is original and is resulted as a group effort of the project members only, unless otherwise specified and in the document and references. Moreover, this work in its entirety is never submitted in any form to quality for another degree or diploma at any university/institution and I/We undertake not to submit the same in future for qualifying and obtaining any other degree from this or any other university/institution.
2. I/We affirm to have applied ethics to the design process and in the selection of the project design. I/We also affirm to have considered the safety of the public to be paramount and have addressed this in the design wherever applicable.

Abdulrahman Hisham

201610191

Niel Carlrich Maritz

201610306

Hassan Ali Jarko

201611039

December 2019

# LIST OF DESIGN CONSIDERATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **The impact of the project on:** | **Aspects** | **Applicability** | **Extent of Applicability** |
| Economy | The project covers feasibility of development, production and operational cost. | Yes No | 1 2 3 4 5 |
| Environment | The project has an effect on environment (negative to positive) | Yes No | 1 2 3 4 5 |
| Ethic | The project does not have any obvious potential conflicts with a professional Code of Ethics arising from the development or use of the product | Yes No | 1 2 3 4 5 |
| Health and Safety | Safety and health of public is being considered while designing and developing the project and it does not violate any existing laws of health and safety | Yes No | 1 2 3 4 5 |
| Society | The project is beneficial for the community and society as a whole. The impact on the society is considered | Yes No | 1 2 3 4 5 |

Abdulrahman Hisham  
20161019

Niel Carlrich Maritz  
201610304

Hassan Ali Jarko  
201611039

December 2019

# CERTIFICATE OF APPROVAL

This is to certify that the project titled “Wireless Accelerator Pedal Control System” accomplished by Abdulrahman Hisham, Niel Carlrich Maritz, Hassan Ali Jarko, Reg. ID. 201610191, 201610306, 201611039, under the supervision of Dr. Khalid Ammar, Ajman University, UAE, is fully acceptable, in quality and scope, as Project1 as required by the study plan of the degree of *B.Sc. in Computer Engineering*.

Project Supervisor:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Khalid Ammar

Assistant Professor

Department of Electrical and Computer Engineering

College of Engineering and Information Technology

Ajman University, UAE

HoD:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Ayman Tawfik

Associate Professor

Department of Electrical and Computer Engineering

College of Engineering and Information Technology

Ajman University, UAE

# ACKNOWLEDGEMENT

*We would like to thank Dr. Khalid Ammar for offering us guidance and support throughout the ideation and execution phases of this project, despite his hectic schedule, he always managed to meet us and discuss both of our accomplishments and shortcomings in the project.*

# ABSTRACT

This paper outlines the details of the continuation of the final project (Project II). The main concept involves controlling and monitoring the speed of cars within a given area, i.e. if a car is on a given road, it will not exceed a given speed x, if the car exceeds the speed x, the driver is promptly alerted of the consequences and possibly penalized. The device works by collecting the speed of the car from 2 sources, both from the GPS module and the OBD II port. The information about the speed is provided by a controlling entity through either the website or the app. The information is retrieved by the device through a GSM module through which general data can be received by the device. The device can also be used to address specific vehicles through their assigned ID, with which the device will alert the driver of any offenses they have committed or potential hazards which they are exposed to. The device communicates with the end user through an LCD display where it displays any warnings.

LIST OF CONTENTS

[PROJECT DECLARATIONS ii](#_Toc28030779)

[LIST OF DESIGN CONSIDERATIONS iii](#_Toc28030780)

[CERTIFICATE OF APPROVAL iv](#_Toc28030781)

[ACKNOWLEDGEMENT v](#_Toc28030782)

[ABSTRACT vi](#_Toc28030783)

[LIST OF CONTENTS vii](#_Toc28030784)

[LIST OF FIGURES viii](#_Toc28030785)

[LIST OF TABLES ix](#_Toc28030786)

[Chapter 1: INTRODUCTION 2](#_Toc28030787)

[1.1 Overview 2](#_Toc28030788)

[1.2 Statement of the Problem 2](#_Toc28030789)

[1.3 Project Specifications 2](#_Toc28030790)

[1.4 Motivation/Justification 2](#_Toc28030791)

[1.5 Project Applications 2](#_Toc28030792)

[1.6 Organization of the Report 3](#_Toc28030793)

[Chapter 2: LITERATURE REVIEW 5](#_Toc28030794)

[2.1 Related Technologies 5](#_Toc28030795)

[2.1.1 OBD II 5](#_Toc28030796)

[2.1.2 ELM327 5](#_Toc28030797)

[2.1.3 Electronic Governor 5](#_Toc28030798)

[2.2 Limitations/Problems in Literature Work 6](#_Toc28030799)

[Chapter 3: HARDWARE AND LOW-LEVEL SOFTWARE IMPLEMENTATION 8](#_Toc28030800)

[3.1 OBD 8](#_Toc28030801)

[3.2 GSM 10](#_Toc28030802)

[3.3 Pedal Control 11](#_Toc28030803)

[3.4 System Integration 15](#_Toc28030804)

[3.5 Block Diagram 16](#_Toc28030805)

[3.6 Flow Diagram 17](#_Toc28030806)

[3.7 Results 18](#_Toc28030807)

[3.7.1 Comparison of Results with Project Specifications 22](#_Toc28030808)

[Chapter 4: HIGH LEVEL SOFTWARE AND DATABASE IMPLEMENTATION 24](#_Toc28030809)

[4 Interaction 24](#_Toc28030810)

[Chapter 5: TOOLS/COMPONENTS 28](#_Toc28030811)

[5.1 Hardware Tools Used 28](#_Toc28030812)

[5.2 Software Tools Used 28](#_Toc28030813)

[Chapter 6: CONCLUSION 30](#_Toc28030814)

[6.1 Conclusion/Summary 30](#_Toc28030815)

[6.2 Future Work 30](#_Toc28030816)

[REFERENCES 31](#_Toc28030817)

# LIST OF FIGURES

[Figure 1: OBD-II Pinout 5](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030715)

[Figure 2: ELM327 connected to OBD port 8](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030716)

[Figure 3: 16-pin OBD port 8](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030717)

[Figure 4: Bluetooth module circuitry 9](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030718)

[Figure 5: Bluetooth Inquiring Process 9](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030719)

[Figure 6: OBD Formatted Response 10](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030720)

[Figure 7: OBD Unformatted Response 10](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030721)

[Figure 8: GSM method looking for the speed limit value in the received messages 11](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030722)

[Figure 9: Old SMS clearing and speed comparison 11](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030723)

[Figure 10: Example throttle valve control system 12](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030724)

[Figure 11: Toyota Sienna, depression angle X APP output voltage 12](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030725)

[Figure 12: Low pass RC filter 13](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030726)

[Figure 13: Voltage output at released position (within speed limit) 13](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030727)

[Figure 14: voltage output at fully depressed position (within speed limit) 13](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030728)

[Figure 15: Voltage output at released position (within the speed limit) 14](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030729)

[Figure 16: Voltage output at fully depressed position (over the speed limit) 14](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030730)

[Figure 17: System block diagram 15](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030731)

[Figure 18: System schematic 16](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030732)

[Figure 19: Block Diagram 16](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030733)

[Figure 20: Flow Diagram 17](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030734)

[Figure 21: Full physical system 18](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030735)

[Figure 22: Speed comparison code 18](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030736)

[Figure 23: Serial monitor output to corresponding speed limit of 2km/h 19](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030737)

[Figure 24: Serial monitor output to corresponding speed limit of 6km/h 19](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030738)

[Figure 25: Serial monitor output of pedal controlling Arduino (under speed limit) 20](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030739)

[Figure 26: Initialization method (setting minimum and maximum values of the pedal AKA. range) 21](#_Toc28030740)

[Figure 27: Serial monitor output of pedal controlling Arduino (over speed limit) 21](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030741)

[Figure 28: Registration/Login Page 24](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030742)

[Figure 29: Stored hashed user identifications 25](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030743)

[Figure 30: Entered credentials must meet standards 25](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030744)

[Figure 31: List of available vehicles for control 25](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030745)

[Figure 32: Vehicle information panel 26](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030746)

[Figure 33: Stored vehicle information 26](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030747)

[Figure 34:Map to visualize vehicle locations 26](https://austae-my.sharepoint.com/personal/201610306_aust_ae/Documents/Project%201%20Report%20Amended.docx#_Toc28030748)

# 

# LIST OF TABLES

[Table 1: APP output voltage value states 13](#_Toc28030764)

**CHAPTER 1**

**INTRODUCTION**

# Chapter 1: INTRODUCTION

## 1.1 Overview

Describe changes & updates.

## 1.2 Statement of the Problem

According to Abu Dhabi Police, the three main causes of road deaths during the third quarter of 2018 were distracted driving, excessive speeding, and lack of courtesy to other road users. That same year the rate of deaths was 3.83 road deaths per 100,000 people, with the goal of reaching 3 road deaths per 100,000 people by 2021[1], our proposed system will help curb one of the main causes of road deaths, excessive speeding.

With our proposed system, authorities will be allowed to control motorists speed wirelessly and enforce a desired speed limit.

## 1.3 Project Specifications

The project is divided into two levels: the user level and the admin level.

On the user side we have a device that will be fitted within their vehicle.  
This device will actively monitor the speed of the car via Satellite and OBD II port. The device will connect to a server through GPRS to collect general information about an area and the speed limit, it will also compare between this speed and the observed speed. If the vehicle is found to be violating the speed limit, the user/driver will be alerted and possibly penalized.

The admin side mostly consists of specifying information for the module to read. There is an app and a website which control the data of either cars in a region or addressing cars when required.

## 1.4 Motivation/Justification

We have pursued the major of engineering to allow us to have the skillset and the deduction abilities to help find problems that we face in our society and daily life, being able to come up with solutions for the betterment of the general public and pushing the boundaries of science and technology to help achieve that, we believe that this project proposal is universal, relatively cheap to produce and distribute commercially, and most importantly, the potential to save human lives, our most valuable assets.

## 1.5 Project Applications

As should be obvious, this project aims to give power to the authorities when controlling traffic. This product would be necessary in all cars by government regulation; a car without this device would not be deemed fit for the road. Since the device works to alert users, it can alert uses of other things, such as dangers on the road, natural disasters and sudden change in the climate.

## 1.6 Organization of the Report

The structure of the report includes 6 chapters talking about distinct parts of the project.

* Chapter 1 is a brief overview.
* Chapter 2 provides the necessary background information to understand the components.
* Chapter 3 explains the hardware implemented and the functions they serve. The third chapter aims to give an overview of the entire system working cohesively. The hardware is divided into two parts, the user section with which the users will be interacting and the controller section which includes all the necessary hardware to control & interface with the system.
* Chapter 4 takes a close look at the software used by the controller hardware to manage and change information.
* Chapter 5 lists all the software and hardware used in the project
* Chapter 6 talks about the conclusions that we drew from this project and contrasts between this project and the last.

**CHAPTER 2**

**LITERATURE REVIEW**

# Chapter 2: LITERATURE REVIEW

## 2.1 Related Technologies

### 2.1.1 OBD II

OBD 2 or On-Board Diagnostics 2 was introduced in 1996 and mandated by the US to be incorporated in all cars sold, it reports specific information about the vehicle like calculated engine load, engine RPM, vehicle speed, throttle position, etc.[4] Additionally the OBD 2 standard specifies the connector standard being used, a 16-pin (2x8) J1962 female connector situated in the car, and can be used by specialized scan tools that utilize that port.

Figure 1[8] shows the pinout of that port. Notably, pin 4 being car chassis ground, 5 as signal ground, and pin 16 being constant 12V battery voltage, with the rest of pins being at the discretion of the vehicle manufacturer or communication lines for the different standards.

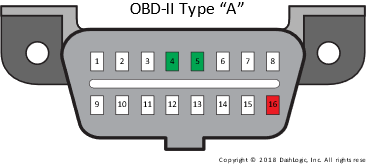


Figure 1: OBD-II Pinout

### 2.1.2 ELM327

The ELM327 Interpreter is developed and produced by ELM Electronics, the ELM327 acts as a bridge between the OBD interface and a standard serial interface be it a wireless serial communication protocol like Bluetooth or a wired serial communication protocol like RS232. The ELM327 can interpret 9 different OBD standards, ELM327 interpreters have been used as diagnostic trouble code readers and as automotive scan tools.[3]

### 2.1.3 Electronic Governor

An electronic governor is essentially the drive-by-wire electronic system found in most if not all modern systems, a servo motor is linked to the throttle valve of the engine, the ECM (Engine Control Module) sends voltage values that are directly proportional to the degree of the throttle valve position, the TPS (Throttle Position Sensor) senses the position of the throttle valve and sends that value back to the ECM to ensure the right mix of air and fuel for the internal combustion engine which is also directly related to rotational speed of the engine.[8]

The Likes of BMW, Mercedes-Benz and Audi have implemented an electronically limited top speed of 155 mph on some of their vehicle offerings.[9]

### 2.1.4 GPS Module

A GPS module was used to be able to track the location of the vehicle as well as determining whether it’s in the geolocation domain set by the authorized personnel or not, the u-blox NEO-6M GPS module, it features an inexpensive ceramic antenna, and a backup battery that can allow the module to sleep instead of a hard shutdown when external power isn’t available, because of the style of the antenna, the GPS signal is a RHCP signal (Right Hand Circular Polarized) as compared to regular antennas (whip antennas) that are used for linear polarized signals.[10]

### 2.1.5 Single Board Computer

A Single Board Computer (SBC) was needed instead of a traditional microcontroller as the device would be expected to do multiple tasks at once, limited of course by the microcontrollers slower clock speed and lack of Operating System. The device which we chose to use was the Raspberry Pi Zero W as it comes equipped with onboard Bluetooth/Wi-Fi along with the necessary pins to communicate with the GSM and GPRS modules.

## 2.2 Limitations/Problems in Literature Work

### Part Documentation

The information about parts online are very specific and finding something that matches the specifications that we need is no problem at all. Acquiring parts on the other hand is often a game of compromise as you must juggle availability, feasibility & cost. We ended up compromising with the MUX by using a 2:4 MUX instead of the desirable 2:2 MUX with a single select pin, this problem was easily solved by connecting one of the select pins to ground.

### The COVID-19 epidemic

The elephant in the room is of course the epidemic happening during this time, while we did not want the epidemic to affect our work on the project, our order for parts was cancelled, face-to-face interaction with teammates was not possible and testing was also beyond reach. Because of this, we decided to complete each section of the project to the point where they could all interface together to work as a system once the opportunity arises.

**CHAPTER 3**

**HARDWARE AND LOW-LEVEL SOFTWARE IMPLEMENTATION**

# Chapter 3: HARDWARE AND LOW-LEVEL SOFTWARE IMPLEMENTATION

## OBD

A picture containing indoor, wall

Description automatically generatedA picture containing indoor, sitting, top, sewing machine

Description automatically generatedWe are interested in retrieving the vehicle speed data, our first step then is to find a way to read vehicle computer to retrieve the desired data, we used an ELM327 Interpreter to do just that by connecting it to the 16-pin OBD port in the vehicle.

Figure 2: ELM327 connected to OBD port

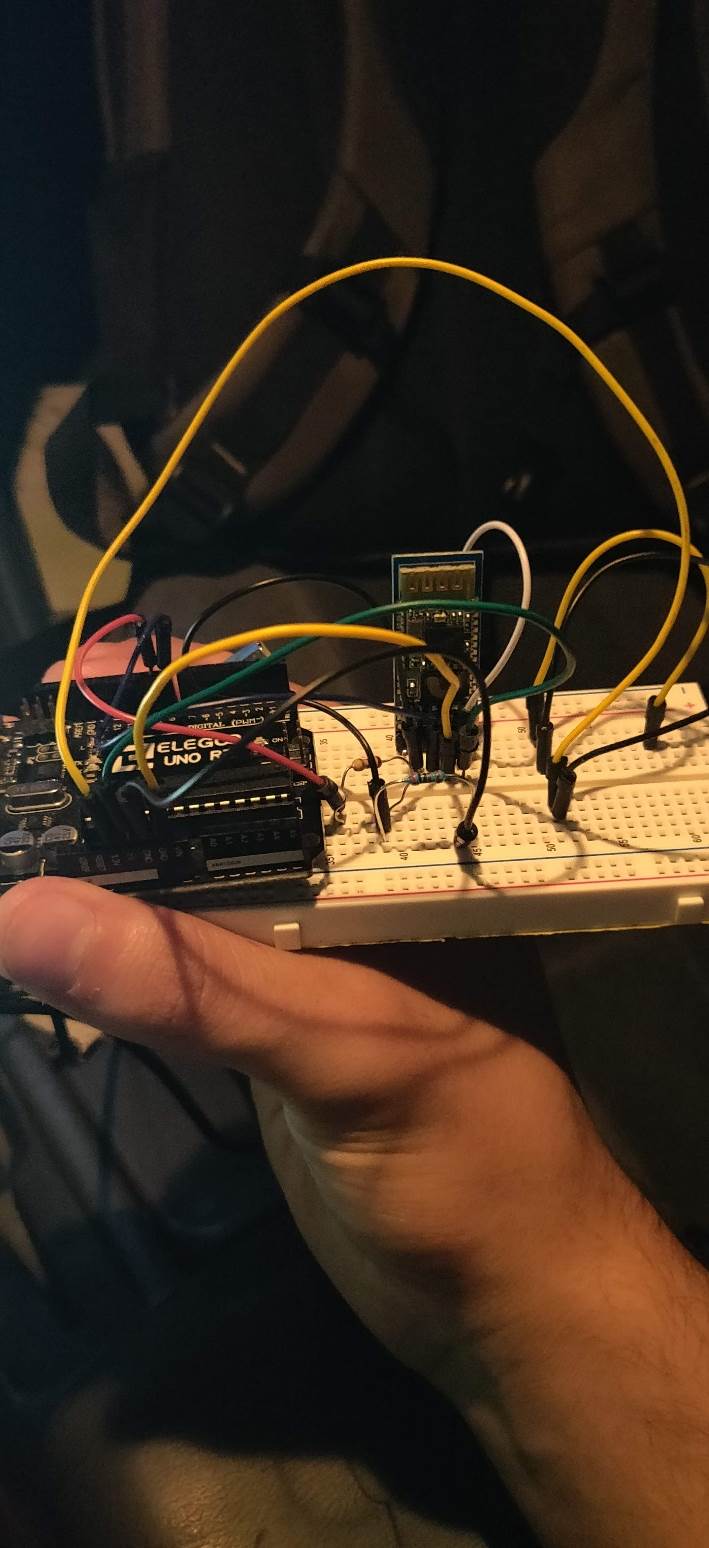
Figure 3: 16-pin OBD port

The next step is to connect our Bluetooth module (HC-05) to the ELM327 interpreter to be able to send it specific commands and retrieve our speed data.

We set up our HC-05 as a master Bluetooth device and the ELM327 as a slave Bluetooth device, after than we had to know the MAC address of our ELM327 interpreter so that our module can pair to it. Our specific interpreter has the address 1D:A5:68:98:8A:1F.

To achieve a successful connection, we had to set up some AT commands for our Bluetooth module [2]

1. **AT+ROLE=1 (to set the module as master)**
2. **AT+CMODE=0 (to connect to one specific address)**
3. **AT+BIND=1DA5,68,988A1F (binding our module to this address)**
4. **AT+INIT (start the inquiring process)**
5. **AT+PAIR=1DA5,68,988A1F,20 (start pairing with that address, a timeout of 20 seconds)**
6. **AT+LINK=1DA5,68,988A1F**

A screen shot of a computer

Description automatically generated

Figure 4: Bluetooth module circuitry

Figure 5: Bluetooth Inquiring Process

After successfully connecting to the ELM327 interpreter we will need to send specific commands to retrieve the speed data[3][4], in our case we have to write a script that continuously transmits 0100 (PIDs supported from 1 to 20) then 010D (vehicle speed), first we have to understand the commands sent.

**01 00**

Underlined in blue is what’s called a service, according to the SAE J1979 standard, 01 is to show current data

Underlined in red is 00, a PID (parameter ID) that returns 4 bytes of data describing the list of parameters that the vehicle supports (from 01 to 20 in hex).

**01 0D**

Underlined in blue as we said earlier is to show current data.

Underlined in red is 0D, a PID that returns 1 byte of data describing the current speed the sensors of the vehicle are reading.

Now the response of the ELM327 interpreter is of 3 bytes

**41 0D 3A**

Underlined in blue is 41, that is the success code that the interpreter returns.

Underlined in red is 0D, the command we sent to the interpreter.

Underlined in Purple is the speed in hex (3A-58Km/H for this example).

Our script will do some string formatting to change that response into a readable human value.

A close up of text on a white background

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Figure 6: OBD Formatted Response

Figure 7: OBD Unformatted Response

## GSM (SMS)

That covers reading the speed value from the car’s computer, now moving to communication. We wanted our system to have a very long range and coverage in case the vehicle needs controlling is in a remote location, luckily in the UAE, cell coverage is very high, for that reason we used a GSM module, more specifically the SIM900A GSM module.

Our script will look for a special string to in the text messages received to differentiate between it and other messages that might be sent to the module, in our case we choose SPDD followed by the speed data, it can be anything we decide on, it’s a way for the system to not look for any integer value and compare it to the ELM327 value, a way to ensure that only the value we set as the speed limit is being compared with the ELM327 value.

Moreover, we also must send some AT commands to the GSM module [7]:

1. **AT+CMGF=1 (will set the module to SMS mode)**
2. **AT+CMGD=? (deletes SMS messages, we** **use it at initialization to ensure that any new incoming messages would be received and not discarded due to insufficient storage)**

A close up of text on a white background

Description automatically generatedA picture containing building

Description automatically generated

Figure 8: GSM method looking for the speed limit value in the received messages

Figure 9: Old SMS clearing and speed comparison

## GSM (GPRS)

The GSM module will use GPRS to connect to the internet using PPP to tunnel the IP to the Raspberry Pi. Once the connection is established and the GSM module is acting as a modem, the raspberry pi can connect to the desired server to retrieve the data such as speed and location.

For general communication, GPRS will be more sustainable as it can communicate 2 ways. The cost of communicating through GPRS is also generally lower than communicating via SMS.

## System Integration

Now that we have a complete picture of each of the parts, let us take a step back and look at an abstract view of the system.

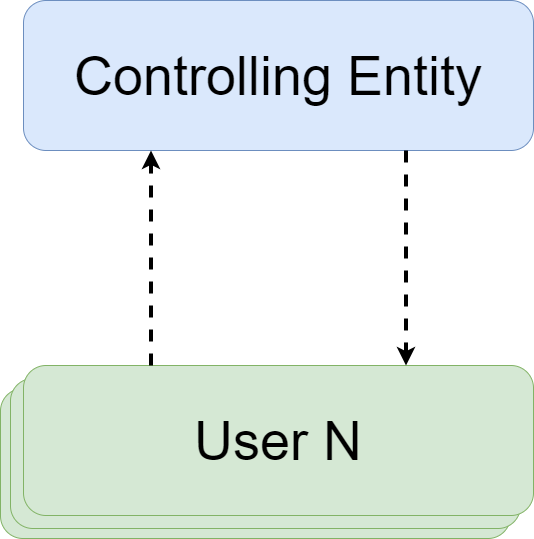


Figure 10: An abstract view of the system and how the 2 parts interact.

The Controlling Entity sends data such as speed and location to the User and in some situations the user returns data if required to. Naturally, the system is designed to communicate with any number of Users but for the purposes of explanation we will only be showing one.

Now that the picture of the system has been established, a closer look into the Controlling Entity can show us how it will be set up. The controlling entity will have a server with which the user will communicate, this server will be connected to two databases, one for the website and one for the app, the data between the two will be kept consistent. The user, in the most probably case an authority figure, will interface with either systems to send out data to the users.



Figure 11: A in-depth view at the controlling entity.

Next, we can look closely at the system on the user end. As we can see the User end consists of 3 parts, the vehicle, the user and the device which will be doing all the work. The vehicle can communicate with the device through the OBDII interface and the user, in this case the driver, will be given information through the device.

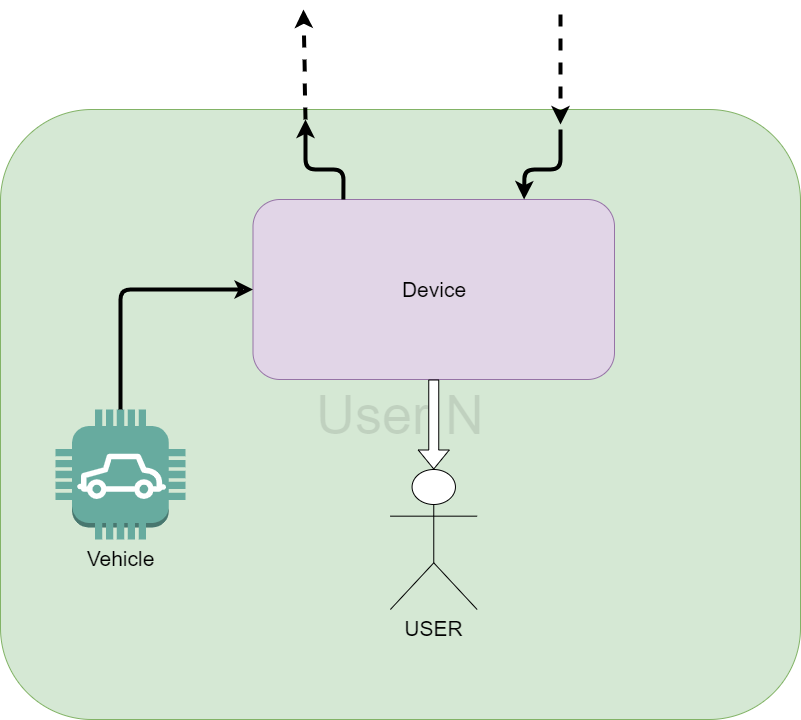
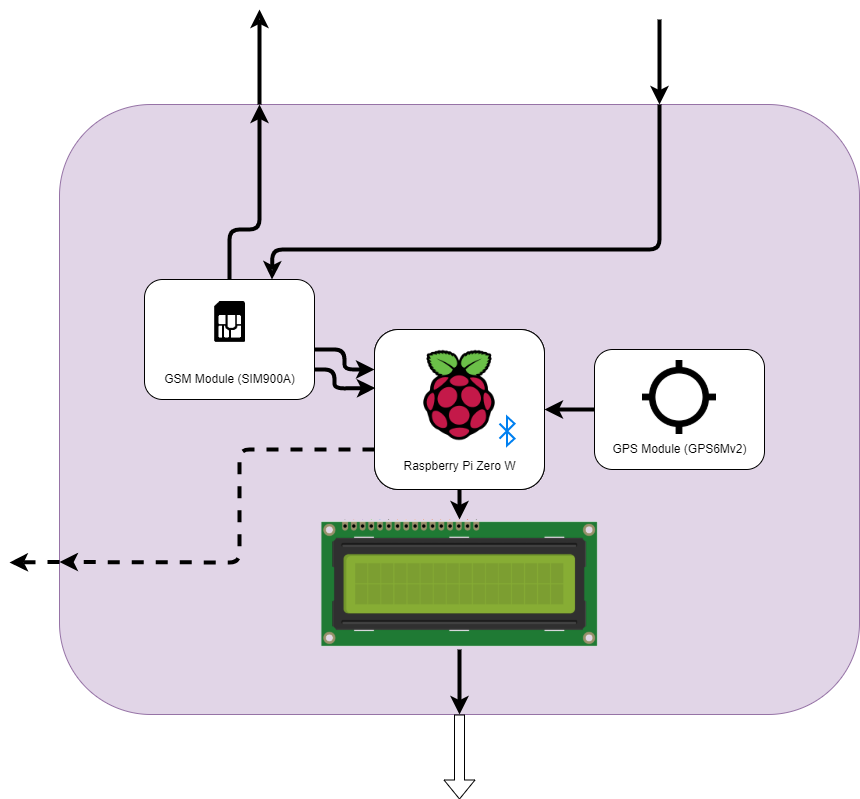


Figure 12: An in-depth view at the user’s section of the system

Finally, we look at the hardware that will be housed inside the device. As we can see, we are using a raspberry pi zero w to interface with a GSM module and a GPS module. The GSM module will be connecting to the server using GPRS and acting as a modem for the raspberry pi to connect to the Internet. The Warnings to the user will be displayed on the LCD display. The built in Bluetooth on the raspberry pi will be used to communicate with the OBDII interpreter.



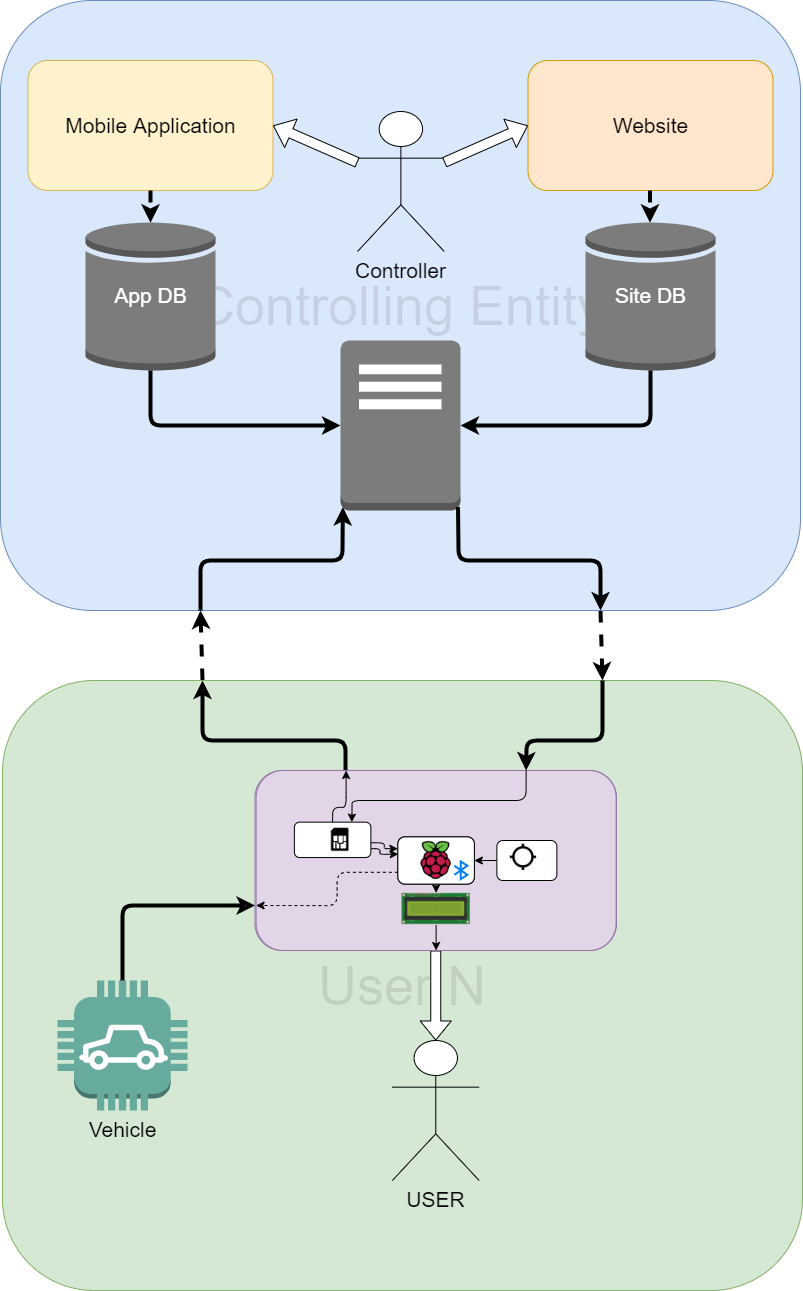


Figure 13: A complete model of the system

## Data Flow Diagram

The below figure shows how the device uses the data that it has access to. The device runs two sections in parallel and

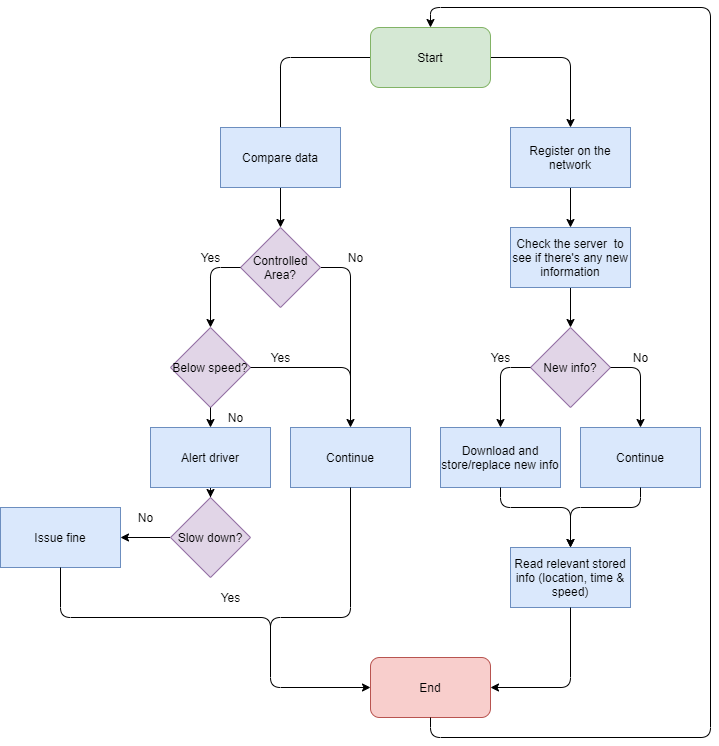


Figure 14: A flow diagram that shows the operations of the device

## Connection Diagram

As we can see below, there consists 4 main components with a minor component being a buzzer to alert the user. As we can see there is a mux connected to the Pi ZW, the GSM module & the GPS module, this is because the Pi ZW only has 2 serial ports, one of which is being used for the Bluetooth module, to circumvent any issues, the serial signal from the pi is multiplexed between the GSM module and the GPS module.

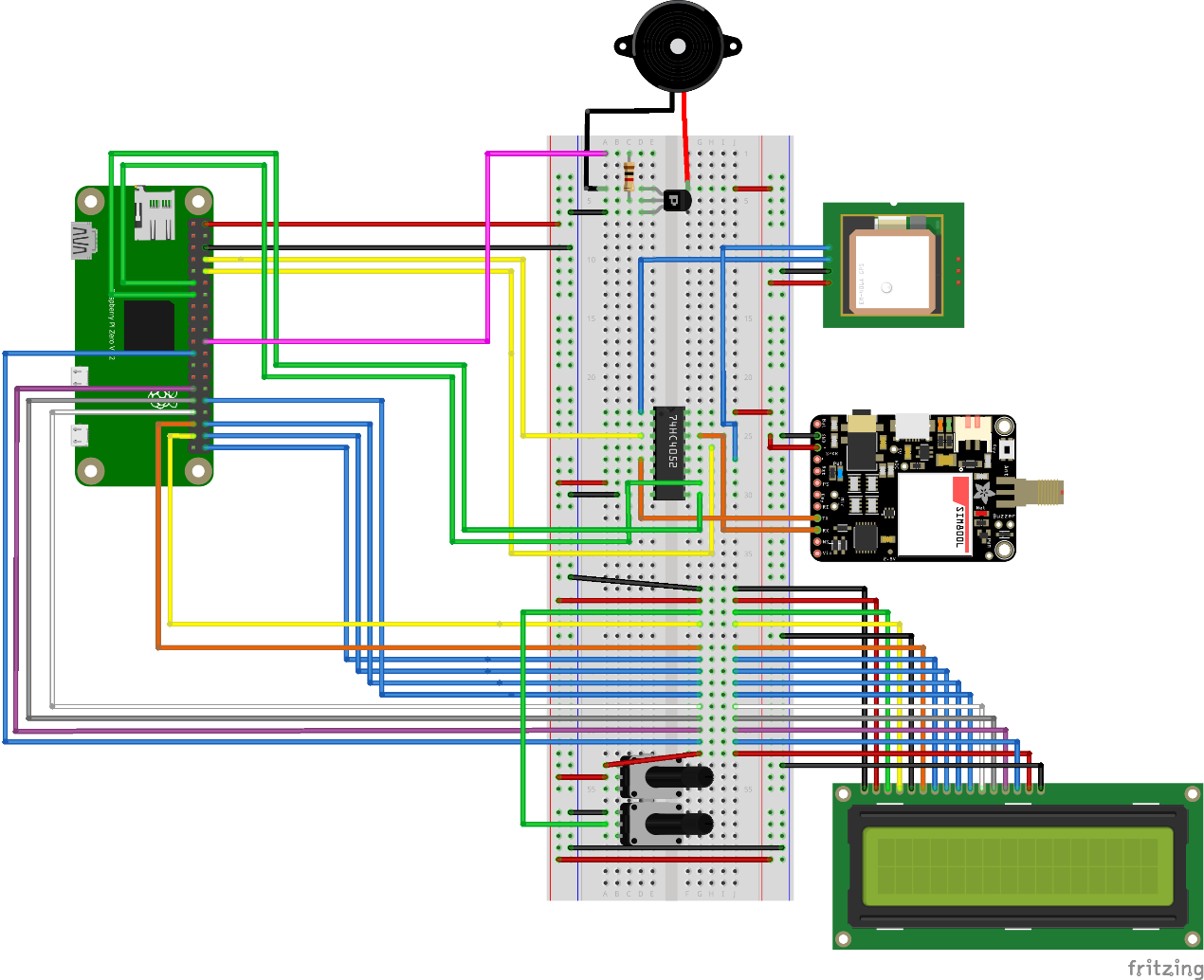


Figure 15: The main components of the Device as they would be wired up.

**CHAPTER 4**

**HIGH LEVEL SOFTWARE AND DATABASE IMPLEMENTATION**

# Chapter 4: HIGH LEVEL SOFTWARE AND DATABASE IMPLEMENTATION

## 4 Interaction

An Android application has been created to provide an intuitive user interface for the authorized personnel to interact with, whilst allowing us to facilitate an environment for us to enable new higher-level features (The ability to add new vehicles, send specific speed and GPS coordinates values to the cars, etc.). The Application was built using Android Studio, written with the Java programming language.

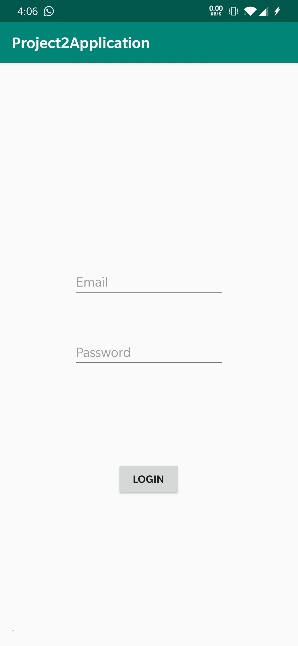
To ensure that security is provided with our system as we don’t want any unauthorized persons to be able to send commands to our alerting system, an authentication procedure had to take place.

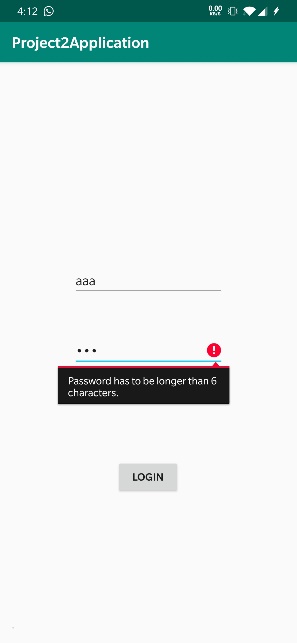
Figure 28 shows the authentication page of our application, users must have an account registered in our database to be able to view and access the list of cars that have the list of the alerting systems installed in.

Furthermore, we must guarantee that our entered credentials meet to some length and complexity criteria to help against security attacks, which is shown in figure 30.

Figure 16: Login Page

Users that have signed into our system can login at any time, their usernames and passwords are hashed and stored in the database, increasing the level of security in the case of an attack that targets the database

Figure 29 shows the email addresses for the signed-up users with their corresponding hashed IDs in the database structure.



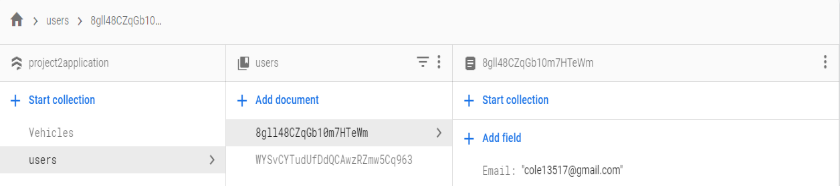


Figure 17: Stored hashed user identifications

Figure 18: Entered credentials must meet standards

Upon login, authorized personnel will be able to see the cars they can alert as well as the ability to add new cars to the database, figure 31 shows these available options, figure 32 shows the fields required to be provided by the authorized personnel for a vehicle to be added and stored in the database.

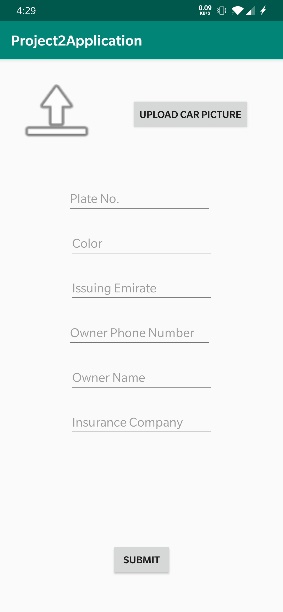


Figure 19 : Add New Car Page

Upon clicking on the view list of cars button, it would navigate to a page where the user would be able to see some information about the car as well as a picture of that vehicle, clicking on a vehicle from the provided list will give us a clearer page with the information of that specific vehicle, as well as the ability to specify a geolocation domain for that vehicle

Figure 31: Given options to authorized personnel

Just like our user/password information being stored in the database, vehicle specific information coupled with the two values (latitude and longitude) that allow us to retrieve the location of the vehicle, are stored in the database (figure 33).

 To make it easier for users to visualize the location of these vehicles, a map was implemented in the application for that reason, the latitude and longitude value for every vehicle would allow us to pin a marker on the map as well as a geolocation domain specified on the map (figure 34).

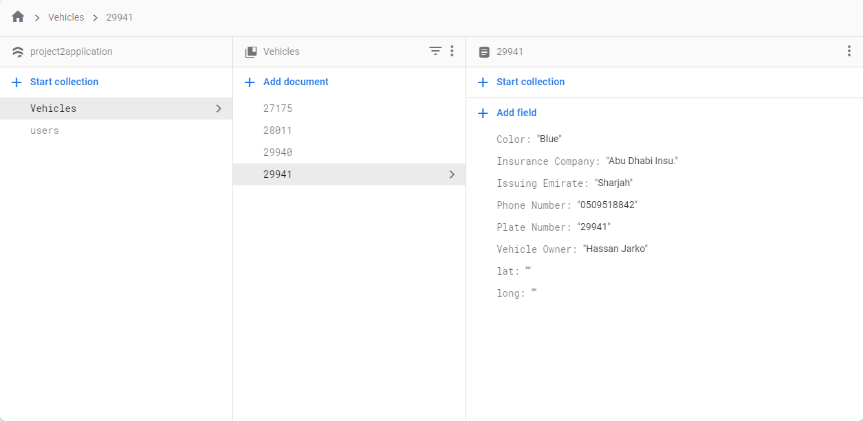


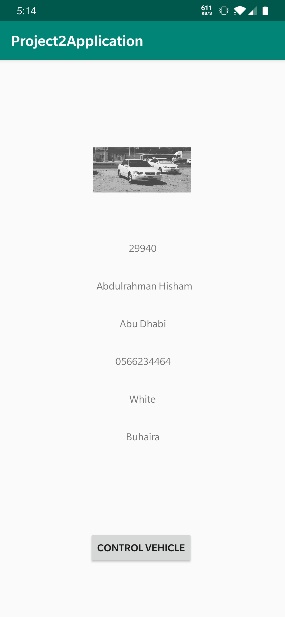
Figure 34:Map to visualize vehicle locations

Figure 33: Stored vehicle information

Our database implementation was done using Google’s Firebase API, for its seamless integration with the android development environment.

The following screenshots are of different parts of the application, figure 35 shows the list of stored vehicles that the authorized personnel can see in the application, clicking on a specific vehicle would focus on the information of that vehicle and provide the option to control the vehicle (figure 36).

As we click on the control vehicle button we are given 8 fields where we can enter the longitude and latitude values for all 4 markers that make up the geolocation domain, figure 37 showcases an example of that with figure 38 showing the coordinates that made up the geolocation domain shown in figure 34.

A screenshot of a cell phone

Description automatically generated

Figure 36: Specific vehicle page

Figure 35: List of available vehicles

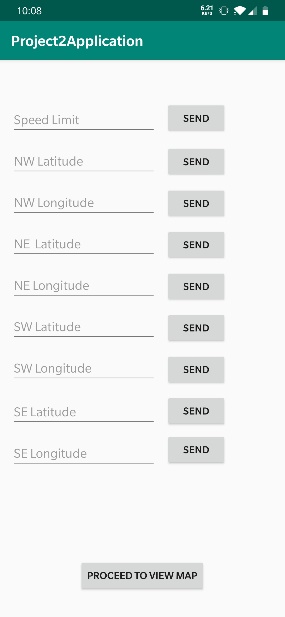
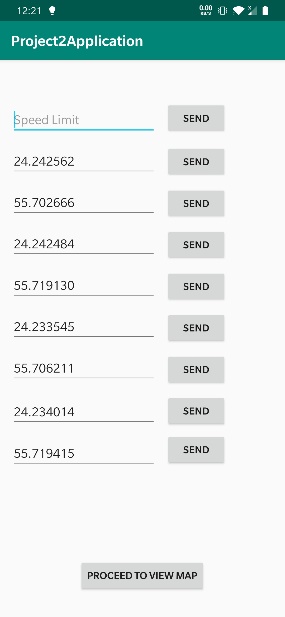
When it comes to software representation, there are many approaches, and various software architectures to choose from.

Figure 38: geolocation domain values (example)

Figure 37: geolocation domain values

Taking in consideration all the required operations the software must perform, using the MVC (Model View Controller) was the best approach for the following reasons, figure 39 shows the software architecture of the website developed.

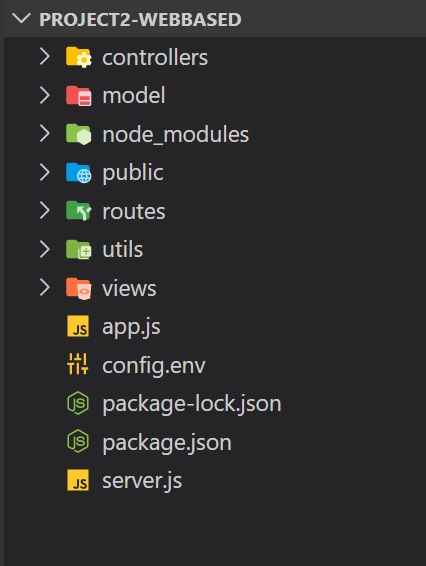


Figure 39: Project Architecture

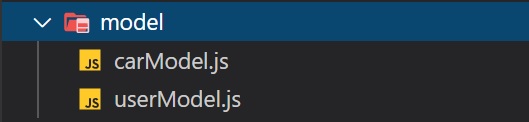


Figure 40: Models

Figure 40 describes the models used for the database design, MongoDB was used to create the models for Cars and Users.

Features we needed includes:

1. The data model available within MongoDB allowed us to represent hierarchical relationships, to store arrays, and other more complex structures more easily.
2. MongoDB uses JSON: This is known as[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)Object Notation (compatible with out code base)
3. Scalability
4. can perform Ad hoc queries: searching by field, range queries, and regular expression searches. Queries can be made to return specific fields within documents.

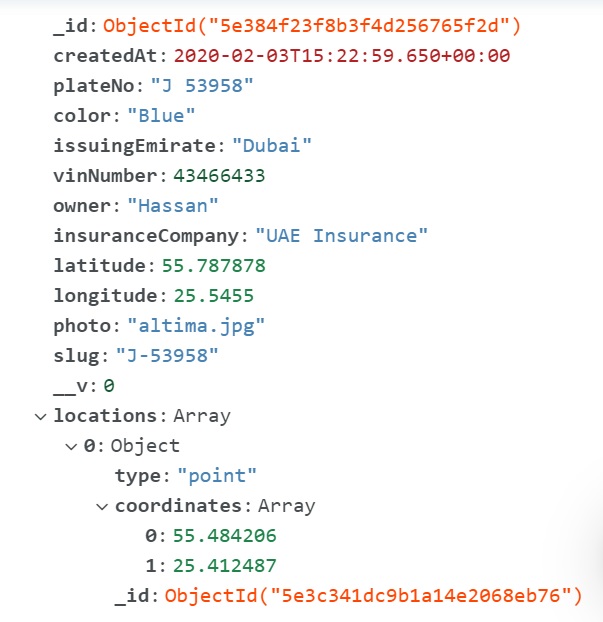


Figure 41: The Car Document Model



Figure 42 The User Document Model

Figures 41 and 42 showcase two examples of the JSON queries that MongoDB records, figure 41 is an example of a car document JSON entry, figure 42 is an example of a user document JSON entry.

**The Controller:** offers Routing Options

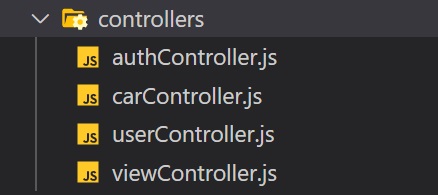


Figure 43: Controller Design

Figure 43 shows the controller design used in the website development process.

Implemented using Node JS and Express JS framework.

Express JS was used to create different routes to navigate through the web application.

Features we needed includes:

**1-**Routing allows a web application to render web page states through their URLs

**2-**Express.js offers a more [efficient routing](https://expressjs.com/en/guide/routing.html) mechanism that can handle highly dynamic URLs.

**3-**the idea of Middleware:

A middleware is a piece of code that has access to a user’s request, the application’s response, and the next middleware to be used

it becomes easy for Express JS to add, remove, or update various features to and from the application.

**4-**templating Engines: Express.js provides a [templating engine](https://expressjs.com/en/guide/using-template-engines.html) that allows web pages to have dynamic content.

**5-**uses async await concept (non-blocking code) so it can handle enormous number of user requests without having a noticeable effect on performance.

**The View:**

Used PUG as our Template Engine.

Features we needed includes:

1-Can inject JavaScript code into html (JSX) JavaScript XML.

2- Pug has powerful features like conditions, loops, includes, mixins using which we can render HTML code based on user input or reference data.

Figure 44 shows the view folder of our MVC architecture with all the different kinds of templates.

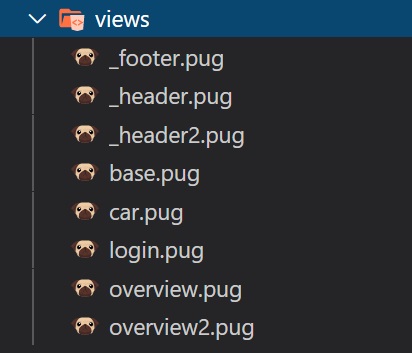


Figure 44: The View Folder

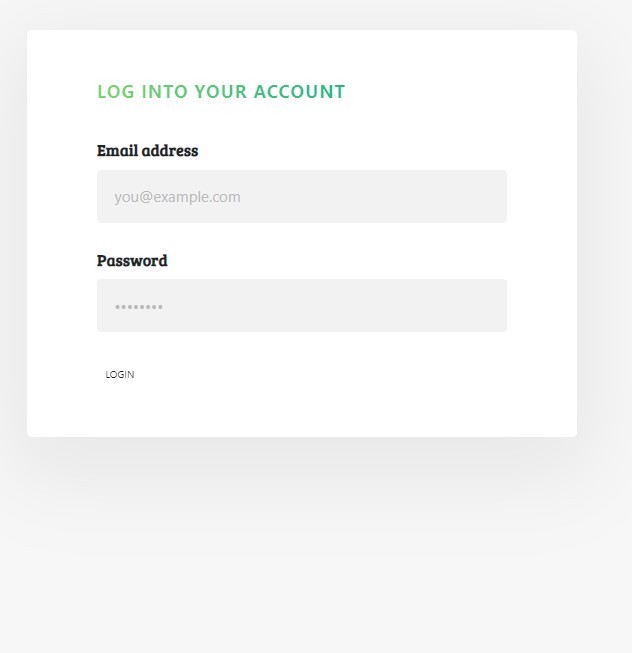
**Features of Our Web Application:**

Figure 45: Log In Screen

Log in and sign up with the ability to assign roles to users:

Figure 45 shows the login page of our website.

Figure 46 shows the user model schema that is behind this visual interface.

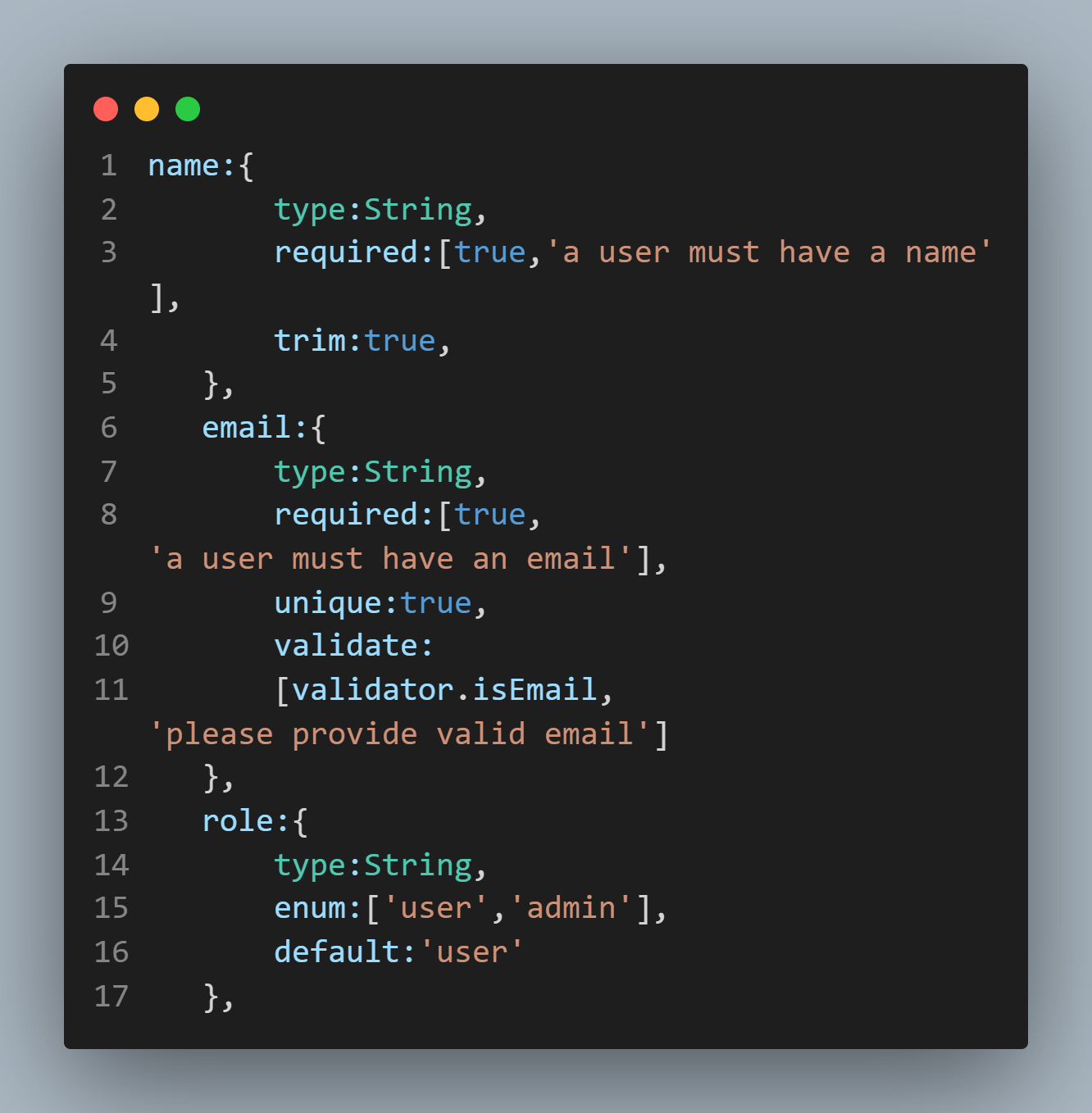


Figure 46: The User Model Schema

Upon logging in the authorized personnel is met with a Cars Dash Board with the ability to filter and search based on certain criteria as shown in figure 47:

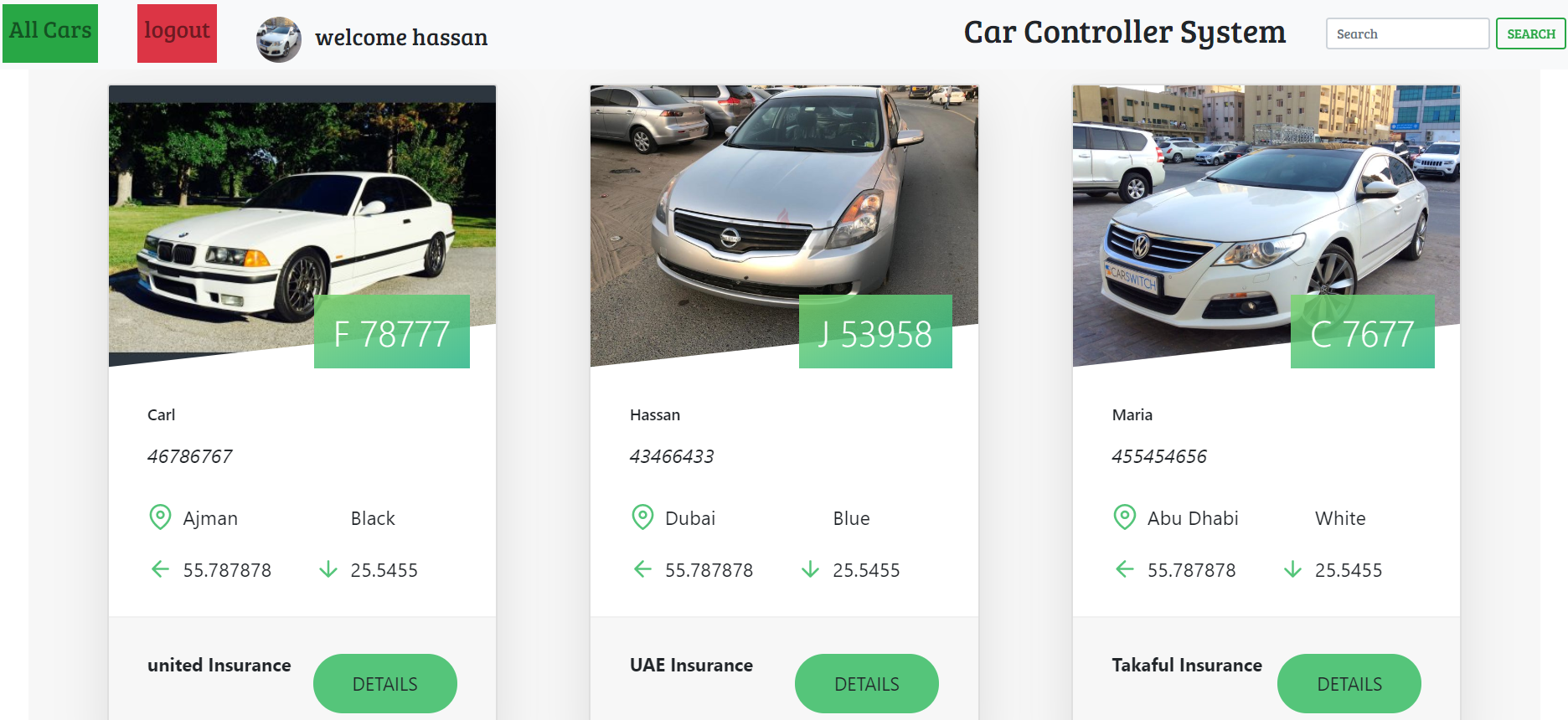


Figure 47: Main Dash Board

Note: to be able to view the whole photo above, please switch your word document into web view

Communication to the specified cars is done through sending SMS using pre implemented message-bird API which will interact directly with the GSM modules inside the systems installed in the cars, figure 48 shows the use of that API.



Figure 48: Message Bird API example

Clicking on a specific vehicle will lead us to a detailed screen with information about the car as shown in figure 49, the ability to send speed limit values as shown in figure 50, geolocation domain coordinates to that car as shown in. figure 51

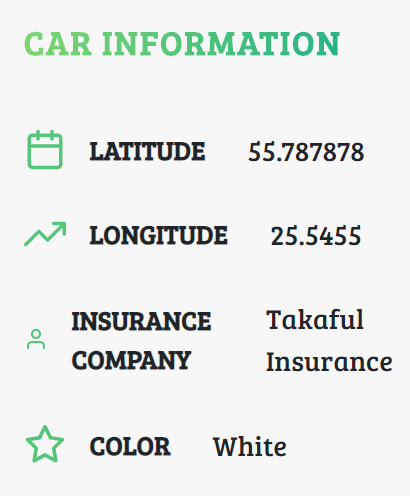


Figure 49: Detailed car controls

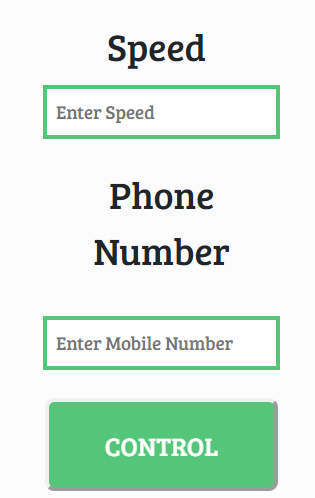


Figure 50 Detailed car Control

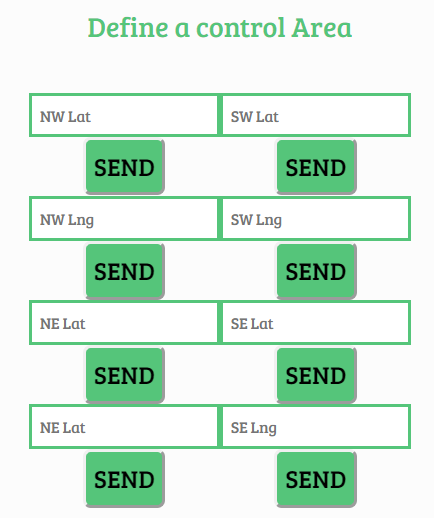


Figure 51 Defined Geolocations

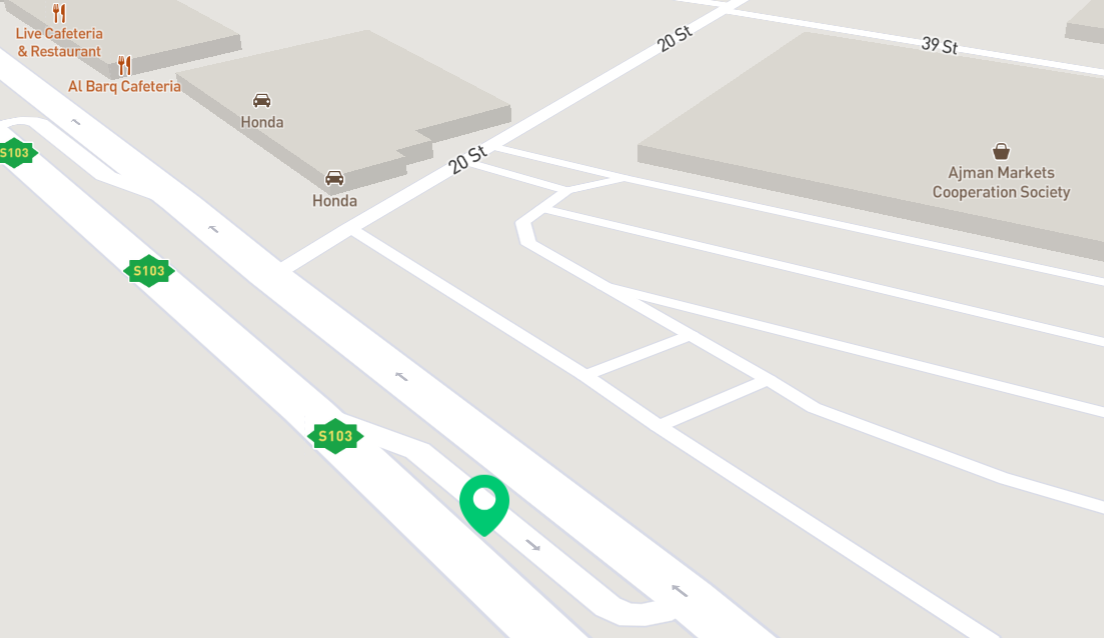


Figure 50: Car live Location on Map

Figure 50 shows the live map that would be displayed with the location of the vehicle,

an outline of the geolocation domain would be shown too, the location of the car is sent from the GPS module to the SMS API periodically.

Figure 51 shows some functions used in the application to enhance the security and protect against common security attacks:

**1-**protect against SQL Injection Attacks.

**2-**protect against XSS.

**3-**Data Sanitization against parameter pollution.

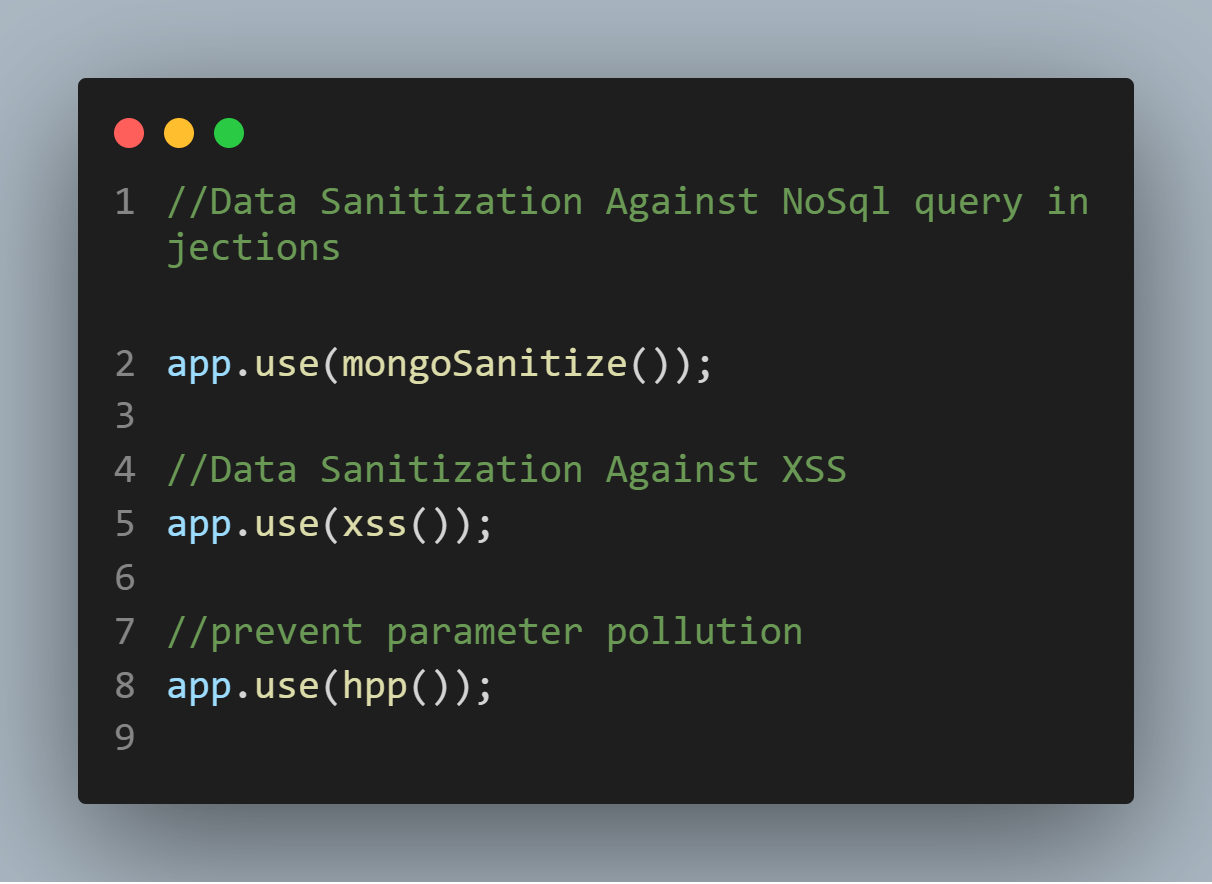


Figure 51 Data Sanitization

**CHAPTER 5**

**TOOLS/COMPONENTS**

# Chapter 5: TOOLS/COMPONENTS

## 5.1 Hardware Tools Used

* Pi Zero W
* Arduino Mega 2560
* Bluetooth Transmitter/Receiver HC-05 Module
* GSM 900A GPRS (General Packet Radio Service) Module
* GPS6MV2
* ELM327 OBD Bluetooth Interpreter
* Resistors
* PNP Transistor
* Buzzer

## 5.2 Software Tools Used

* Microsoft Visual Studio
* Arduino IDE
* Python (Programming Language)
* Google Firebase API
* Android Studio
* PUG
* NPM
* Node.js
* Postman

**CHAPTER 6**

**CONCLUSION**

# Chapter 6: CONCLUSION

## 6.1 Conclusion/Summary

After finishing the first project and controlling the pedal position, the system was still not able to slow down drivers as the lowered pedal position would simply prevent an increase acceleration. The second project aims to improve on the first project by implementing a passive system in the car that would warn users of any violations they are committing and if they do not cease, punish them accordingly. This system would be controlled by the authorities giving them control over cars. The cars would still be individually addressable via SMS which means that the authorities can still inform or warn any vehicle on the road. We see this system as a far more realistic option as it would be more effective than speed cameras and would work in any vehicle.

The second project also saw the implementations of most the goals from the first as the device can now communicate with a server for any application necessary.

# REFERENCES

[1] Serkal, M. A. (2019, September 25). Revealed: 3 main causes for road deaths in UAE. Retrieved from <https://gulfnews.com/uae/transport/revealed-3-main-causes-for-road-deaths-in-uae-1.1569392463691>.

[2] HC-03/05 Embedded Bluetooth Serial Communication Module AT command set . (2011). Retrieved 1, 2019, from <http://www.linotux.ch/arduino/HC-0305_serial_module_AT_commamd_set_201104_revised.pdf>.

[3] ELM Electronics . (n.d.). ELM327 OBD to RS232 INTERPRETER DATASHEET . Retrieved December 1, 2019, from <https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf>.

[4] Vehicle E E System Diagnostic Standards Committee. (2017). SAE J1979: E/E Diagnostic Test Modes. SAE International. doi: 10.4271/J1979\_201702

[5] Shen, T., Zhang, J., & Jiao, X. (2014, November). An Adaptive Servo Control Strategy for Automotive Electronic Throttle and Experimental Validation. Retrieved December 1, 2019, from <https://www.researchgate.net/figure/Schematic-of-the-electronic-throttle-control-system_fig1_262938429>.

[6] Toyota Sienna Service Manual: Throttle / Pedal Position Sensor/ Switch. (n.d.). Retrieved December 1, 2019, from <http://www.tsienna.net/throttle_pedal_position_sensor_switch-2255.html>.

[7] SimCom. (2014, October 20). Sim900 AT Commands Manual\_V1.11. Retrieved December 1, 2019, from [https://simcom.ee/documents/SIM900/SIM900\_AT Command Manual\_V1.11.pdf](https://simcom.ee/documents/SIM900/SIM900_AT%20Command%20Manual_V1.11.pdf).

[8] How does a small engine governor work? (n.d.). Retrieved December 10, 2019, from <https://www.briggsandstratton.com/na/en_us/support/faqs/browse/governor-system.html>.

[9] van Gorp, Anke. "Ethical Issues in Engineering Design; Safety and Sustainability" page 16. Published by 3TU Ethics, 2005. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [9090199071](https://en.wikipedia.org/wiki/Special:BookSources/9090199071), 9789090199078 . ISSN 1574-941X

[10] NEO-6 u-blox GPS Modules data sheet. (n.d.). Retrieved from <https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf>