



USMAN INSTITUTE OF TECHNOLOGY

Affiliated with NED University of Engineering & Technology, Karachi

Department of Computer Science

B.S. Computer Science / Software Engineering

FINAL YEAR PROJECT REPORT

Batch-2019

VEHICLE DETECTION SYSTEM

By

ADNAN SAMAD 19B-011-SE

M.OMER SIDDIQUI 19B-004-SE

ALISHA HAMEED 19B-017-SE

HAFIZ TAHA SHARIF 19B-035-SE

Supervised by

DR SHARAF HUSSAIN

ST-13, Block 7, Gulshan-e-Iqbal, Abul Hasan Ispahani Road, Opposite Safari Park, P.O. Box 75300, Karachi, Pakistan. Phone: 34978274-5; 34994305; 34982476; <http://www.uit.edu>

Declaration

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at USMAN INSTITUTE OF TECHNOLOGY or other institutions.

ADNAN SAMAD (19B-011-SE)

Signature: _____

M.OMER SIDDIQUI (19B-004-SE)

Signature: _____

ALISHA HAMEED (19b-017-SE)

Signature: _____

HAFIZ TAHA SHAREEF (19B-035-SE)

Signature: _____

Date: _____

Acknowledgments

Abstract

Table of Contents

Declaration.....	2
Acknowledgments	3
Abstract	4
CHAPTER 1.....	11
INTRODUCTION.....	11
1.1 INTRODUCTION	12
1.2 PROBLEM STATEMENT.....	13
1.3 PROPOSED SOLUTION.....	13
1.4 PROJECT SCOPE	14
1.4.1 IN SCOPE:.....	14
1.4.2 OUT OF SCOPE:.....	14
1.5 MOTIVATION.....	14
1.6 SYSTEM DIAGRAM	15
1.7 BUSINESS VALUE OR BENEFITS OF PROJECT	15
1.8 CONCLUSION OR FUTURE SCOPE.....	16
CHAPTER 2.....	17
BACKGROUND AND LITERATURE REVIEW.....	17
2.1 BACKGROUND AND LITERATURE REVIEW	18
2.2 SIMILAR APPLICATIONS.....	18
2.3 RFID SYSTEM (FLEET MANAGEMENT SYSTEM) PRODUCT LINKS AND COMMON COMPARISON .	18
2.4 STUDY FOR RFID IN CONTAINER AREA	19
2.5 ARCHITECTURE DESIGN OF THE VEHICLE TRACKING SYSTEM BASED ON RFID.....	19
2.6 RFID TRACKING SYSTEM FOR VEHICLES (RTSV).....	21
2.7 RFID-BASED VEHICLE MONITORING SYSTEM.....	22
2.8 RFID INTRODUCTION	23
2.9 ANTI-THEFT SYSTEM	23
2.10 ANTI-THEFT SYSTEM 2.....	23
CHAPTER 3.....	25
AIM AND STATEMENT OF THE PROBLEM.....	25
3.1 EXPLAINATION OF PROJECT	26
3.2 DEFINATION OF THE PROBLEM.....	26
3.3 LIMITATION OF SCOPE	26

3.4 SOLUTION OF THE PROBLEM	26
3.5 LACK OF TIME	27
3.6 SCOPE CREEP	27
CHAPTER 4.....	28
HARDWARE, SOFTWARE ANALYSIS & REQUIREMENTS	28
4.1 FACT FINDING.....	29
4.1.1 BENEFITS.....	29
4.1.2 DRAWBACKS.....	30
4.2 HARDWARE	30
4.2.1 RFID	30
4.2.1.1 HOW RFID WORKS	31
4.2.1.2 ACTIVE RFID	31
4.2.1.3 PASSIVE RFID.....	31
4.2.3 TAG	33
4.2.4 ESP32.....	34
4.2.5 GPS MODULE NEO6M V2 U BLOX.....	35
4.3 HARDWARE COST	35
4.3.2 WORKING ASSUMPTION.....	35
4.3.3 RFID VS AI (ARTIFICIAL INTELLIGENCE).....	36
4.4 UML DIAGRAMS	37
4.4.1 OPERATIONAL DIAGRAM	37
4.4.2 ACTIVITY DIAGRAM	38
4.4.3 USECASE DIAGRAM.....	41
CHAPTER 5.....	42
SOFTWARE DESIG AND MODELING	42
5.1 ARCHITECTURE	43
5.2 DIAGRAMS	44
5.2.1 USECASE DIAGRAM.....	44
5.2.2 OBJECT DIAGRAM	46
5.2.3 ACTIVITY DIAGRAM	47
5.2.4 CLASS DIAGRAM.....	56
5.3 NON-WORKING PROTOTYPING	58
CHAPTER 6.....	64

ALGORITHM ANALYSIS AND COMPLEXITY	64
6.1 ALGORITHM	65
6.2 PSEUDOCODE	65
CHAPTER 7.....	66
IMPLEMENTATION.....	66
7.1 INTRODUCTION	67
7.2 CODING	67
7.2.1 HTML'S	67
7.2.2 MODELS	82
7.2.3 FORMS.....	83
7.2.4 APP	84
7.3 OPERATIONAL DIAGRAM	89
7.4 COMPONENT DIAAGRAM	89
7.5 DEPLOYMENT DIARGAM.....	90
CHAPTER 8.....	91
TESTING.....	91
8.1 INTRODUCTION.....	92
8.2 WHITE BOX TESTING	92
8.1.1 TEST CASE (USER LOGIN)	92
8.2.3 TEST CASES (SEARCH VEHICLE OWNER USING CNIC)	95
8.2.4 TEST CASES (SEARCH VEHICLE LOCATION USING MAP)	96
8.2.5 TEST CASES (VIEW ALL VEHICLES LIST)	97
8.2.6 TEST CASES (REGISTER A VEHICLE).....	98
8.2.7 TEST CASES (ADD NEW USER).....	99
8.3 BLACK BOX TESTING	100
8.3.1 TEST CASES	101
CHAPTER 9.....	105
CONCLUSION	105
9.1 INTRODUCTION	106
9.2 CONCLUSION	106
CHAPTER 10.....	107
FUTURE WORK	107
10.1 INTRODUCTION	108

10.2 FUTURE ENHANCEMENT	108
CHAPTER 11	109
ACHIEVEMENT.....	109
11.1 INTRODUCTION	110
11.2 ACTIVITIES	110
11.2.1 PROJECT PLANNING.....	110
11.2.2 SYSTEM ANALYSIS AND DESIGN.....	110
11.2.3 SYSTEM IMPLEMENTATION	110
11.2.4 SYSTEM TESTING AND VALIDATION	110
11.2.5 DOCUMENTATION	110
11.2.6 PRESENTATION AND DEMONSTRATION	110
11.3 IMPROVEMENT.....	111
11.3.1 TECHNICAL SKILLS.....	111
11.3.2 PROBLEM-SOLVING ABILITIES.....	111
11.3.3 INDUSTRY-RELEVANT EXPERIENCE	111
11.3.4 TEAMWORK AND COMMUNICATION	111
11.3.5 ACADEMIC RECOGNITION.....	111
APPENDIX A	112
APPENDIX B	113
APPENDIX C	113
APPENDIX D	114
REFERENCES.....	115

TABLE OF FIGURE

Figure 1 Working system diagram for the project	15
Figure 2 Basic Components of an RFID System	19
Figure 3 Vehicle system layer	20
Figure 4 Google maps.....	20
Figure 5 How the system works.....	21
Figure 6 Component Diagram	22
Figure 7 Anti thief Diagram.....	24
Figure 8 RFID	31
Figure 9 Passive RFID.....	32
Figure 10 MFRC522 RC522 RFID READER	33
Figure 11 RFID Tag	34
Figure 12 ESP32.....	34
Figure 13 GPS MODULE NEO6M V2 U BLOX	35
Figure 14 Operational Diagram	37
Figure 15 Activity Diagram 4.1	38
Figure 16 Activity Diagram 4.2	39
Figure 17 Activity Diagram 4.3	39
Figure 18 Activity Diagram 4.4	40
Figure 19 Usecase Diagram 4.1.....	41
Figure 20 MVC	44
Figure 21 Usecase Diagram 5.1.....	44
Figure 22 Object Diagram	46
Figure 23 Activity Diagram 5.1	47
Figure 24 Activity Diagram 5.2	48
Figure 25 Activity Diagram 5.3	49
Figure 26 Activity Diagram 5.4	50
Figure 27 Sequence Diagram 5.1	51
Figure 28 Sequence Diagram 5.2	52
Figure 29 Sequence Diagram 5.3.....	53
Figure 30 Sequence Diagram 5.4	54
Figure 31 Sequence Diagram 5.5	55
Figure 32 Class Diagram.....	56
Figure 33 ER Diagram	57
Figure 34 OPERATIONAL DIAGRAM	89
Figure 35 COMPONENT DIAGRAM.....	90
Figure 36 DEPLOYMENT DIAGRAM	90

TABLE LIST

Table 1 Cost estimation	36
-------------------------------	----

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

As Technology is being used in our everyday life to overcome the problems. We use different things together to create inventions. There are a lot of technologies to control over crimes some are perfect and cheap solution but some are very expensive solution. A car detection system is a technology used to identify and track vehicles. It is typically used in traffic monitoring and control, as well as in automotive safety. Car detection systems can be used to monitor speed, identify vehicles that are in violation of traffic regulations, and track stolen vehicles. RFID technology is used for various purposes. For Example

- Asset management and inventory
- Patient identification
- Medical device tracking
- Control of infectious diseases
- Vehicle Detection system

RFID

RFID is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is an object that can be applied to or incorporated into a product, animal, or person for the purpose of identification using radio waves. There are several types of RFID tags, but the most common is called passive RFID tags. Passive RFID tags have no internal battery, so power is supplied by the reader . They are much cheaper than active RFID tags and are the preferred choice for many applications. RFID is used in many applications including toll plaza, parking, access control, supply chain management, livestock tracking and many other tracking and identification applications.

RFID is a technology that uses radio waves to identify objects. RFID In car detection system is a system that uses RFID technology to detect the presence of cars. RFID In car detection system can be used to detect the presence of cars in a parking lot, garage, or other area. RFID In car detection system can also be used to track the movements of cars.

RFID works in car detection system by sending a radio signal to interrogate an electronic tag attached to a vehicle. The tag replies with a serial number, identifying the vehicle. The tag contains electronically stored information that can be read from a distance.

RFID can be used in car detection system in various ways. For example, it can be used to track the location of vehicles in a parking lot. It can also be used to identify vehicles that are entering or exiting a parking lot. Additionally, RFID can be used to track the movements of vehicles in a traffic congestion area.

1.2 PROBLEM STATEMENT

RFID car detection system is to design a system which can detect the presence of a car using RFID tags. The system should be able to distinguish between different cars and should be able to provide information about the car such as the make, model and year. The system will also be able to provide information about the location of the car.

RFID tag which can be attached to a car. The tag should be able to withstand the harsh environment and should be able to communicate with the reader even when the car is moving.

A reader which can be installed to reader the RFID tags should be able to read the tags of the cars and should be able to provide information about the car to the system.

A system which can process the information received from the reader and can provide information about the car to the user.

1.3 PROPOSED SOLUTION

In this project we will install the RFID reader on different instances which will be able to read the RFID tags install in cars. Every RFID tags will have their unique number once this RFID tag will be passes from any of the RFID reader the scanner will record the time, location and RFID Tag number and will send it to the database at the backhand. So that if the concern data will be required it can be access easily.

1. The car detection system should be able to detect cars in real-time.
2. The system should be able to identify the make, model, and color of the car.
3. The system should be able to track the car's movements.

4. The system should be able to automatically record the license plate number of the car.
5. The system should be able to generate a report.
6. Implement an RFID tag system on all vehicles.
7. Use RFID reader to scan for tags.
8. Record the tag number and time of entry for each vehicle.

1.4 PROJECT SCOPE

The RFID scanner can scan the number and match the information of the owner and match this to the excise database. Moreover, the location of the vehicle will be sent to the control center, which will be helpful to identify the actual location in the city.

The scope of this project includes and excludes the following items.

1.4.1 IN SCOPE:

- To secure the vehicles in a way that if it is stolen, it will be detected through scanner installed on different location with the help of RFID tags and map it on Google Map with the last scanned location.
- To provide the relevant information to the excise department of vehicle and taxation so that they could know the location of their vehicle.
- To provide all the information of the vehicles to excise department so that no-one could be able to run the vehicle without registering & without paying tax of the vehicle.

1.4.2 OUT OF SCOPE:

- Hand-held scanning devices for concern officers.
- Tax can be paid.
- Concern officers can add Tax challan of the vehicle.

1.5 MOTIVATION

The RFID car detection system is designed to automatically detect the presence of vehicles in a given area. The system can be used to monitor the flow of traffic in a parking lot or garage, or to

track the movements of vehicles in a fleet. The system is based on the principle of radio frequency identification (RFID), which uses radio waves to identify objects. RFID tags are attached to objects, and an RFID reader is used to read the tags. In the case of the car detection system, the reader is connected to a computer that is programmed to automatically detect the presence of vehicles in the given area. The system is designed to reduce the need for manual monitoring of traffic flow.

1.6 SYSTEM DIAGRAM

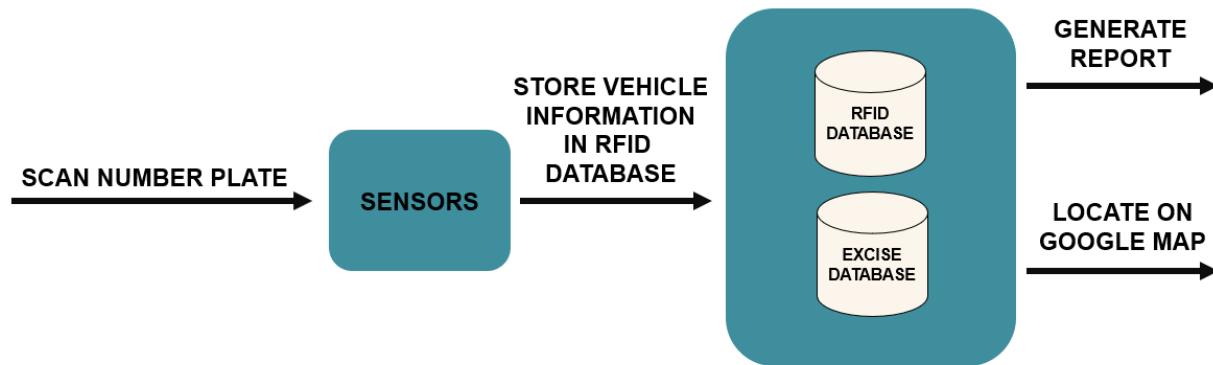


Figure 1 Working system diagram for the project

At first the scanner scans rfid card which has been pivoted in number plates. This sensor information then be sent to the server (database). The other taxation database stores information related to owner. At last the information generated through google maps and report.

1.7 BUSINESS VALUE OR BENEFITS OF PROJECT

The RFID car detection system provides a number of benefits for businesses, including:

1. Increased security: The system can help businesses to keep track of their vehicles and prevent theft.
2. Reduced costs: The system can help businesses to save money on insurance and fuel costs.
3. Improved efficiency: The system can help businesses to improve their fleet management and operations.

4. Enhanced customer service: The system can help businesses to provide better customer service by tracking vehicles and providing information about their location.
5. Increased sales: The system can help businesses to increase their sales by providing information about their vehicles' location and availability.

1.8 CONCLUSION OR FUTURE SCOPE

This system is using one of the most advanced technologies, Radio Frequency Identification or RFID. This technology is able to detect car, and to identify it. The system is able to detect a car from a marked point.

This system can be further improved by adding features like storing the count of vehicle in a non-volatile memory. This helps to count vehicles even if the system is switched off.

CHAPTER 2

BACKGROUND AND

LITERATURE REVIEW

2.1 BACKGROUND AND LITERATURE REVIEW

Our system rely on RFID (Radio frequency identification) chips and scanner which is integrated with gps and this scanner senses the rfid number which is pivoted in the license plate and sends longitude and latitude information against this vehicle rfid and points this location on google maps similarly more scanners have to be deployed to track vehicle with the points and gives the path from where the vehicle passes. And verifies license plate RFID with the owner information. This will check the tax information which is connected with the excise database.

2.2 SIMILAR APPLICATIONS

A fleet management system is a system used to track and optimize the performance of a fleet of vehicles. The system typically includes GPS tracking, vehicle diagnostics, driver behavior analytics, and more. In our project, The goal is same to locate and track the vehicles but not for single company or organization but to provide solution for vehicle tracking in all over Pakistan.

YouTube video explanation of a similar project (1566) IVIS RFID License Plate - YouTube

(1566) Fleet Management System with Live GPS Tracking in PHP | Free Source Code Download - YouTube [1] (Gps) Fleet Management System with Live GPS Tracking in PHP Free Download is a Vehicle Management System for managing vehicles. Fleet Management System with Live GPS Tracking is system can able to manage vehicles and drivers, customers, keep a track of income and expenses and get detailed reports. Real time GPS tracking and Geofence is available in the system, which help to track location. Geofence help to track eye on inbound and outbound of particular marked location. This system gives a unique tracking URL of trip and can able to share, the URL to customer to keep tracking.

HOME - RFID PAKISTAN [2]

RFID vehicle tracking system, Vehicle asset tracking | RAMP (ramprfid.com) [3]

2.3 RFID SYSTEM (FLEET MANAGEMENT SYSTEM) PRODUCT LINKS AND COMMON COMPARISON

Title (archive.org) (history of rfid)

GPS vs RFID asset tracking system - WandResearch (wand-research.com)

RFID in Motion: High-Speed Applications - atlasRFIDstore

RFID vehicle tracking system, Vehicle asset tracking | RAMP (ramprfid.com)

2.4 STUDY FOR RFID IN CONTAINER AREA

The paper presents a new method for image retrieval based on the use of a bag of words representation. The method is tested on a database of images of outdoor scenes. The results show that the proposed method outperforms the state of the art in terms of accuracy and efficiency. Rfid adoption for the container area where it discusses about the test that passes by an rfid.

A study of RFID adoption for vehicle tracking in a container terminal (upc.edu) [4]

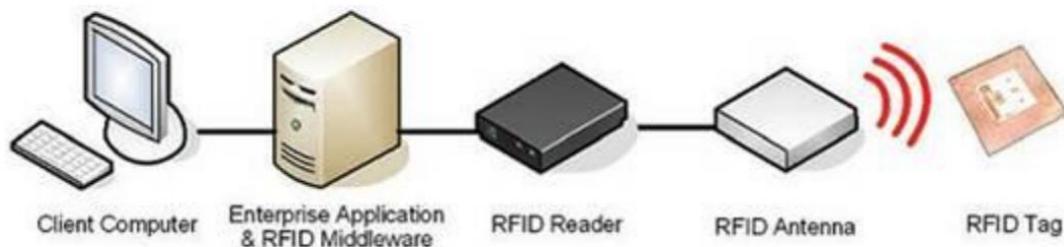


Figure 2 Basic Components of an RFID System

2.5 ARCHITECTURE DESIGN OF THE VEHICLE TRACKING SYSTEM BASED ON RFID

These are the samples of rfid vehicle tracking in the research article which is about cargo tracking but is similar to the vehicle tracking system which we are about to create in our system. Similarly, the scanners being placed on different locations to be able to track the vehicle at particular point. When the vehicle passes through this point scanners pins its location and inform the database that this vehicle passes by me at this time. And we are able to draw location on google map with the help of these points.

2317-3152-1-PB.pdf

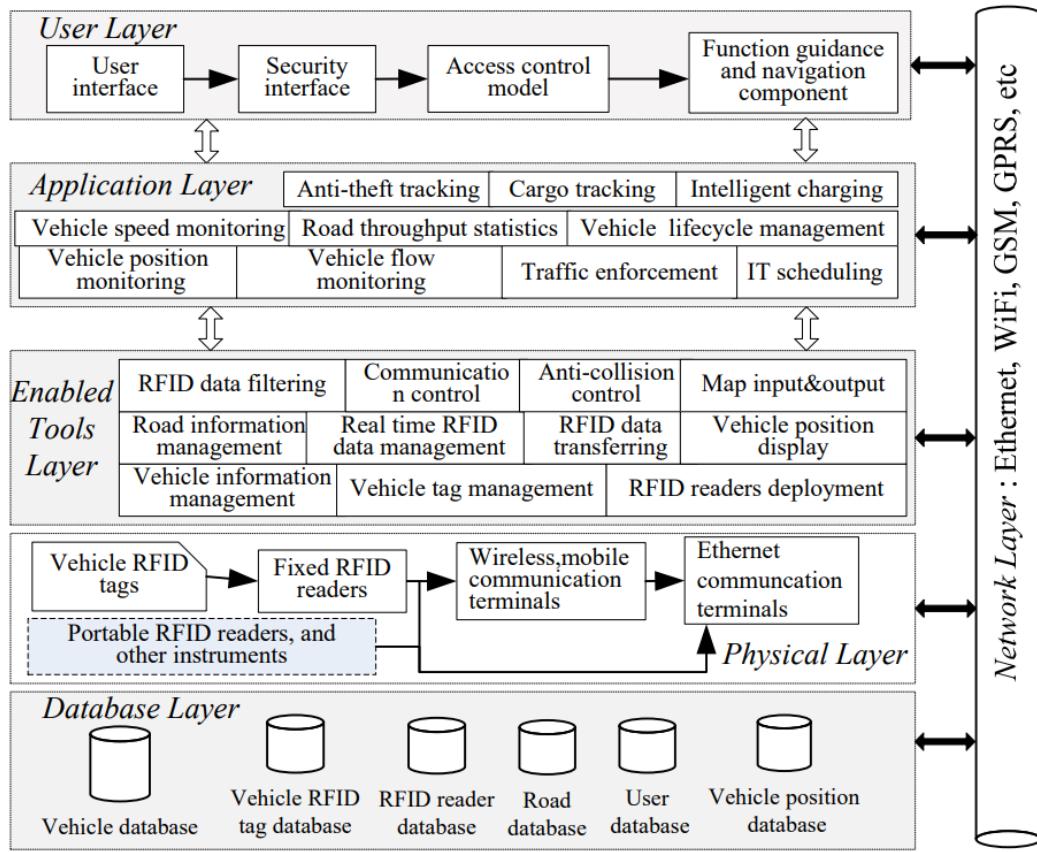


Figure 3 Vehicle system layer



Figure 4. RFID reader deployment sample



Figure 5. Cargo tracking results

Figure 4 Google maps

2.6 RFID TRACKING SYSTEM FOR VEHICLES (RTSV)

RFID tracking system for vehicles (RTSV) is a system that uses radio frequency identification (RFID) technology to track the movement of vehicles. It is typically used to track the location of vehicles in a fleet, or to track the movement of vehicles through a toll road or other restricted area. The system consists of a network of RFID readers, which are installed at strategic locations along the route that the vehicles will be travelling. The readers are connected to a central database, which stores the information collected by the readers. When a vehicle passes one of the readers, the reader will capture the vehicle's identification information, as well as the time and location of the reading. This information is then transmitted to the central database, where it is stored. The system can be used to track the location of individual vehicles, or to track the movement of vehicles in a fleet. The system can also be used to monitor the performance of a fleet, or to track the usage of a toll road or other restricted area.

RFID Tracking System for Vehicles (RTSV) (researchgate.net) [5]

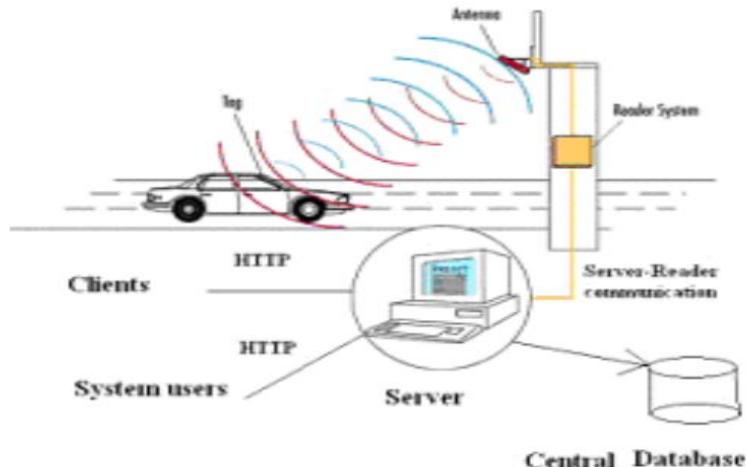


Figure 5 How the system works

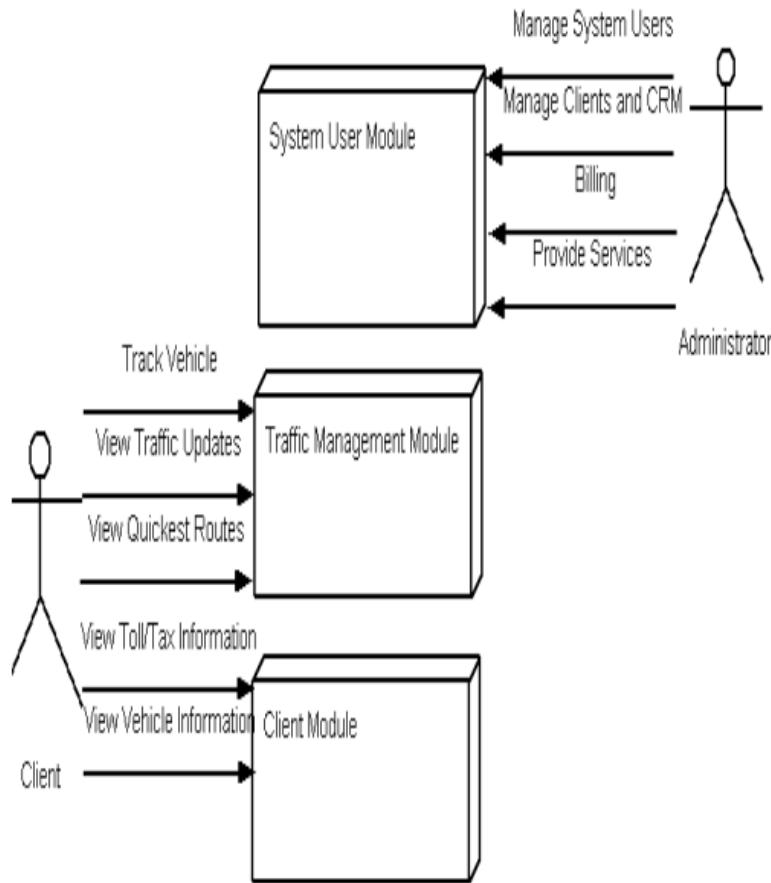


Figure 6 Component Diagram

2.7 RFID-BASED VEHICLE MONITORING SYSTEM

This research paper includes the literary work on monitoring inside a boundary with the rfid entering system. The paper presents a system for monitoring vehicles using radio frequency identification (RFID) tags. The system consists of an RFID reader, a microcontroller, and a GPS module. The RFID reader is used to read the RFID tags attached to the vehicles. The microcontroller is used to process the data from the RFID reader and the GPS module. The system is able to track the location of the vehicles and to provide information about the vehicles' status.

Paper Title (use style: paper title) ([researchgate.net](https://www.researchgate.net))

2.8 RFID INTRODUCTION

This paper describes how rfid works and the deployment issues that are being faced by the programmer at the time of implementation. Understanding of Radio frequency has been a great source when we are working on rfid implantation solutions.

untitled (roywant.com)

2.9 ANTI-THEFT SYSTEM

This paper is about the vehicle anti-theft location which is kind of similar to our project but uses different methodology of GPS and GSM integrated with android mobile and pyroelectric infrared sensor. The idea of this is to lock via rfid and if intruder still be managed to enter the vehicle, then pyroelectric sensor works and be able to identify by the heat produced by the person and pins the location of the vehicle with the help of android and GPS. This will send latitude and longitude information to the owner via SMS

untitled (xilirprojects.com)

2.10 ANTI-THEFT SYSTEM 2

In Bangladesh, Vehicle theft is a major problem. In this paper they propose a Smart Anti-theft vehicle tracking system. The entire system is used in two phases. In the first phase, they used a device that is embedded on a vehicle. The device collects data from the sensors and sends it to the cloud. The second phase uses the data to determine the location of the vehicle and uses GPS (Global Positioning System) to track the location of the vehicle. This system is based on Internet of things (IoT), which includes sensors, cloud, and GPS. The system is tested on the Android platform with a range of sensors. It is observed that the accuracy of the system is 100% in the case of vehicle theft.

https://www.researchgate.net/profile/Maruf-Islam/publication/322516489_Smart_anti-theft_vehicle_tracking_system_for_Bangladesh_based_on_Internet_of_Things/links/5c93134145851506d71f7728/Smart-anti-theft-vehicle-tracking-system-for-Bangladesh-based-on-Internet-of-Things.pdf). [6]

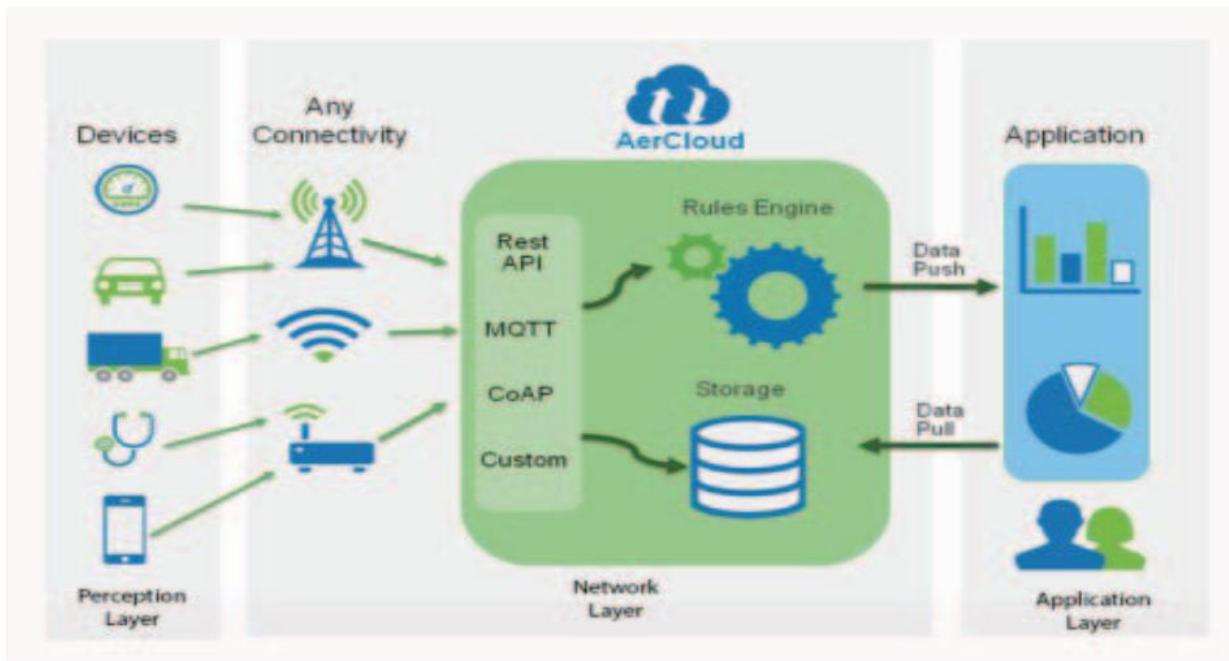


Figure 7 Anti thief Diagram

CHAPTER 3

AIM AND STATEMENT OF THE PROBLEM

3.1 EXPLAINATION OF PROJECT

The vehicle tracking system with RFID uses radio frequency identification (RFID) technology to track the location of vehicles. RFID tags are attached to each vehicle, and these tags emit radio signals that can be detected by RFID readers. The readers then relay the information to a central system, which can track the location of the vehicles in real-time. This system can be used to monitor the movement of vehicles in a fleet, or to track the location of individual vehicles.

3.2 DEFINATION OF THE PROBLEM

Now a days crime is increasing day by day similarly the vehicle are stolen rapidly and can't be recovered easily. There must be proper autonomous system which can track these activities and inform to the concern authorities. Today almost 169 cars have been stolen and not been recovered yet this report has been published by CPLC (citizen polices Liasion committee) these stats clearly showing us the deficiency of management of these vehicles.

3.3 LIMITATION OF SCOPE

The main limitation of the vehicle tracking system with RFID is the range of the RFID tags. The tags need to be within a certain range of the reader in order to be read, so if a vehicle goes outside of that range it will not be tracked. Another limitation is that RFID tags can be blocked by metal or other materials, so if a vehicle is parked in a garage or near a metal structure, the tag may not be readable. And RFID tags can be easily removed or tampered with, which could allow vehicles to avoid being tracked.

3.4 SOLUTION OF THE PROBLEM

1. The vehicle tracking system with RFID uses a tag or transponder that is attached to the vehicle.
2. The tag contains information about the vehicle, such as the make, model, and license plate number.
3. The RFID reader scans the tag and decodes the information.
4. The tracking system then uses the information to track the vehicle's location.

3.5 LACK OF TIME

One of the main challenges that the team may face is a lack of time to properly test and implement the system. This could lead to potential issues with the system's accuracy and reliability. If the team does not have enough time to work on the project, the quality of the RFID Vehicle detection system may be compromised. The team may not be able to finish developing and testing the system before the deadline. If the RFID Vehicle detection system project is not given enough time to be completed, it is likely that the system will not work as intended. The system may have errors, or it may not be able to properly detect vehicles. It is likely that it will not be completed correctly or to the fullest extent. This can cause major problems and setbacks for the company or organization that is implementing the system.

3.6 SCOPE CREEP

If the RFID vehicle detection system is changed, it may affect the project in a number of ways. For example, if the system is changed to a different type of RFID system, it may be incompatible with the existing hardware and software, and may require new hardware and software to be installed. Additionally, the change may require new training for personnel, and may cause disruptions to the project schedule. The RFID vehicle detection system may affect the budget and schedule for the project. The RFID vehicle detection system is used to track vehicles as they move through the detection zone. If the system is changed, it could affect the accuracy of the vehicle tracking and the overall performance of the project. Also the cost of the project will increase because the new system will be more expensive. The schedule for the project will be delayed because the new system will take longer to install. And the scope of the project will change because the new system will have different capabilities than the old system.

CHAPTER 4

HARDWARE, SOFTWARE ANALYSIS & REQUIREMENTS

4.1 FACT FINDING

The goal of this project is to detect vehicles with the help of RFID. The first step is to find an RFID tag that can be used to detect vehicles. The second step is to find an RFID reader that can be used to read the tag. The third step is to find an antenna that can be used to read the tag. The fourth step is to find a power source that can be used to power the tag and the reader. The fifth step is to find a way to connect the tag and the reader to the power source. The sixth step is to find a way to connect the tag and the reader to the antenna. The seventh step is to find a way to connect the tag and the reader to the vehicle. The eighth step is to find a way to connect the tag and the reader to the ground. The ninth step is to find a way to connect the tag and the reader to the computer. The tenth step is to find a way to connect the tag and the reader to the Internet.

The purpose of this fact finding is to determine the feasibility of using an RFID system for vehicle detection. RFID systems are already in use for a variety of applications, but their use for vehicle detection is relatively new. This fact finding will examine the potential benefits and drawbacks of using an RFID system for vehicle detection.

4.1.1 BENEFITS

1. RFID systems can provide real-time tracking of vehicles. This information can be used to improve traffic flow and reduce congestion.
2. RFID systems can be used to identify stolen vehicles.
3. RFID systems can be used to track the location of vehicles in case of an emergency.
4. RFID systems can be used to monitor the speed and movement of vehicles. This information can be used to improve safety on the roads.
5. RFID systems can be used to automatically toll vehicles. This can reduce the need for human toll collectors and improve efficiency.

4.1.2 DRAWBACKS

1. RFID systems require the installation of infrastructure, such as antennas and readers. This can be costly and time-consuming.
2. RFID systems can be susceptible to interference from other electronic devices.
3. RFID systems can be hacked

4.2 HARDWARE

The main hardware components used in a Vehicle Detection system with RFID would include an RFID reader, an antenna, and a tag. Additionally, you might also need a computer or other device to store and process the data collected by the RFID reader.

RFID

MFRC522 RC522 RFID READER

TAG

ESP32

GPS MODULE NEO6M V2 U BLOX

4.2.1 RFID

RFID is a short-range wireless communications technology that uses electromagnetic fields to identify and track objects. RFID technology can be used in a variety of applications, including access control, inventory management, asset tracking, and vehicle detection.

RFID technology can be used to detect vehicles in a variety of ways. For example, RFID tags can be affixed to the outside of vehicles, or RFID readers can be installed in strategic locations to detect the presence of vehicles. RFID technology can also be used to track the movement of vehicles in real-time, allowing for better traffic management and route planning. [1]



Figure 8 RFID

4.2.1.1 HOW RFID WORKS

Radio-frequency identification (RFID) is a technology that uses radio waves to identify and track objects. RFID tags are small devices that contain a chip and antenna that emit a signal that can be read by an RFID reader.

RFID tags are attached to objects such as vehicles, and the signal is used to identify and track the object. RFID readers can be handheld, mobile, or fixed, and they can be used to track objects in real time. [7]

4.2.1.2 ACTIVE RFID

Active RFID tag is an active transponder with a built-in antenna that can be read at long ranges. Active RFID tags use battery power to transmit a signal to a reader. The battery life of an active tag is usually 5 to 10 years. Active RFID tags are used in applications where long read ranges are required, such as in warehouse management and tracking high-value assets.

RFID is an automated identification technology that uses radio waves to identify people or objects. RFID technology can be used for a variety of purposes, including access control, inventory management, and asset tracking.

4.2.1.3 PASSIVE RFID

Passive RFID technology is used in a system where the reader does not require a power source. Instead, the tag is powered by the electromagnetic field generated by the reader. The power for the tag is typically supplied by a battery, which is used to store the data. The tag does not have its own power source, so it cannot be used to track the movements of an object.

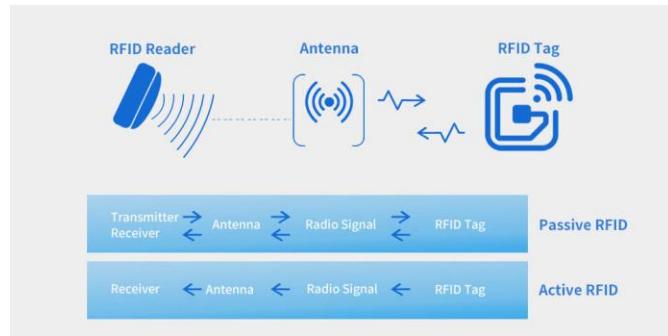


Figure 9 Passive RFID

4.2.2 MFRC522 RC522 RFID READER

RFID (Radio Frequency Identification) is a technology that allows us to identify certain objects without physical contact. The way it works is very simple. There is a tag or card which contains a chip with a unique serial number. This serial number is read with a reader, which is connected to a computer.

The use of RFID is not new. This technology has been used for many years in the field of logistics and transportation, where it has been used to track and trace products and goods. However, the use of RFID is not limited to logistics and transportation. It can be used in many other fields, such as healthcare, retail, security, and so on.

The main advantages of RFID are its speed and accuracy. With RFID, you can track and trace objects in a matter of seconds, which is not possible with other technologies such as barcodes. In addition, RFID is not affected by dirt, dust, or water, which means that it can be used in harsh environments.

The main disadvantage of RFID is its cost. RFID tags and readers are not cheap, and the infrastructure required to use RFID can be costly. In addition, RFID is a complex technology, and it requires trained personnel [8].

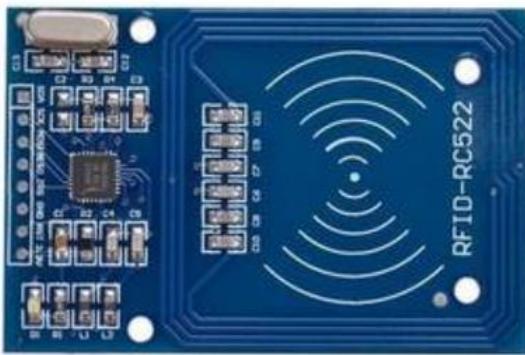


Figure 10 MFRC522 RC522 RFID READER

4.2.3 TAG

RFID tag is a kind of tag which uses RFID technology to facilitate the identification of objects. RFID tag comprises an antenna, a semiconductor chip and a substrate. The antenna is electrically connected to the RFID chip and is used to receive the signal transmitted by the reader, and then transmit the signal to the RFID chip; the RFID chip comprises a memory space, and the control unit includes an address space and a command space; the data in the memory space can be modified through the command space, and the address space is used to store the address of the command space; the substrate is used to fix the antenna and the RFID chip on the object to be identified.

RFID tag is a non-contact automatic identification technology. RFID tag does not need to be in visual line of sight, which can be identified at any time and any place. It is mainly based on radio frequency signal to automatically identify target and read and write data. It is mainly used in identification and tracking of products, people, vehicles, and animals in various fields. It is an important part of radio frequency identification technology.

RFID tag has three major components: antenna, chip and substrate. Antenna is mainly used to receive and transmit electromagnetic [8].



Figure 11 RFID Tag

4.2.4 ESP32

The ESP32 is a powerful, low cost, Wi-Fi, and Bluetooth system on a chip. It is a single chip solution for low power consumption wireless communication. It is designed for high performance applications such as Internet of Things (IoT), machine-to-machine (M2M), and wearable electronics. The ESP32 is an advanced, low power, wireless SoC with integrated dual-mode Bluetooth and Wi-Fi. It supports a range of peripherals, including UART, SPI, I2C, PWM, ADC, digital microphone, and touch sensor. It is an excellent choice for embedded applications that require robust networking and low power consumption. The ESP32 is ideal for applications that require high performance and low power, such as home automation, security systems, and IoT devices.

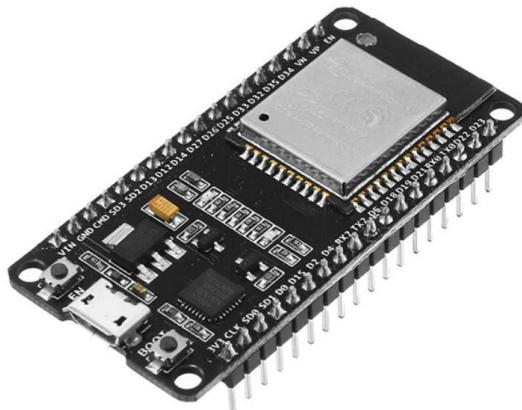


Figure 12 ESP32

4.2.5 GPS MODULE NEO6M V2 U BLOX

The NEO-6M GPS module is a highly sensitive and low power GPS receiver. It is based on the U-BLOX NEO-6M GPS module and has the same performance and features. The module is suitable for a wide range of applications including navigation, tracking, timing, surveying, and much more. It has an item size of 25*25mm and a very low power consumption of only 6mA. It also has an integrated antenna and a high level of accuracy with up to 10 meters. It is designed to be used in a wide range of applications and is compatible with most microcontrollers. It is designed to provide a reliable, accurate, and cost-effective solution for GPS applications.

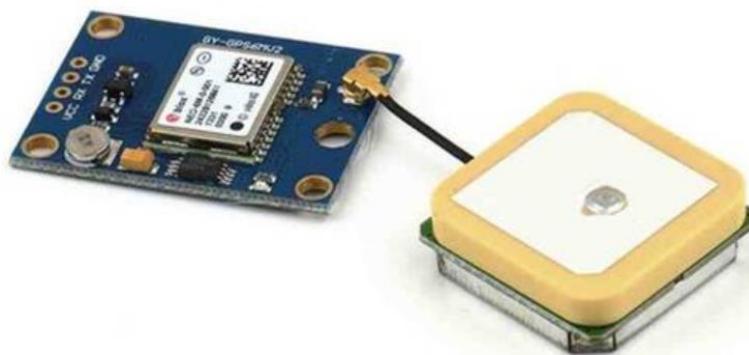


Figure 13 GPS MODULE NEO6M V2 U BLOX

4.3 HARDWARE COST

In this section we will discuss the main and important points related to our project and compare the equipment, discuss detail further to explain why we use this kind of product in our project.

4.3.2 WORKING ASSUMPTION

How the system works integrated we use the Raspberry Pi can be used as a GPS tracker by connecting it to a GPS module. The GPS module will then be able to track the location of the Raspberry Pi. To send the location of the Raspberry Pi, you will need to use a communication module such as a cellular modem or Wi-Fi. The location can then be sent over the internet to a tracking service or website. We will also need to install a software package called gpsd, which provides a daemon that communicates with GPS modules and provides GPS data to applications.

Products	Estimated prices
RFID CHIP	50X10=500
RFID READER	350x1050
ESP32	950x3=2850
GPS MODULE NEO6M V2 U BLOX	1000x3=3000
JUMPER WIRES	5x200=1000
MB102 BREADBOARD	160x3=480

Table 1 Cost estimation

4.3.3 RFID VS AI (ARTIFICIAL INTELLIGENCE)

There is no clear winner when it comes to RFID vs Artificial Intelligence advantages and disadvantages. Each technology has its own set of advantages and disadvantages that need to be considered when deciding about which technology to use.

RFID advantages include its ability to track and manage inventory, its low cost, and its durability. RFID disadvantages include its potential for abuse and its reliance online-of-sight technology.

Artificial intelligence advantages include its ability to learn and improve over time, its ability to make decisions based on data, and its ability to automate tasks. Artificial intelligence disadvantages include its lack of common sense and its potential for misuse. And it is much expensive than RFID tags because of its computation and processing of data and equipment like expensive cameras to record high quality video.

4.4 UML DIAGRAMS

4.4.1 OPERATIONAL DIAGRAM

RFID's to be installed in number plate of cars and scanners are installed on various points of the city which then scan these chips. And send the scanned rfid to the database with its location which then be mapped on to google maps. The Rfid database store's location with date and time of the vehicle which it passes. this all information can be accessed by a dashboard. In the end authenticated by excise database.

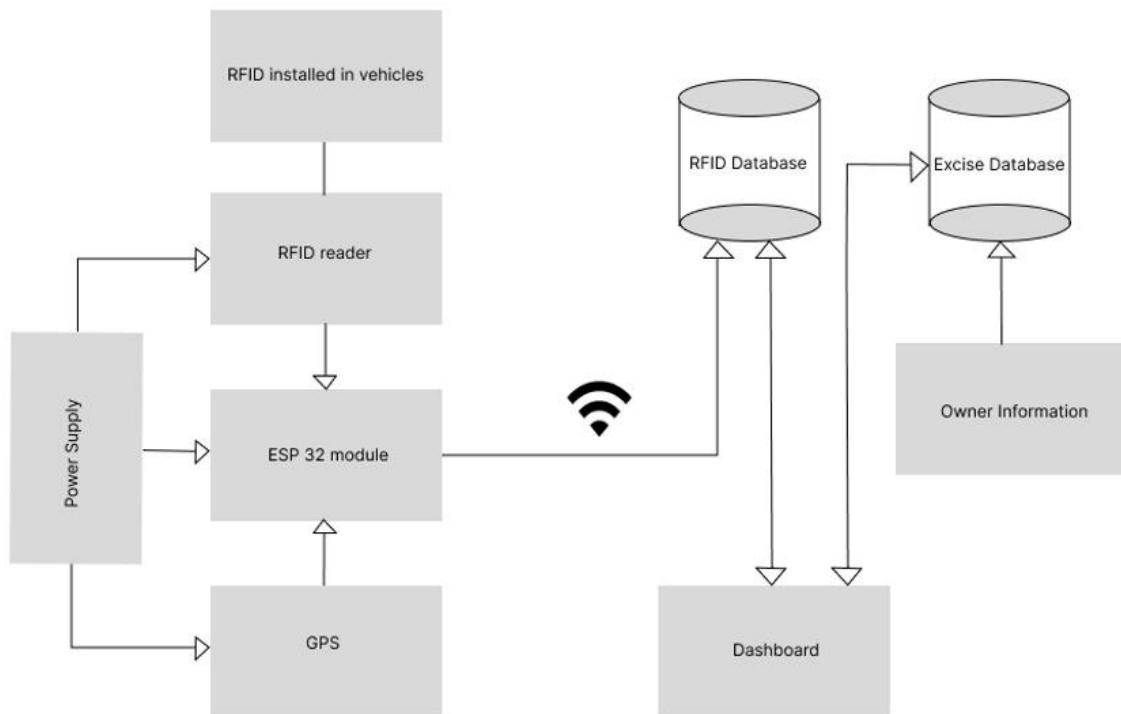


Figure 14 Operational Diagram

4.4.2 ACTIVITY DIAGRAM

The activity diagram for a vehicle tracking system with RFID would show the various activities that need to be performed to track a vehicle. These activities would include reading the RFID tag on the vehicle, determining the location of the vehicle, and then tracking the movement of the vehicle.

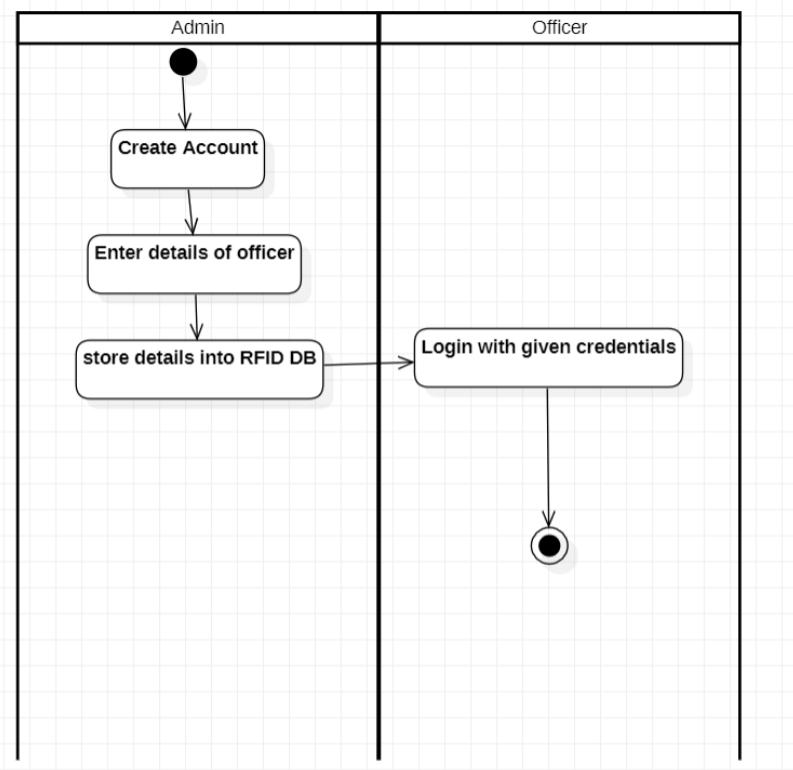


Figure 15 Activity Diagram 4.1

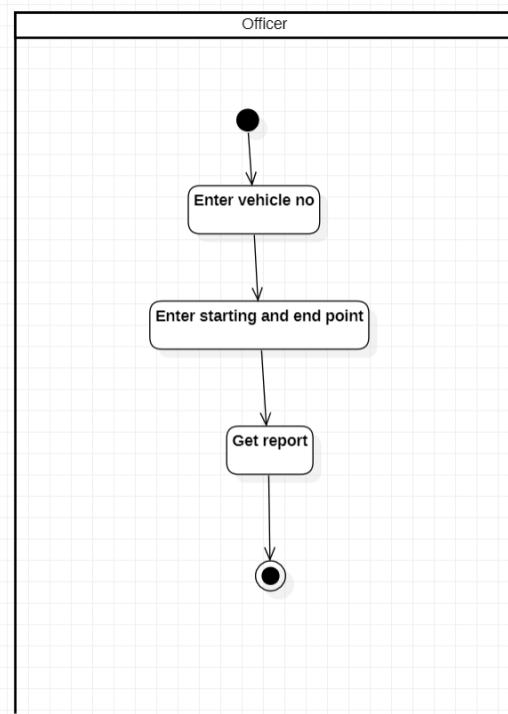


Figure 16 Activity Diagram 4.2

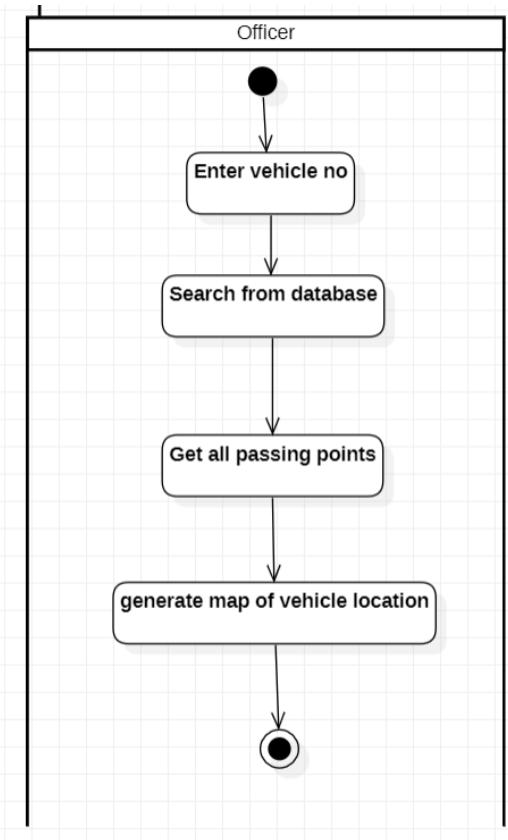


Figure 17 Activity Diagram 4.3

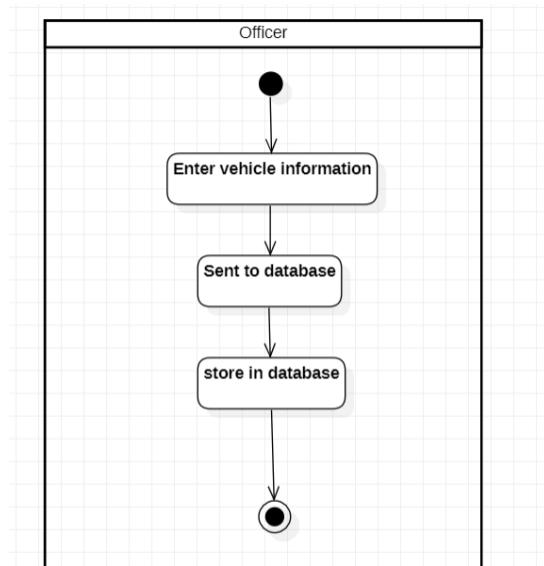


Figure 18 Activity Diagram 4.4

4.4.3 USECASE DIAGRAM

There are two users Admin who administer whole system including search,register,getting location and owner data. The other user is officer who maintain logs of admin and create or remove admins from the existing system.

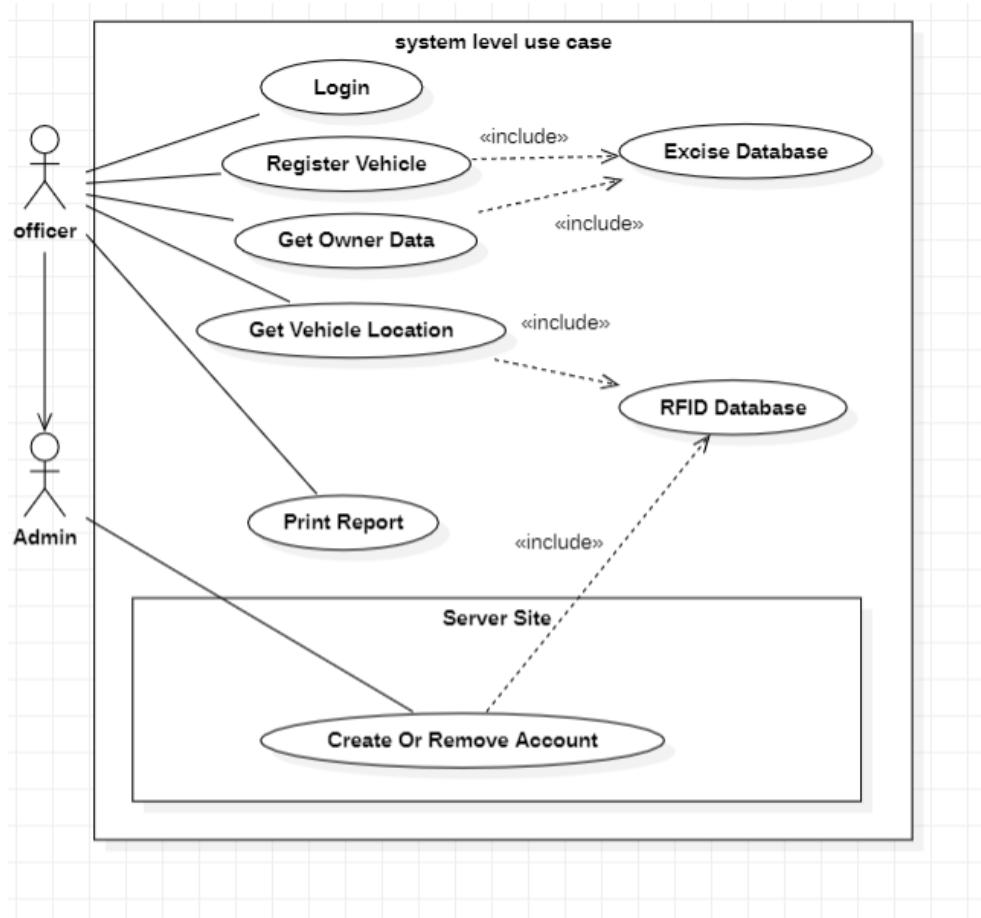


Figure 19 Usecase Diagram 4.1

CHAPTER 5

SOFTWARE DESIGN AND

MODELING

5.1 ARCHITECTURE

MVC (Model View Controller):

The Model-View-Controller (MVC) architecture is a design pattern that is commonly used to organize the code of an application. It is particularly useful for complex systems such as a vehicle detection system using RFID and GPS, as it allows for the separation of different aspects of the system into distinct components.

The Model is responsible for handling the data and logic of the system. In this case, it would handle the information received from RFID and GPS sensors and would process this data to determine if a vehicle is present. It would also handle any algorithms or logic required for the detection system to function properly.

The View is responsible for displaying the information to the user. In this case, it would handle the presentation of the vehicle detection data on a map or in a list. The view would also handle the user interface elements such as buttons, menus, and other controls that the user interacts with.

The Controller is responsible for handling the flow of information between the Model and the View. It would take the information from the Model and pass it to the View, and would also take user input from the View and pass it to the Model for processing. The controller would also handle any updates to the view based on new information from the Model.

By separating the system into these three components, it becomes easier to understand and maintain the code. Additionally, if one component needs to be updated or changed, it can be done independently of the other components, making the system more flexible and easier to scale.

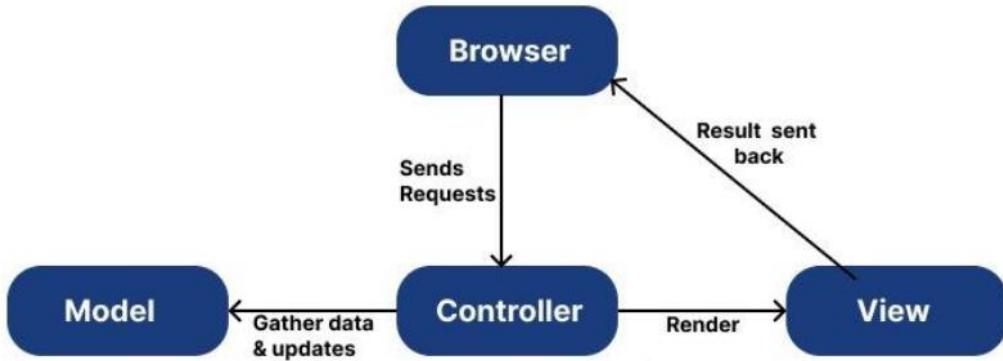


Figure 20 MVC

5.2 DIAGRAMS

5.2.1 USECASE DIAGRAM

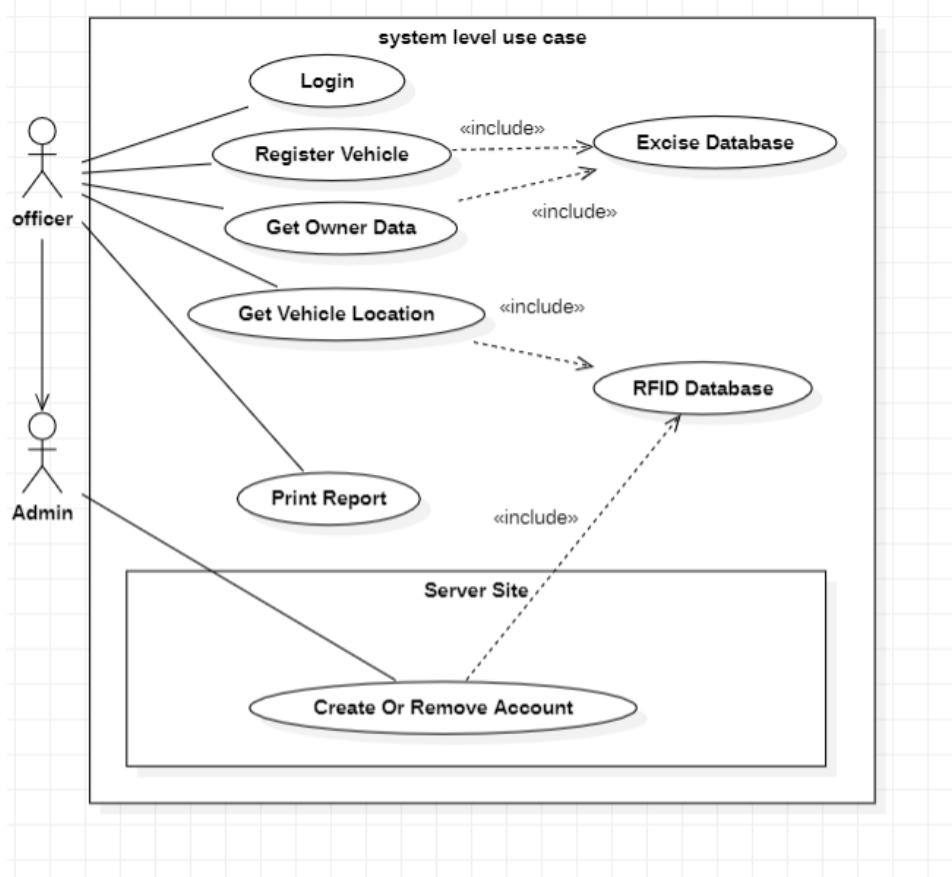


Figure 21 Usecase Diagram 5.1

A use case diagram for a vehicle detection system using RFID would show several actors and use cases. The actors in this system would include:

- Admin: responsible for creating and removing accounts for officers and managing the system's settings.
- Officer: responsible for registering vehicles, tracking vehicles using RFID technology, generating reports, and managing the system's data.

The use cases in this system would include:

- Create/Remove Officer Account: The admin can create or remove an account for an officer.
- Login: The officer can log in to the system using their credentials.
- Register Vehicle: The officer can register a vehicle by inputting the vehicle's registration number and other relevant information.
- Get Owner Data: The officer can retrieve the owner's information by inputting the vehicle's registration number.
- Get Vehicle Location: The officer can track the vehicle's location by using the RFID technology
- Print Report: The officer can generate, and print reports based on the system's data.

All these use cases are connected by the system and can be performed by the officer after successful login and by admin to manage the system.

5.2.2 OBJECT DIAGRAM

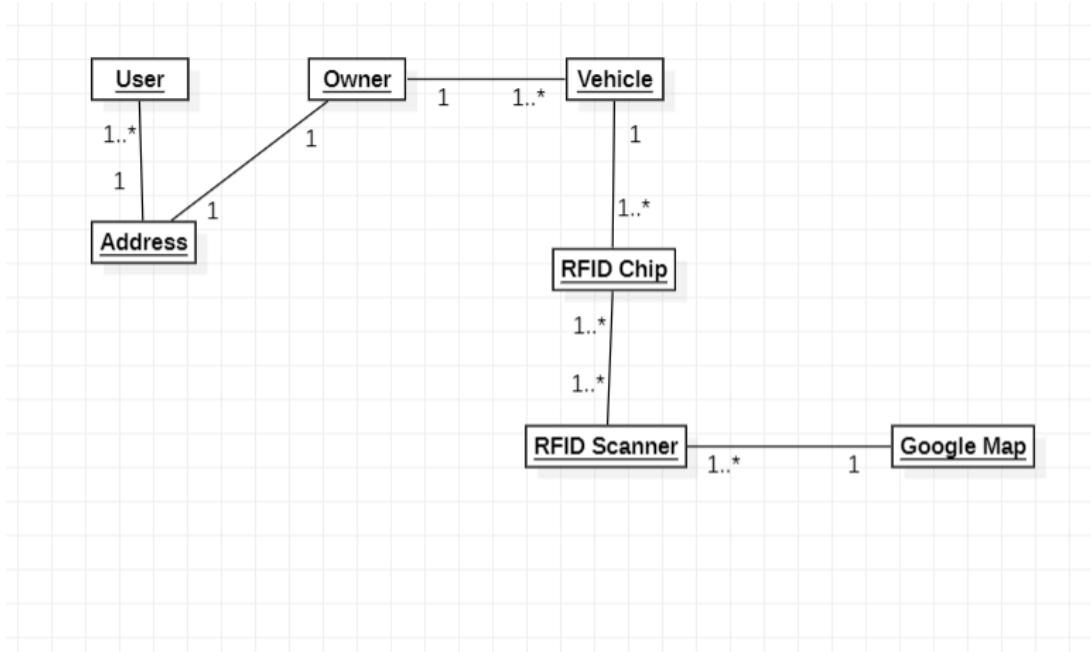


Figure 22 Object Diagram

Object diagrams in the vehicle detection system are used to model the static structure of a system. This diagram is used to show the objects and the relationships between them.

The main objects in the vehicle detection system include User, Owner, Vehicle, RFID Scanner, and software. Each of these objects has its own attributes such as location, performance, type of sensor

The relationships between these objects include the data flow between them. For example, the data from the sensors is sent to the processor, which then sends it to the software.

5.2.3 ACTIVITY DIAGRAM

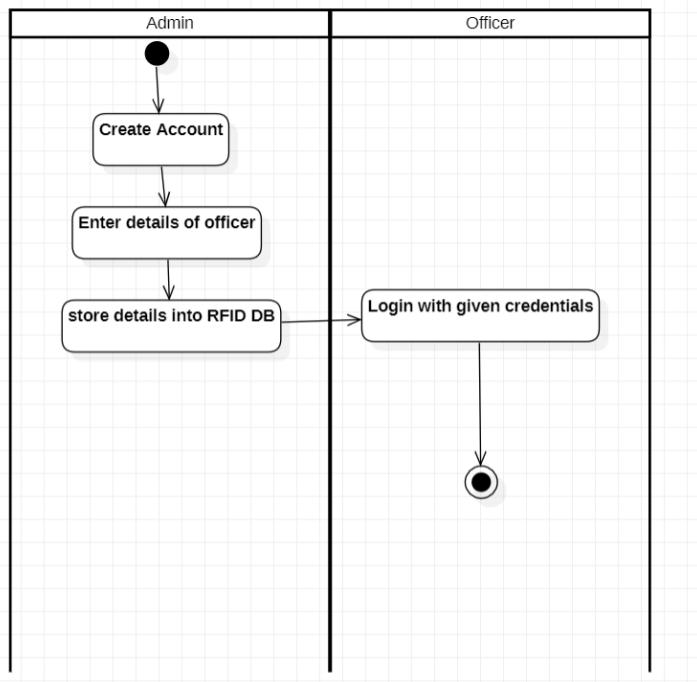


Figure 23 Activity Diagram 5.1

In this activity admin can create an account for an officer and the officer can login the account with the given credentials would involve several steps. The first step would be for the admin to input the officer's information, such as name, username, CNIC number, phone number and password. This information would then be stored in a database.

Once the account is created, the officer would be able to use their username and password to login to the system. The officer would be prompted to enter their credentials, such as username and password. This information would then be verified against the database to ensure that the officer has the correct credentials.

Once the officer is logged in, the officer would have access to the system's functions, such as registering vehicles, generating reports, and tracking vehicles.

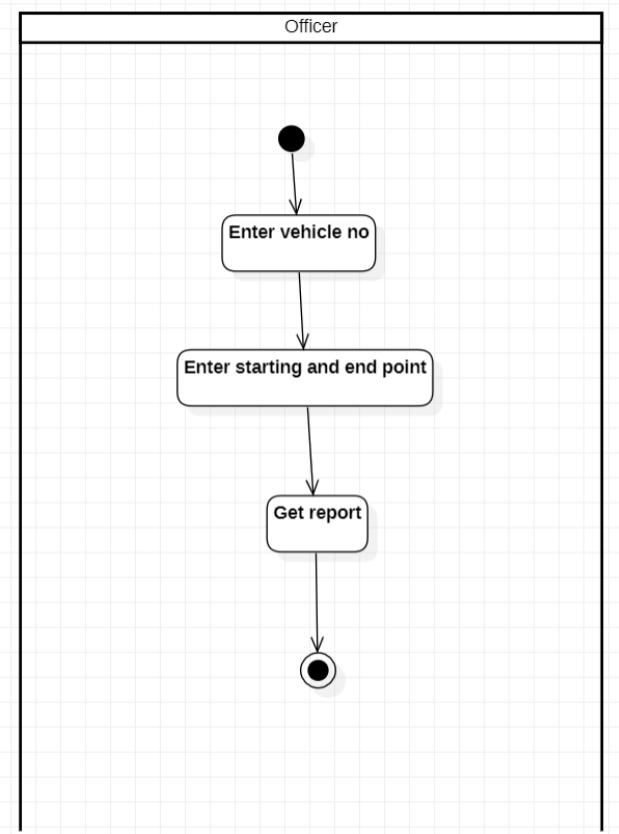


Figure 24 Activity Diagram 5.2

An activity diagram for a vehicle detection system using RFID would involve several steps. The first step would be for an officer to enter the vehicle number. This information would then be verified and matched with a database of registered vehicles.

Once the vehicle has been identified, the officer would enter the starting and end point for the vehicle's journey. This information would also be verified and matched with a database of authorized routes.

Once the starting and end point have been confirmed, the vehicle would be allowed to proceed, and its location would be tracked using RFID technology. This information would be recorded and stored in a database.

Finally, the officer would be able to generate reports based on the collected data, such as a list of vehicles that have passed through a certain point, or the routes taken by a specific vehicle.

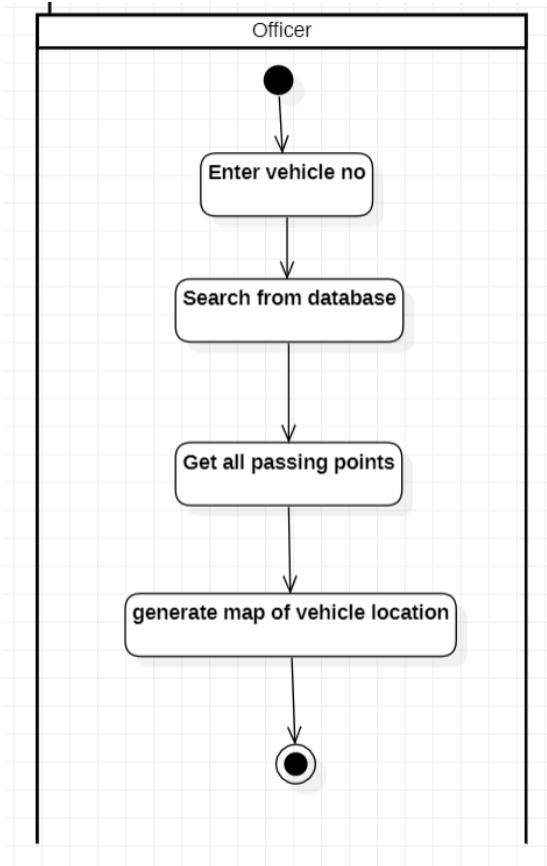


Figure 25 Activity Diagram 5.3

An activity diagram for a vehicle detection system using RFID would involve several steps. The first step would be for an officer to enter the vehicle number using an input device, such as a keyboard or scanner. This information would then be verified and matched with a database of registered vehicles.

Once the vehicle has been identified, the officer would enter the starting and end point for the vehicle's journey. This information would also be verified and matched with a database of authorized routes.

Once the starting and end point have been confirmed, the vehicle would be allowed to proceed, and its location would be tracked using RFID technology. This information would be recorded and stored in a database.

Finally, the officer would be able to generate map based on the collected data, such as a list of vehicles that have passed through a certain point, or the routes taken by a specific vehicle.

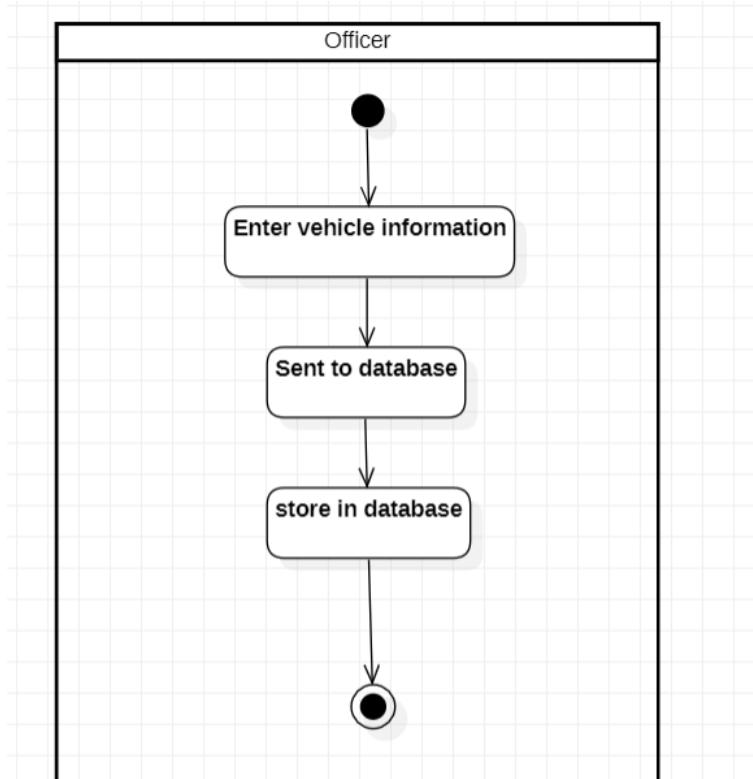


Figure 26 Activity Diagram 5.4

In this activity officer can register vehicles using a vehicle registration number would involve several steps. This information would then be verified and matched with a database of registered vehicles.

If the vehicle is not already registered, the officer would be prompted to enter additional information about the vehicle, such as make, model, and owner's details. This information would also be stored in the database.

Once the registration process is complete, the officer would assign a RFID tag to the vehicle, which would be affixed to the vehicle. The RFID tag would contain unique identification number that corresponds to the vehicle registration number and all the other details.

5.2.4 SEQUENCE DIAGRAM

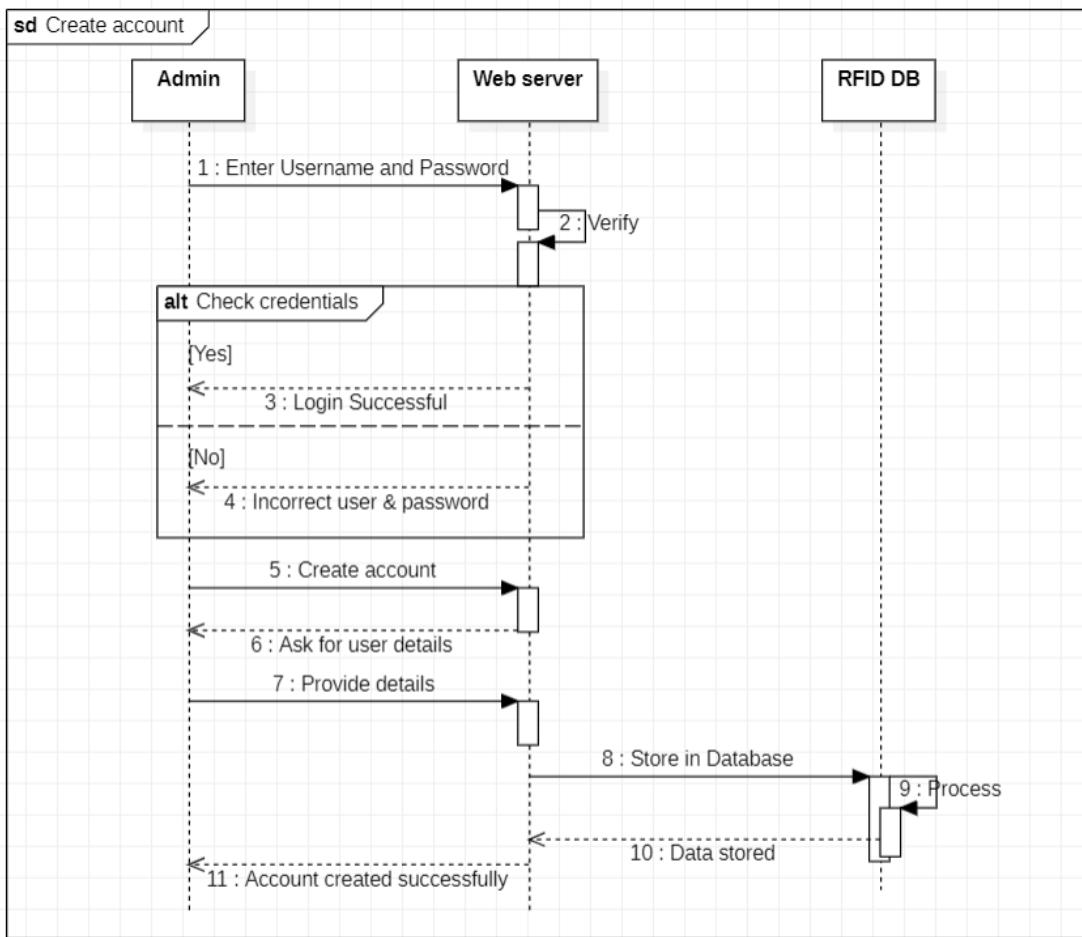


Figure 27 Sequence Diagram 5.1

When new officer will join the department, he/she will request the admin of the system to create his or her account the admin will get his personal information like CNIC number, address, phone number the system will then store his information in the database and create account so that a user can use the system by providing USERID and password. Now whenever the user enters his id password into the system the system will first verify the user and then will give access if valid.

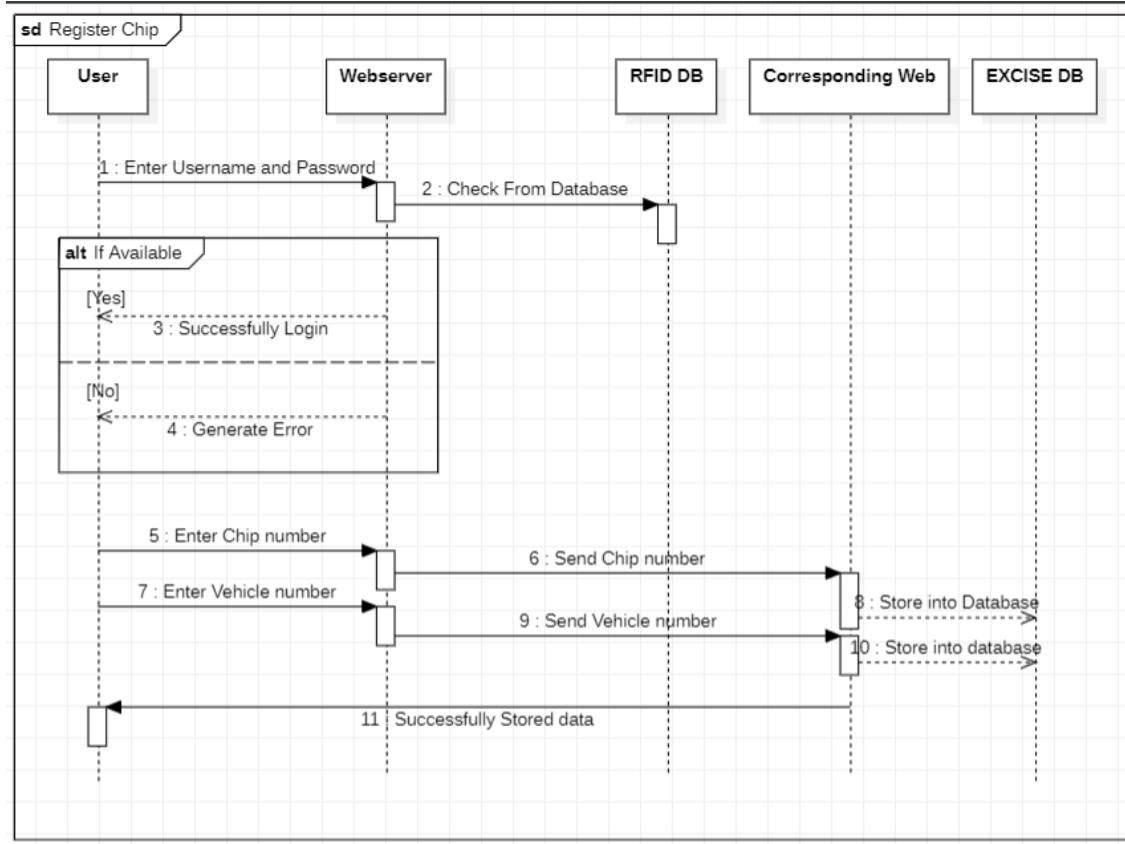


Figure 28 Sequence Diagram 5.2

The officer (User) can register the chip by sending the information of RFID chip and vehicle to the Excise database. The Excise Database will verify the data and registered the vehicle if valid information is provided

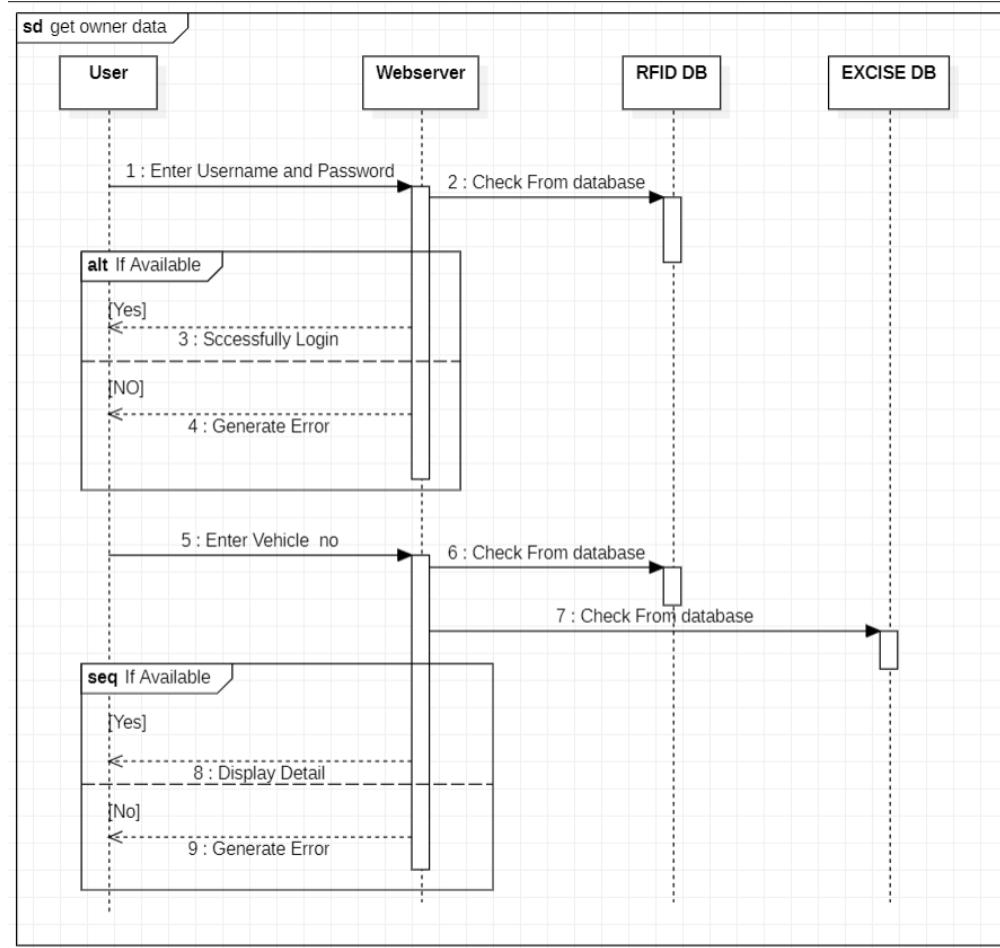


Figure 29 Sequence Diagram 5.3

The officer will login in to the system by entering his username and password the system will first verify the username and password from database then will act accordingly. For getting the owner data the officer will first enter the vehicle number in to the website the web application then fetch data from the RFID as well as EXCISE database if the data will be available then it will be displayed on screen

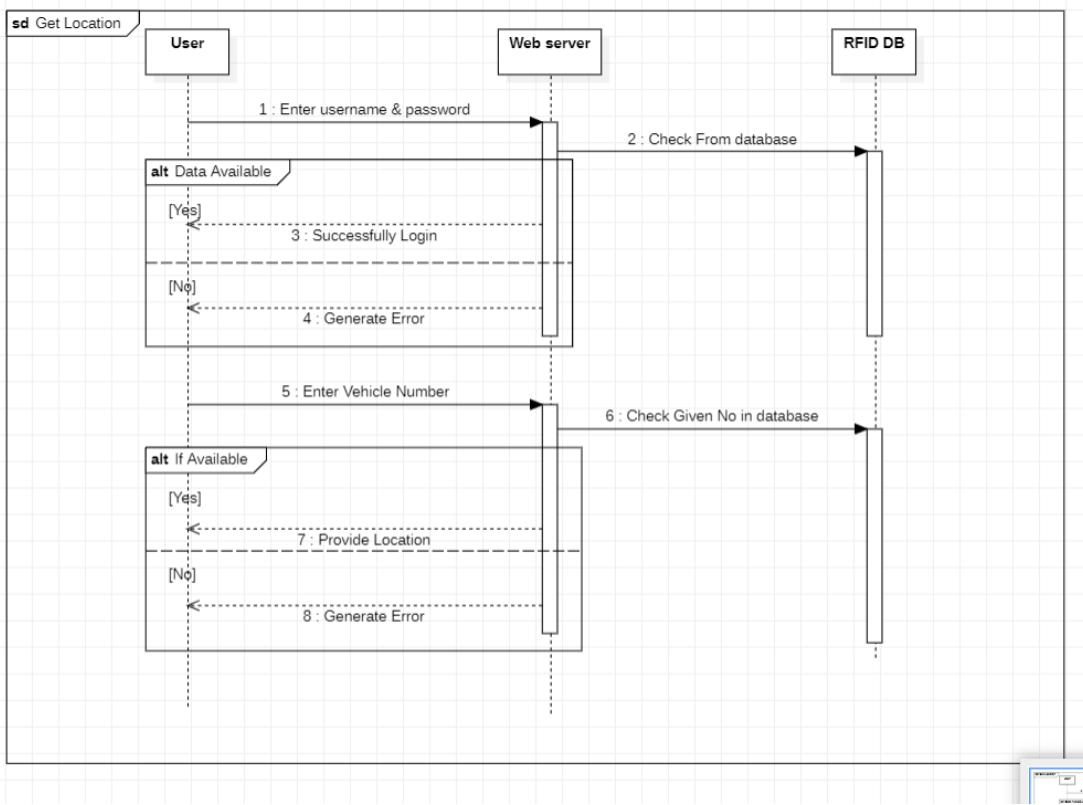


Figure 30 Sequence Diagram 5.4

In the same way the officer (User) can get the location of the vehicle. Whenever the Officer enter the data of the vehicle like Vehicle number & date the system will fetch the Location from the database that we created (RFID DATABASE) where the location is saved by RFID scanner.

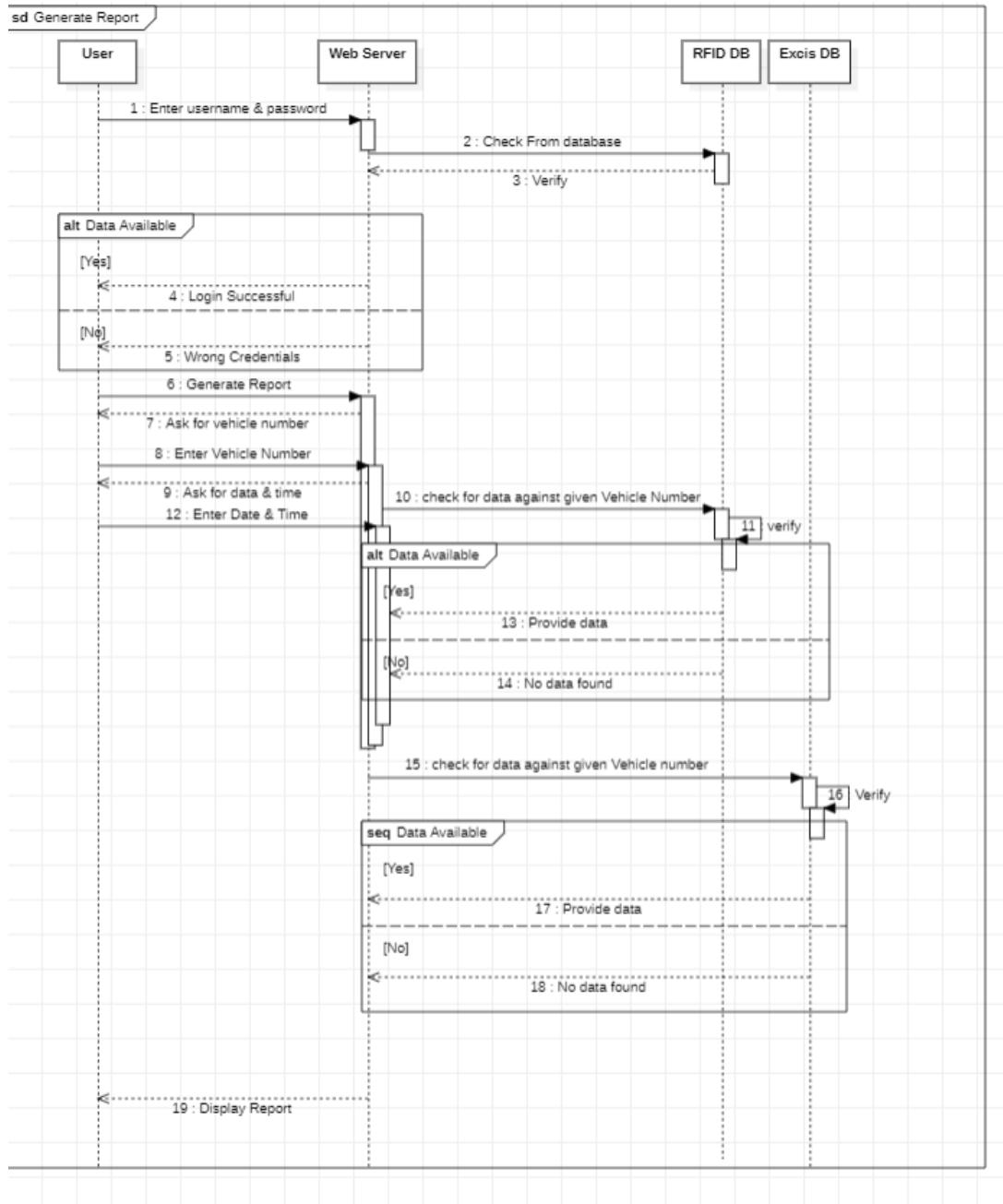


Figure 31 Sequence Diagram 5.5

The officer will first login in to the system for printing the report the user will enter vehicle number and date and time into the web server the web will check the data into the database if available then will provide the information and will display on the screen

5.2.4 CLASS DIAGRAM

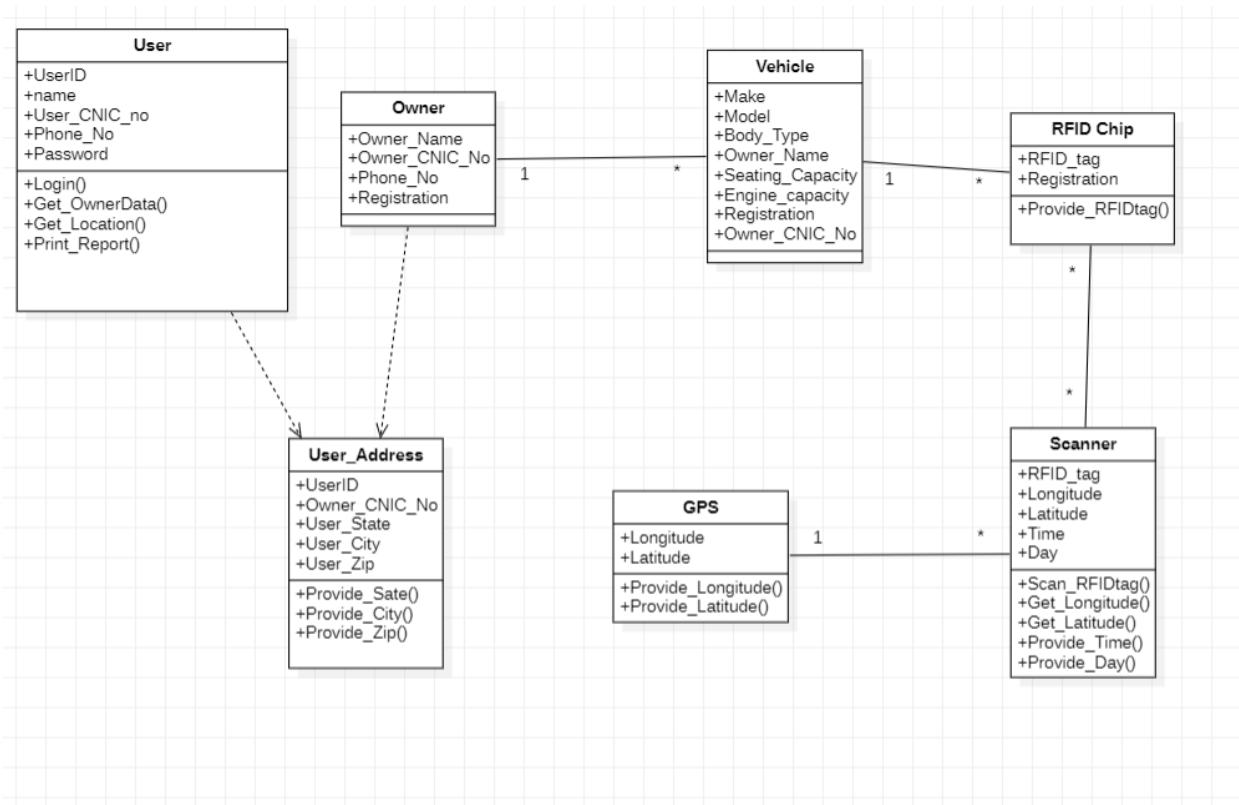


Figure 32 Class Diagram

In this Project Vehicle detection system the main objectives are Owner, User, Vehicle, RFID Chip, GPS, and Scanner

The above diagram shows that the User is able to Login in to the system, He can get the information of the Owner, Also get the location of the car using the system and he can also print the report of the data.

The scanner will scan the RFID Chip attach to the system and will provide the giving information like longitude, Latitude which will be provided by GPS module. Also, date and time.

5.2.5 ER DIAGRAM

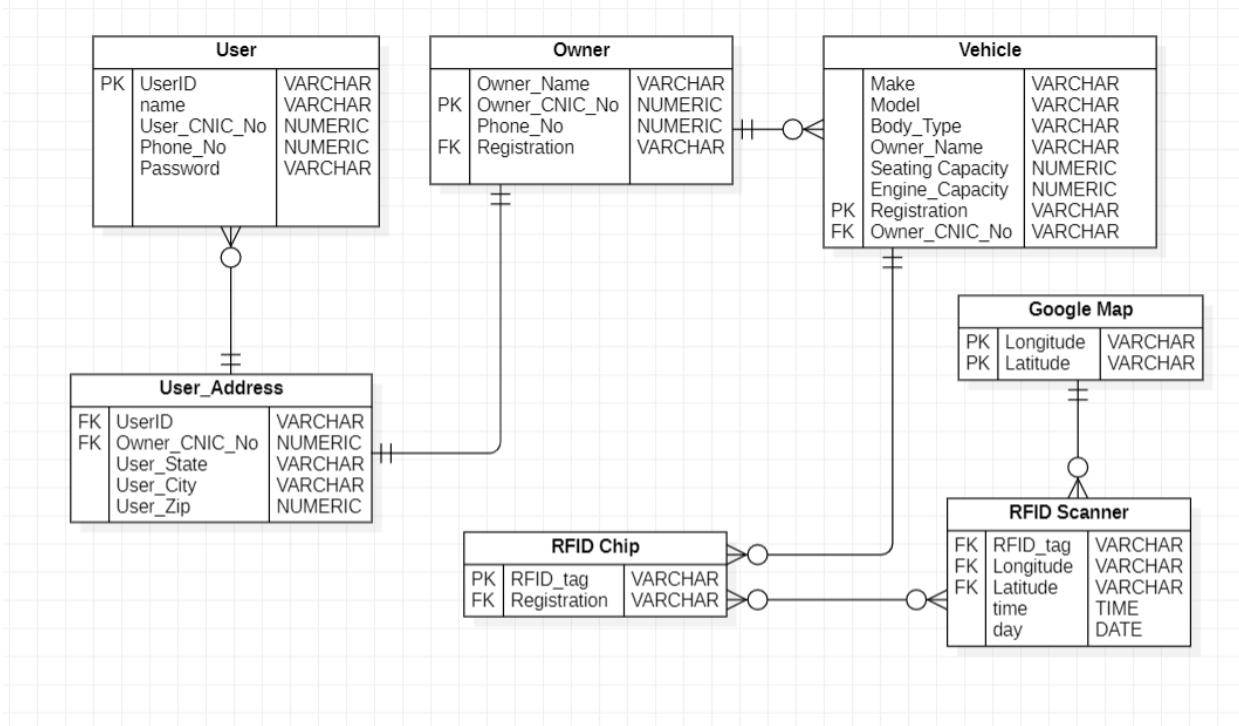
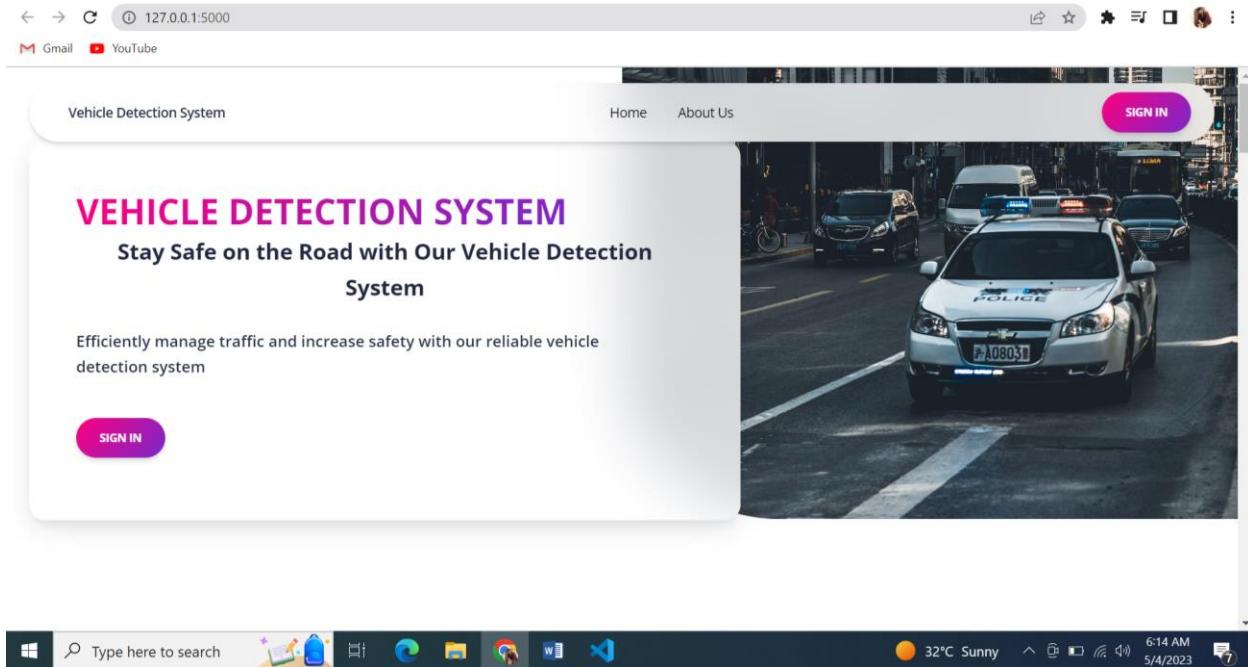


Figure 33 ER Diagram

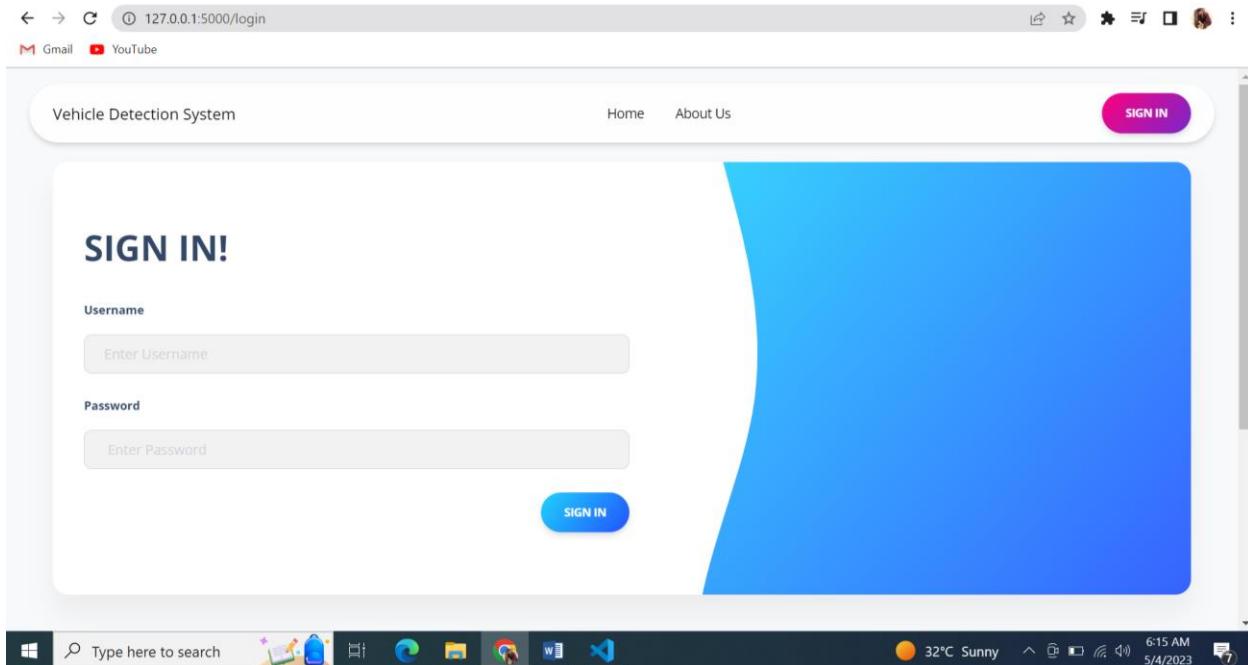
The ER Diagram shows that there will be multiple User using the system. Multiple User can have single address e.g., two brother living in the same house working with the same system. In the same way Multiple vehicles can have single owner and single owner will have single address.

One Vehicle can have multiple chips install in it. E.g., installing RFID chips into number plate as well as install in card dashboard. Multiple scanners can scan multiple RFID chip at the same time. Similarly Multiple Scanner can be installed at the same place means having the same Longitude and Latitude.

5.3 NON-WORKING PROTOTYPING



MAIN HOME PAGE



LOGIN PAGE

← → ⌛ ① 127.0.0.1:5000/admin_dashboard

Gmail YouTube

Vehicle Detection System Register Search Owner Search Vehicle Vehicle List Vehicle Location **Create Officer** SIGN OUT

WELCOME ADMIN

ID	Username	Email	Is Admin	Action	Action
1	adnansamad	adnansamad073@gmail.com	True	Edit	Delete
3	alisha	alishahameed12@gmail.com	True	Edit	Delete
4	Taha	tahaasaim19@gmail.com	True	Edit	Delete
5	zafarabas	zafarabas1@gmail.com	True	Edit	Delete
6	Aizahkhan	aizahkhanpunjaban@gmail.com	False	Edit	Delete
7	omer	omer.sidd93@gmail.com	False	Edit	Delete

Windows Type here to search 32°C Sunny 6:16 AM 5/4/2023

ADMIN DASHBOARD

← → ⌛ ① 127.0.0.1:5000/user_main

Gmail YouTube

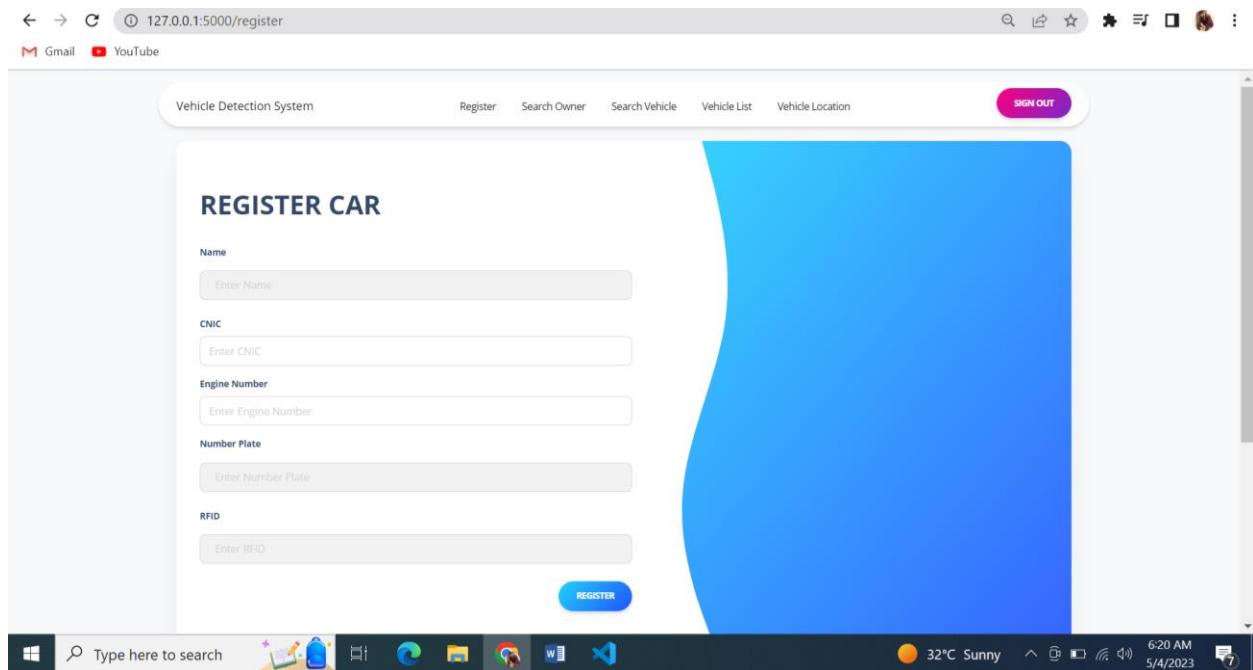
Vehicle Detection System Register Search Owner Search Vehicle Vehicle List Vehicle Location SIGN OUT

 EXCISE, TAXATION AND NARCOTICS CONTROL DEPARTMENT
GOVERNMENT OF SINDH

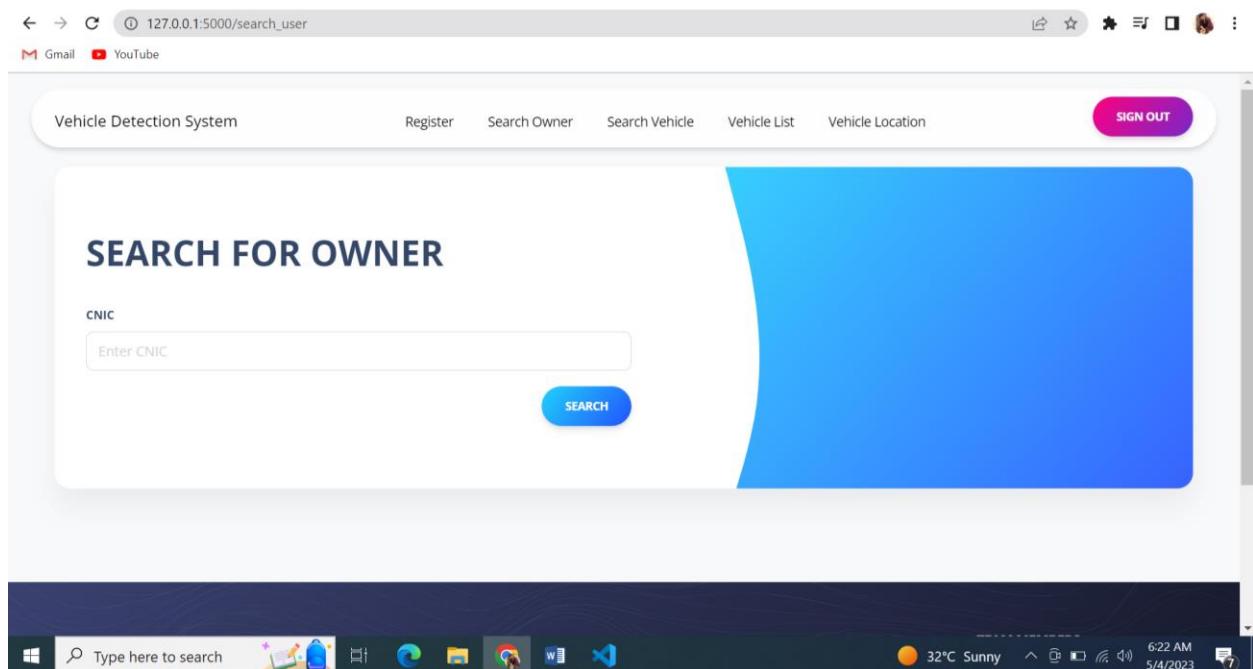
WELCOME TO VEHICLE DETECTION SYSTEM

Windows Type here to search 32°C Sunny 6:19 AM 5/4/2023

OFFICER DASHBOARD



REGISTER CAR PAGE



SEARCH PAGE FOR OWNER

OWNER INFORMATION

Name	CNIC	Engine Number	Number Plate	RFID
Adnan Samad	4220143196615	eng1122	BLN698	RF3

OWNER INFORMATION PAGE

SEARCH FOR VEHICLE

Number Plate

SEARCH

SEARCH PAGE VEHICLE

The screenshot shows a web browser window with the URL 127.0.0.1:5000/search. The page title is "Vehicle Detection System". The main content area is titled "RFID INFORMATION" and displays a table with the following data:

Number Plate	RFID tag	Latitude	Longitude	Timestamp
ABC123	rfid_001	24.9134	67.0309	67.0309

The browser toolbar at the top includes links for "Gmail" and "YouTube". The taskbar at the bottom shows various pinned icons and the system tray indicating a temperature of 32°C, sunny weather, and the date 5/4/2023.

RFID INFORMATION PAGE

The screenshot shows a web browser window with the URL 127.0.0.1:5000/vehicle_location. The page title is "Vehicle Detection System". The main content area displays a table with the following data:

Latitude	Longitude	Timestamp
24.8607	67.0011	2023-04-25 10:30:00
24.9134	67.0309	2023-04-25 12:00:00
25.0133	67.0232	2023-04-25 12:00:00

The browser toolbar at the top includes links for "Gmail" and "YouTube". The taskbar at the bottom shows various pinned icons and the system tray indicating a temperature of 32°C, sunny weather, and the date 5/4/2023.

VEHICLE LOCATION PAGE

Vehicle Detection System

Register Search Owner Search Vehicle Vehicle List Vehicle Location SIGN OUT

ALL VEHICLES

Name	CNIC	Engine Number	Number Plate	RFID
Muhammad Omer Siddiqui	42201-9017078-2	expy13	OMI123	rfid_004
adnan	42201-9017078-1	expy13	XYZ123	rfid_009
Taha	42201-9017078-3	expy14	ZHI-123	rfid_010
Taha Sharif	4220173163869	eng1112	AAC577	RF2
Adnan Samad	4220143196615	eng1122	BLN698	RF3

Type here to search 32°C Sunny 6:30 AM 5/4/2023

VEHICLE LIST PAGE

CHAPTER 6

ALGORITHM ANALYSIS AND COMPLEXITY

6.1 ALGORITHM

In our project we are using Reverse Geocoding because reverse geocoding is a process that takes geographic coordinates and uses them to find a descriptive address or place name. It is the opposite of forward geocoding, which attempts to find geographic coordinates based on an address or place name. Reverse geocoding is used in many applications, such as locating nearby amenities, finding nearby addresses and landmarks, and even pinpointing cell phone locations.

6.2 PSEUDOCODE

```
# Initialize RFID scanner and GPS module

rfid_scanner = RFIDScanner()

gps = GPS()

# Connect to database

db = Database()

while True:

    # Wait for a vehicle to pass by the scanner

    vehicle = rfid_scanner.wait_for_vehicle()

    # Get the registration number of the vehicle from RFID

    reg_num = vehicle.get_registration_number()

    # Get the current location of the vehicle from GPS

    location = gps.get_location()

    longitude = location.longitude

    latitude = location.latitude

    # Store the information in the database

    db.store_vehicle_data(reg_num, longitude, latitude)

    # To retrieve the information from the database

    def retrieve_vehicle_data(reg_num):

        data = db.get_vehicle_data(reg_num)

        return data
```

CHAPTER 7

IMPLEMENTATION

7.1 INTRODUCTION

In this chapter we will discuss the source code of our chapter. This chapter outlines the specific steps taken to create the system.

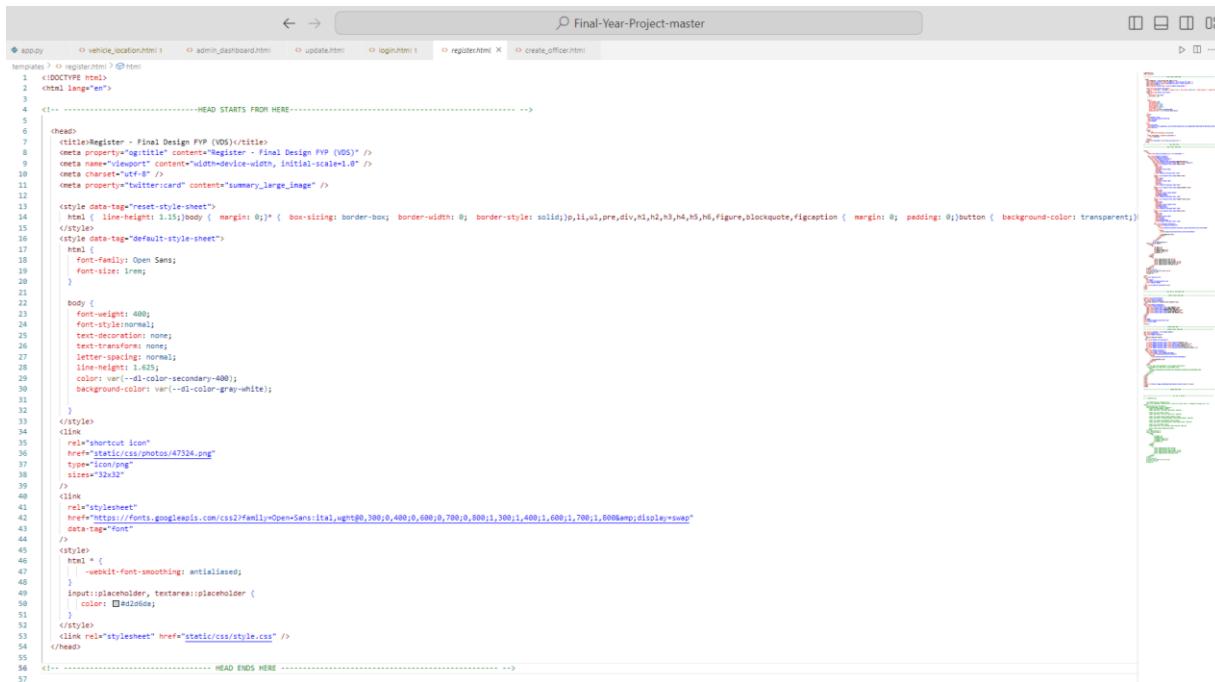
7.2 CODING

The Source code of our project is as follows.

7.2.1 HTML'S

7.2.1.1 BASE HTML

Base Html is created to give a common design to all of templates



```
app.py
templates > register.html > vehicle_location.html > admin_dashboard.html > update.html > login.html > register.html > create_officer.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta name="og:title" content="Register - Final Design FYP (VOS)" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta charset="utf-8" />
    <meta property="twitter:card" content="summary_large_image" />
    <style data-tag="reset-style-sheet">
        html { line-height: 1.15; margin: 0; } body { box-sizing: border-box; border-width: 0; border-style: solid; } p, li, ul, pre, div, h1, h2, h3, h4, h5, h6, figure, blockquote, figcaption { margin: 0; padding: 0; } button { background-color: transparent; }
    </style>
    <link rel="shortcut icon" href="static/css/photos/47324.png" type="image/png" sizes="32x32" />
    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,600;0,700;0,800;1,300;1,400;1,600;1,700;1,800&ampampdisplay=swap" data-tag="font" />
    <style>
        html {
            -webkit-font-smoothing: antialiased;
        }
        input::placeholder, textarea::placeholder {
            color: #A9A9A9;
        }
    </style>
    <link rel="stylesheet" href="static/css/style.css" />
</head>
<!-- ----- HEAD ENDS HERE ----- -->
```

HEAD OF ALL THE TEMPLATES

Final-Year-Project-master

```

templates > register.html > register.html
198 <!-- .....HEADER STRATS FROM HERE .....-->
199 <div data-role="Header" class="header-header">
200   <nav class="header-nav">
201     <div class="header-container">
202       <a href="#">Vehicle Detection System</a>
203     </div>
204     <div class="header-full-container">
205       <a class="header-navlink1 Large" href="/register">Register </a>
206       <a class="header-navlink1 Large" href="/search_user">Search Owner </a>
207       <a class="header-navlink1 Large" href="/search">Search Vehicle </a>
208       <a class="header-navlink1 Large" href="/all_vehicles">Vehicle List </a>
209       <a class="header-navlink1 Large" href="/vehicle_location">Vehicle Location </a>
210     </div>
211     <div class="header-container1">
212       <div class="header-container2">
213         <a href="logout" class="header-navlink6">
214           <div class="primary-pink-button-container">
215             <button class="primary-pink-button-button button ButtonSmall">
216               <span>SIGN OUT</span>
217             </button>
218           </div>
219         </a>
220       </div>
221     </div>
222     <div data-role="BurgerMenu" class="header-burger-menu">
223       <svg viewBox="0 0 1024 1024" class="header-icon">
224         <path d="M128 256h768v86h-768v-86zM128 554v-84h768v84h-768zM128 768v-86h768v86h-768z" />
225     </svg>
226   </div>
227   </div>
228   </nav>
229   </div>
230   </div>
231   </div>
232   </div>
233   </div>
234   </div>
235   </div>
236   </div>
237   </div>
238   </div>
239   </div>
240   <script src="https://unpkg.com@teleporta/teleport-custom-scripts"></script>
241 </body>
242 </html>
243
244 <!-- ..... HEADER ENDS HERE .....-->

```

HEADER OF ALL THE TEMPLATES

```

171 <!-- .....FOOTER STARTS FROM HERE .....-->
172
173 <div class="footer-footer">
174   <div class="footer-container">
175     <div class="footer-container1">
176       | <span>This project is created by UIT Students</span>
177     </div>
178     <div class="footer-container2">
179       <div class="footer-container3">
180         <span class="footer-text1 Large">TEAM MEMBERS</span>
181         <span class="footer-text2 Large">ADNAN SAMD</span>
182         <span class="footer-text3">MUHAMMAD OMER SIDDIQUI</span>
183         <span class="footer-text4 Large">ALISHA HANEED</span>
184         <span class="footer-text5 Large">HAFIZ TAHA SHARIF</span>
185       </div>
186     </div>
187   </div>
188   <div>
189     
190   </div>
191 </div>
192 </div>
193 <!-- .....FOOTER ENDS HERE .....-->
194
195
196
197

```

FOOTER OF ALL THE TEMPLATES

7.2.1.2 LOGIN HTML

```
Final-Year-Project-master
app.py vehicle_location.html 1 admin_dashboard.html update.html login.html 1 register.html create_officer.html ...
```

templates > login.html > html > body > div

```
56
57
58 <!------- BODY STARTS FROM HERE ----->
59
60 <body>
61   <script type="text/javascript" src="{{ url_for('static', filename='script.js') }}"/></script>
62   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}"/>
63   <div>
64     <link href="static/css/sign-in.css" rel="stylesheet" />
65
66   <div class="sign-in-container">
67     <div class="sign-in-contact">
68       <div class="sign-in-container1">
69         <div class="sign-in-form1">
70           <h2 class="sign-in-text HeadingOne">SIGN IN!</h2>
71           {% with messages = get_flashed_messages() %}
72             {% if messages %}
73               <div class="alert alert-dismissible alert-danger">
74                 <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
75                 {% for message in messages %}
76                   <p>{{ message }}</p>
77                 {% endfor %}
78               {% endif %}
79             {% endifwith %}
80             <form class="sign-in-form1" method="POST" action="{{ url_for('login') }}">
81               <label class="sign-in-text1 Label">Username</label>
82               <input
83                 type="text"
84                 required=""
85                 placeholder="Enter Username"
86                 id="username"
87                 name="username"
88                 class="sign-in-textinput Small input"
89               />
90               <label class="sign-in-text2 Label">Password</label>
91               <input
92                 type="password"
93                 required=""
94                 placeholder=" Enter Password"
95                 id="password"
96                 name="password"
97                 class="sign-in-textinput Small input"
98               />
99             <div class="sign-in-container2">
100               <div class="sign-in-container3">
101                 <a href="user_main" class="sign-in-navlink">
102                   <div
103                     class="primary-blue-button-container primary-blue-button-root-class-name"
104                   >
105                     <button
106                       class="primary-blue-button-button button ButtonSmall"
107                     >
108                       <span>SIGN IN</span>
109                     </button>
110                   </div>
111                 </a>
112               </div>
113             </div>
114           </div>
115         </div>
116       </div>
117     </div>
118     <div class="sign-in-info">
119       
124       <div class="sign-in-container4"></div>
125     </div>
126   </div>
127 </div>
128 <!------- BODY ENDS HERE ----->
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
220
```

7.2.1.3 ADMIN DASHBOARD HTML

```
Final-Year-Project-master
app.py      vehicle_location.html 1  admin_dashboard.html ●  update.html  login.html 1  register.html  create_officer.html
templates > admin_dashboard.html > html > body > div > div.register-container > div.register-contact > div.register-container1 > div.register-form
58  <!-- ----- BODY STRATS FROM HERE ----->
59
60
61  <body>
62    <div>
63      <link href="static/css/register.css" rel="stylesheet" />
64
65      <div class="register-container">
66        <div class="register-contact">
67          <div class="register-container1">
68            <div class="register-form">
69              <h2 class="register-text HeadingOne">WELCOME ADMIN</h2>
70              <form class="register-form1" method="POST" action="{{ url_for('admin_dashboard') }}">
71                <table id="data-table">
72                  <thead>
73                    <tr>
74                      <th>ID</th>
75                      <th>Username</th>
76                      <th>Email</th>
77                      <th>Is Admin</th>
78                      <th>Action</th>
79                      <th>Action</th>
80                  </tr>
81                </thead>
82                <tbody>
83                  {% for user in users %}
84                  <tr>
85                    <td>{{ user.id }}</td>
86                    <td>{{ user.username }}</td>
87                    <td>{{ user.email }}</td>
88                    <td>{{ user.is_admin }}</td>
89                    <td>
90                      <a href="{{ url_for('update', id=user.id) }}">Edit</a>
91                    </td>
92                    <td>
93                      <form action="{{ url_for('delete', id=user.id) }}" method="post">
94                        <input type="hidden" name="_method" value="DELETE">
95                        <button type="submit">Delete</button>
96                      </form>
97                    </td>
98                  </tr>
99                  {% endfor %}
100                </tbody>
101              </table>
102            </form>
103
104        </div>
105        <div class="register-info">
106          
111          <div class="register-container4"></div>
112        </div>
113      </div>
114    </div>
115  </div>
116
117 <!-- ----- BODY ENDS HERE ----->
-->
```

7.2.1.4 ADMIN UPDATE INFORMATION HTML

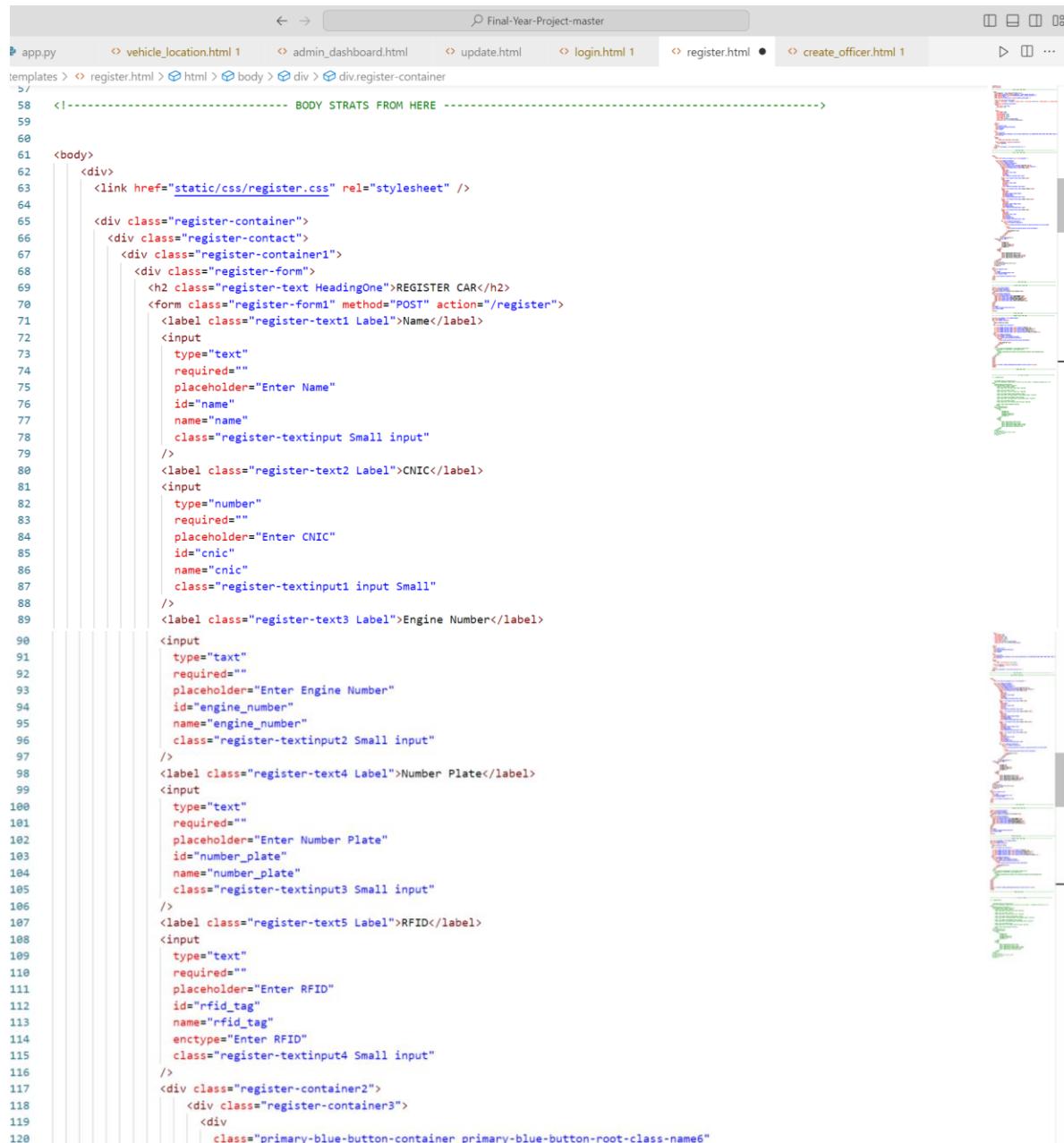
```
Final-Year-Project-master
app.py      vehicle_location.html 1  admin_dashboard.html  update.html  login.html  register.html  create_officer.html  ...
templates > update.html > html > body > div.register-container > div.register-contact > div.register-container1
57
58  <!-- ----- BODY STRATS FROM HERE ----->
59
60
61  <body>
62    <div>
63      <script type="text/javascript" src="{{ url_for('static', filename='script.js') }}></script>
64      <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}>
65
66    </div >
67    <div class="register-container">
68      <div class="register-contact">
69        <div class="register-container1">
70          <div class="register-form">
71            <h2 class="register-text HeadingOne">UPDATE INFO</h2>
72            <form class="register-form1" method="POST">
73              <label class="sign-in-text1 Label">Username</label>
74              <input
75                type="text"
76                required=""
77                placeholder="Enter Username"
78                value="{{ user.username }}"
79                name="username"
80                class="sign-in-textinput Small input"
81              />
82              <label class="sign-in-text2 Label">Password</label>
83              <input
84                type="password"
85                required=""
86                placeholder=" Enter Password"
87                value="{{ user.password }}"
88                name="password"
89                class="sign-in-textinput Small input"
90            />
91            <label class="sign-in-text2 Label">Email</label>
92            <input
93              type="email"
94              required=""
95              placeholder=" Enter Email"
96              id="email"
97              value="{{ user.email }}"
98              name="email"
99              class="sign-in-textinput Small input"
100        />
101        <input type="checkbox" name="is_admin" {% if user.is_admin %}checked{% endif %}
102          required=""
103          class="sign-in-textinput Small input"
104        >
105        <div class="register-container2">
106          <div class="register-container3">
107            <div
108              class="primary-blue-button-container primary-blue-button-root-class-name6"
109            >
110              <button
111                class="primary-blue-button-button button ButtonSmall"
112              >
113                <span>UPDATE</span>
114              </button>
115            </div>
116          </div>
117        </div>
118      </form>
119    </div>
120    <div class="register-info">
121      
126      <div class="register-container4"></div>
127    </div>
128  </div>
129</div>
130  </div>
131</div>
132<!-- ----- BODY ENDS HERE ----->
```

7.2.1.5 CREATE OFFICER HTML

```
Final-Year-Project-master
app.py    vehicle_location.html 1    admin_dashboard.html    update.html    login.html    register.html    create_officer.html 1

templates > create_officer.html > html
58  <!----- BODY STARTS FROM HERE ----->
59
60  <body>
61      <script type="text/javascript" src="{{ url_for('static', filename='script.js') }}></script>
62      <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}>
63      <div>
64          <link href="static/css/sign-in.css" rel="stylesheet" />
65
66          <div class="sign-in-container">
67              <div class="sign-in-contact">
68                  <div class="sign-in-container1">
69                      <div class="sign-in-form">
70                          <h2 class="sign-in-text HeadingOne">REGISTER SYSTEM USER!</h2>
71                          {% with messages = get_flashed_messages() %}
72                          {% if messages %}
73                              <div class="alert alert-dismissible" %if 'success' in messages[0] %alert-success% else %alert-danger% endif %>
74                                  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
75                                  {% for message in messages %}
76                                      <p>{{ message }}</p>
77                                  {% endfor %}
78                          {% endif %}
79                          {% endif %}
80                          {% endwith %}
81
82
83
84
85          <form class="sign-in-form1" method="POST" action="{{ url_for('create_officer') }}>
86              <label class="sign-in-text1 Label">Username</label>
87              <input
88                  type="text"
89                  required=""
90
91                  placeholder="Enter Username"
92                  id="username"
93                  name="username"
94                  class="sign-in-textinput Small input"
95
96              <label class="sign-in-text2 Label">Password</label>
97              <input
98                  type="password"
99                  required=""
100                 placeholder=" Enter Password"
101                 id="password"
102                 name="password"
103                 class="sign-in-textinput1 Small input"
104
105             <label class="sign-in-text2 Label">Email</label>
106             <input
107                 type="email"
108                 required=""
109                 placeholder=" Enter Email"
110                 id="email"
111                 name="email"
112                 class="sign-in-textinput1 Small input"
113
114         <div class="sign-in-container2">
115             <div class="sign-in-container3">
116                 <a href="main-admin-page.html" class="sign-in-navlink">
117                     <div
118                         class="primary-blue-button-container primary-blue-button-root-class-name"
119                     >
120                         <button
121                             class="primary-blue-button-button button ButtonSmall"
122                         >
123                             <span>SIGN IN</span>
124                         </button>
125                     </div>
126                 </a>
127             </div>
128         </div>
129     <div class="sign-in-info">
130         
135     <div class="sign-in-container4"></div>
136     </div>
137     </div>
138     </div>
139     </div>
140
141 <!----- BODY ENDS HERE ----->
```

7.2.1.6 REGISTER CAR HTML



```
58 <!------- BODY STRATS FROM HERE ----->
59
60
61 <body>
62     <div>
63         <link href="static/css/register.css" rel="stylesheet" />
64
65         <div class="register-container">
66             <div class="register-contact">
67                 <div class="register-container1">
68                     <div class="register-form">
69                         <h2 class="register-text HeadingOne">REGISTER CAR</h2>
70                         <form class="register-form1" method="POST" action="/register">
71                             <label class="register-text1 Label">Name</label>
72                             <input
73                                 type="text"
74                                 required=""
75                                 placeholder="Enter Name"
76                                 id="name"
77                                 name="name"
78                                 class="register-textinput Small input"
79                             />
80                             <label class="register-text2 Label">CNIC</label>
81                             <input
82                                 type="number"
83                                 required=""
84                                 placeholder="Enter CNIC"
85                                 id="cnic"
86                                 name="cnic"
87                                 class="register-textinput1 input Small"
88                             />
89                             <label class="register-text3 Label">Engine Number</label>
90                             <input
91                                 type="text"
92                                 required=""
93                                 placeholder="Enter Engine Number"
94                                 id="engine_number"
95                                 name="engine_number"
96                                 class="register-textinput2 Small input"
97                             />
98                             <label class="register-text4 Label">Number Plate</label>
99                             <input
100                                type="text"
101                                required=""
102                                placeholder="Enter Number Plate"
103                                id="number_plate"
104                                name="number_plate"
105                                class="register-textinput3 Small input"
106                            />
107                            <label class="register-text5 Label">RFID</label>
108                            <input
109                                type="text"
110                                required=""
111                                placeholder="Enter RFID"
112                                id="rfid_tag"
113                                name="rfid_tag"
114                                encType="Enter RFID"
115                                class="register-textinput4 Small input"
116                            />
117                            <div class="register-container2">
118                                <div class="register-container3">
119                                    <div
120                                        | class="primary-blue-button-container primary-blue-button-root-class-name6"
```

```

121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169 <!-- -----BODY ENDS HERE ----->

```

7.2.1.7 SEARCH OWNER HTML

```
app.py          vehicle_location.html 1  search_user.html ●
templates > < search_user.html > html > head > style
56
57     <!-- ----- BODY STRATS FROM HERE ----->
58
59     <body>
60         <div>
61             <link href="static/css/register.css" rel="stylesheet" />
62
63             <div class="register-container">
64                 <div class="register-contact">
65                     <div class="register-container1">
66                         <div class="register-form">
67                             <h2 class="register-text HeadingOne">SEARCH FOR OWNER </h2>
68
69                             {% if error %}
70                             <p style="color: red">{{ error }}</p>
71                             {% endif %}
72
73                         <form class="register-form1" id="search-form" method="POST" action="{{ url_for('search_user') }}">
74                             <label class="register-text1 Label">CNIC</label>
75                             <input
76                                 type="number"
77                                 required=""
78                                 placeholder="Enter CNIC"
79                                 id="cnic"
80                                 name="cnic"
81                                 class="register-textinput Small input"
82                             />
83                             <div class="register-container2">
84                                 <div class="register-container3">
85                                     <div
86                                         class="primary-blue-button-container primary-blue-button-root-class-name6"
87                                     >
88                                         <button
89                                             class="primary-blue-button-button button ButtonSmall"
90                                         >
91                                             <span>SEARCH</span>
92                                         </button>
93                                     </div>
94                                 </div>
95                             </div>
96                         </form>
97
98
99                         {% if registrations %}
100                         <table id="data-table">
101                             <thead>
102                                 <tr>
103                                     <th>Name</th>
104                                     <th>CNIC</th>
105                                     <th>Engine Number</th>
106                                     <th>Number Plate</th>
107                                     <th>RFID</th>
108                                 </tr>
109                             </thead>
110                             <tbody>
111                                 <tr>
112                                     <td>{{ registrations.name }}</td>
113                                     <td>{{ registrations.cnic }}</td>
114                                     <td>{{ registrations.engine_number }}</td>
115                                     <td>{{ registrations.number_plate }}</td>
116                                     <td>{{ registrations.rfid_tag }}</td>
117                                 </tr>
118                             </tbody>
119                         </table>
120                         {% elif error %}
121                         {% endif %}
122
123                     </div>
124                     <div class="register-info">
125                         
130                         <div class="register-container4"></div>
131                     </div>
132                 </div>
133             </div>
134
135             <!-- ----- BODY ENDS HERE ----->
```

7.2.1.8 OWNER INFORMATION HTML

```
Final-Year-Project-master
app.py vehicle_location.html 1 view_user.html
templates > view_user.html > html > body > div > div.register-container > div.header-header > nav.header-nav > div.header-container > div.header-container1 > div.header-cont
58
59     <!-- ----- BODY STRATS FROM HERE ----->
60
61
62     <body>
63         <div>
64             <link href="static/css/register.css" rel="stylesheet" />
65
66             <div class="register-container">
67                 <div class="register-container1">
68                     <div class="register-form">
69                         <h2 class="register-text HeadingOne">OWNER INFORMATION</h2>
70                         <div class="search-user-container5">
71                         <div class="search-user-container6">
72
73                         <table>
74                             <thead>
75                                 <tr>
76                                     <th>Name</th>
77                                     <th>CNIC</th>
78                                     <th>Engine Number</th>
79                                     <th>Number Plate</th>
80                                     <th>RFID</th>
81
82                             </tr>
83                         </thead>
84                         <tbody>
85                             <tr>
86                                 <td>{{ registrations.name }}</td>
87                                 <td>{{ registrations.cnic }}</td>
88                                 <td>{{ registrations.engine_number }}</td>
89                                 <td>{{ registrations.number_plate }}</td>
90                                 <td>{{ registrations.rfid_tag }}</td>
91
92                         </tbody>
93                     </div>
94                 </div>
95             </div>
96             <div class="register-info">
97                 
102                 <div class="register-container4"></div>
103             </div>
104         </div>
105     </div>
106
107     <!-- -----BOX AND ITS INFO ENDS HERE ----->
```

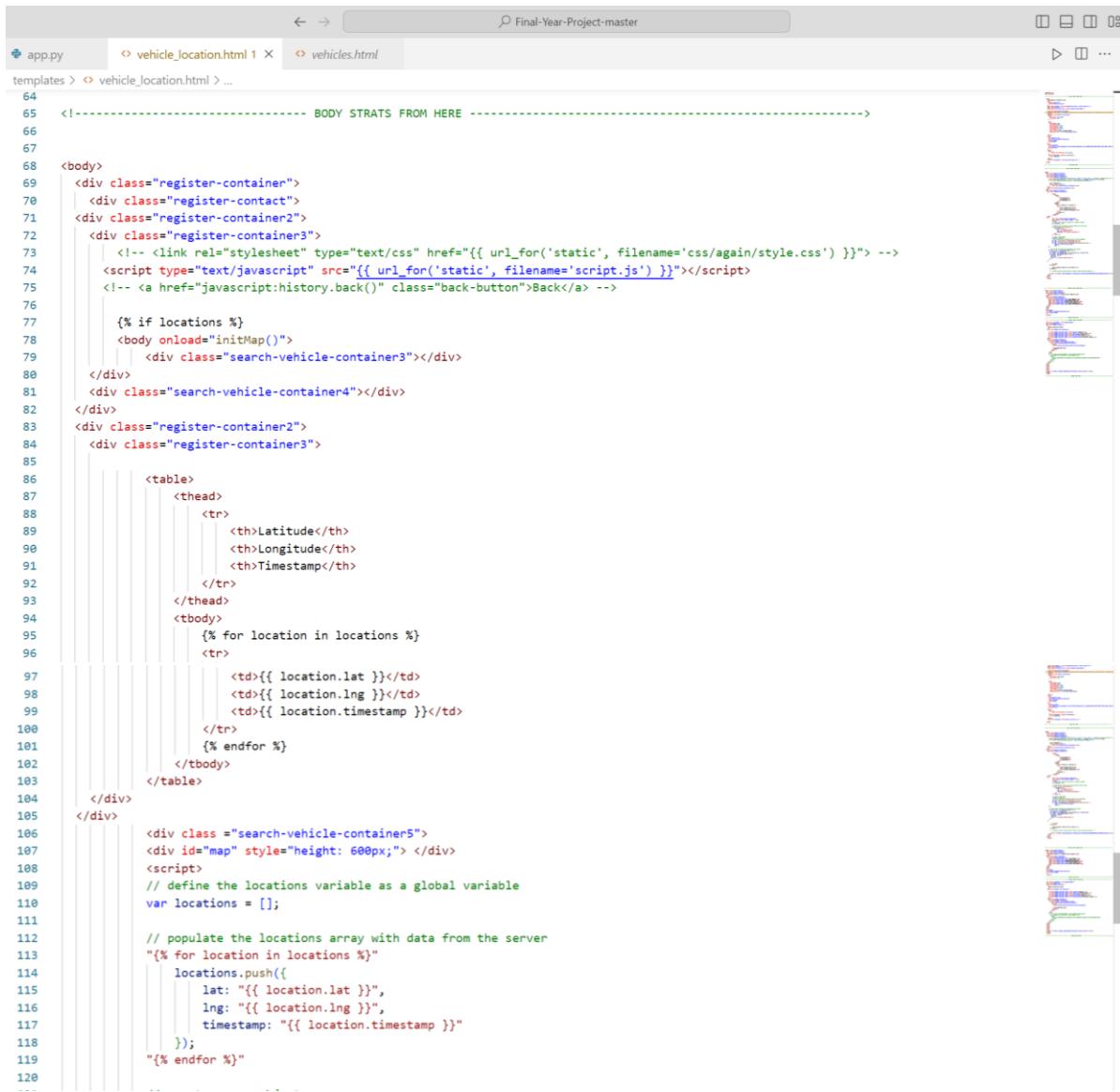
7.2.1.9 SEARCH VEHICLE HTML

```
Final-Year-Project-master
app.py vehicle_location.html 1 search.html
templates > search.html > html > body > div > div.register-container > div.header-header > nav.header-nav > div.header-container > div.header-container1 > div.header-container2 >
58 <!-- ----- BODY STRATS FROM HERE ----->
59
60
61 <body>
62   <script type="text/javascript" src="{{ _url_for('static', filename='script.js') }}></script>
63   <div>
64     <link href="static/css/register.css" rel="stylesheet" />
65
66   <div class="register-container">
67     <div class="register-contact">
68       <div class="register-container1">
69         <div class="register-form">
70           <h2 class="register-text HeadingOne">SEARCH FOR VEHICLE </h2>
71
72           {% if error %}
73             <p style="color: red">{{ error }}</p>
74           {% endif %}
75           <form method="POST" class="register-form1">
76             <label class="register-text1 Label" for="number_plate">Number Plate</label>
77             <input
78               type="text"
79               required=""
80               id="number_plate"
81               name="number_plate"
82               placeholder="Enter Number Plate"
83               class="register-textinput Small input"
84             />
85             <div class="register-container2">
86               <div
87                 class="primary-blue-button-container primary-blue-button-root-class-name6"
88               >
89                 <button type="submit" class="primary-blue-button-button button ButtonSmall">
90                   Search
91                 </button>
92               </div>
93             </div>
94
95           </form>
96
97
98           {% if registrations %}
99             <table id="data-table">
100               <thead>
101                 <tr>
102                   <th>Name</th>
103                   <th>CNIC</th>
104                   <th>Engine Number</th>
105                   <th>Number Plate</th>
106                   <th>RFID</th>
107                 </tr>
108               </thead>
109               <tbody>
110                 <tr>
111                   <td>{{ registrations.name }}</td>
112                   <td>{{ registrations.cnic }}</td>
113                   <td>{{ registrations.engine_number }}</td>
114                   <td>{{ registrations.number_plate }}</td>
115                   <td>{{ registrations.rfid_tag }}</td>
116                 </tr>
117               </tbody>
118             </table>
119             {% elif error %}
120             {% endif %}
121
122           </div>
123           <div class="register-info">
124             
129             <div class="register-container4"></div>
130           </div>
131         </div>
132       </div>
133     </div>
134   <!-- ----- BODY ENDS HERE ----->
```

7.2.1.10 RFID INFORMATION HTML

```
Final-Year-Project-master
app.py      vehicle_location.html  view_rfid.html
templates > view_rfid.html > html > body > div > div.register-container
57
58  <!-- ----- BODY STARTS FROM HERE ----->
59
60  <body>
61    <link href="static/css/register.css" rel="stylesheet" />
62
63  <div class="register-container">
64    <div class="register-contact">
65      <div class="register-container1">
66        <div class="register-form">
67          <h2 class="register-text HeadingOne">RFID INFORMATION</h2>
68          <table>
69            <thead>
70              <tr>
71                <th>Number Plate</th>
72                <th>RFID tag</th>
73                <th>Latitude</th>
74                <th>Longitude</th>
75                <th>Timestamp</th>
76              </tr>
77            </thead>
78            <tbody>
79              <tr>
80                <td>{{ vehicle.number_plate }}</td>
81                <td>{{ vehicle.rfid_tag }}</td>
82                <td>{{ location.lat }}</td>
83                <td>{{ location.lng }}</td>
84                <td>{{ location.lng }}</td>
85              </tr>
86            </tbody>
87          </table>
88
89  {% if location %}
90
91
92  {% else %}
93    <p>No location data available for this vehicle.</p>
94  {% endif %}
95  <!-- <a href="{{ url_for('search') }}">Back to search</a> -->
96
97
98  {% if error %}
99    <!-- <p style="color:red">{{ error }}</p> -->
100   <p>{{ error }}</p>
101  {% endif %}
102
103
104
105
106  </div>
107  |  <div class="register-info">
108  |    
113  |    <div class="register-container4"></div>
114  |  </div>
115  |  </div>
116  </div>
117
118  <!-- ----- BOX AND ITS INFO ENDS HERE ----->
```

7.2.1.11 VEHICLE LOCATION HTML



The screenshot shows a code editor interface with two tabs open: 'vehicle_location.html' and 'vehicles.html'. The 'vehicle_location.html' tab is active, displaying the following code:

```
64 65  <!-- BODY STRATS FROM HERE -->
66
67
68  <body>
69    <div class="register-container">
70      <div class="register-contact">
71      <div class="register-container2">
72        <div class="register-container3">
73          <!-- <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/again/style.css') }}" -->
74          <script type="text/javascript" src="{{ url_for('static', filenames='script.js') }}">></script>
75          <!-- <a href="javascript:history.back()" class="back-button">Back</a> -->
76
77          {% if locations %}
78            <body onload="initMap()">
79              <div class="search-vehicle-container3"></div>
80            </div>
81            <div class="search-vehicle-container4"></div>
82          </div>
83          <div class="register-container2">
84            <div class="register-container3">
85
86              <table>
87                <thead>
88                  <tr>
89                    <th>Latitude</th>
90                    <th>Longitude</th>
91                    <th>Timestamp</th>
92                  </tr>
93                </thead>
94                <tbody>
95                  {% for location in locations %}
96                  <tr>
97                    <td>{{ location.lat }}</td>
98                    <td>{{ location.lng }}</td>
99                    <td>{{ location.timestamp }}</td>
100                  </tr>
101                  {% endfor %}
102                </tbody>
103              </table>
104            </div>
105          </div>
106          <div class="search-vehicle-container5">
107            <div id="map" style="height: 600px;"></div>
108            <script>
109              // define the locations variable as a global variable
110              var locations = [];
111
112              // populate the locations array with data from the server
113              "% for location in locations %"
114              locations.push({
115                lat: "{{ location.lat }}",
116                lng: "{{ location.lng }}",
117                timestamp: "{{ location.timestamp }}"
118              });
119            "% endfor %"
120          </div>
```

```

121     // create a map object
122     function initMap() {
123         // get latitude and longitude values from the database
124         var lat = parseFloat("{{ locations[0].lat }}");
125         var lng = parseFloat("{{ locations[0].lng }}");
126         var map = new google.maps.Map(document.getElementById('map'), {
127             center: { lat: lat, lng: lng },
128             zoom: 12
129         });
130         // loop through locations and add markers to the map
131         for (var i = 0; i < locations.length; i++) {
132             var location = locations[i];
133             var latlng = new google.maps.LatLng(location.lat, location.lng);
134             var marker = new google.maps.Marker({
135                 position: latlng,
136                 map: map,
137                 title: '{{ locations.number_plate }}'
138             });
139         }
140     }
141     </script>
142     {% else %}
143         <p>No locations found for this vehicle.</p>
144     {% endif %}
145     <!-- <button onclick="location.href='/search'">Search Vehicle</button> -->
146     <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCcREyA3Z0ALYAaciq1043Fya3-2dSL1E"></script>
147     </div>
148     </div>
149     </div>
150     </body>
151
152 <!-- -----BODY ENDS HERE ----->

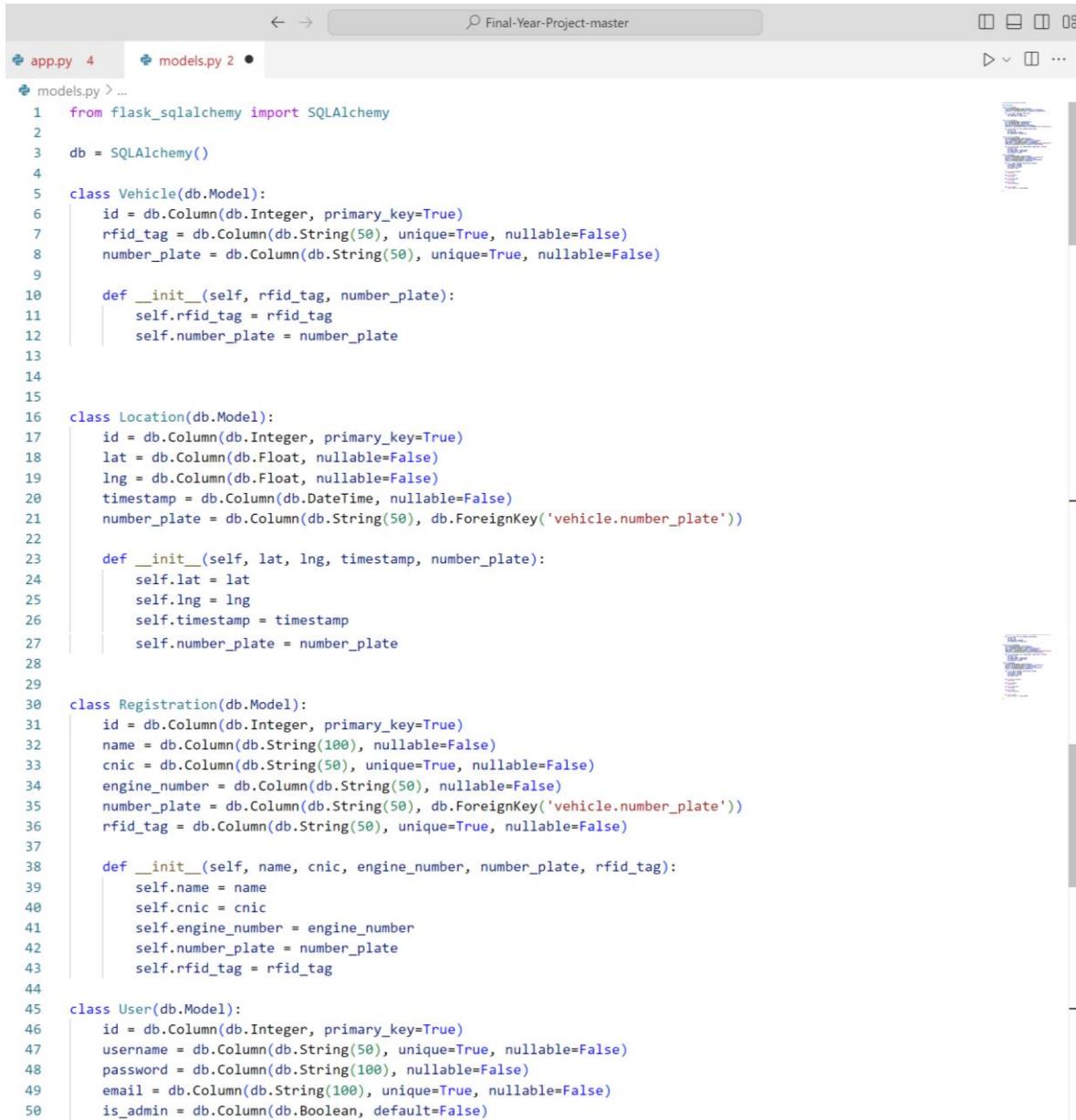
```



7.2.1.12 VEHICLE LIST HTML

```
Final-Year-Project-master
app.py      vehicle_location.html 1  vehicles.html
templates > vehicles.html > html > head > style > input:placeholder > textarea:placeholder
59  <!-- ----- BODY STRATS FROM HERE ----->
60
61
62  <body>
63    <div>
64      <link href="static/css/register.css" rel="stylesheet" />
65
66      <div class="register-container">
67        <div class="register-contact">
68          <div class="register-form">
69            <h2 class="register-text HeadingOne">ALL VEHICLES</h2>
70            <table class="table">
71              <thead>
72                <tr>
73                  <th>Name</th>
74                  <th>CNIC</th>
75                  <th>Engine Number</th>
76                  <th>Number Plate</th>
77                  <th>RFID</th>
78                </tr>
79              </thead>
80              <tbody>
81                {% for vehicle in vehicles %}
82                  <tr>
83                    <td>{{ vehicle.name }}</td>
84                    <td>{{ vehicle.cnic }}</td>
85                    <td>{{ vehicle.engine_number }}</td>
86                    <td>{{ vehicle.number_plate }}</td>
87                    <td>{{ vehicle.rfid_tag }}</td>
88                  </tr>
89                {% endfor %}
90              </tbody>
91            </table>
92
93
94
95        </div>
96        <div class="register-info">
97          
102        <div class="register-container4"></div>
103      </div>
104    </div>
105  </div>
106
107 <!-- ----- BOX AND ITS INFO ENDS HERE ----->
```

7.2.2 MODELS



The screenshot shows a code editor window with two tabs: 'app.py' and 'models.py'. The 'models.py' tab is active, displaying Python code for SQLAlchemy models. The code defines four classes: Vehicle, Location, Registration, and User. Each class has attributes like id, rfid_tag, number_plate, lat, lng, timestamp, name, cnic, engine_number, and is_admin, with various constraints like primary_key=True, unique=True, and nullable=False. The code uses db.Column and db.ForeignKey from flask_sqlalchemy.

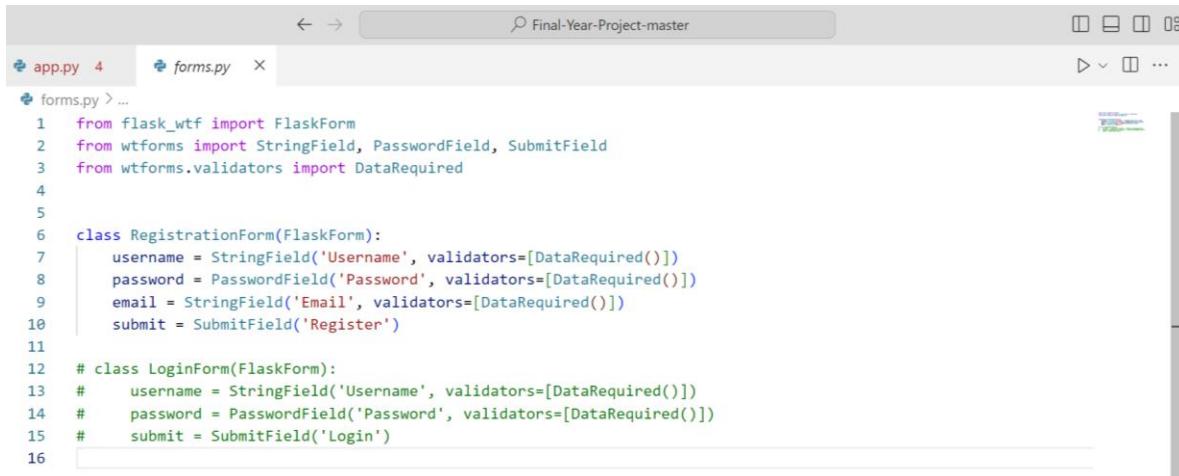
```
from flask_sqlalchemy import SQLAlchemy
db = SQLAlchemy()
class Vehicle(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    rfid_tag = db.Column(db.String(50), unique=True, nullable=False)
    number_plate = db.Column(db.String(50), unique=True, nullable=False)
def __init__(self, rfid_tag, number_plate):
    self.rfid_tag = rfid_tag
    self.number_plate = number_plate
class Location(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    lat = db.Column(db.Float, nullable=False)
    lng = db.Column(db.Float, nullable=False)
    timestamp = db.Column(db.DateTime, nullable=False)
    number_plate = db.Column(db.String(50), db.ForeignKey('vehicle.number_plate'))
def __init__(self, lat, lng, timestamp, number_plate):
    self.lat = lat
    self.lng = lng
    self.timestamp = timestamp
    self.number_plate = number_plate
class Registration(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    cnic = db.Column(db.String(50), unique=True, nullable=False)
    engine_number = db.Column(db.String(50), nullable=False)
    number_plate = db.Column(db.String(50), db.ForeignKey('vehicle.number_plate'))
    rfid_tag = db.Column(db.String(50), unique=True, nullable=False)
def __init__(self, name, cnic, engine_number, number_plate, rfid_tag):
    self.name = name
    self.cnic = cnic
    self.engine_number = engine_number
    self.number_plate = number_plate
    self.rfid_tag = rfid_tag
class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(50), unique=True, nullable=False)
    password = db.Column(db.String(100), nullable=False)
    email = db.Column(db.String(100), unique=True, nullable=False)
    is_admin = db.Column(db.Boolean, default=False)
```

```

51
52     def __init__(self, username, password, email, is_admin):
53         self.username = username
54         self.password = password
55         self.is_admin = is_admin
56         self.email = email
57
58
59     def is_authenticated(self):
60         return True
61
62     def is_active(self):
63         return True
64
65     def is_anonymous(self):
66         return False
67
68     def get_id(self):
69         return str(self.id)
70
71
72     def __repr__(self):
73         return '<User %r>' % self.username

```

7.2.3 FORMS

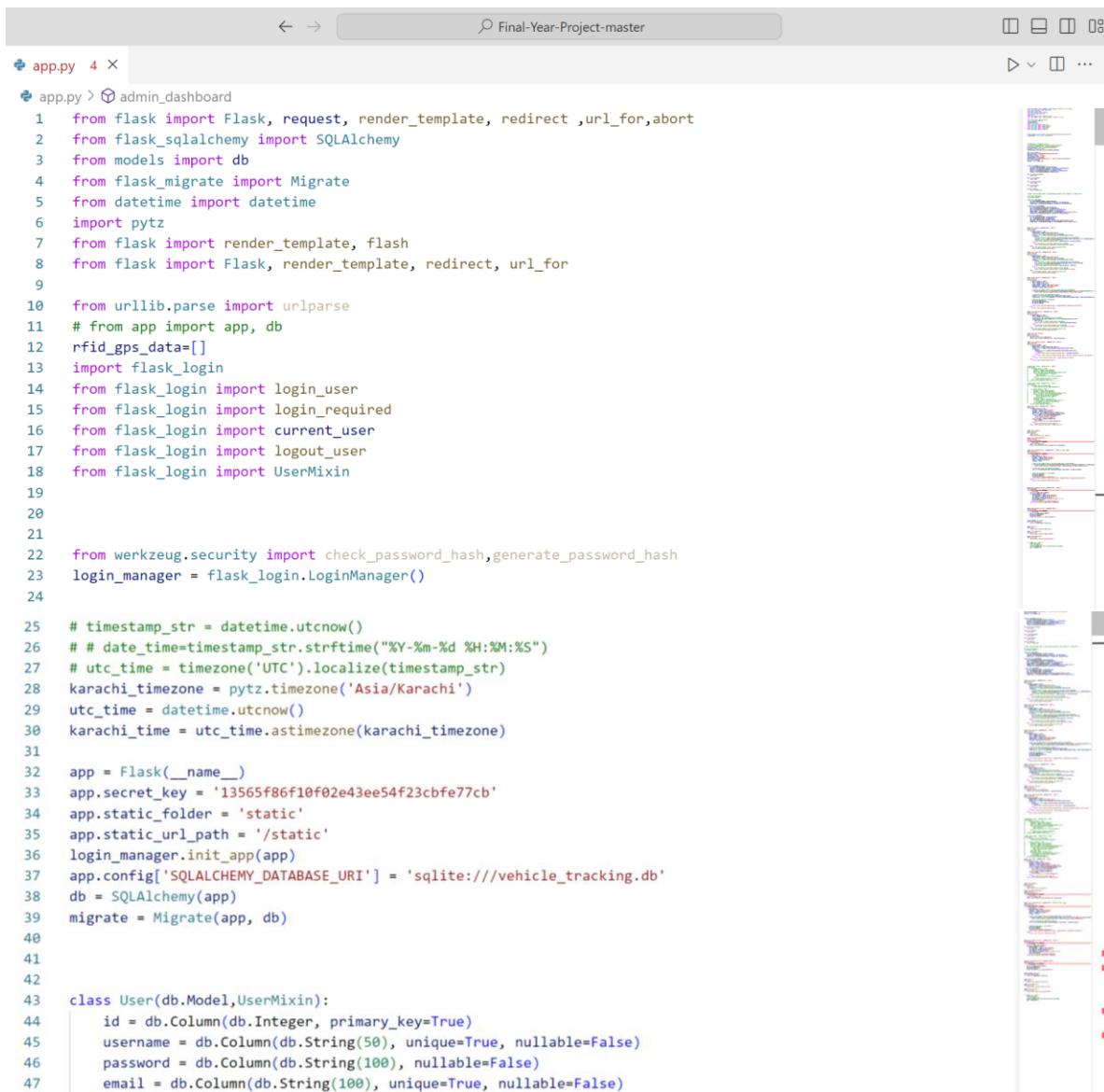


```

app.py 4 forms.py X
forms.py > ...
1  from flask_wtf import FlaskForm
2  from wtforms import StringField, PasswordField, SubmitField
3  from wtforms.validators import DataRequired
4
5
6  class RegistrationForm(FlaskForm):
7      username = StringField('Username', validators=[DataRequired()])
8      password = PasswordField('Password', validators=[DataRequired()])
9      email = StringField('Email', validators=[DataRequired()])
10     submit = SubmitField('Register')
11
12 # class LoginForm(FlaskForm):
13 #     username = StringField('Username', validators=[DataRequired()])
14 #     password = PasswordField('Password', validators=[DataRequired()])
15 #     submit = SubmitField('Login')
16

```

7.2.4 APP



The screenshot shows a code editor window with the file 'app.py' open. The code is written in Python and defines a Flask application for vehicle tracking. It includes imports for Flask, SQLAlchemy, and various Flask extensions like Flask-Migrate and Flask-Login. The code sets up the application, configures the database, and defines a User model with fields for id, username, password, and email. The code is annotated with line numbers from 1 to 47.

```
app.py 4 ×
app.py > admin_dashboard
1  from flask import Flask, request, render_template, redirect ,url_for,abort
2  from flask_sqlalchemy import SQLAlchemy
3  from models import db
4  from flask_migrate import Migrate
5  from datetime import datetime
6  import pytz
7  from flask import render_template, flash
8  from flask import Flask, render_template, redirect, url_for
9
10 from urllib.parse import urlparse
11 # from app import app, db
12 rfid_gps_data=[]
13 import flask_login
14 from flask_login import login_user
15 from flask_login import login_required
16 from flask_login import current_user
17 from flask_login import logout_user
18 from flask_login import UserMixin
19
20
21
22 from werkzeug.security import check_password_hash,generate_password_hash
23 login_manager = flask_login.LoginManager()
24
25 # timestamp_str = datetime.utcnow()
26 # # date_time=timestamp_str.strftime("%Y-%m-%d %H:%M:%S")
27 # utc_time = timezone('UTC').localize(timestamp_str)
28 karachi_timezone = pytz.timezone('Asia/Karachi')
29 utc_time = datetime.utcnow()
30 karachi_time = utc_time.astimezone(karachi_timezone)
31
32 app = Flask(__name__)
33 app.secret_key = '13565f86f10f02e43ee54f23cbfe77cb'
34 app.static_folder = 'static'
35 app.static_url_path = '/static'
36 login_manager.init_app(app)
37 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///vehicle_tracking.db'
38 db = SQLAlchemy(app)
39 migrate = Migrate(app, db)
40
41
42
43 class User(db.Model,UserMixin):
44     id = db.Column(db.Integer, primary_key=True)
45     username = db.Column(db.String(50), unique=True, nullable=False)
46     password = db.Column(db.String(100), nullable=False)
47     email = db.Column(db.String(100), unique=True, nullable=False)
```

```

48     is_admin = db.Column(db.Boolean, default=False)
49
50     def is_authenticated(self):
51         return True
52
53     def is_active(self):
54         return True
55
56     def is_anonymous(self):
57         return False
58
59     def is_admin(self):
60         return True
61
62     def get_id(self):
63         return str(self.id)
64
65
66 # admin = User(username='admin', password=admin_password, email='a@g.com', is_admin=True)
67
68 # db.session.add(admin)
69 # db.session.commit()
70
71 class Vehicle(db.Model):
72
73     id = db.Column(db.Integer, primary_key=True)
74     rfid_tag = db.Column(db.String(50), unique=True, nullable=False)
75     number_plate = db.Column(db.String(50), unique=True, nullable=False)
76
77     class Registration(db.Model):
78         id = db.Column(db.Integer, primary_key=True)
79         name = db.Column(db.String(100), nullable=False)
80         cnic = db.Column(db.String(50), unique=True, nullable=False)
81         engine_number = db.Column(db.String(50), nullable=False)
82         number_plate = db.Column(db.String(50), db.ForeignKey('vehicle.number_plate'))
83         rfid_tag = db.Column(db.String(50), unique=True, nullable=False)
84
85     class Location(db.Model):
86         id = db.Column(db.Integer, primary_key=True)
87         lat = db.Column(db.Float, nullable=False)
88         lng = db.Column(db.Float, nullable=False)
89         timestamp = db.Column(db.DateTime, nullable=False, default=karachi_time)
90         number_plate = db.Column(db.String(50), db.ForeignKey('vehicle.number_plate'))
91
92
93
94 @app.route('/search', methods=['GET', 'POST'])
95 @login_required
96 def search():
97     if request.method == 'POST':
98         number_plate = request.form['number_plate']
99         # retrieve the vehicle information from the database
100        vehicle = Vehicle.query.filter_by(number_plate=number_plate).first()
101        if vehicle:
102            # if the vehicle is found, retrieve its latest location from the database
103            location = Location.query.filter_by(number_plate=number_plate).order_by(Location.timestamp.desc()).first()
104            # display the vehicle and location information
105            return render_template('view_rfid.html', vehicle=vehicle, location=location)
106        else:
107            # if the vehicle is not found, display an error message
108            return render_template('search.html', error='Vehicle not found')
109    else:
110        # if the request method is GET, display the search form
111        return render_template('search.html')
112
113
114 @app.route('/view_rfid', methods=['GET', 'POST'])
115 @login_required
116 def view_rfid():
117     if request.method == 'POST':
118         number_plate = request.form['number_plate']
119         # retrieve the vehicle information from the database

```

```

120     vehicle = Vehicle.query.filter_by(number_plate=number_plate).first()
121     if vehicle:
122         # if the vehicle is found, retrieve its associated RFID from the database
123         rfid = Registration.query.filter_by(number_plate=number_plate).first().rfid_tag
124         # display the vehicle and RFID information
125         return render_template('view_rfid.html', vehicle=vehicle, rfid=rfid)
126     else:
127         # if the vehicle is not found, display an error message
128         return render_template('view_rfid.html', error='Vehicle not found')
129     else:
130         # if the request method is GET, display the search form
131         return render_template('view_rfid.html')
132
133
134
135 @app.route('/register', methods=['GET', 'POST'])
136 @login_required
137 def register():
138     if request.method == 'POST':
139         name = request.form['name']
140         cnic = request.form['cnic']
141         engine_number = request.form['engine_number']
142         number_plate = request.form['number_plate']
143         rfid_tag = request.form['rfid_tag']
144
145         # check if the number plate or rfid tag already exist in the database
146         if db.session.query(Vehicle).filter_by(number_plate=number_plate).first() or db.session.query(Registration).filter_by(rfid_tag=rfid_tag).first():
147             return render_template('register.html', error='Vehicle or RFID already exists')
148
149         # create new vehicle and registration objects
150         vehicle = Vehicle(rfid_tag=rfid_tag, number_plate=number_plate)
151         registration = Registration(name=name, cnic=cnic, engine_number=engine_number, number_plate=number_plate,
152
153             # add the new objects to the database
154             db.session.add(vehicle)
155             db.session.add(registration)
156             db.session.commit()
157
158             return render_template('register.html', success='Vehicle registered successfully')
159         else:
160             return render_template('register.html')
161
162
163 @app.route('/search_user', methods=['GET', 'POST'])
164 @login_required
165 def search_user():
166     if request.method == 'POST':
167         cnic = request.form['cnic']
168         # retrieve the user information from the database
169         registrations = db.session.query(Registration).filter_by(cnic=cnic).first()
170         if registrations:
171             # if the user is found, display their information
172             return render_template('view_user.html', registrations=registrations)
173         else:
174             # if the user is not found, display an error message
175             return render_template('search_user.html', error='User not found')
176     else:
177         # if the request method is GET, display the search form
178         return render_template('search_user.html')
179
180
181 @app.route('/all_vehicles')
182 @login_required
183 def all_vehicles():
184     vehicles = Registration.query.all()
185     return render_template('vehicles.html', vehicles=vehicles)
186
187
188 @app.route('/vehicle_location', methods=['GET', 'POST'])
189 @login_required
190 def vehicle_location():
191     if request.method == 'POST':
192         number_plate = request.form['number_plate']
193         vehicle = Vehicle.query.filter_by(number_plate=number_plate).first()
194         if vehicle:

```

```

195     locations = Location.query.filter_by(number_plate=number_plate).all()
196     if locations:
197         return render_template('vehicle_location.html', locations=locations)
198     else:
199         return render_template('vehicle_location.html', error='No locations found for this vehicle.')
200     else:
201         return render_template('search.html', error='Vehicle not found')
202     else:
203         return render_template('search.html')
204
205 @app.route('/login', methods=['GET', 'POST'])
206 def login():
207     if request.method == 'POST':
208         username = request.form['username']
209         password = request.form['password']
210         remember_me = bool(request.form.get('remember_me'))
211         user = User.query.filter_by(username=username).first()
212         if user is None or not user.password==password:
213             flash('Invalid username or password')
214             return redirect(url_for('login'))
215         login_user(user, remember=remember_me)
216         if user.is_admin:
217             return redirect(url_for('admin_dashboard'))
218         else:
219             return redirect(url_for('user_main'))
220
221     return render_template('login.html', title='Sign In')
222
223
224
225 @app.route('/logout')
226 @login_required
227 def logout():
228     logout_user()
229     return redirect(url_for('index'))
230
231 @app.route('/admin_dashboard')
232 @login_required
233 def admin_dashboard():
234     if not current_user.is_admin:
235         abort(403)
236     users = User.query.all()
237     return render_template('admin_dashboard.html', users=users)
238
239
240
241 @app.route('/create_officer', methods=['GET', 'POST']) # type: ignore
242 @login_required
243 def create_officer():
244     if not current_user.is_admin:
245         abort(403)
246     if request.method == 'POST':
247         username = request.form['username']
248         password = request.form['password']
249         email = request.form['email']
250         is_admin = False
251
252         # check if the number plate or rfid tag already exist in the database
253         if db.session.query(User).filter_by(username=username).first() or db.session.query(User).filter_by(email=email).first():
254             return render_template('create_officer.html', error='username or email already exists')
255
256         # create new vehicle and registration objects
257         user = User(username=username, password=password, email=email, is_admin=is_admin)
258
259
260         # add the new objects to the database
261         db.session.add(user)
262         db.session.commit()
263         flash('Officer created successfully!')
264         return render_template('create_officer.html', success='Officer registered successfully')
265     else:
266         return render_template('create_officer.html')

```

```

269 @app.route('/update/<int:id>', methods=['GET', 'POST'])
270 def update(id):
271     if not current_user.is_admin:
272         abort(403)
273     user = User.query.get_or_404(id)
274     if request.method == 'POST':
275         user.username = request.form['username']
276         user.password = request.form['password']
277         user.email = request.form['email']
278         user.is_admin = request.form.get('is_admin') == 'on'
279         db.session.commit()
280     return redirect(url_for('admin_dashboard'))
281 return render_template('update.html', user=user)
282
283
284
285
286 @app.route('/delete/<int:id>', methods=['POST'])
287 def delete(id):
288     if not current_user.is_admin:
289         abort(403)
290     user = User.query.get_or_404(id)
291     db.session.delete(user)
292     db.session.commit()
293     return redirect(url_for('admin_dashboard'))
294
295
296 @login_manager.user_loader
297 def load_user(user_id):
298     return User.query.get(int(user_id))
299
300
301 @app.route('/')
302 def index():
303     return render_template('index.html')
304
305 @app.route('/about_us')
306 def about_us():
307     return render_template('about_us.html')
308
309 @app.route('/user_main')
310 def user_main():
311     return render_template('user_main.html')
312
313
314 if __name__ == '__main__':
315     # db.create_all()
316     # app.run(debug=True)
317     # app.run(debug=True, host='192.168.43.206', port=5000)
318     app.run(debug=True)

```

7.3 OPERATIONAL DIAGRAM

RFID's to be installed in number plate of cars and scanners are installed on various points of the city which then scan these chips. And send the scanned rfid to the database with its location which then be mapped on to google maps. The Rfid database store's location with date and time of the vehicle which it passes. this all information can be accessed by a dashboard. In the end authenticated by excise database.

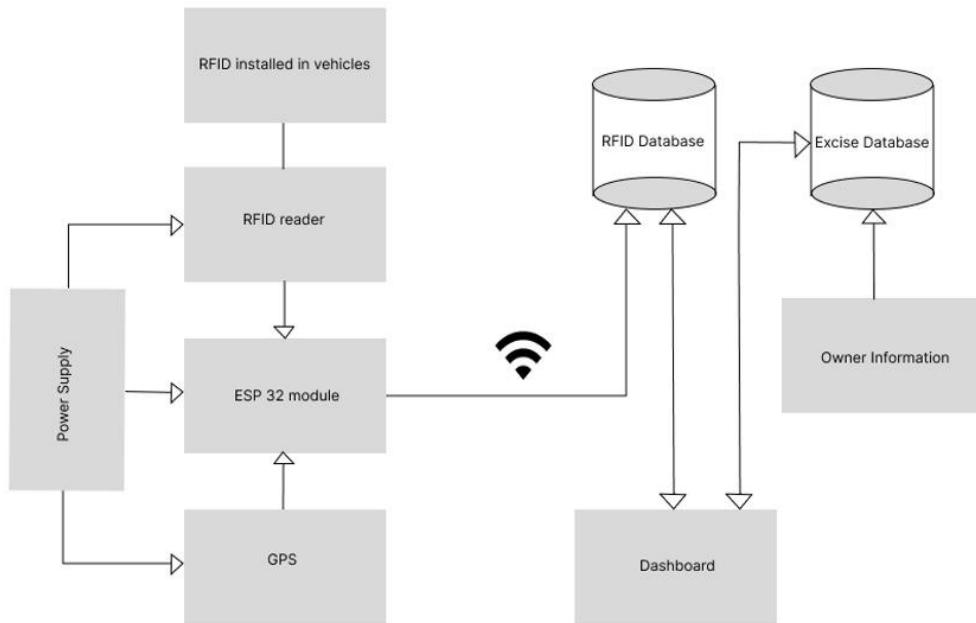


Figure 34 OPERATIONAL DIAGRAM

7.4 COMPONENT DIAAGRAM

A component diagram in a Vehicle Detection System (VDS) with RFID is a graphical representation that shows the various software and hardware components that make up the system, and how they interact with each other. The component diagram helps to visualize the overall architecture of the system and the dependencies between the various components, making it easier to understand the system's design and implementation. It also helps to identify potential areas of improvement or modifications needed to the system.

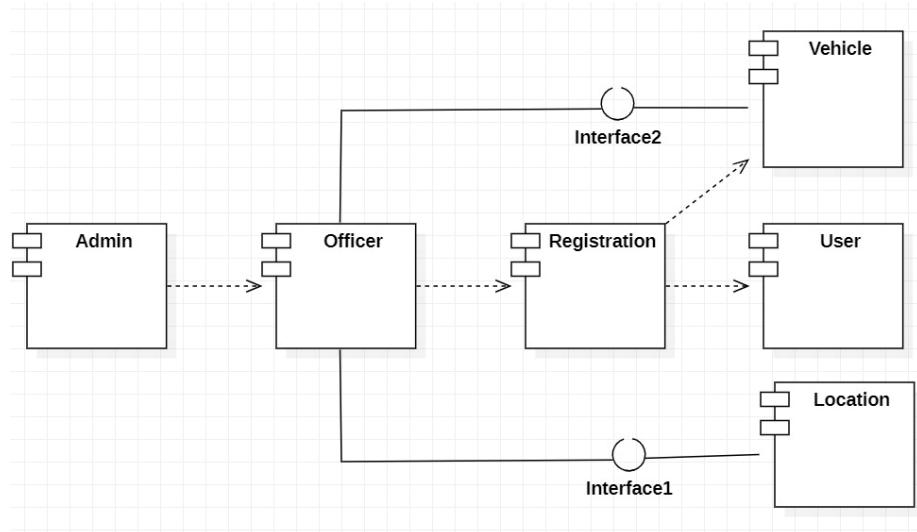


Figure 35 COMPONENT DIAGRAM

7.5 DEPLOYMENT DIARGAM

A deployment diagram in a Vehicle Detection System (VDS) with RFID is a graphical representation that shows the physical deployment of the system's components, such as hardware and software, on various computing nodes and devices. The deployment diagram helps to visualize the physical deployment of the system's components and the communication channels between them, making it easier to understand how the system is deployed in the physical environment. It also helps to identify potential issues related to the deployment, such as network latency, bandwidth constraints, or hardware limitations.

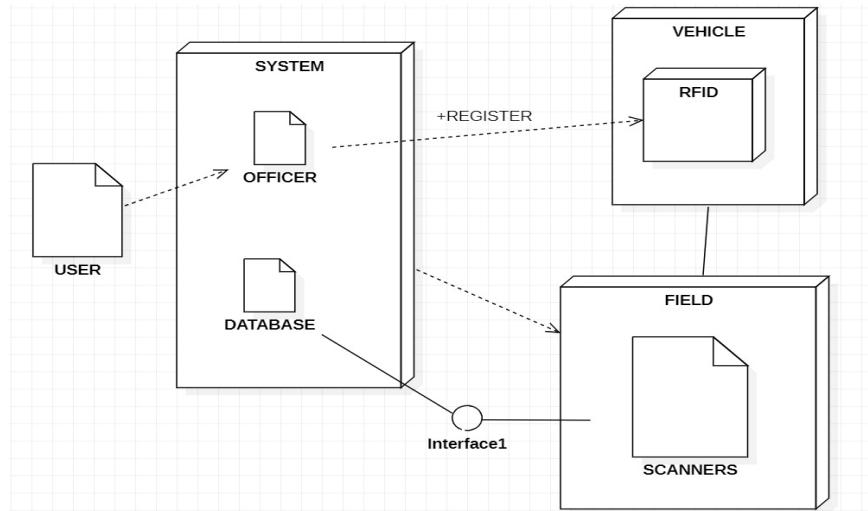


Figure 36 DEPLOYMENT DIAGRAM

CHAPTER 8

TESTING

8.1 INTRODUCTION

Testing is the process of evaluating a software application or system to ensure that it meets its intended requirements and functions correctly. Testing is typically performed by a software tester or a team of testers, who use various techniques and tools to identify defects, bugs, and other issues in the software. The goal of testing is to find and fix any defects before the software is released to end-users, to ensure that it is reliable, secure, and performs as expected.

8.2 WHITE BOX TESTING

White box testing is a type of software testing that involves testing the internal workings or code of a software application. In this type of testing, the tester has access to the source code and is able to see the software's internal structures and processes. The purpose of white box testing is to evaluate the internal logic, structure, and design of the software to ensure that it functions correctly, efficiently, and reliably. It is also known as structural testing, clear-box testing, or open-box testing. White box testing can be conducted at various levels of the software development lifecycle, including unit testing, integration testing, and system testing.

8.1.1 TEST CASE (USER LOGIN)

TEST CASE	PRE-CONDITION	TEST STEP	EXPECTED RESULT	SUCCESS CRITERIA
To ensure that the user can successfully sign-in to the system using the credentials provided by the admin.	The system is installed and configured properly. The database is set up and contains the necessary user information. The user has been provided with valid credentials by the admin.	Launch the system and navigate to the user sign-in page. Enter the valid username and password provided by the admin. Click on the "Sign In" button. Verify that the user is	The user should be able to log in to the system using valid credentials. The user should not be able to log in to the system using invalid credentials. Appropriate error messages should be displayed for invalid credentials.	The test case passes if the user is able to log in to the system using valid credentials, and appropriate error messages are displayed for invalid credentials. The test case fails if the user is unable to log in to the

		<p>successfully logged in to the system and the user's dashboard is displayed.</p> <p>Log out from the user's account.</p> <p>Repeat the above steps with invalid username and password combinations.</p> <p>Verify that appropriate error messages are displayed for incorrect credentials.</p>	<p>invalid credentials.</p>	<p>system using valid credentials or no error messages are displayed for invalid credentials.</p>
--	--	--	-----------------------------	---

8.2.2 TEST CASES (SEARCH VEHICLE USING NUMBER PLATE)

TEST CASE	PRE-CONDITION	TEST STEP	EXPECTED RESULT	SUCCESS CRITERIA
To verify that the system can accurately search for a vehicle using its number plate.	<p>The system is installed and configured properly.</p> <p>The user has valid login credentials to access the system.</p>	<p>Log in to the system with valid login credentials.</p> <p>Navigate to the "Search Vehicle" option.</p> <p>Enter a valid vehicle</p>	<p>The system should accurately display the details of the searched vehicle for a valid number plate.</p> <p>An appropriate error message</p>	<p>The test case passes if the system accurately displays the details of the searched vehicle for a valid number plate, and appropriate error messages</p>

	<p>There are one or more vehicles registered in the system.</p>	<p>number plate in the search field.</p> <p>Click on the "Search" button.</p> <p>Verify that the details of the searched vehicle are displayed, including its make, model, color, and other relevant information.</p> <p>Repeat the above steps with an invalid number plate format (e.g., wrong pattern, special characters, etc.).</p> <p>Verify that an appropriate error message is displayed for invalid number plate format.</p> <p>Repeat the above steps with a non-existent number plate.</p> <p>Verify that an appropriate</p>	<p>should be displayed for invalid number plate format.</p> <p>An appropriate message should be displayed stating that the number plate was not found for a non-existent number plate.</p>	<p>are displayed for an invalid number plate format and non-existent number plates. The test case fails if the system does not display the details of the searched vehicle for a valid number plate or does not display an appropriate error message for an invalid number plate format or non-existent number plates.</p>
--	---	--	--	--

		message is displayed stating that the number plate was not found.		
--	--	---	--	--

8.2.3 TEST CASES (SEARCH VEHICLE OWNER USING CNIC)

TEST CASE	PRE-CONDITION	TEST STEP	EXPECTED RESULT	SUCCESS CRITERIA
verify that the system can accurately display the owner details of a vehicle using their CNIC number.	The system is installed and configured properly. The user has valid login credentials to access the system. The owner of the vehicle whose details are to be searched has a valid CNIC number registered in the system.	Log in to the system with valid login credentials. Go to the "Search Vehicle Owner" option. Enter a valid CNIC number in the search field. Click on the "Search" button. Verify that the owner details of the searched CNIC number are accurately displayed. Repeat the above steps with an	The system should accurately display the owner details of the searched CNIC number. An appropriate error message should be displayed for an invalid CNIC number.	The test case passes if the system accurately displays the owner details of the searched CNIC number and an appropriate error message is displayed for an invalid CNIC number. The test case fails if the system does not display the owner details accurately or does not display an appropriate error message for an invalid CNIC number.

		<p>invalid CNIC number.</p> <p>Verify that an appropriate error message is displayed for an invalid CNIC number.</p>		
--	--	--	--	--

8.2.4 TEST CASES (SEARCH VEHICLE LOCATION USING MAP)

TEST CASE	PRE-CONDITION	TEST STEP	EXPECTED RESULT	SUCCESS CRITERIA
To verify that the system can accurately display the location of a vehicle on a map.	<p>The system is installed and configured properly.</p> <p>The user has valid login credentials to access the system.</p> <p>The vehicle whose location is to be searched has a RFID chip installed and is registered to the system.</p>	<p>Log in to the system with valid login credentials.</p> <p>Go to the "Search Vehicle option" page.</p> <p>Enter a valid vehicle number plate in the search field.</p> <p>Click on the "Search" button.</p> <p>Verify that the location of the searched vehicle is accurately displayed on a map.</p>	<p>The system should accurately display the location of the searched vehicle on a map for a valid number plate.</p> <p>An appropriate error message should be displayed for an invalid number plate.</p>	<p>The test case passes if the system accurately displays the location of the searched vehicle on a map for a valid number plate and an appropriate error message is displayed for an invalid number plate. The test case fails if the system does not display the location of the searched vehicle accurately or does not display an</p>

		<p>Repeat the above steps with an invalid number plate.</p> <p>Verify that an appropriate error message is displayed for an invalid number plate.</p>		appropriate error message for an invalid number plate.
--	--	---	--	--

8.2.5 TEST CASES (VIEW ALL VEHICLES LIST)

TEST CASE	PRE-CONDITION	TEST STEP	EXPECTED RESULT	SUCCESS CRITERIA
To verify that the system can accurately display the list of all vehicles to both the user and the admin.	<p>The system is installed and configured properly.</p> <p>The user has valid login credentials to access the system.</p> <p>The admin has valid login credentials to access the system.</p> <p>There are multiple vehicles registered in the system.</p>	<p>Log in to the system with valid login credentials.</p> <p>Go to the "View All Vehicles" option.</p> <p>Verify that the list of all vehicles is accurately displayed.</p> <p>Verify that the user can only view the list and cannot edit or delete any vehicle details.</p>	<p>The system should accurately display the list of all vehicles to both the user and the admin.</p> <p>The user and admin should able to view the list and vehicle details.</p>	<p>The test case passes if the system accurately displays the list of all vehicles to both the user and the admin.</p> <p>The test case fails if the system does not display the list of all vehicles accurately.</p>

8.2.6 TEST CASES (REGISTER A VEHICLE)

TEST CASE	PRE-CONDITION	TEST STEP	EXPECTED RESULT	SUCCESS CRITERIA
To verify that the system can accurately register a new vehicle by providing the name, CNIC, engine number, number plate, and RFID number.	<p>The system is installed and configured properly.</p> <p>The user has valid login credentials to access the system.</p> <p>The user has the necessary information, including the name, CNIC, engine number, number plate, and RFID number, for registering a new vehicle.</p>	<p>Log in to the system with valid login credentials.</p> <p>Navigate to the "Register a Vehicle" page.</p> <p>Enter the name, CNIC, engine number, number plate, and RFID number of the new vehicle.</p> <p>Click on the "Register" button.</p> <p>Verify that the new vehicle is successfully registered and the system displays a success message.</p> <p>Navigate to the "View All Vehicles" page and verify that the new vehicle is accurately displayed.</p>	<p>The system should successfully register the new vehicle and display a success message.</p> <p>The new vehicle should be accurately displayed on the "View All Vehicles" page.</p> <p>An appropriate error message should be displayed for invalid or incomplete information.</p>	<p>The test case passes if the system successfully registers the new vehicle, accurately displays it on the "View All Vehicles" page, and displays an appropriate error message for invalid or incomplete information.</p> <p>The test case fails if the system fails to register the new vehicle, displays incorrect information, or does not display an appropriate error message for invalid or incomplete information.</p>

		<p>Repeat the above steps with invalid or incomplete information.</p> <p>Verify that the system displays an appropriate error message for invalid or incomplete information.</p>		
--	--	--	--	--

8.2.7 TEST CASES (ADD NEW USER)

TEST CASE	PRE-CONDITION	TEST STEP	EXPECTED RESULT	SUCCESS CRITERIA
To verify that the system can accurately add a new user to the system by the admin.	<p>The system is installed and configured properly.</p> <p>The admin has valid login credentials to access the system.</p> <p>The admin has the necessary information, including name, email, and password, for adding a new user to the system.</p>	<p>Log in to the system with valid admin login credentials.</p> <p>Navigate to the "Add New User" page.</p> <p>Enter the necessary information, including name, email, and password, of the new user.</p> <p>Click on the "Add User" button.</p>	<p>The system should successfully add the new user and display a success message.</p> <p>An appropriate error message should be displayed for invalid or incomplete information.</p>	<p>The test case passes if the system successfully adds the new user and displays a success message. An appropriate error message should be displayed for invalid or incomplete information.</p> <p>The test case fails if the system fails to add the new user, displays incorrect</p>

		<p>Verify that the new user is successfully added to the system and the system displays a success message.</p> <p>Repeat the above steps with invalid or incomplete information.</p> <p>Verify that the system displays an appropriate error message for invalid or incomplete information.</p>		information, or does not display an appropriate error message for invalid or incomplete information.
--	--	---	--	--

8.3 BLACK BOX TESTING

Black box testing is a technique used in software testing to check the functionality of a software application without knowing how it works internally. It is done by testers who only focus on making sure that the software meets its intended functional requirements. To achieve this, testers use different techniques such as equivalence partitioning, boundary value analysis, and decision table testing. The main objective of black box testing is to ensure that the software is reliable, robust, and meets the needs of its users by checking its behavior without any knowledge of its internal workings.

8.3.1 TEST CASES

Case	Input	Expected Output	Actual Output	Status
Verify that a user can register and login to their account with all the required information.	<p>Navigate to the user registration page.</p> <p>Fill in all the required fields, including name, email, password, and contact information.</p> <p>Click the "Register" button.</p> <p>Verify that the system displays a success message confirming that the user account has been created.</p> <p>Navigate to the user login page.</p> <p>Enter the email and password used during registration.</p> <p>Click the "Login" button.</p> <p>Verify that the system redirects the user to their</p>	<p>The user is able to successfully register and login to their account with all the required information.</p>	<p>The user is able to successfully register and login to their account with all the required information.</p>	Valid

	dashboard page.			
Verify that users can register their vehicles with all the required information.	<p>Navigate to the vehicle registration option.</p> <p>Fill in all the required fields, Click the "Register Vehicle" button.</p> <p>Verify that the system displays a success message confirming that the vehicle has been registered.</p>	<p>The user is able to successfully register their vehicle with all the required information.</p>	<p>The user is able to successfully register their vehicle with all the required information.</p>	Valid
Verify that searching for a vehicle using number plate is working as intended, and the system returns the correct vehicle information.	<p>Navigate to the search option.</p> <p>Enter the number plate of the vehicle.</p> <p>Click the "Search" button.</p> <p>Verify that the system returns the correct vehicle information, including the name, CNIC, engine number, number plate, and RFID number.</p>	<p>The system should return the correct vehicle information for the given search criteria.</p>	<p>The system returns the correct vehicle information for the given search criteria.</p>	Valid

<p>Verify that adding new users to the system by an admin is working as intended, and admins can create new user accounts with all the required information.</p>	<p>Navigate to the user management page.</p> <p>Click the "Add New User" option.</p> <p>Fill in all the required fields, including the name, email, username, password.</p> <p>Click the "Create User" button.</p> <p>Verify that the system displays a success message confirming that the user has been created.</p>	<p>The admin should be able to create a new user account with all the required information.</p>	<p>The admin is able to create a new user account with all the required information.</p>	<p>Valid</p>
<p>Verify that users and admins can view the list of all registered vehicles in the system.</p>	<p>Navigate to the vehicle list option.</p> <p>Verify that the system displays a list of all registered vehicles in the system, including their name, CNIC, engine number, number plate, and RFID number.</p>	<p>The system should display a list of all registered vehicles in the system, and users and admins should be able to view accurate and up-to-date information for each vehicle.</p>	<p>The system displays a list of all registered vehicles in the system, and users and admins can view accurate and up-to-date information for each vehicle.</p>	<p>Valid</p>

	<p>Check that the list is properly sorted and searchable.</p> <p>Verify that the system displays accurate and up-to-date information for each vehicle.</p>			
--	--	--	--	--

CHAPTER 9

CONCLUSION

9.1 INTRODUCTION

A Vehicle Detection System (VDS) is a technology used to detect the presence of vehicles entering and exiting a particular location. VDS can use various technologies such as Radio Frequency Identification (RFID), ultrasonic sensors, infrared sensors, or magnetic sensors to detect the vehicles.

In the case of RFID-based VDS, a passive RFID tag is attached to the vehicle, and an RFID reader is installed at the entry and exit points of the location. As the vehicle passes through the reader, the RFID tag transmits a unique identifier to the reader, which is then processed by the VDS software.

The VDS software can then track the movement of the vehicle, providing real-time information about the vehicle's entry and exit times. The system can also generate reports on vehicle traffic patterns, parking availability, and revenue generation.

9.2 CONCLUSION

a Vehicle Detection System (VDS) with RFID technology is an innovative solution that can significantly improve the efficiency of traffic management and vehicle tracking. With its ability to automatically detect and identify vehicles using RFID tags, the VDS can provide accurate and real-time information about vehicle movements, entry and exit times, and parking availability.

By implementing the VDS, organizations such as traffic police, parking management authorities, and transportation agencies can optimize their operations, reduce congestion, and enhance security. The system is also user-friendly, with a simple and intuitive graphical user interface that makes it easy for operators to monitor and manage the system.

Furthermore, the VDS is a cost-effective solution that requires minimal maintenance and can be easily integrated with other systems. With its high reliability and accuracy, the VDS can provide a significant return on investment and contribute to the overall success of an organization.

Overall, the VDS with RFID technology is a powerful tool that can improve the efficiency and effectiveness of traffic management and vehicle tracking. Its numerous benefits make it an essential system for any organization that wants to optimize its operations and enhance its services.

CHAPTER 10

FUTURE WORK

10.1 INTRODUCTION

VDS with RFID is the project that can be further modified in the future like we can also use this system in parking areas. The parking system uses RFID technology to monitor parking facilities and provide real-time information about available parking spaces. It can also be used to enforce parking regulations and improve parking management. Similar, it can be used to detect who didn't pay the tax.

10.2 FUTURE ENHANCEMENT

Mobile app integration: The VDS software could be integrated with a mobile app, allowing users to view real-time information about vehicle traffic patterns and parking availability from their smartphones.

Machine learning-based analytics: By leveraging machine learning algorithms, the VDS software could provide more accurate and insightful data analysis, such as predicting peak traffic periods and optimizing parking allocation.

Multi-site integration: The VDS system could be expanded to cover multiple sites, providing a centralized view of vehicle traffic across multiple locations.

Integration with smart city systems: VDS could be integrated with smart city systems, allowing for better coordination of traffic flow and optimizing overall city operations.

Integration with autonomous vehicles: As autonomous vehicles become more common, VDS could be modified to support automatic detection and tracking of these vehicles.

Overall, these enhancements could provide even greater benefits in terms of traffic management, security, and revenue generation, making VDS with RFID technology an even more valuable investment for organizations in the future.

CHAPTER 11

ACHIEVEMENT

11.1 INTRODUCTION

In this chapter we will discuss the skill experience that we gain by completing such project.

11.2 ACTIVITIES

11.2.1 PROJECT PLANNING

This involves defining the project goals, objectives, scope, and timeline, as well as identifying the resources required and the risks involved. Which we have done at the start of the project.

11.2.2 SYSTEM ANALYSIS AND DESIGN

This involves analyzing the requirements of the VDS and designing the system architecture, components, and functionalities. Also include selecting the appropriate RFID hardware and software, database management, and user interface design. We had several meeting with our supervisor to select appropriate hardware

11.2.3 SYSTEM IMPLEMENTATION

This involves developing and coding the software, hardware components, and integrating them into a working system. All of our team members sit together to work on the project and to overcome the problems

11.2.4 SYSTEM TESTING AND VALIDATION

This involves testing the VDS system to ensure that it meets the specified requirements and functions as intended. All of our members discuss the project to make case studies and test cases accordingly

11.2.5 DOCUMENTATION

This involves creating documentation, such as project plans, technical specifications, user manuals, and system documentation, to support the development and implementation of the VDS. The documentation had place from the very first day of the project.

11.2.6 PRESENTATION AND DEMONSTRATION

This involves presenting the VDS project to the project stakeholders, demonstrating the functionality of the system, and obtaining feedback on the project outcomes. We demonstrate our working in very milestone.

11.3 IMPROVEMENT

11.3.1 TECHNICAL SKILLS

By completing a VDS project improve experience in designing and developing complex software and hardware systems, which will help us to develop technical skills in areas such as RFID technology, database management, and system integration.

11.3.2 PROBLEM-SOLVING ABILITIES

Developing a VDS project requires a significant amount of problem-solving and critical thinking skills, which students can apply to other real-world scenarios and challenges. By completing VDS project our skills of Problem solving has improved

11.3.3 INDUSTRY-RELEVANT EXPERIENCE

The VDS project is a practical and industry-relevant project that prepares students for future career opportunities in areas such as transportation, logistics, and security.

11.3.4 TEAMWORK AND COMMUNICATION

By completing VDS project we gain valuable experience working in a team and communicating effectively with team members, stakeholders, and clients.

11.3.5 ACADEMIC RECOGNITION

Completing a VDS project also provide us with academic recognition in the form of course credits, grades, and awards, which can enhance our academic and professional credentials.

APPENDIX A

EXCISE, TAXATION AND NARCOTICS CONTROL DEPARTMENT

A Vehicle Detection System (VDS) with RFID technology can be helpful for the Excise, Taxation and Narcotics Control Department in Pakistan in several ways:

Vehicle Registration: VDS with RFID technology can help the department manage vehicle registration by automating the process of capturing vehicle data, such as make, model, and registration number. This can help the department maintain accurate records of registered vehicles and reduce the risk of fraud.

Vehicle Taxation: VDS with RFID technology can help the department monitor vehicle usage and calculate taxes based on distance traveled or fuel consumption. This can help the department collect taxes more efficiently and reduce the risk of tax evasion.

Anti-Smuggling: VDS with RFID technology can be used to track the movement of vehicles suspected of smuggling illicit goods, such as drugs or contraband. This can help the department identify and intercept illegal activity more effectively.

Security and Surveillance: VDS with RFID technology can be used for security and surveillance purposes, such as monitoring the movement of vehicles near sensitive locations or tracking the movement of suspected criminals. This can help the department improve public safety and reduce crime.

Overall, a Vehicle Detection System with RFID technology can help the Excise, Taxation and Narcotics Control Department in Pakistan manage vehicle registration and taxation more efficiently, reduce the risk of fraud and tax evasion, and improve security and surveillance.

Real-Time Traffic Monitoring: VDS with RFID technology can provide real-time data on vehicle movement, which can help the traffic police department monitor traffic flow and identify congestion hotspots. This can help them make informed decisions about traffic management and deploy resources more effectively.

Automated Traffic Violation Detection: VDS with RFID technology can be integrated with automated traffic violation detection systems, such as speed cameras and red light cameras, to

identify and penalize traffic violators. This can help the traffic police department enforce traffic laws and improve road safety.

Parking Management: VDS with RFID technology can be used to manage parking facilities, such as public parking lots and street parking zones, by tracking the number of vehicles entering and leaving the facility and enforcing parking regulations. This can help the traffic police department regulate parking and reduce traffic congestion caused by illegal parking.

Emergency Response: VDS with RFID technology can help the traffic police department respond more quickly to emergencies, such as accidents and road closures, by providing real-time data on traffic flow and congestion. This can help them divert traffic away from affected areas and minimize disruption to the public.

APPENDIX B

ASSET TRACKING SYSTEM

Asset Management: An ATS can help organizations manage their assets more efficiently by providing real-time information about their location and status. This can help reduce the risk of theft or loss and improve asset utilization.

Inventory Management: An ATS can be used to track inventory in real-time, allowing organizations to optimize their inventory levels and reduce waste.

Maintenance Management: An ATS can be used to track the maintenance history of assets, such as equipment, and schedule preventive maintenance. This can help organizations reduce downtime and extend the lifespan of their assets.

Compliance Management: An ATS can help organizations comply with regulations related to asset tracking and management, such as those related to environmental or safety standards.

APPENDIX C

HARDWARE PACIFICATION

RFID Tags: Passive RFID tags can be attached to the vehicles, and active RFID tags can be installed in the vehicles. These tags contain a unique identifier that can be read by RFID readers.

RFID Readers: RFID readers can be installed at the entry and exit points of the parking facility or on the gantries of the highway. These readers can communicate with the RFID tags and collect data about the location and movement of the vehicles.

Antennas: RFID antennas are used to transmit and receive signals between the RFID readers and tags. They can be installed in various locations to ensure maximum coverage of the facility or highway.

Control Unit: The control unit is used to manage and process the data collected by the RFID readers. It can be a standalone device or integrated with a software application.

Software: The software application is used to manage and analyze the data collected by the RFID readers. It can provide real-time information about the location and status of vehicles and generate reports for parking management or traffic control.

Network Infrastructure: A network infrastructure is required to connect the RFID readers, control unit, and software application. This can be a wired or wireless network, depending on the location and size of the facility or highway.

APPENDIX D **DESIGN DOCUMENTS**

System Design Document: This document provides an overview of the system architecture, components, and interfaces. It includes a detailed description of the hardware and software components required for the VDS, such as RFID tags, readers, antennas, control unit, software, and network infrastructure. It also outlines the data flow and communication protocols between the components.

User Interface Design Document: This document describes the graphical user interface (GUI) of the VDS software. It includes wireframes, mockups, and user interface specifications. The document defines the layout, navigation, and functionality of the GUI.

Database Design Document: This document outlines the database schema and data model used by the VDS software. It includes a description of the database tables, fields, and relationships. The document also defines the data validation rules and constraints.

Network Design Document: This document outlines the network infrastructure required for the VDS. It includes a description of the network topology, devices, and protocols. The document also defines the security policies and access controls.

Integration Design Document: This document describes the integration of the VDS with other systems, such as a parking management system or traffic control system. It includes a description of the data exchange protocols, data format, and data transformation rules.

Test Plan Document: This document outlines the testing strategy and test cases for the VDS. It includes a description of the testing environment, test data, and test procedures. The document also defines the acceptance criteria and quality metrics.

REFERENCES

- [1] [\(1566\) IVIS RFID License Plate - YouTube](#)
 - [2] [HOME - RFID PAKISTAN](#)
 - [3] [RFID vehicle tracking system, Vehicle asset tracking | RAMP \(ramprfid.com\)](#)
 - [4] [A study of RFID adoption for vehicle tracking in a container terminal \(upc.edu\)](#)
 - [5] [RFID Tracking System for Vehicles \(RTSV\) \(researchgate.net\)](#)
 - [6] https://www.researchgate.net/profile/Maruf-Islam/publication/322516489_Smart_anti-theft_vehicle_tracking_system_for_Bangladesh_based_on_Internet_of_Things/links/5c93134145851506d71f7728/Smart-anti-theft-vehicle-tracking-system-for-Bangladesh-based-on-Internet-of-Things.pdf
 - [7] <https://www.asiarfid.com/what-is-rfid.html>
 - [8] <https://www.sparkfun.com/rfid>
 - [9] <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>
 - [10] [Fareastson 1-3meters Long Range Scanner Impinj R2000 Chip Handheld Wireless Portable Uhf Rfid Reader To Connect Android Phone - Buy Bluetooth Rfid Reader,Rfid Reader To Android,Uhf Rfid Handheld Reader Product on Alibaba.com](#)

Turnitin Originality Report

Processed on: 15-Jan-2023 18:10 PKT

ID: 1992324011

Word Count: 7850

Submitted: 2

report By Alisha Hameed

Similarity Index	Similarity by Source
15%	Internet Sources: 10%
	Publications: 5%
	Student Papers: 8%

2% match (Internet from 09-Mar-2016)

<http://louisdl.louislibraries.org/cdm/compoundobject/collection/p120701coll25/id/101780/rec/11>

1% match (student papers from 11-Jan-2023)

[Submitted to Higher Education Commission Pakistan on 2023-01-11](#)

1% match (Internet from 29-May-2022)

<https://www.campcodes.com/projects/php/fleet-management-system-with-live-gps-tracking/>

1% match (Internet from 05-Mar-2014)

http://www.infonomics-society.org/IJRFIDSC/IJRFIDSC_Paper%201.pdf

1% match (Internet from 20-Nov-2022)

<https://worldwidescience.org/topicpages/f/field+uhf+rfid.html>

< 1% match (student papers from 17-Apr-2018)

[Submitted to Higher Education Commission Pakistan on 2018-04-17](#)

< 1% match (Internet from 20-Dec-2022)

https://www.researchgate.net/publication/269309084_Design_and_implementation_of_vehicle_tracking_system_using_GPSGSMGPRS_technology_and