WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

Submitted to Department of Software Engineering in Partial Fulfilment of the Requirement for the Degree of Bachler Science in Software Engineering

Title: Smart Vehicle Tracking System for Wolkite University

No	Team members	ID
1	Azimeraw Taye	251/10
2	Seble Sefineh	301/10
3	Kebron Sebsibe	281/10

Advisor Mrs. Marta S.

Tuesday, May 26, 2021

Wolkite University, Wolkite, Ethiopia

DECLARATION

This is to declare that this project work, which is done under the supervision of Mrs. Marta and having the title Smart Vehicle Tracking System, is the sole contribution of Seble Sefineh, Azimeraw Taye, and Kebron Sebsibe. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if the violation of this declaration is proven.

Date: Tuesday, August 23, 2021

Group Members:

No	Full Name	Signature
1	Azimeraw Taye	----------
2	Seble Sefineh	----------
3	Kebron Sebsibe	----------

APPROVAL FORM

This is to confirm that the project report entitled Smart Vehicle Tracking System submitted to Wolkite University, College of Computing and Informatics Department of Software Engineering by Azmeraw Taye, Seble Sefineh, and Kebron Sebsibe is approved for submission.

Advisor Name	Signature	Date
---	---	---
Department Head Name	Signature	Date
---	---	---
Examiner 1 name	Signature	Date
---	---	---
Examiner 2 name	Signature	Date
---	---	---
Examiner 3 name	Signature	Date
---	---	---

ACKNOWLEDGMENT

First, we want to thank God for helping us to live and to do all this. We thank our Advisor, Mrs. Marta, for guiding, correcting our mistakes, constantly monitoring, encouraging us to do more, and providing necessary information about the project. We have also special thanks for Ph.D. Candidate and Researcher Negalign Wake Hundera helps with title selection and function of the project, Mr. Natinael helps with a general idea, Mr. Fouad help for document organization and IoT material information, we would like to express our gratitude to the Dean of the College, the Head of the Department, our teachers, our classmates, and all individuals those who contributed to the success of this project by giving moral and intellectual support.

Table of Contents

CHAPTER ONE 1
1 INTRODUCTION 1
1.1 Background of the Organization 2
1.2 Statement of the problem 2
1.3 Objective of the project 2
1.3.1 General objective 2
1.3.2. Specific objective 2
1.4. Feasibility analysis 3
1.4.1. Operational feasibility 3
1.4.2. Technical feasibility 3
1.4.3. Economic feasibility 3
1.4.4. Schedule feasibility 4
1.4.5. Political feasibility 4
1.5. Scope and limitation of the Project 5
1.6. Significance of the project 5
1.7. Beneficiary of the project 5
1.8. Methodology of the Project 6
1.8. 1. Data collection techniques 6
1.8.2. System analysis and design 6
1.8.3. System development model 6
1.8.4. System testing methodology 6
1.8.5 System development tool 7
1.10. Document organization 8
CHAPTER TWO 10
2. Description of the Existing System 10
2.1. Introduction of Existing System 10
2.2. Users of Existing System 10
2.3. Major Functions of the Existing System 10
2.4. Forms and other Documents of the Existing Systems 10
2.5. Drawbacks of the Existing System 10
2.6. Business Rules of the Existing System 11
CHAPTER THREE 12
3. PROPOSED SYSTEM 12
3.1. Functional Requirements 12

3.2. Non-Functional Requirements 12
3.2.1. User Interface and Human Factors 12
3.2.2. Hardware Consideration 13
3.2.3. Security Issues 14
3.2.4. Performance Consideration 14
3.2.5. Error Handling and Validation 14
3.2.6. Quality Issues 14
3.2.7. Physical Environment 14
3.2.8. Documentation 14
CHAPTER FOUR 15
4. SYSTEM ANALYSIS 15
4.1. System model 15
4.1.1. Use case model 15
4.1.1.1. Use case diagram 15
4.2 Object Model 26
4.2.1 Class Diagram 26
4.2.2 Data Dictionary 27
4.3 Dynamic Model 28
4.3.1 Sequence Diagram 28
4.3.2 Activity Diagram 32
4.3.3 State Chart Diagram 33
CHAPTER FIVE 37
5. SYSTEM DESIGN 37
5.1. Design Goals 37
5.2. Current System Architecture 38
5.3. Proposed system architecture 38
5.3.1 Sub System Decomposition 39
5.3.2. Hardware/Software mapping 40
5.3.4. Detailed Class Diagram 41
5.3.5. Persistent data management 42
5.3.6. Access Control and Security 43
5.4. Packages 43
5.5. Algorithm Design 44
5.6. User Interface Design 45
CHAPTER SIX 46
6. IMPLEMENTATION AND TESTING 46

6.1 Implementation of the Database 46
6.2 Implementation of the Class Diagram 46
6.3 Configuration of the Application Server 47
6.4 Configuration of Application Security 47
6.5 Implementation of User Interface 48
6.4 Testing 51
6.4.1 Test Case 51
6.4.2 Testing Tools and Environment 51
6.4.3 Unit Testing 51
6.4.4 Integration Testing 52
6.4.5 System Testing 52
6.4.6 Acceptance Testing 52
CHAPTER SEVEN 53
7. CONCLUSION AND RECOMMENDATION 53
7.1 Conclusion 53
7.2 Recommendation 53
References 54

List of Table

Table1. 1 Hardware tool 7
Table1. 2 Front end 8
Table1. 3 Back End 8
Table 4. 1 use case description for login 19
Table 4. 2 use case description for manage user..... 20
Table 4. 3 use case description for register vehicle 21
Table 4. 4 use case for sense request 22
Table 4. 5 Give response..... 23
Table 4. 6 use case for view location and speed 24
Table 4. 7 use case for display location 25
Table 4. 8 use case Theft detection 26
Table 4. 9 User Data Dictionary..... 30
Table 4. 10 Vehicle Dictionary 30
Table 4. 11 Fleet Admin Dictionary..... …31
Table 4. 12 Driver Dictionary 31

List of Figure

Figure 4. 1 use case diagram..... 18
Figure 4. 2 Analysis label class diagram 29
Figure 4. 3 Sequence Diagram for login..... 32
Figure 4. 4 Sequence Diagram for vehicle registration..... 33
Figure 4. 5 Sequence diagram for vehicle tracking..... 34
Figure 4. 6 Admin activity Diagram..... 35
Figure 4. 7 Fleet Admin Activity diagram..... 36
Figure 4. 8 Login state chart diagram .....
37 Figure 4. 9 Vehicle Register state chart Diagram ..... 38 Figure 4. 10 GPS State chart Diagram ..... 39 Figure 4. 11 View location state chart Diagram..... 40
Figure 5. 1 Proposed architecture of SVT system..... 43
Figure 5. 2 Subsystem Decomposition Diagram 45
Figure 5. 3 Hardware/Software Diagram .....
46 Figure 5. 4 Detail class Diagram.....
47 Figure 5. 5 Persistent Data Diagram.....
49 Figure 5. 6 Package Diagram ..... 50 Figure 5. 7 Simulation Diagram for Tracking 52
Figure 5. 8 Login user interface .....
53
Figure 6. 1 Login page for smart vehicle tracking system website..... 49
Figure 6. 2 live streaming of 49
Figure 6. 3 Block Diagram for smart vehicle tracking system 51

Abbreviations

No	Acronyms	Meaning
1	**SVT**	Smart vehicle tracking
2	**IoT**	Internet of Thing
3	**WKU**	Wolkite University
4	**OO**	Object-Oriented
5	**OOA**	Object-Oriented Analysis
6	**OOD**	Object-Oriented Design
7	**UML**	Unified Modeling Language
8	**IDE**	Integrated Development Environment
9	**GPS**	Global Positioning system

Abstract

The vehicle tracking system is designed to know the position, speed and live streaming of the vehicle. A tracking system is developed by using GPS, MEMS, Arduino and ESP2666 to locate the user's vehicle easily. GPS module and MEMS are used to track the location of the vehicle in the form of values such as latitude and longitude and get the speed of the vehicle. The authorized access can track the location and speed of the vehicle view on the navigation map and the engine controller (driver) can start the vehicle by using authenticate figure print sensor.

Keyword: GPS, MEMS, ESP2666

CHAPTER ONE

1 INTRODUCTION

There are several ways in which a vehicle can ease your day-to-day activities. Whenever you want to go to work every day and you have got your vehicle then you will have an easy time arriving at your destination. Vehicles will help you cover long distances over a short period of time and you can visit a lot of places within a short while. This does not mean that a vehicle is a basic need. However, it is turning out to be very essential nowadays in this modern world and very soon everyone will be possessing vehicles. A vehicle will always help you transport heavy loads from one destination.

The vehicle tracking system is a well-established technology in this era which is used by the fleet system and owner of the vehicle all over the world. It is a very safe and reliable technology used to determine the location details, speed, distance traveled, which can be viewed on a digital mapping with the help of software via the Internet of a vehicle. (engg, 2013)

In the present-day vehicle, tracking is becoming essential for improving our living conditions. In the existing system of WKU there are many problems employers no way to track the bus and not finish the work, and difficult to plan the activity. In addition to that wisely use problem, so our proposed system convenience and ease of using the vehicle is what WKU‘s vehicle tracking is offering and WKU vehicles are difficult to track and request the vehicles.

Vehicle tracking offers a futuristic way of life in which the WKU gets to control their vehicle using smart vehicle tracking system is used for, tracking of vehicles is a process in which we track the vehicle location in form of Latitude and Longitude (GPS coordinates). GPS Coordinates are the value of a location.

In this project, we are going one-step ahead with GPS building a GPS-based vehicle tracking system using Arduino, We view the position and speed of the vehicle on Google map. and live streaming can tell you where each vehicle is at any time and real-time camera lets you look inside the vehicle remotely.

The vehicle tracking knows a day-use buying device but there are many limitations i.e. not extend other functionality, costly, and issue of security not know about devices.

The proposed system may difficult to solve if not given the hardware materials on time from WKU and the system are use new technology which leads to not get enough knowledge from others.

SVT tracks the exact location of the vehicles - moving or stationary and personal tracking and control their vehicle from their Desktop/Laptop/ Smart mobile phone with them irrespective of the location of the vehicle by applying features.

1.1 Background of the Organization

The Wolkite university vehicle system includes different vehicle types such as bus, minibus, ambulance, motor, etc., and each vehicle orders with Fleet Admin and next office, that can give service with all Passenger in WKU members.

1.2 Statement of the problem

The global issue related to a constantly increasing crime rate needs to be urgently addressed by both developed and developing countries. In Ethiopia, many cases of vehicle theft are reported each year, and the number is still increasing. If not recovered soon, stolen vehicles are generally sold, revamped, or even burned if the resale price is considered too low. Once a vehicle is stolen, it becomes hard to locate it and track it, which considerably decreases the chances of recovering it.

- ✓ Fleet admin and passenger difficult to track vehicles.
- ✓ Difficulty to succeed activity of the Fleet Admin.
- ✓ Passengers are difficult to know where the vehicle is, and wait a long time.
- ✓ If the vehicle is being stolen, the owner cannot track their vehicle, and the way to get their vehicle back is very difficult. It makes it so hard to track the vehicle when there is no evidence.

1.3 Objective of the project

1.3.1 General objective

The main objective of this project is to develop an embedded smart vehicle tracking system for Wolkite University.

1.3.2. Specific objective

- ✓ Gather and analysis of the current system
- ✓ Design the proposed system.
- ✓ Implement the designed (proposed) system
- ✓ Test and evaluate the implemented system
- ✓ Deployment of the tested system

1.4. Feasibility analysis

A feasibility study is a test of the embedded system proposal according to its IoT, web application, the importance of vehicle tracking in WKU uses the vehicle the employers, ability to meet needs, and effective use of the related vehicle resources. The objective is to determine whether the proposed system is feasible or not through the following feasibility study mechanisms.

1.4.1. Operational feasibility

Our system is simple, accurate, active, easy to use, and solves the business of the organization (WKU) side; since the system is the web application platform-independent, which indicates that the system will be operationally feasible.

1.4.2. Technical feasibility

IoT has able to solve the problem with existing technology and WKU can able to the technology and support the requirements. It is planned to implement the proposed system using Windows operating systems, to develop web applications, the necessary hardware has WKU and software are available for the development and implementation of the proposed system so the application will technically feasible.

1.4.3. Economic feasibility

The proposed system will need additional costs for tools and costs for use of hardware devices that we accomplish but not, much cost and Open Source software programming language in our project development, when we compare the importance of the project and minimize additional cost vehicle-related wastage so will be economically feasible.

The cost-benefit analysis is a commonly used method in evaluating the effectiveness of the system. The cost of developing this system currently includes the minimum software cost to develop and run the system.

Tangible benefit*:* -: The team calculated the corresponding tangible benefits based on the technique called the Time Value of Money (TVM).

Proposed system Smart vehicle tracking			
Hardware materials	Cost in ETB	unit	Total price/unit
Arduino Uno	600	(600)1	600
GPS	550	550(1)	550

Fingerprint sensor	1100	(1100)1	1100
Accelerometer	200	(200)1	200
Camera	700	(700)1	700
ESP2666 module	1000	1	1000
Relay circuit	800	1	800
Total			4950

Estimate Existing system cost

Hardware material	Cost in ETB	Unit	Total price
GPS	5500	1	5500
Ardino uno	30,000	1	30,000
Accelerometer	4,0354	1	4,0354
Camera	1000	1	1000
Fingerprint sensor	3,303	1	3,303
ESP2666module	1500	1	1500
Total			81,657

As we have seen in the above table the cost of hardware which are required to build the proposed system is feasible as compared with the hardware materials that are present in market currently

1.4.4. Schedule feasibility

If the material is given a specific time, our project will accomplish and delivered within the given time. Because there is a work division in our group. Due to this reason the team will deliver based on the given time scope.

1.4.5. Political feasibility

The system we are going to develop is not in conflict and violates any rule with any government since it gives services for the WKU employers, vehicle managers, and drivers. All the stakeholders also will have agreed before the system is developed. Therefore, the application we are going to develop is free from government conflict, in this case, the government is profitable and the system will be politically feasible.

1.5. Scope and limitation of the Project

There are several problems related to vehicles in WKU, but this project is particularly focused on track the exact location, speed, and live streaming of the vehicle and start the engine with enrolled fingure print.

Limitation of the project

- ✓ The proposed system does not notify when the accident has occurred during driving
- ✓ The system also not report the fuel status of the vehicle

1.6. Significance of the project

The newly proposed system allows

- ✓ To tracking vehicle with remote area
- ✓ Live streaming of the vehicle
- ✓ To protect vehicles from theft.
- ✓ To improve safety, not only will drivers be more responsible because they are aware of the GPS monitoring, the Fleet admin manages exactly where a vehicle is it require any assistance, theft recovery, lower operational costs and increase productivity.
- ✓ To increase economic development by wisely use of vehicles, vehicles resource, time and budget
- ✓ Self-service customer support
- ✓ To improve the administration process to know the driver's trust.

1.7. Beneficiary of the project

WKU

- ✓ Used when the employers in every work are coming punctually of time due to vehicle proper work.
- ✓ Effectively manage their employees to minimize idle time thus enhancing productivity

Fleet Admin

- ✓ Used to control all activity and access information in a remote area of university vehicles.
- ✓ Will have full information about the vehicle’s history.

Passengers

- ✓ Track their vehicle
- ✓ They safely travel without hitch.
- ✓ They train punctuality.

1.8. Methodology of the Project

In a newly proposed system, we use different methods to develop the system, to gather information or data on the existing system.

1.8. 1. Data collection techniques

Data collection is the most important part of the project to find the main requirement of the system and to understand how the system works. We are going to use different methods to collect data. Among the methods, we use the following:

Interviews: In our project, we will use structural interview types. We use standardized questions to find points that we need to know all about existing vehicles tracking that, we can also semi-structured interviews when things go another direction of our question.

1.8.2. System analysis and design

We will use object-oriented software engineering methodology (approaches) to develop the system. because it is a popular technical approach for analyzing, designing an application system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to brief better stakeholder communication and software product quality. We choose Object-Oriented Analysis because it has so many advantages and which can make the system more effective.

- ✓ Systems can be added gradually one subsystem at a time.
- ✓ Reuse of previously written software is a possibility.
- ✓ It is acceptable to tackle difficult problems first

1.8.3. System development model

In the proposed system, we will use iterative software development models because our system uses hardware materials (IoT) that may support multi functionalities then we can add other features on the embedded system and also use to testing and debugging during smaller iteration is easy and we are built and improve the product step by step.

1.8.4. System testing methodology

1.8.4.1 Unit testing

Used to test the circuit, Arduino, web to check the input-output data is valid or not and modify it to the correct way.

- ✓ White box testing: use to test the functionality of each hardware.
- ✓ Gray box testing: test the circuit of the hardware before the power supply and upload the code.

1.8.4.2 Integration testing

- ✓ Integrated hardware and software environment to verify the behavior of the complete system.
- ✓ This test is used in vehicle tracking, client view can be integrated.

1.8.4.3 System testing

In this level of testing process, we have examined how the whole smart vehicle tracking subsystems came together to achieve the desired goal (user's requirements of the system). Sample Tests

- ✓ Our system can be integrated with configured IoT circuit and web application that implements the problem.
- ✓ Evaluate the functionality of the subsystem after a combination of individual subsystems whether it works correctly or not.
- ✓ Verify the system completeness-based user's requirement.

1.8.5 System development tool

1.8.5.1 Hardware Tool

Table1. 1 Hardware tool

No	Name	Reasons
21	Arduino Uno (Microcontroller)	The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with.it is used and documented board of the whole Arduino family. (Saniah Ahmed, 2018)
2	GPS module	Used to detect the Latitude and Longitude of any location on the Earth, with exact Universal Time Co-ordinate time.
3	ESP2666 module	IoT device enables microcontrollers to connect to Wi-Fi
4	Accelerometer(MEMS)	To measures and reports the speed of a vehicle
5	Relay circuit	Used to control engine by opening and closing contacts in another circuit.
6	Figureprint sensor	Used to detect the engine controller that can start the vehicle

7	Camera	Used to control the live streaming of vehicle

1.8.5.2 Software Tool

1.9.2.1 Front end

Table1. 2 Front end

No.	Name	Version	Reasons
1	Arduino IDE	1.8.13	It is used to write and upload programs to Arduino compatible boards.
1	Enterprise Architect	12	That is capable of drawing types of UML diagrams.
2	HTML	HTML5	Will be used to structure a web page and its content in the web part of ours project.
3	Proteus software		simulating the circuits containing the microcontrollers where we can simulate the circuit

1.9.2.2 Back end

Table1. 3 Back End

No.	Name	Version	Reasons
1	Django	3.2	Python 3.9.6 is the newest major release of the Python support Django and it's an easy and fast way to develop a good website that is why we choose Django web development.
2	MYSQLite	3.33. 0	store data in structured manner,

1.10. Document organization

The proposed system document contains seven-chapter each chapter describe it in the following manner:

- ✓ Chapter 1: - In chapter one we describe the introduction of the existing system, the problem of the existing system, the objective (general and specific objective), and methodology we followed during data collection and analysis.
- ✓ Chapter 2: - This chapter describes what the existing system looks like in detail (who uses the existing system with major function and drawback of the existing system described in this chapter in detail).
- ✓ Chapter 3: - This chapter is about the proposed system, which includes the functional and non-functional requirements of the proposed system.
- ✓ Chapter 4: - Describe the analysis of the proposed system consists of the flow of events in the scenario, use case model with its description of major use case, detail model, and dynamic model of the proposed system.
- ✓ Chapter 5: - Deals with system design. Which includes the overview of the system, design consideration, design goal, design tradeoffs, the architecture of the System, subsystem decomposition, persistent data management, and user interface design.
- ✓ Chapter 6:- Describe the implementation and testing strategy of the system.
- ✓ Chapter 7:- in chapter we conclude the complete system and give recommendation that are done in the future.

CHAPTER TWO

2. Description of the Existing System

2.1. Introduction of Existing System

The existing system in WKU Vehicle tracking systems track indirectly since some activities and tasks can be taken care of by the Fleet Admin, Instruction driver: -based on the requested service the manager order the driver by giving the word command or written command.

2.2. Users of Existing System

The followings are users in the existing system:

- ✓ Fleet admin (Simrit kifl): Are used to controlling all activity and access information of university vehicles.
- ✓ The driver of the vehicle: the driver has the responsibility to travel according to their schedule.
- ✓ Passengers**:** Includes all lecturers, administrative workers, and students on the campus that gets the vehicle service.

2.3. Major Functions of the Existing System

Vehicle allocation

- ✓ Giving service for administrative worker
- ✓ Allocating the bus for a student for apparent and trip programs

Driver allocation

- ✓ Arrange drivers to short distance middle distance or long distance through their order.

Vehicle management

- ✓ Manages the vehicle that needs oil and also need maintenance

2.4. Forms and other Documents of the Existing Systems

- ✓ We have put some forms in the appendix below.

2.5. Drawbacks of the Existing System

This manual system gives a variety of service for the distribution office, but it has its drawback since its response time not efficient and makes workers impressed due to a lot of heavy-duty, the major drawback is:

- ✓ The vehicle tracking way manual or there is no way to track vehicles

- ✓ Their no way to get stolen vehicles electronically or simple manual finding is tedious to get vehicles and less probability getting chance.

2.6. Business Rules of the Existing System

The business rule tells the organization what is expected from them to use these systems.it is a rule that everybody must follow to use the system and service from it, the business rule of the existing system.

Table 2. 1 Business rule of the existing system

Name	**ID**	**Description**
Membership	BR1	The member of the university community who is willing to do in the system can use the system
Post schedule	BR2	The transport manager post schedule weekly for the customer
Approving request	BR3	During an emergency, the transport manager must approve the request of the customer.
Requesting the vehicle	BR4	The request must be performed before one month for a trip and apparent
Sending fuel coupon	BR5	The transport managers must send the message for fuel information when fuel is needed for the vehicle.

CHAPTER THREE

3. PROPOSED SYSTEM

The proposed system can be highly useful for various transport companies to keep track of their vehicles. This system can be designing WKU vehicle location and speed tracking and start and stop vehicle.

3.1. Functional Requirements

The system shall be expected to provide vehicle tracking services and functionalities:

- ✓ Able to track, the current location of the vehicle
- ✓ able to show the speed of the vehicle that has traveled
- ✓ Users can navigate the location by open vehicle map, but display on web because it need Google map API key.
- ✓ Live streaming of vehicle cameras can automatically and save the view of your vehicles.
- ✓ Able to start and stop vehicle by using Figureprint, when an unauthorized person are detect, only legal person (driver) are registered and recognized.
- ✓ Able to manages for the system users, i.e. Fleet Admin, passengers, and driver

Proposed hardware (IoT device)

This device is designed using the hardware components that are configured and set on inside the vehicle. It has the following abilities: -

- ✓ IoT device is when on auto-connect and monitor and disconnect when off.
- ✓ It is fast performance in the case of monitoring data in a few seconds.
- ✓ It controls the engine state.
- ✓ Getting location position, speed, and push data to servers.

3.2. Non-Functional Requirements

3.2.1. User Interface and Human Factors

The interface of the proposed system is simple to understand; easy to use and user-friendly interface and users of the system easily use and perform their tasks. Everything stems from knowing our users, including understanding their goals, skills, preferences, and tendencies. Once we know about our user, we consider the following when designing our interface:

- ✓ Keep the interface simple. The best interfaces are almost invisible to the user.

- ✓ Create consistency and use common UI elements. By using common elements in our UI, users feel more comfortable and can get things done more quickly, layout and design throughout the site to help facilitate efficiency.

3.2.2. Hardware Consideration

Our system will compatible with different sensor and hardware devices and perform different functionality are done with adding hardware materials, we use different IoT devices to implement smart vehicle tracking systems such as power supply, Arduino, GPS, GSM, wireless network sensor, and accelerometer is the main device. (Saniah Ahmed, 2018)

3.2.2.1 Arduino

Arduino is open-source physical processing that is based on a microcontroller board and an incorporated development environment for the board to be programmed. Arduino gains a few inputs, like switches or sensors, and controls a few multiple outputs. It is an open-source stage focused around a straightforward microcontroller board and an environment for composing programs for the board.

3.2.2.2 GPS module

The global positioning system is a satellite communication system. We can use GPS technology when we are driving. With a GPS receiver, we have an amazing amount of information readily available. Here are just a few examples of how we can use GPS technology. Know precisely how far you have run and at what pace while tracking your path so you can find your vehicle. (How Does Live GPS Tracking Work?, 2020)

3.2.2.3 Figure print

A Control of the Engine (Start / Stop), you can immobilize or Stop a vehicle. This ensures that there is better Safety of the vehicle.

3.2.2.4 ESP2666 module

The ESP8266 is a low-cost module built by Espressif Systems. Its popularity has been growing among the hardware community thanks to it's nice features and stability, to the point that it can be easily programmed using your Arduino IDE,enables microcontrollers to connect to Wi-Fi.

3.2.2.5 Accelerometers/ MEMS

MEMS accelerometers allow to measure speed that are related to their movement and position in the real world.

3.2.3. Security Issues

The system is much secured based on the username and password for all user activity. Nobody can access the system without the authorized person. We custom the Django python framework for backend web development it allows us to develop highly secured applications.

3.2.4. Performance Consideration

The web-based system should have easy and efficient code manipulation and have a clear database. Thus, the response time should be very small i.e. not more than 0.187 seconds. The one that may affect our system performance is when the Fleet Admin adds a new vehicle and it more depends on the user's computer performance.

3.2.5. Error Handling and Validation

IoT sends URL that broadcast in the mobile phone that touches automatically open GPS maps without any problems. On the other Django gives us good URL management (even we can use regular expression for URL) so the URL entered by the user checked and return for a specific page or return page doesn’t exist if the user enters the wrong URL and mentioned to enter correct path. The system will check user inputs to the system to handle the error. It handles and shows errors in a user-friendly manner, without exaggerating the user. We will handle this error on the client-side by using a javascript validator. We will use front-end validation to reduce the loading time of pages. If there is any server-side error, it is only visible on the server log, they are not visible to the user.

3.2.6. Quality Issues

Several problems faced the quality of the system like duplicate data, unstructured data, security issue, hidden data, and inaccurate data these are will be solved by using id, arranged stored in a structured way, simply solved by a framework, use easy UI and accessed by id from database constantly respectively.

3.2.7. Physical Environment

The web application of the proposed system is deployed on a server that is accessible over the internet at a specific address, there is no need for a special hardware device to work on any computer to use the web application.

3.2.8. Documentation

A user-level document is provided that the user can read the document to know how to use the system and what functionality should the system support the user. also, the development process is provided that the user can read to know about the process and what type of model the developer uses to develop the system.

CHAPTER FOUR

4. SYSTEM ANALYSIS

In this chapter, we describe the system model, object model, and dynamic model. In our system Analysis model, are contains three models: Functional, object, and dynamic models. Use case diagrams can describe the functional model. Class diagrams and object diagram describes the object model. The dynamic model can also be described in terms of sequence, activity, and state chart diagrams.

4.1. System model

To describe the abstract models of our system we use the following use case model with each of these use case model presenting of developing abstract models of the system. We can represent our system by using different system models such as use case models, object models, dynamic models, that describe the problem to be solved, and as system models are represented graphically, they are more understandable than more detailed natural language descriptions of the system requirement.

4.1.1. Use case model

The Use Case Model (use case diagram, use case description, use case scenario) is used to define the core elements and processes that make up our system. This Use Case Model captures the functional system components.

4.1.1.1. Use case diagram

It is the diagrammatic representation of our system functionality or the service provided by the system, in the proposed system we identified both actor and use case (service, task, or functionality) as the following.

- ✓ Admin
- ✓ Fleet admin
- ✓ Passenger
- ✓ Driver
- ✓ IoT device
- ✓ Google map

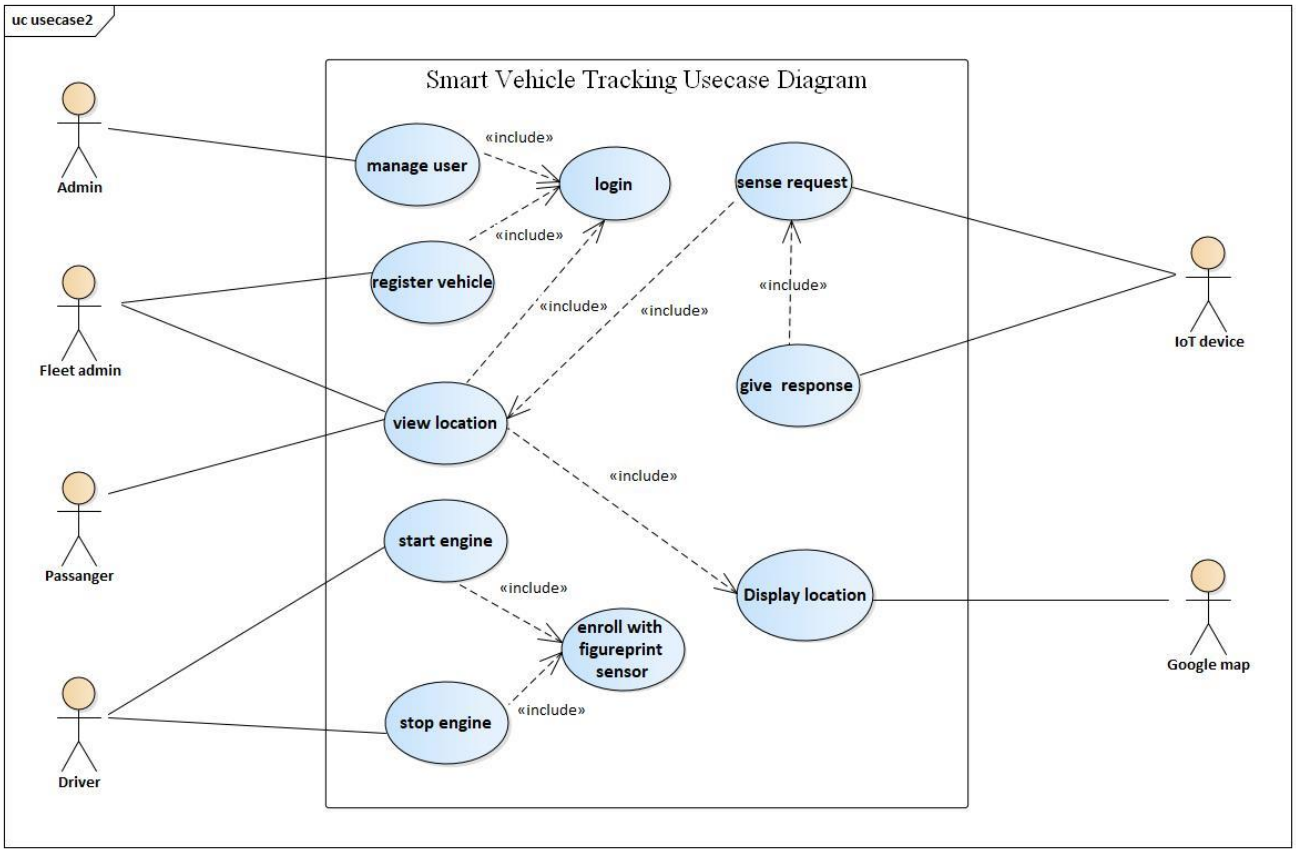


Figure 4. 1 use case diagram

4.1.1.2. Use case description

It is initiated by an actor, provides value to that actor, and is a goal of the actor working in that system. Below, you see a use case description that documents how the actor identified, interacts with the proposed system to perform user and system functionality.

Table 4. 1 Use case Description for Login

Use case id	UC_01
Use case title	Login
Use case description	This use case describes how the user uses the proposed system to log in to the system.
Actors	Admin, Fleet Admin, Passenger and Driver
Pre-condition	Users should have a valid username and password (valid account).
Basic flow of events	1. The use case begins when the user opens the system 2. The user clicks “login” 3. The application show form to enter the user name and password. 4. The user fill username and password 5. Username and password sent to the controller; the controller replies with a go/no go reply telling if the username and password are ok. 6. The system redirects to the user's homepage according to their privilege. 7. The use case ends successfully
Alternative flow	If the entered username and/or password are not correct. 1. The system shows the wrong username or password error. 2. The system does not grant permission to access. 3. The system informs the user to reenter username and/or password. 4. The use case resumes at step 4 of the main sequences.

Post condition	1. Successful completion 2. Failure condition a. The user does not have access to the application if there is no internet connection. b. The system shows an error message for an invalid username or password and stays on the login page.

Table 4. 2 use case description for manage user

Use case id	UC_02
Use case title	Manage user
Use case description	Controlling and identifying the user of the system.
Actors	Admin
Pre-condition	There should be registered user that needs privilege to use the system
The basic flow of events	1.Admin login into the system 2.click manage user button 3.the system displays add user, delete and update user 4. the user is managed successfully 5. Use case ends.
Alternative flow	If the admin enters invalid username and password the system shows an error message

Table 4. 3 Use case Description for Register the vehicle

Use case id	UC_03
Use case title	Register the vehicle

Use case description	This use case describes how the Fleet Admin uses this system to register the vehicle
Actors	Fleet Admin
Precondition	open in the web browser
Basic flow	1.Fleet Admin login 2.the system displays the vehicle registration page 3.Fleet Admin register vehicle with full information including plate number(id), type, and model 4.the vehicle is resisted successfully 5.use case end
Alternative flow	If the entered information is a validation error 2. The system informs the user to fills all required fields correctly.
Postcondition	1. Successful completion of the application creates and adds vehicle to the database. 2. Failure condition the application shows an error Message for invalid personal information.

Table 4.4 Use case Description for Sense request

Use case id	UC_04
Use case title	Sense request
Use case description	This use case describes how the hardware device can sense the request
Actors	Hardware device
Precondition	The hardware device should be fit into the vehicle
Basic flow of events	1The hardware device and the Arduino code connected 2.The hardware device sense the instruction 3.The use case ends successfully
Postcondition	1. Successful completion of the hardware device sense and response to the instruction

Table 4. 5 Use case Description for Give response

Use case id	UC_05
Use case title	Give response
Use case description	This use case describes how the hardware device can response the request
Actors	Hardware device
Precondition	The hardware device should be fit into the vehicle
Basic flow of events	1The hardware device and the Arduino code connected 2.The hardware device response request 3.The use case ends successfully
Postcondition	1. Successful completion of the hardware device response to the instruction

Table 4. 6 use case for view location and speed

Use case id	UC_06
Use case title	View location and speed(track)
Use case description	This use case describes how the Fleet Admin and Passenger track the vehicle
Actors	Fleet Admin, Passenger
Precondition	open in the web browser
Basic flow of events	1. The device fits into the vehicle and captures the GPS location information 2. GPS sending the GPS data to Arduino 3. Arduino generate the raw data to latitude and longitude and send to the website 4. The system will send location details to the Fleet Admin and other authenticate persons. 5. User track the vehicle on google maps. 6. The use case ends successfully

Table 4. 7 use case for display location

Use case id	UC_07
Use case title	Display location
Use case description	This use case describes how the user navigates or see the vehicle location and speed with a google map
Actors	Fleet Admin and Passenger(Employee)
Precondition	The user is should be registered.
Basic flow of events	1.the user login into the website to track its vehicle 2. Press the view location button. 3.the system displays the current vehicle location and speed
Post condition	1. Successful completion and view location

Table 4. 7 use case Engine control

Use case id	UC_08
Use case title	Engine control(Start engine with Figureprint to protect from theft
Use case description	This use case describes how driver are start the vehicle with Figureprint
Actors	Driver
Precondition	The driver is should be a legal person authorized by the Fleet Admin
Basic flow of events	1. Engine controller are enrolled with figure print sensor 2.The hardware device makes sense and verify 3.the engine are start
Postcondition	1. Successful completion

4.1.1.3. Use case scenario

They describe the steps, events, and/or actions that occur during the interaction. Their focus is on understandability. The following Scenarios describe the interaction between the user and the smart vehicle tracking system.

1.Login Scenario Scenario name: Login

Participating actors: MR. Kebede

Initial assumption: power and connection are available. also, the browser is open.

The normal flow of events:

- ✓ Mr. Kebede clicks on the login link on the home page.
- ✓ The System display the login page.
- ✓ Mr. Kebede fulfills the username, password, and then clicks on the Login Button.
- ✓ The System validates the input value.

- ✓ Mr. Kebede has access to his account (logged in).
- ✓ End-use case.

Alternative flow:

The System displays Messages that contain “field cannot be empty or you entered wrong username or password” will be displayed.

Scenario name: vehicle registration

Participating actors: Mr. Abraham

Initial assumption: smart vehicle tracking system should be first displayed.

Normal flow of events:

- ✓ Mr. Abraham clicks on the vehicle register link on the home page.
- ✓ The System display the vehicle registration page.
- ✓ Mr. Abraham fulfils the information that the system request and click on the (vehicle register) button.
- ✓ The System validates entered value from user.
- ✓ The system Registers Samuel and displays a successfully registered message.
- ✓ The System display the home page.
- ✓ Use case end Alternative flow:
- ✓ Mr. Abraham enters incorrect information and the System displays Messages that contain “field cannot be empty or you entered wrong value” will be displayed.

Scenario name: vehicle track.

Participating actors: Mr. Abraham.

- ✓ Mr.Abraham login into the system with the proper username and password
- ✓ The system opens the website with view location button
- ✓ Track the vehicle on vehicle maps.
- ✓ Use case end.

Scenario name: Engine Controller

- ✓ Participating actors: Ato Teshome.
- ✓ Ato Teshome are enrolled his fingure with Figureprint sensor
- ✓ The sensor verify or much his fingure print
- ✓ Ato Teshome can start the vehicle
- ✓ If the Teshome tries to steal with try to open them Unregistered Figure print

- ✓ The sensor makes sense and unable to open the vehicle
- ✓ Use case end

4.2 Object Model

To describe our system in terms of objects and classes. We define the interfaces or interactions between different models, inheritance, encapsulation, and other object-oriented interfaces and features. In the proposed system, we use object models like the class diagrams and data dictionaries.

4.2.1 Class Diagram

Our project class consists of a collection of attributes and methods that determine the state and the behavior of its instances (objects); we identify Admin, Fleet Admin, Passenger, and Driver, user, and IoT device class for the proposed system.

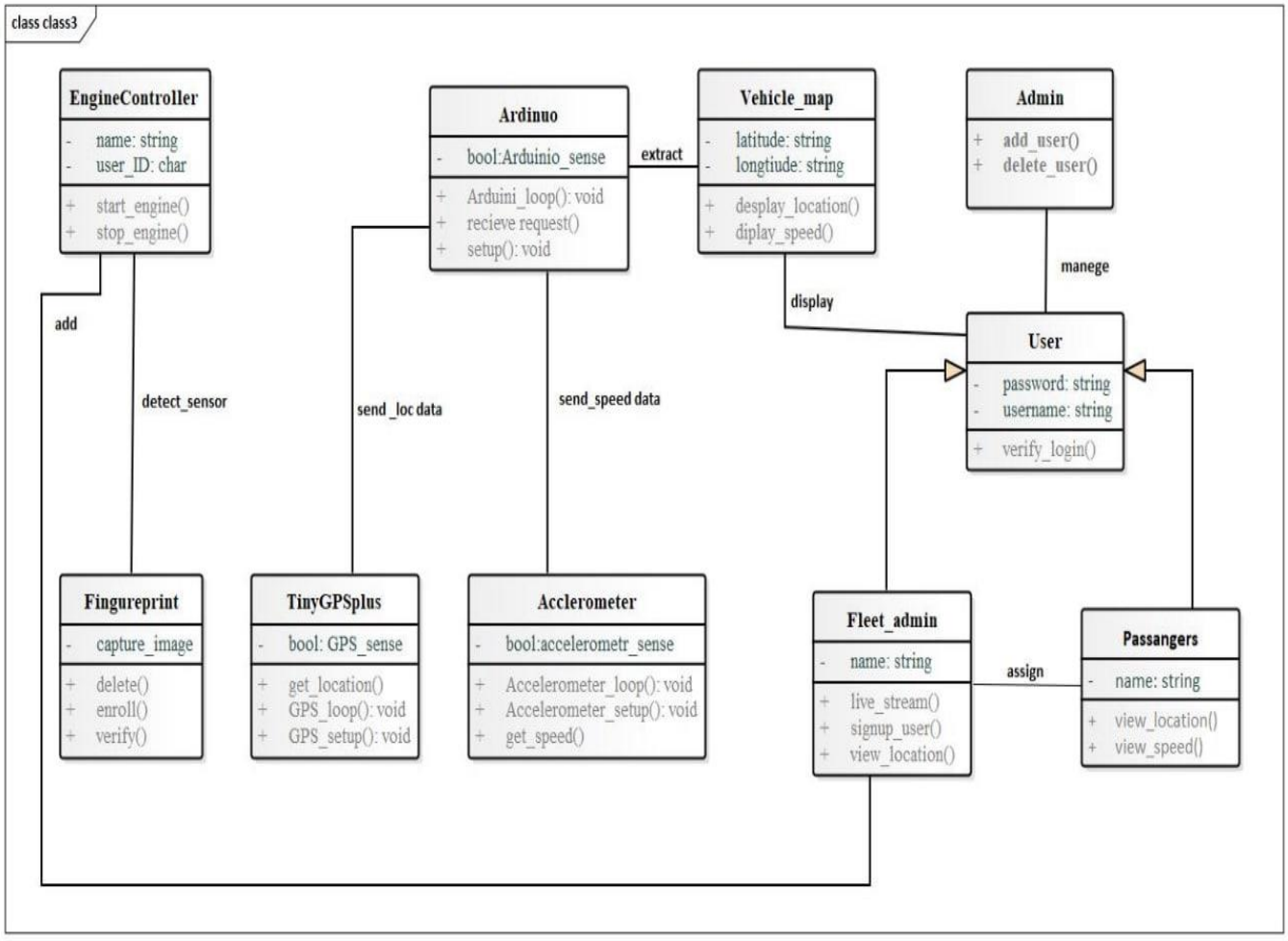


Figure 4. 2 Analysis label class diagram

4.2.2 Data Dictionary

A collection descriptions of our system data objects or items for the benefit of who need to refer to them, objects with which users interact is to identify each object and its relationship to other objects and description of the database table of the system such as functionality, data types, data size, contents item and table relations follows:

Table name: User

Primary key: User id

Description: Used to store account information for the user

Table 4. 9 User Data Dictionary

Field-name	**Data type**	**Constraints**	**Description**
User-id	Int	Primary key	Unique identification of the user
User-name	String (50)	Not null	Unique user name for each user used to login to track and detect the vehicle.
User-password	Varchar (50)	Not null	Associated password for each user

Table name: vehicle

Primary key: vehicle plate-no

Description: Used to store vehicle information

Table 4. 10 Vehicle Dictionary

Field-name	**Data type**	**Constraints**	**Description**
vehicle –id	Int	Primary key	Unique identification of the vehicle
vehicle –name	String (50)	Not null	Unique vehicle name for each vehicle.
vehicle –type	Varchar (500)	Not null	Describe the model for each vehicle

vehicle -model	Varchar (50)	Not null	Describe the model for each vehicle

Table name: Fleet Admin

Primary key: Fleet Admin-id

Description: Used to store Fleet Admin information

Table 4. 11 Fleet Admin Dictionary

Field-name	**Data type**	**Constraints**	**Description**
Fleet Admin –id	Int	Primary key	Unique identification of the user
Fleet Admin– name	String (50)	Not null	Unique-Fleet Admin name for the system is used

Table name: Driver

Primary key: Driver -id

Description: Used to store Driver information

Table 4. 12 Engine controller Dictionary

Field-name	**Data type**	**Constraints**	**Description**
Engine controller –id	Int	Primary key	Unique identification of the Engine controller
Engine controller – name	String (50)	Not null	Driver name for the system is used
Engine controller – fingure image	image	Not null	The fingure print

4.3 Dynamic Model

In this section, we document the behavior of the object model, in terms of sequence, activity, and state chart diagrams.

4.3.1 Sequence Diagram

In this section, we illustrate (diagrammatically) a sequential logic, in effect, and the time ordering of messages of the functionality of the project.

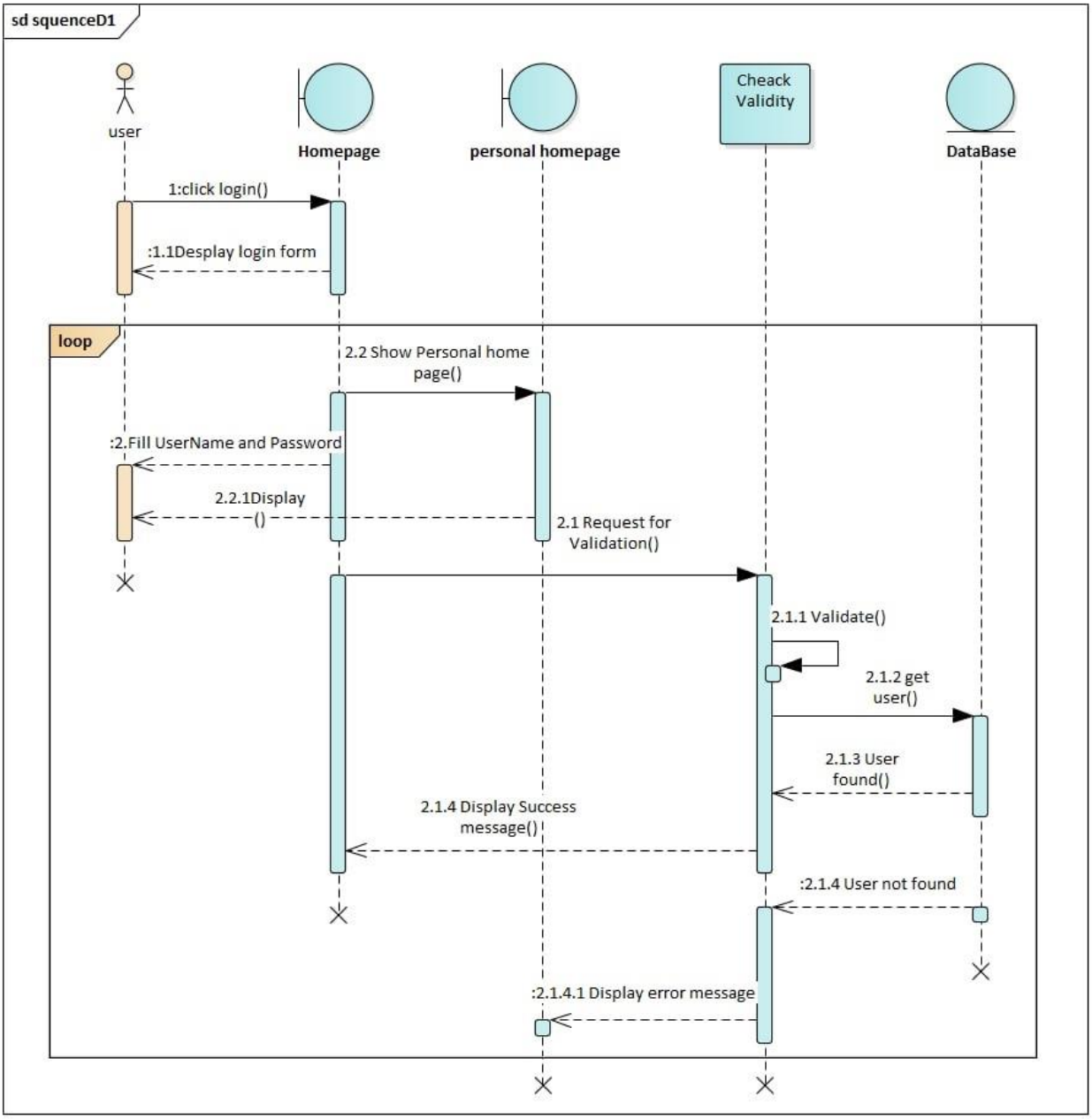


Figure 4. 3 Sequence Diagram for login

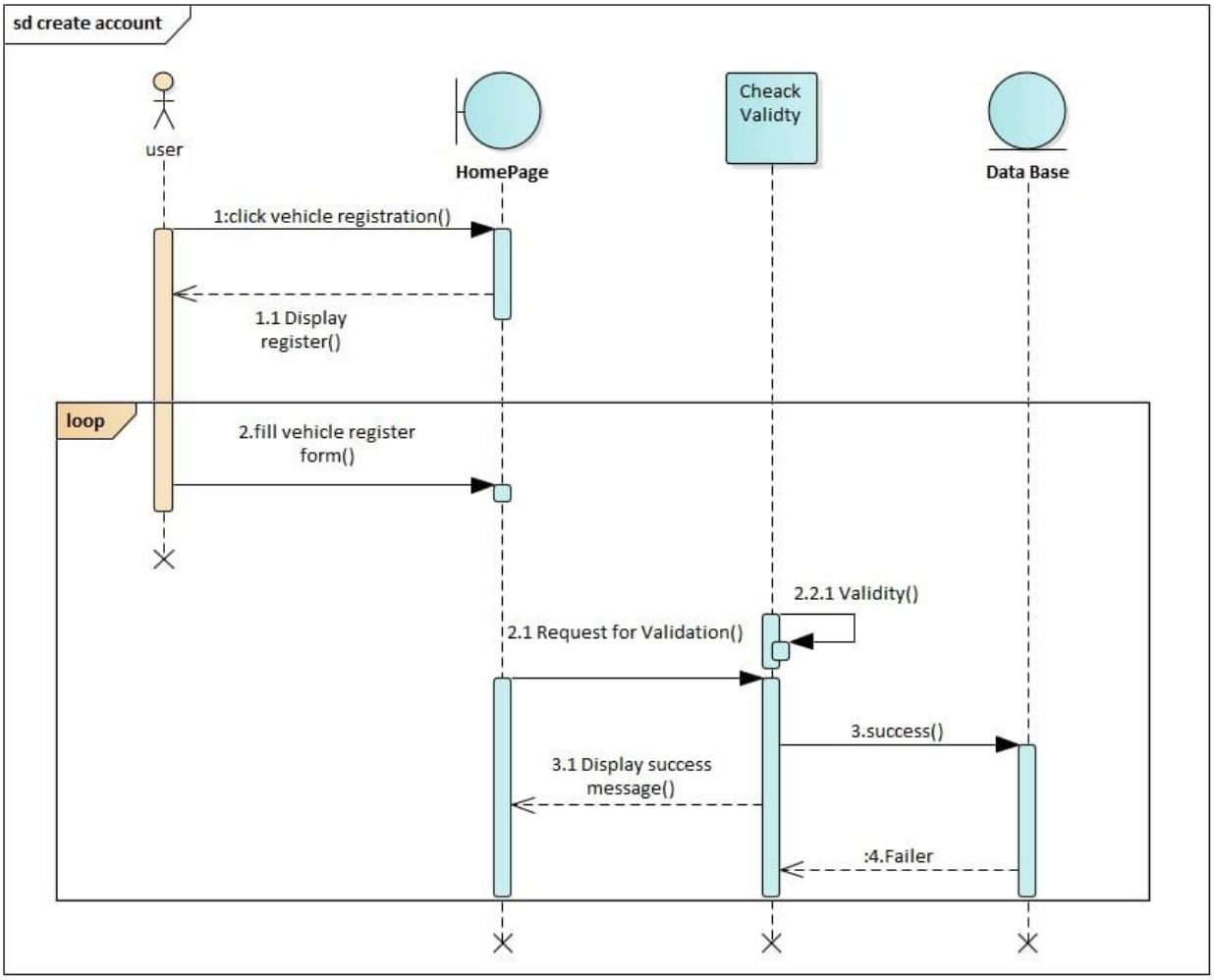


Figure 4. 4 Sequence Diagram for vehicle registration

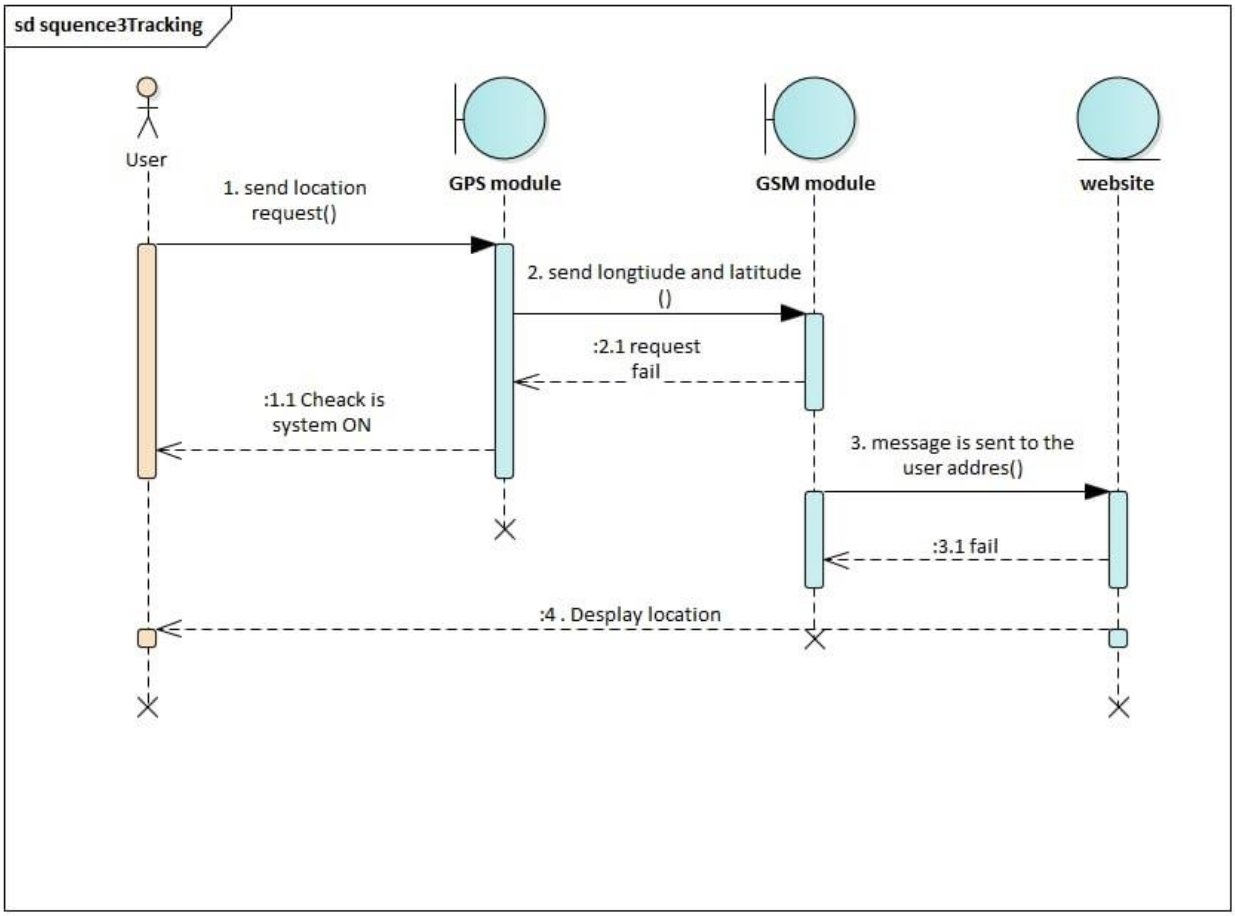


Figure 4. 5 Sequence diagram for vehicle tracking

4.3.2 Activity Diagram

In this section, we are expected to illustrate graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency.

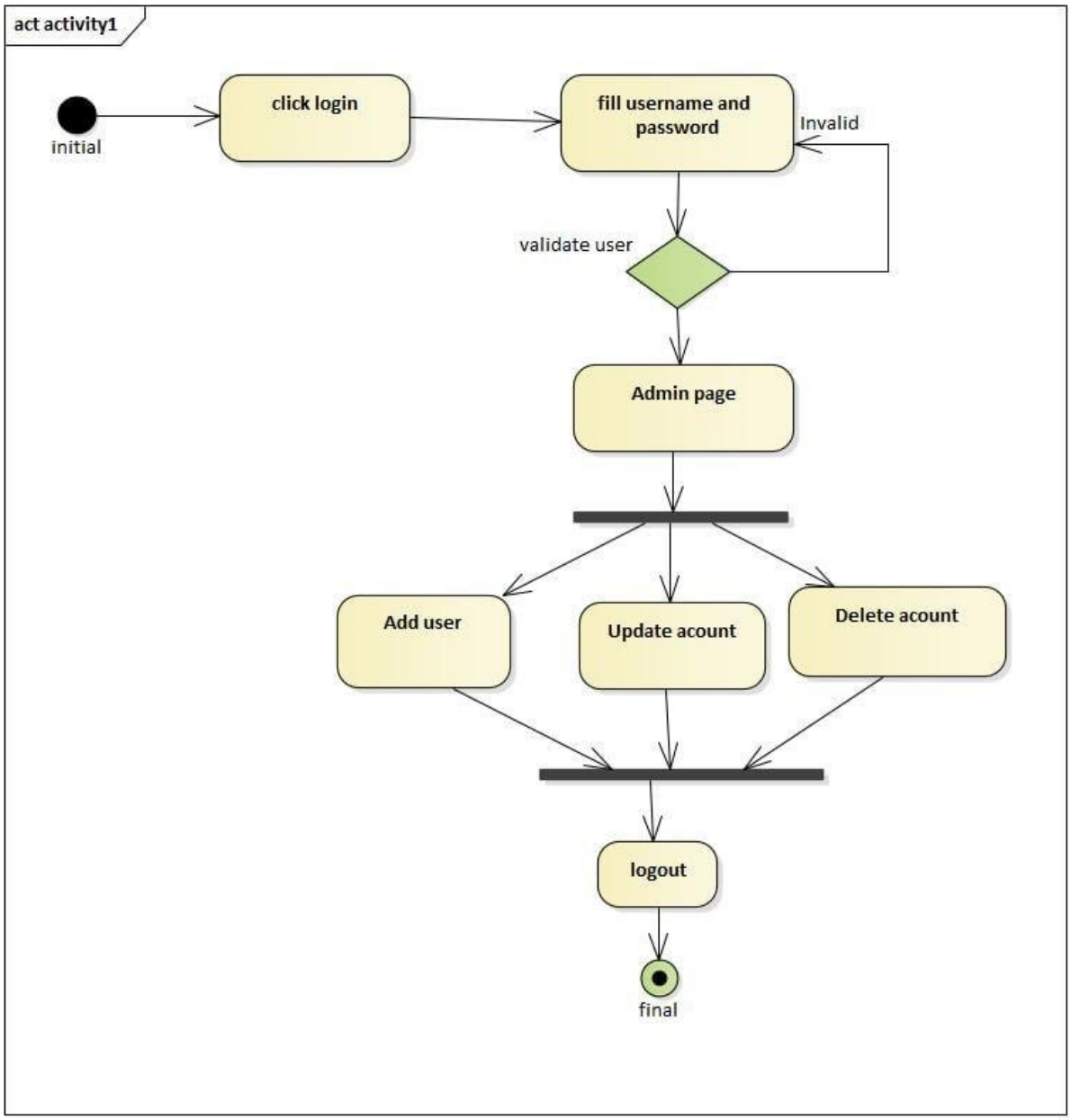


Figure 4. 6 Admin activity Diagram

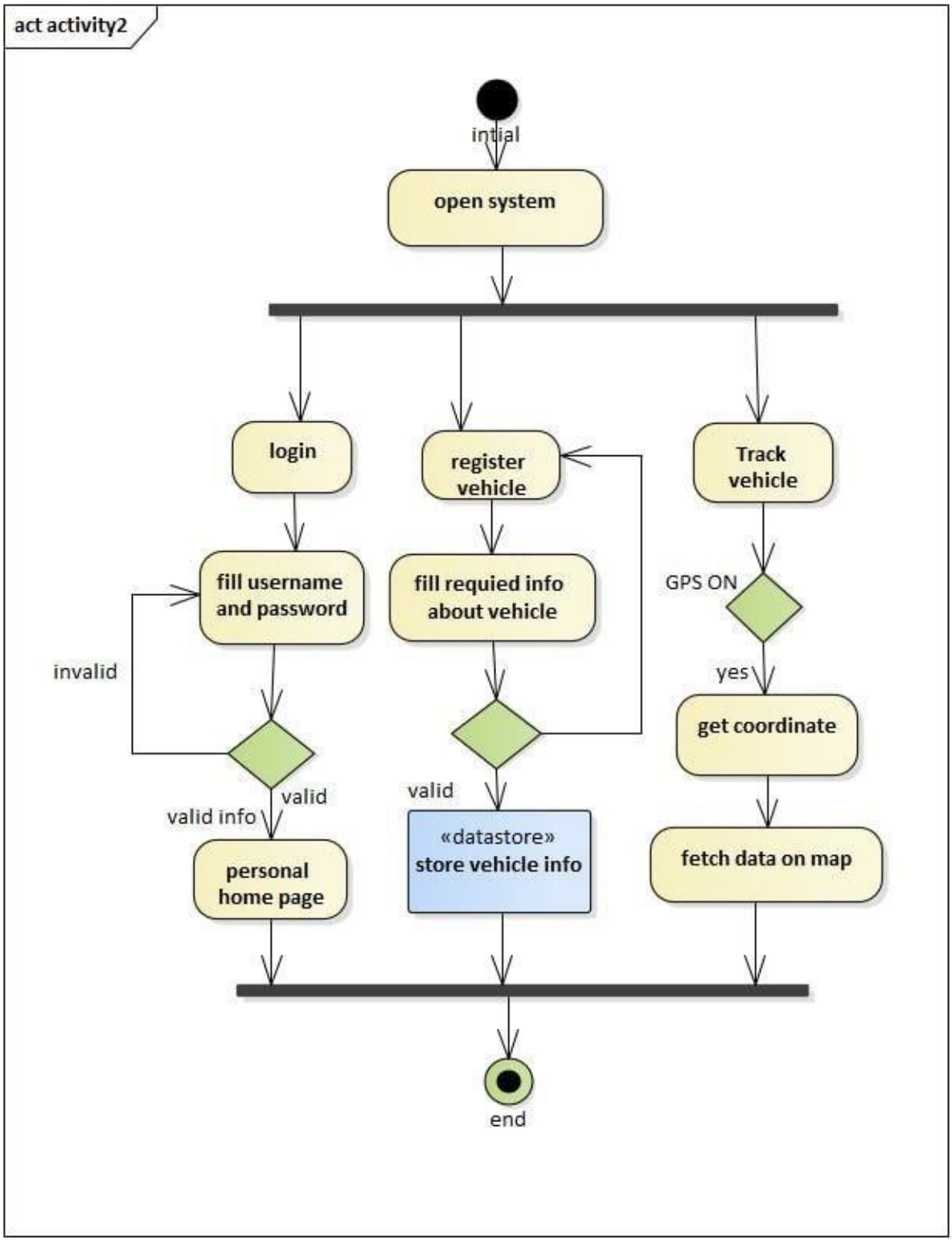


Figure 4. 7 Fleet Admin Activity diagram

4.3.3 State Chart Diagram

Here we are expected to define different states of an object during its lifetime and these states are changed by events.

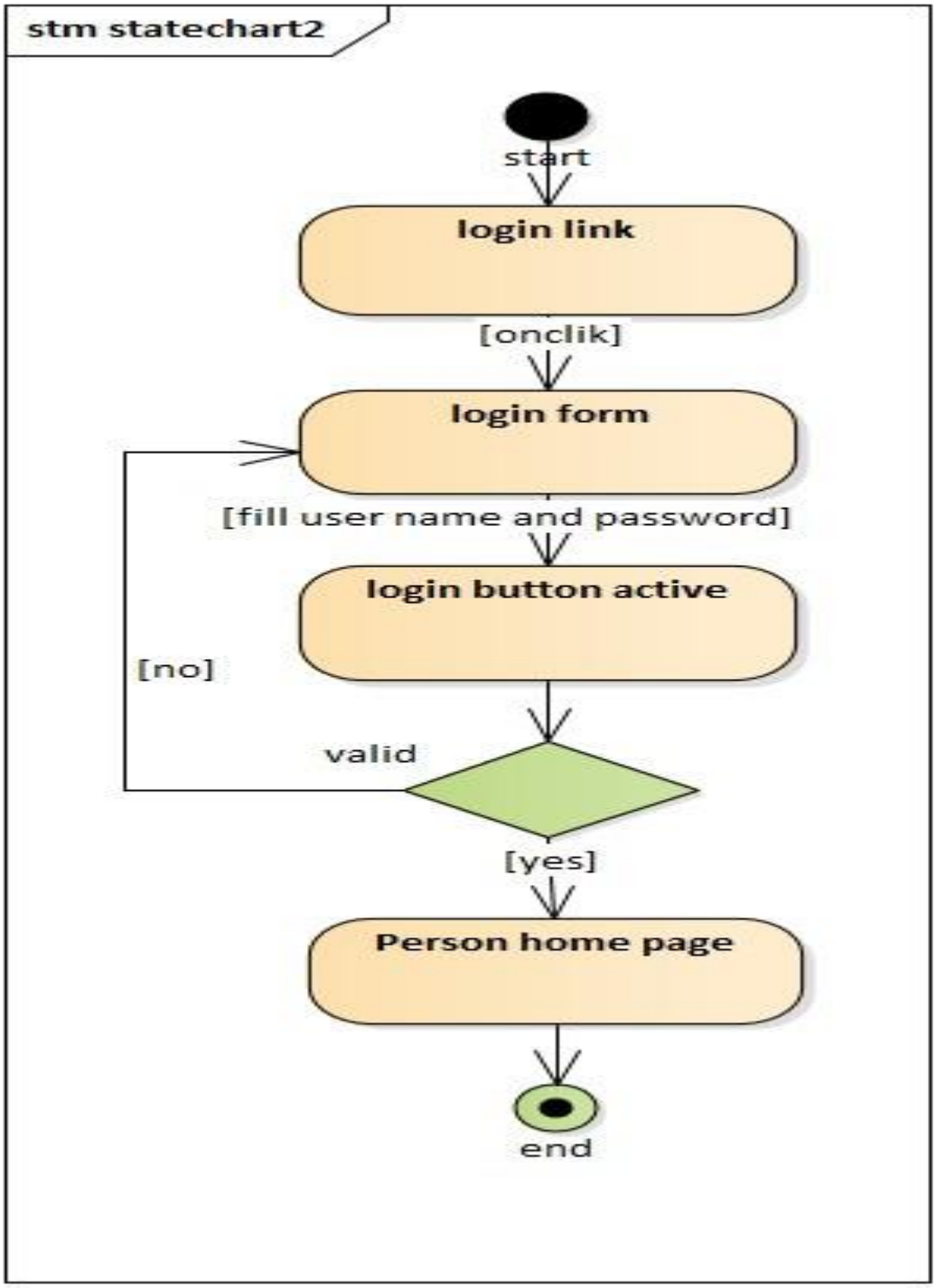


Figure 4. 8 Login state chart diagram

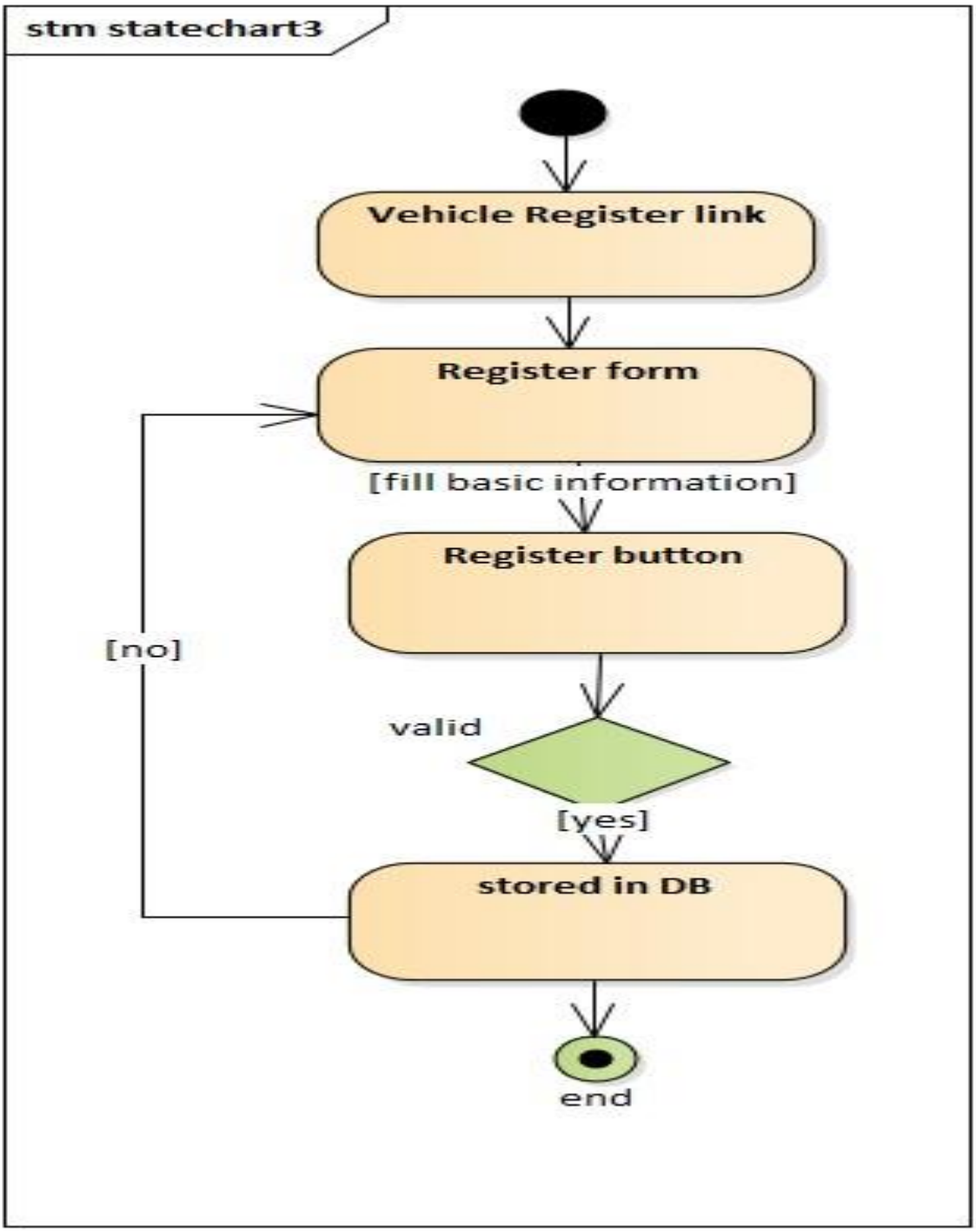


Figure 4. 9 Vehicle Register state chart Diagram

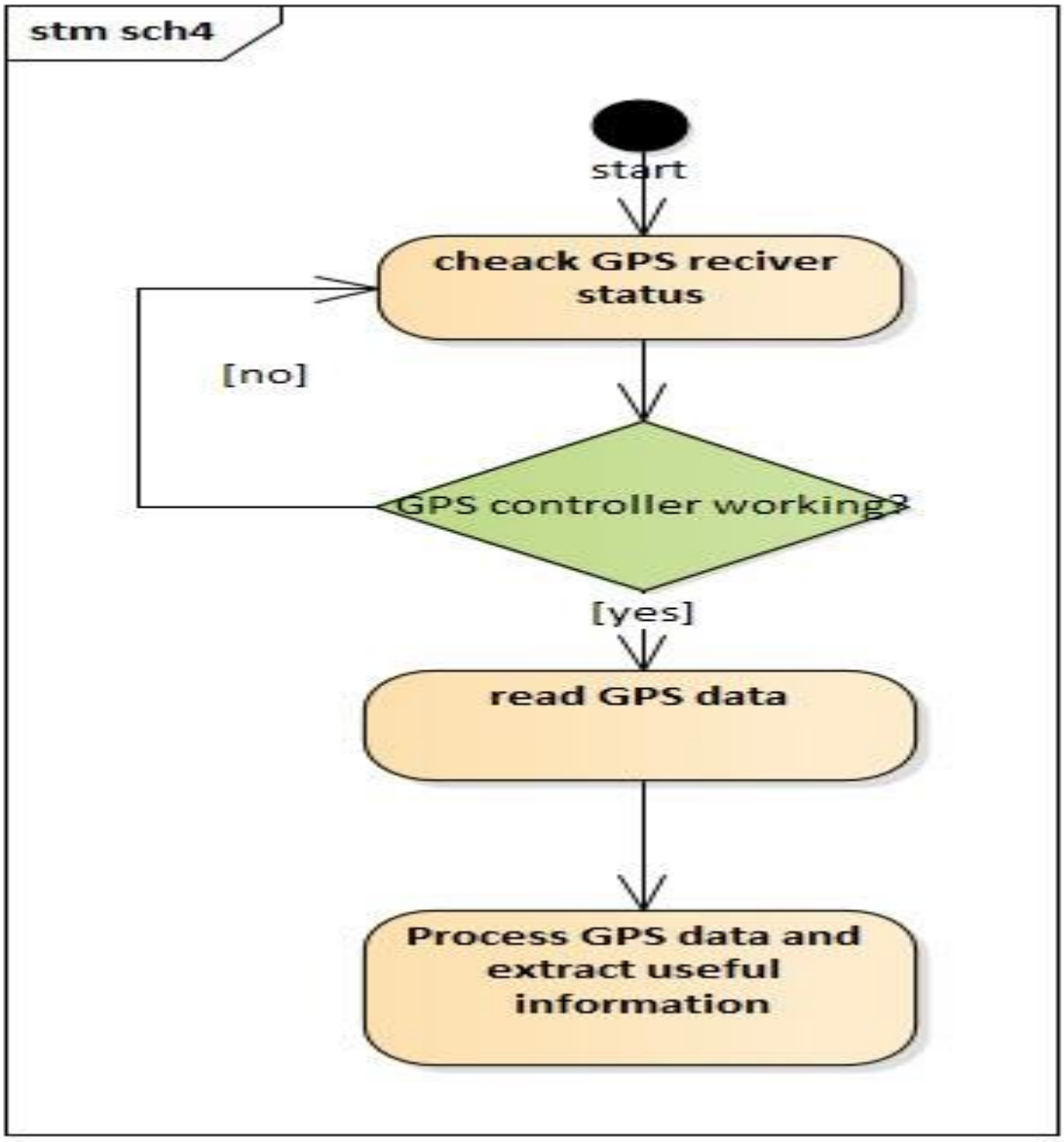


Figure 4. 10 GPS Statechart Diagram

CHAPTER FIVE

5. SYSTEM DESIGN

In the system design, the design aspects of the proposed system will be described. Those are design goals, system architecture, package algorithm design, and user interface design.

5.1. Design Goals

The proposed system design goal is derived from the non-functional requirements of the system, which were stated in chapter three of this document. They describe what the system should focus on. This includes:

- ✓ **User interface:** The user interface of the system should be easy to use by each user of the system with little training. Our interfaces contain Different sizes, fonts, and arrangement of the text, which guides the users on what they need to click. So the system's interface is easy to go through. We also used different icons, buttons, frames, and colors to make the interface more attractive.
- ✓ **Security:** The system will cannot access the system without an authorized person. Passwords that are not visible cannot be accessed by anyone because the passwords are encrypted and protect from hacker using Cross site scripting (XSS), Cross site request forgery (CSRF), SQL injection, Clickjacking, SSL/HTTPS, Host header validation, Referrer policy, Session security.
- ✓ But only legally person use the system
- ✓ **Error Handling and Validation**: Django create URL so, this used to handle the file not found. It handles and shows errors in a user-friendly manner, without exaggerating the user. We should handle this error on the client side by using a JavaScript validator. We should use front-end validation to reduce the loading time of pages. If there is any server-side error, it is only visible on the server log, they are not visible to the user.
- ✓ **End-user Criteria:** Usability is the extent to which specified users achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use can use a product. From the end users' perspective, the system should be designed in such a way that it is easy to learn and use efficiently.

5.2. Current System Architecture

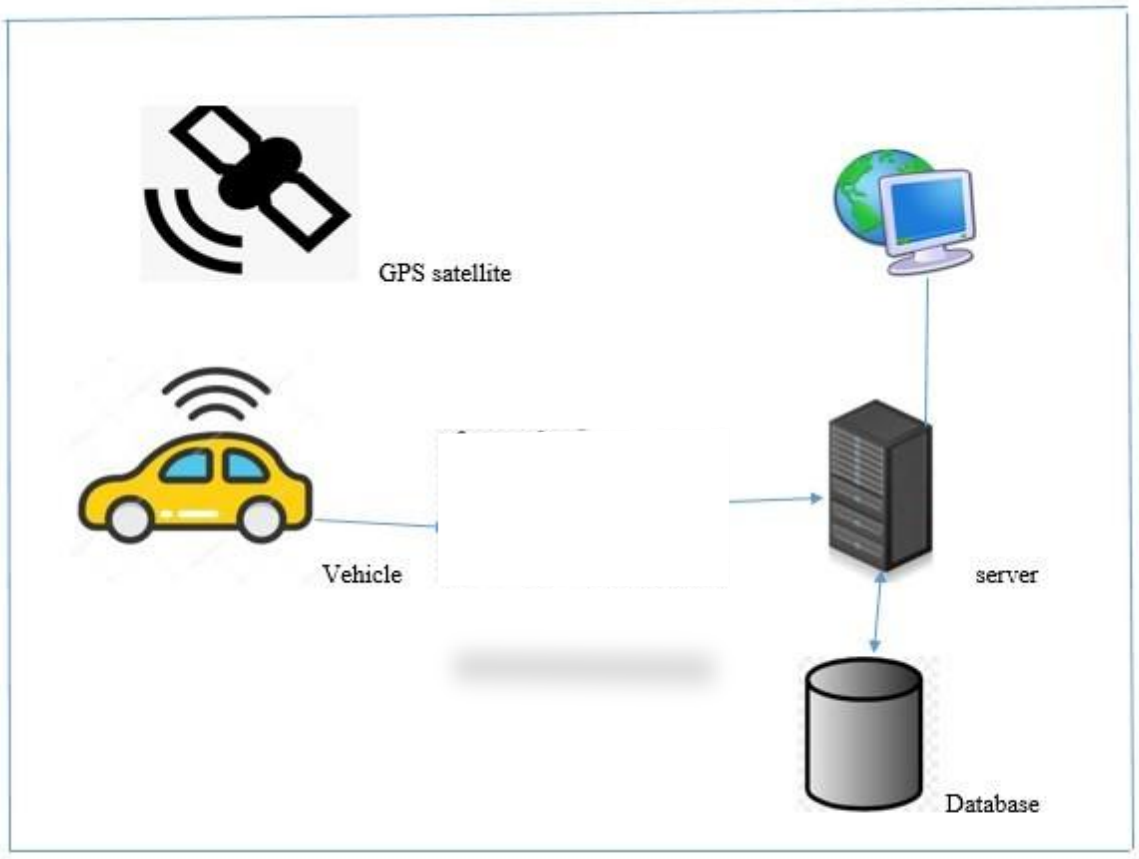
Currently, there is no system architecture because vehicle tracking is carried out manually means. The Fleet Admin carries out vehicle registration, deployment, and all other tasks manually and there is no way for passengers and /or other individuals where the vehicle is.

5.3. Proposed system architecture

Our proposed system is consisting Model View Templet design pattern: The Model helps to handle database, which is used to store data entered by the user, it is the storage of this new vehicle tracking system and stores real-time data for the process of tracking vehicles. User information. And the vehicle database records the information of tracking and tracked vehicles, referring as the number, the type, the manager, and other information related to a vehicle, which can directly come from the basic data of a vehicle presented by its manufacturer In general, a client of our system use browser to access the system using the internet. In this case when the user enters input and takes certain action application server process client request to interact with the database server below is proposed system architecture.

Which handles the data. Template, which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data, it is the interface to our system, which takes information from the user.

Figure 5. 1 Proposed architecture of SVT system



5.3.1 Sub System Decomposition

The proposed system components and their relationships with each other following subcomponent is:

Fleet Admin subsystem- responsible for managing overall activities.

- ✓ Register vehicles
- ✓ Add vehicles
- ✓ update info
- ✓ view

Track subsystem-responsible for detecting the location of the vehicle.

- ✓ Track location
- ✓ Track speed
- ✓ Live streaming

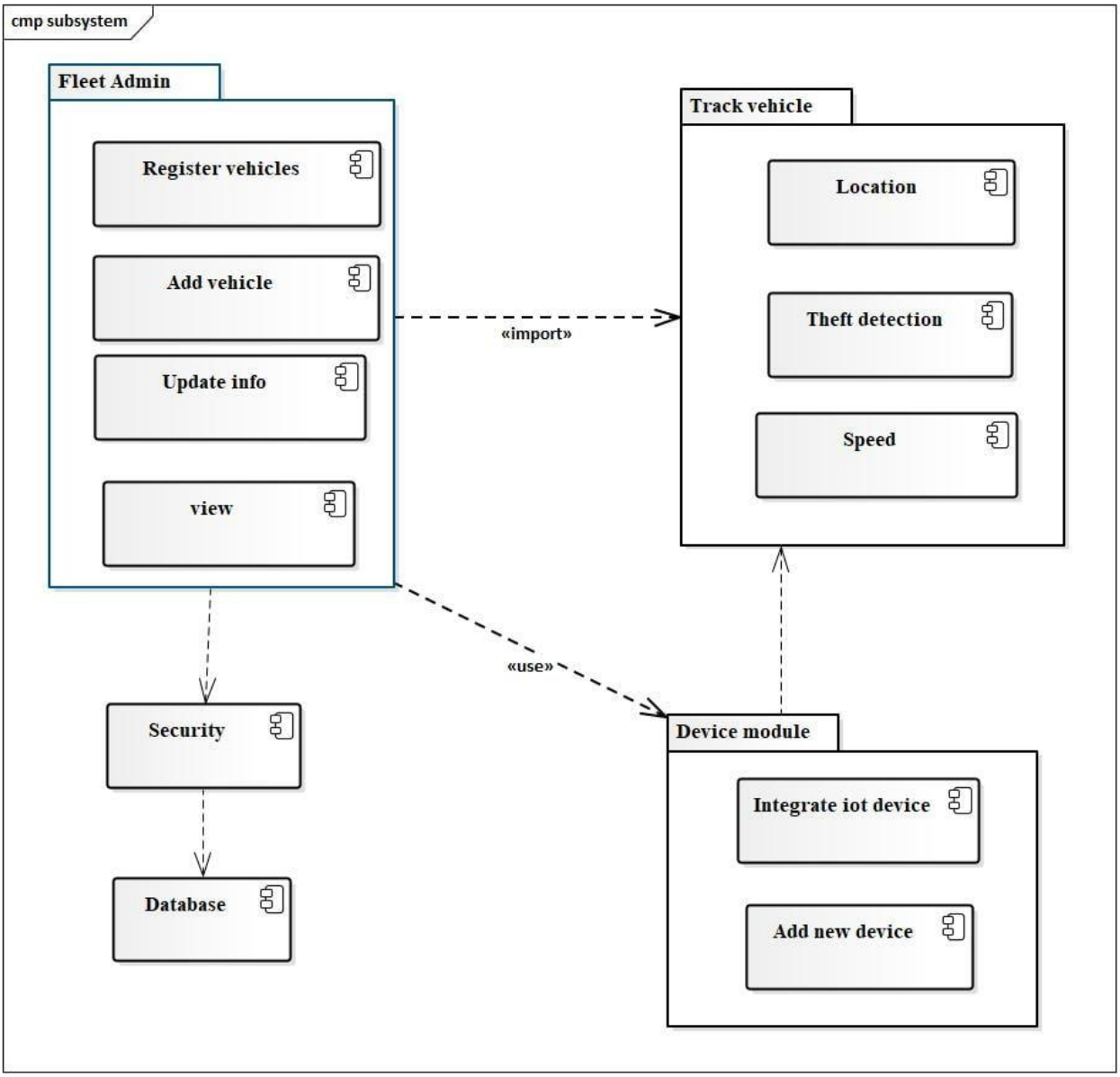


Figure 5. 2 Subsystem Decomposition Diagram

5.3.2. Hardware/Software mapping

To shows how the software and hardware components work together. Deployment diagram is used to show the hardware of the system, the software that is i **Live streaming** , and the middleware that is used to connect the disparate machines to or hows the deployment diagram of the proposed system.

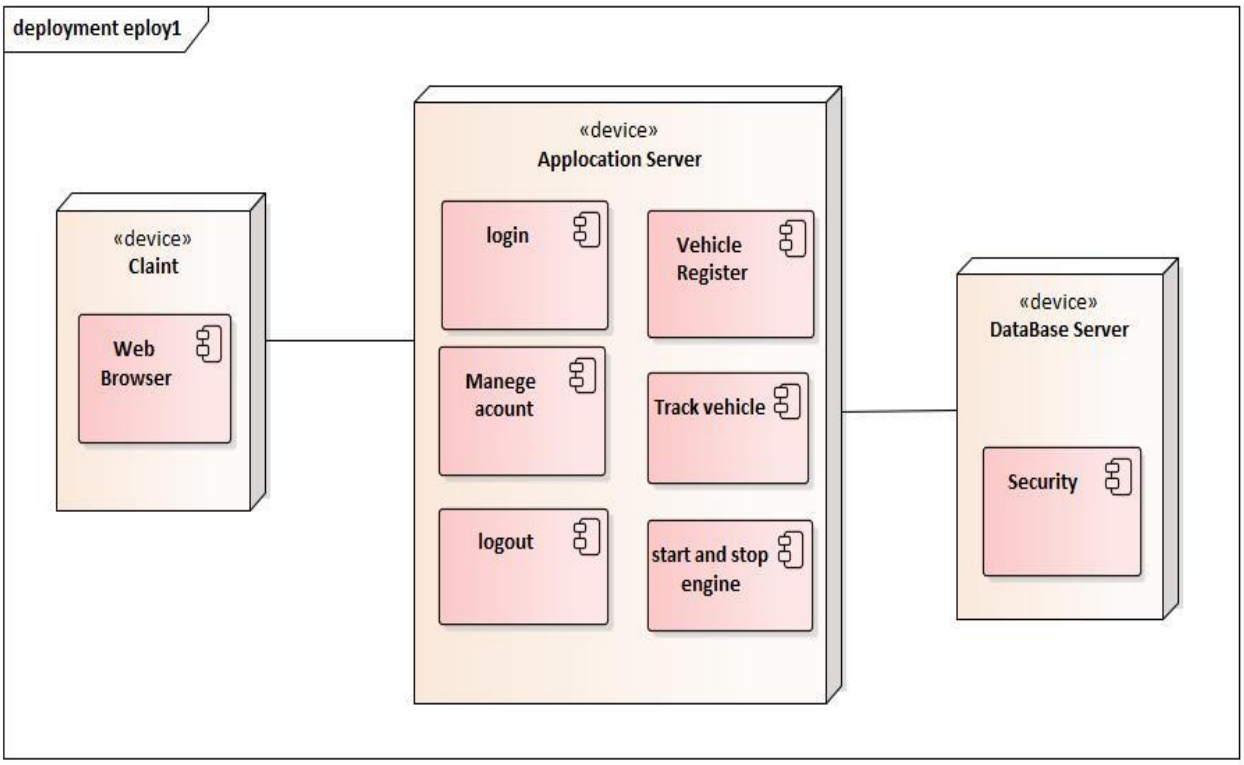


Figure 5. 3 Hardware/Software Diagram

5.3.4. Detailed Class Diagram

The proposed system detail class diagram are includes attributes, methods, attribute data types, visibility of attributes and methods, inheritance, association, and municipality (cardinality and optimality). The following figure uses the UML attributes and operations with their visibility information. The following figure is the detailed class diagram of our system

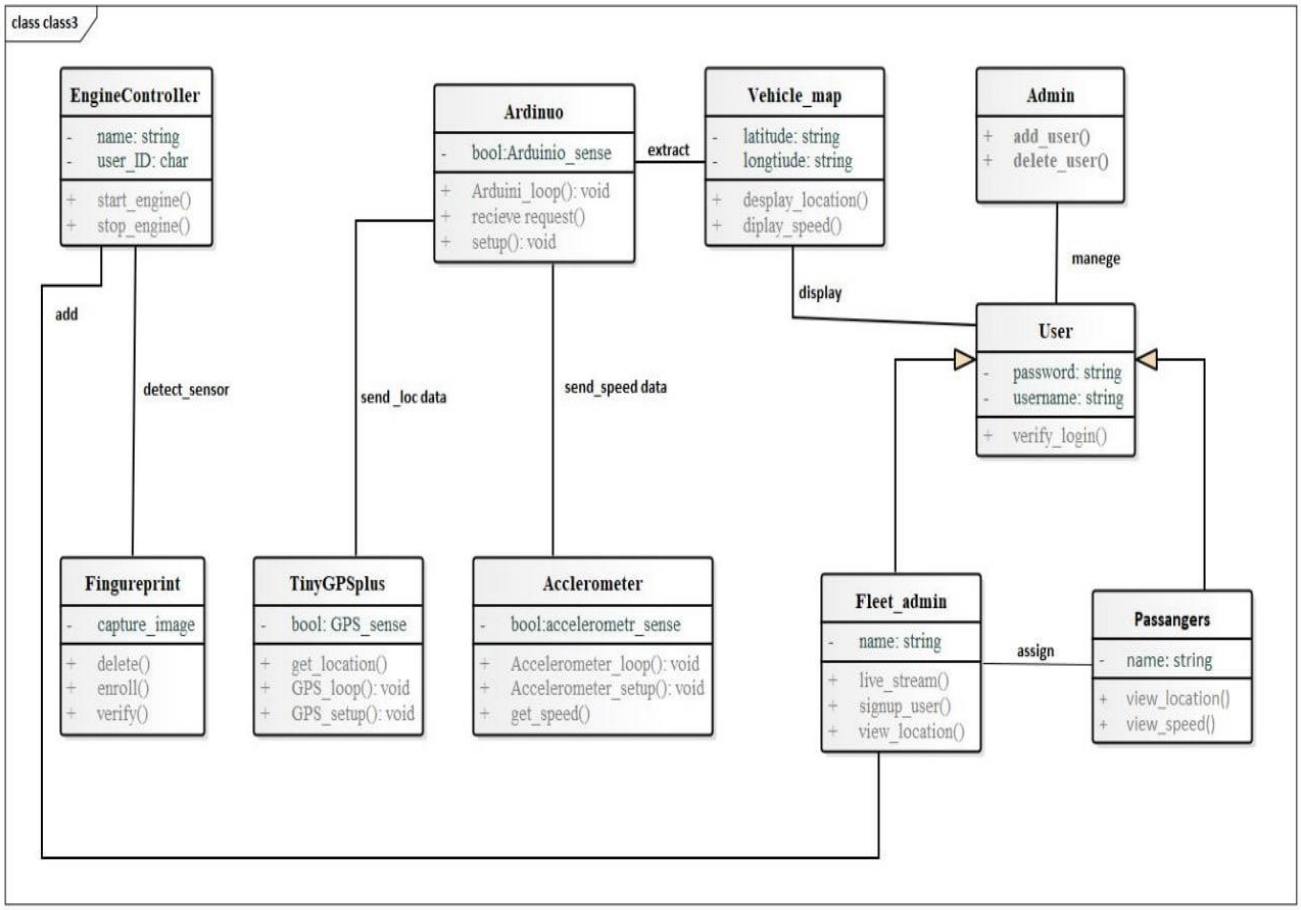
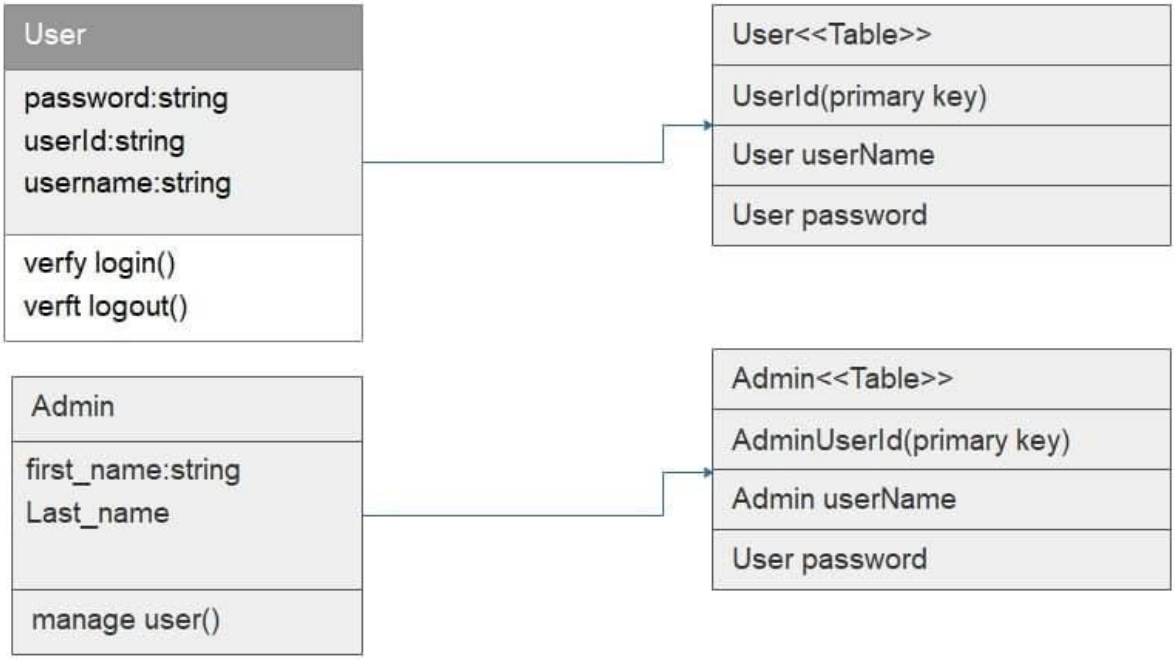


Figure 5. 4 Detail class Diagram

5.3.5. Persistent data management



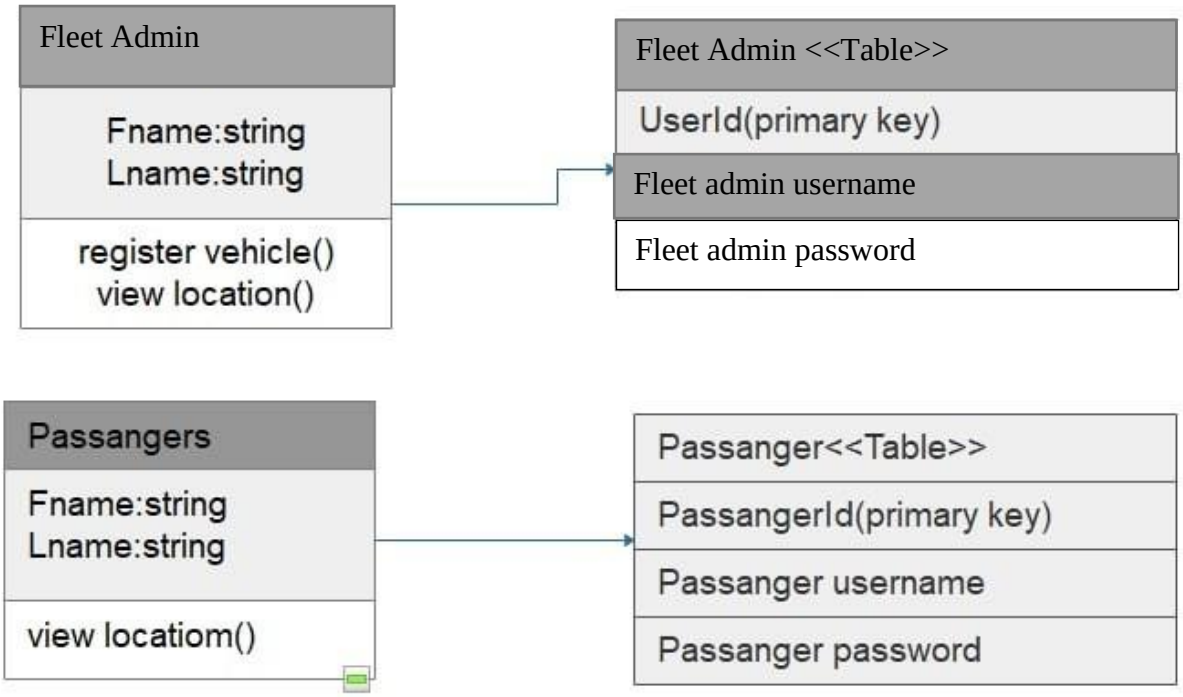


Figure 5. 5 Persistent Data Diagram

5.3.6. Access Control and Security

In our system, different actors have access to different information and data. Access control and security specify what the user can access or what cannot perform by some users. This access control is verified by username and password. System admin represents an authenticated user. Different actors have access to different functions and data. Then it must be having: -

- ✓ Confidentiality: Only an authorized person can see the information. Private data is kept private; personal privacy is respected.
- ✓ Integrity: There are limits on who can change the data in this system.
- ✓ Availability: The system is available at all times to authorized users.

5.4. Packages

In the proposed system we category into User management package, vehicle tracking package, security management package, and database management package.

- ✓ Database management package: - The database subsystem will be implemented by the relational database management system which is used to store the persistent data of the proposed system.
- ✓ Track vehicle package: - It is a subsystem that helps to track the vehicle with its location, speed, and theft detection with GPS and GSM /GPRS,

- ✓ Security management package: - It is a subsystem that helps to secure the proposed system.

The following figure represents the general package diagram of the proposed system.

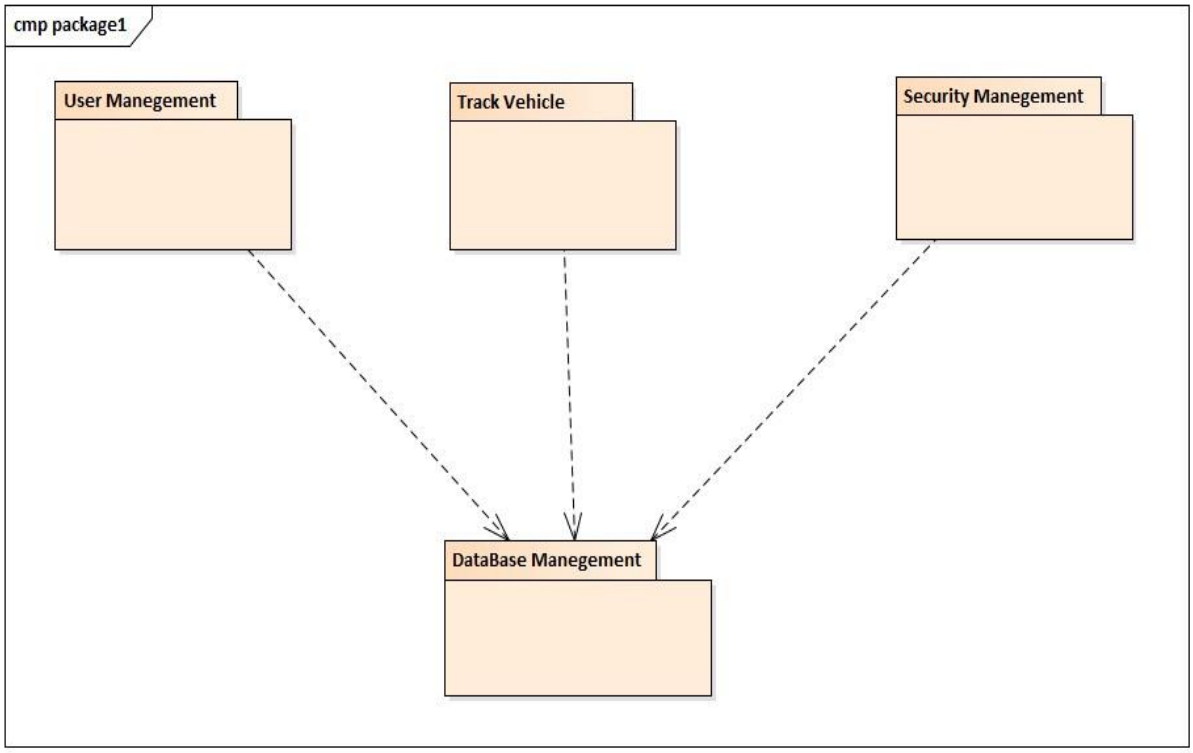


Figure 5. 6 Package Diagram

5.5. Algorithm Design

Login

Home page displayed.

User click on sign in link

System display sign-in page.

User enters user email and password.

If the user email and password are correct,

If **a** user is Fleet Admin, then the system display the Fleet Admin page.

Else if **a** user is Admin**,** then the system display the Admin page.

Else user's email and password are not correct, then the system display an error message and redisplay the login page.

5.6. User Interface Design

To make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design) in our project smart vehicle tracking system we have designed user interfaces that increase the user experience.

Figure 5. 8 Login user interface

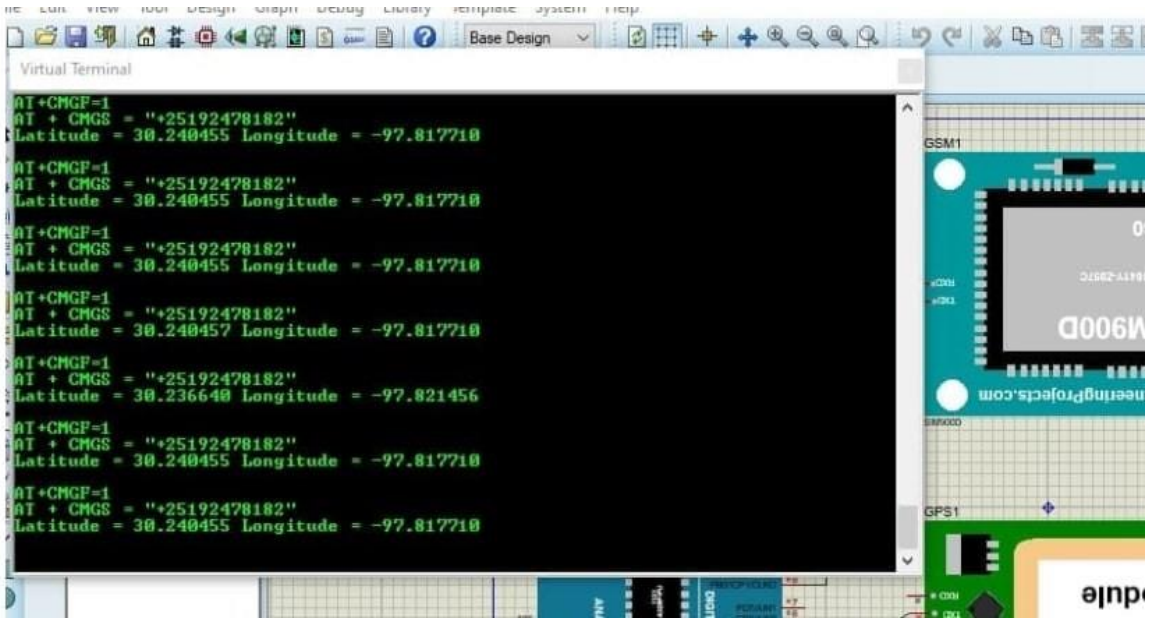


Figure 5. 7 Simulation Diagram for Tracking

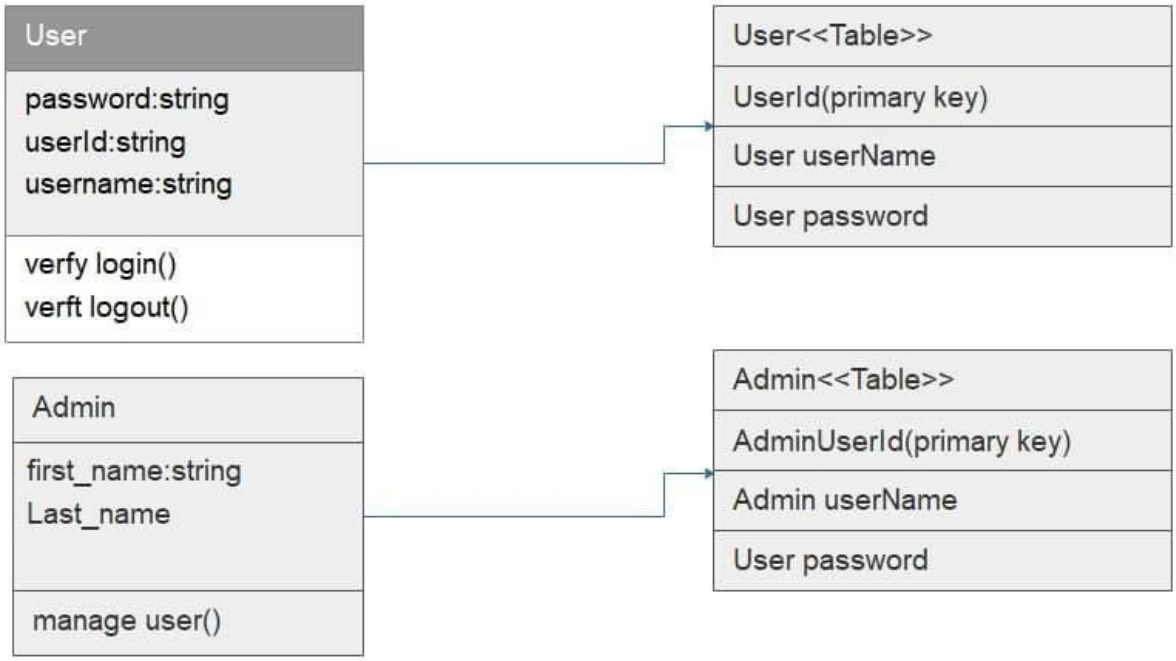
CHAPTER SIX

6. IMPLEMENTATION AND TESTING

During this phase we design physical specification is turned into working computer code, and provide help for current and future users and take care of the system. Then the code is tested until most of the errors have been detected and corrected. The purpose of this activity is to convert the final physical system specification into working model with reliable software and hardware.

6.1 Implementation of the Database

We have use SQLite database management system for the implementation of the database. Because SQLite is most stable, secure, reliable and higher performance than the rest; it takes lower time



6.2 Implementation of the Class Diagram

In class implementation the following activities we have perform:

- ✓ We define all methods with appropriate return type, parameters and the corresponding data types.
- ✓ Implementation of Figureprint class that are used engine controller that start the engine with Figureprint method are:

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

```
void finger_setup()
{
enroll_setup();
match_setup();
delet_setup();
}
void finger_loop (){
enroll_loop();
match_loop();
delet_loop();
}
void enroll_setup(){
}
void enroll_loop(){
}
void match_setup(){
}
void match_loop(){
}
void delet_setup(){
}
void delet_loop(){
}
```

6.3 Configuration of the Application Server

We have been used SQLite as application server because it is an incredibly popular single connection, local database because it provides a stable, flat-file database solution for application developers. SQLite is an in-process library that implements a self-contained, server-less, zero-

configuration, transactional SQL database engine. We choose SQLite because it is a very high-level language , and SQLite are transactional. SQLite is often faster than direct low-level input output

6.4 Configuration of Application Security

Since our system involves storing of some data, we put some security mechanisms like unauthorized person cannot login into the system because the system requires a user name and password. Since we use Django framework to develop the web part of our system. So securing this website mean make the whole system.. Since the system developed to the users who may seneor to computer or may professional on computer systems so all inputs must be implemented easily and simple to use. When the user enters invalid inputs or empty, the system notifies to the user to inter valid inputs. In order to secure our system, we have been performed the following activities: -

- ✓ User accounts was assigned with necessary access privileges
- ✓ Django provides a flexible password storage system and uses PBKDF2 by default.
- ✓ We use fingure print

6.5 Implementation of User Interface

Regarding to the user interface, we have to apply the following.

- ✓ The user interface that we develop is user adjusted design. It means we have to make our user interface be attractive and eye catching for users by developing compatible, well matched and friendly user interface.
- ✓ The user interface that we develop is consistent and dependable
- ✓ Reduce user's memory load –our user interface can be easily remembered for users in long period of usage and it does not require any retention to use
- ✓ The user interface is consistent and stable which does not create any confusion for users easily understandable with clear and steady navigation.

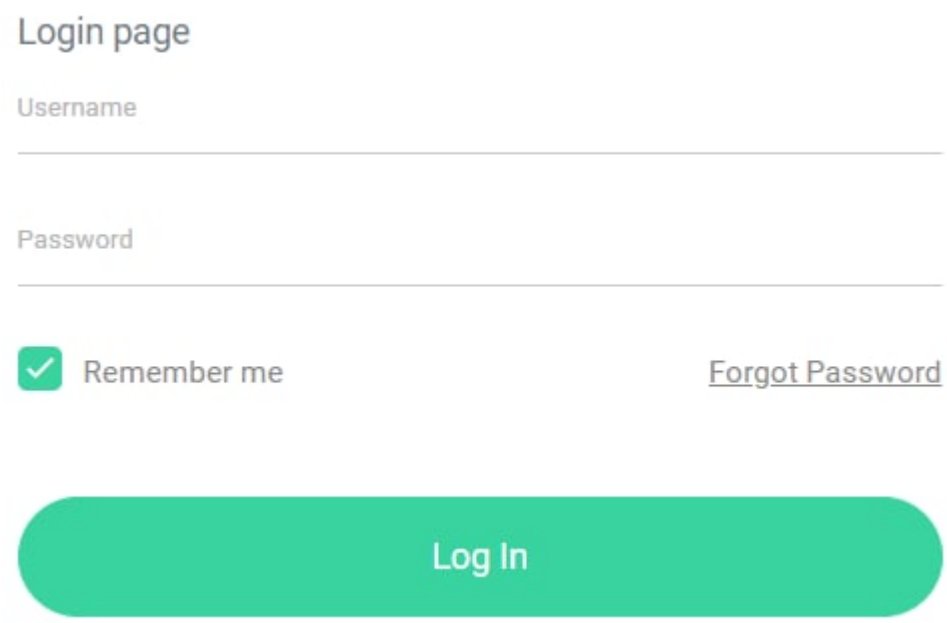
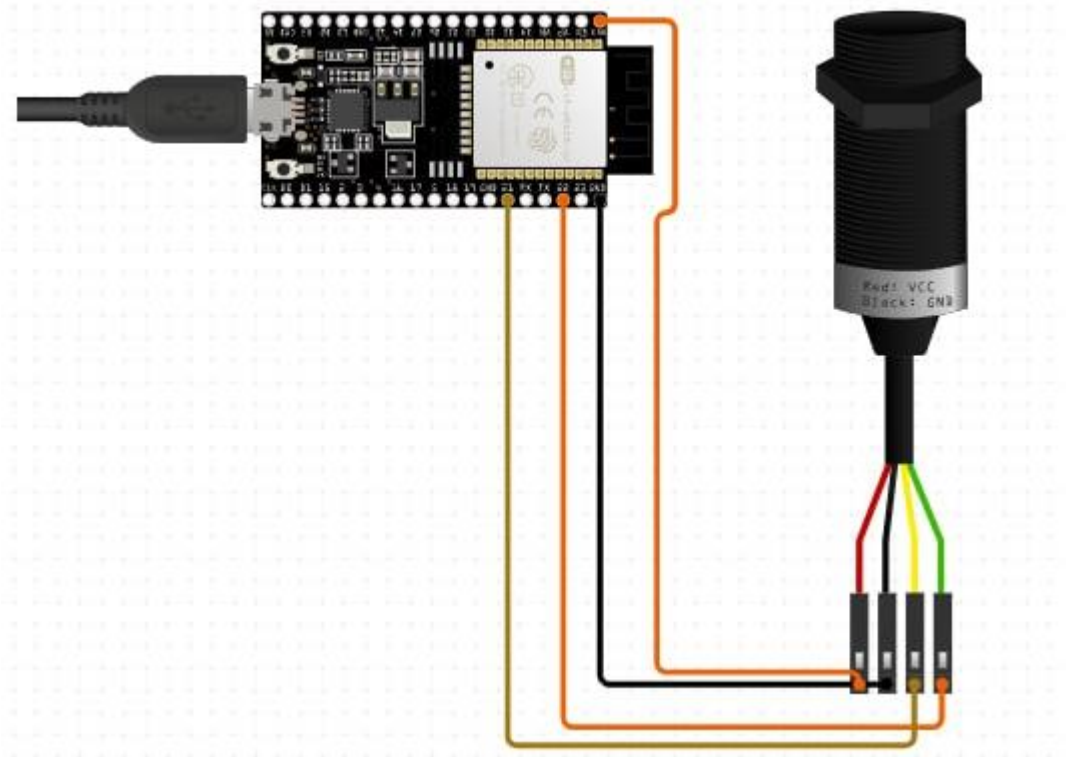


Figure 6. 1 Login page for smart vehicle tracking system website

Figure 6. 2 live streaming of the vehicle


Red: VCC
Black: GND

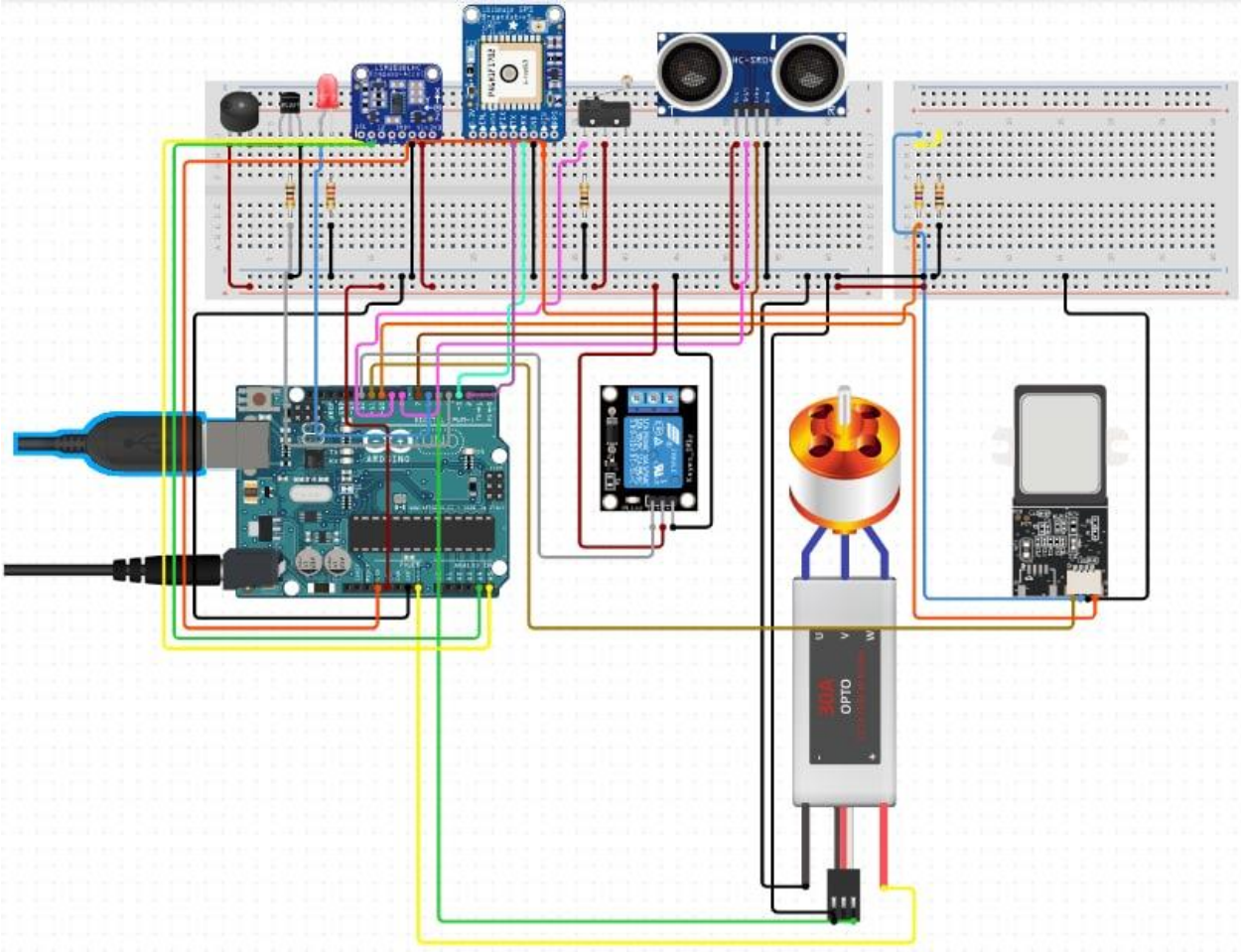

OPTO

Figure 6. 3 Block Diagram for smart vehicle tracking system

6.4 Testing

In this section and subsections, you are expected to describe which kind of testing technique you are going to apply and should mention the test results.

6.4.1 Test Case

In our Project Features we have tested following.

- ✓ Input output functions. Checking what type of our system should be taken as input and produce as output and checking input is produce expected output.
- ✓ Hardware materials are well configured.
- ✓ Security. Identifying security of our system by identifying only identified user allowed access our system and ensuring password of each individual user not seen by another. This thing is the thing we consider on login capability.
- ✓ User interface and database interaction. Database is one that store data and user interface one that user data is entered, so during our test we try to identify data entered on user interface stored in database and another crud.

6.4.2 Testing Tools and Environment

Hardware testing tools

- ✓ Computer to run and execute to test the system
- ✓ Volt meter to measure the voltage of device power
- ✓ Web browser to test the website

Software is testing tools

- ✓ Circuit. I/O website to test is an online tool for designing electronic circuits.
- ✓ Operating system and Windows 10

6.4.3 Unit Testing

Sample we test in our system in unit testing are:

- ✓ Test the circuit of the hardware before the power supply and upload the code.
- ✓ Test input/output configuration
- ✓ We verify each Arduino code
- ✓ use to test the functionality of each hardware
- ✓ Check whether entered input is validated correctly.
- ✓ Check how the sub procedures or functions are call correctly.

- ✓ Check if the correct output is produced for different inputs.
- ✓ Check the input data that we write on the GUI must be submitted to the database.
- ✓ Check the GUI can access the privileged data from the database.

6.4.4 Integration Testing

In this level of testing, we have examined how the different procedures work together to achieve the goal of the subsystem. Therefore, we integrate each component from single function to the main function incrementally.

Used to test the circuit, Arduino, web to check the input-output data is valid or not and modify it to the correct way.

Sample Tests

- ✓ Check the location tracking device send the longitude and latitude website or mobile app map.
- ✓ The Arduino chekESP32CAM are connect with Wi-Fi or not.
- ✓ Check the ESP32CAM live streaming status red or not and store data in local DB within 5 second.

6.4.5 System Testing

In this level of testing process, we have examined how the whole smart vehicle tracking subsystems came together to achieve the desired goal (user's requirements of the system).

Sample Tests

- ✓ Check the overall functionality of systems that achieves the user's requirement.

Sample Tests

- ✓ Our system can be integrated with configured IoT circuit and web application that implements the problem.
- ✓ Evaluate the functionality of the subsystem after a combination of individual subsystems whether it works correctly or not.
- ✓ Verify the system completeness-based user's requirement.

6.4.6 Acceptance Testing

The customer will participate on the acceptance testing of the system. A system is said to be accepted if and only if the user of the system is satisfied. According to system requirements and

other resources (documentation, source code) test cases are generated to determine (validation and verification) whether the system satisfies users need and expectation to solve user's problem.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1 Conclusion

The project entitled smart vehicle tracking system for WKU was successfully executed using Arduino, GPS modules, Figureprint, and MEMS, ESP2666, and camera. In this project, a device was developed that would track vehicle location and send the information to a control unit at the user side.

The main objective of this project work, which is design and construction of smart vehicle tracking system, are achieved. From the results obtained, the system can be seen to effectively track vehicle location, speed and GPS tracking can tell you where each vehicle is at any time and real-time camera lets you look inside the vehicle remotely. In order to achieve security vehicle, the system will provide Figureprint authenticate with legal person that start the engine of the vehicle. This system is becoming increasingly important in large cities and it is more secured than other systems.

7.2 Recommendation

Based on the title of smart vehicle tracking system, the following recommendations were suggested for further project;

- ✓ The documentation format will have its own guidelines, IoT based with Block diagram and hardware and software consideration but not the same existing web format.
- ✓ If you are interested to done a Project with IoT, first you consider the hardware materials exist or able to buy form market.

The smart vehicle tracking system will add the features:

- ✓ With the help of high sensitivity vibration sensors, able to detect the accident. Whenever vehicle unexpectedly had an accident on the road with help of vibration sensor, we can detect the accident and we can send the location to the owner.
- ✓ The system report the fuel status of the vehicle.
- ✓ Able to detect alcohol intake of the driver

References

engg, N. D. (2013, June 14). Retrieved from VEHICLE TRACKING SYSTEM USING GPS AND GSM MODEM: https://www.projectsof8051.com/vehicle-tracking-system-using-gps-and-gsm-modem/

How Does Live GPS Tracking Work? (2020, 03 20). Retrieved from https://www.trackyourtruck.com/blog/how-does-live-gps-tracking-work/: https://www.trackyourtruck.com/blog/how-does-live-gps-tracking-work/

RAKESH, K. (2014). VEHICLE TRACKING AND ACCIDENT ALERT. *National Institute Of Technology Rourkela*, 34.

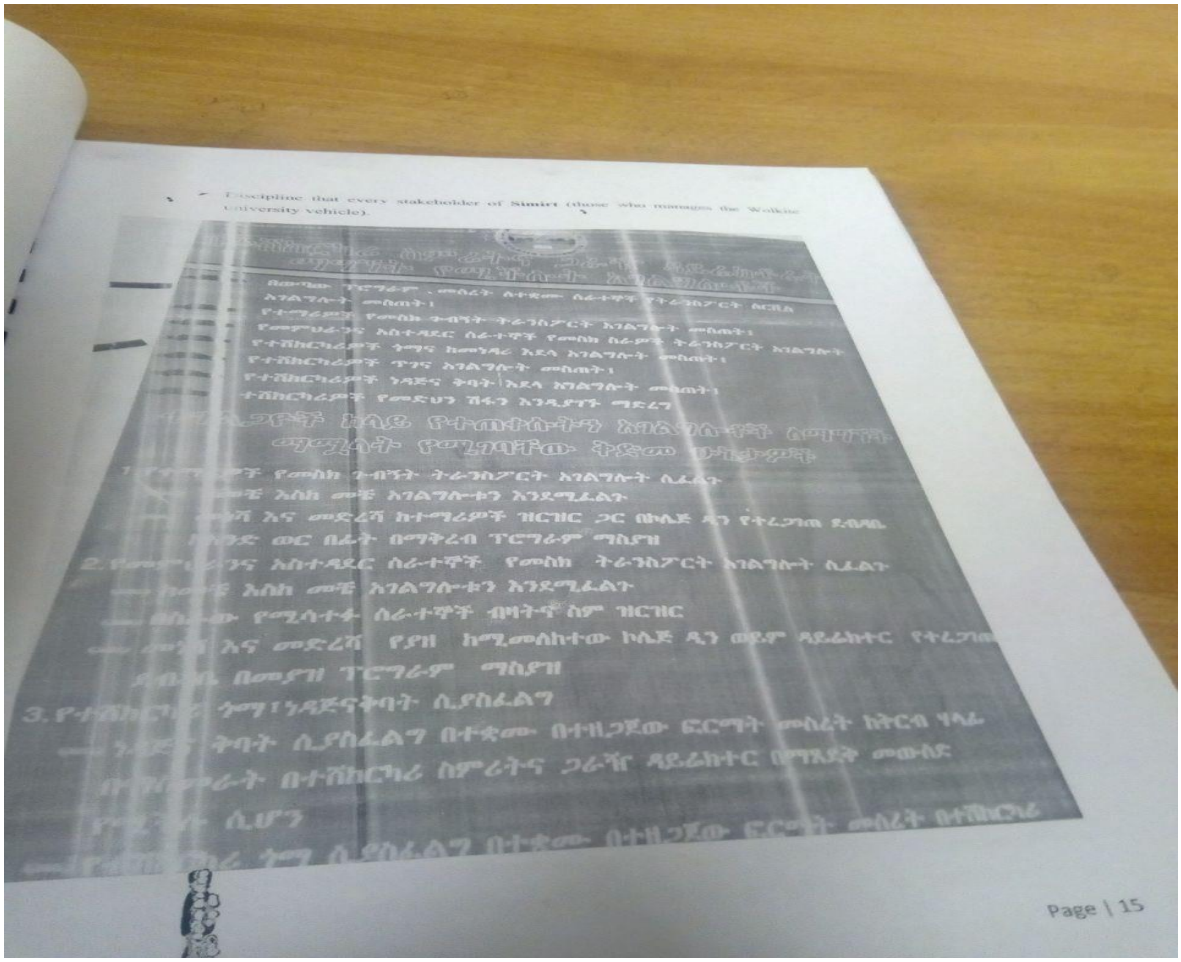
Salunkhe1, M. V. (n.d.). GSM and GPS Based Real Time Vehicle Theft. *International Advanced Research Journal in Science, Engineering and Technology*, 4.

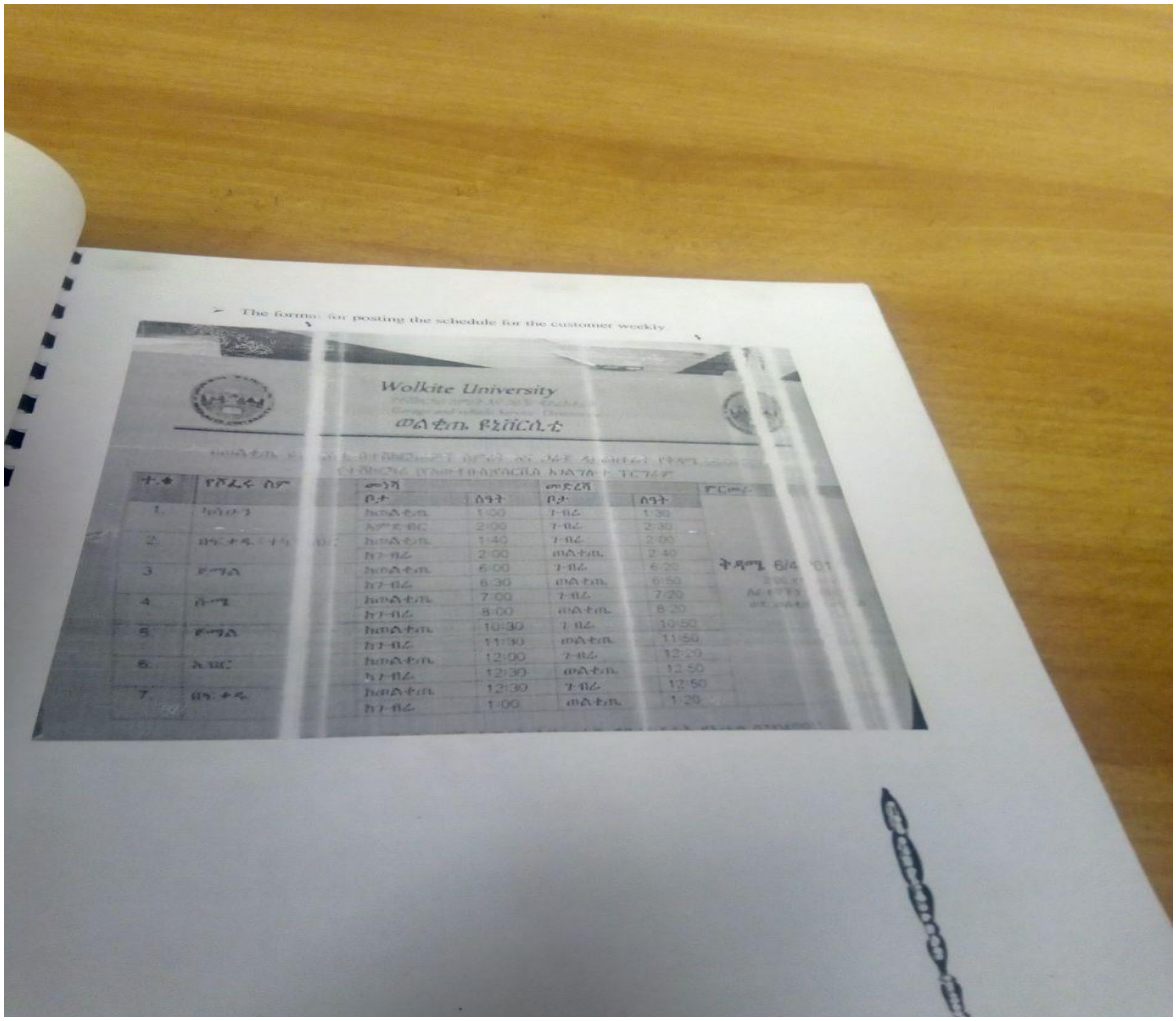
Saniah Ahmed, S. R. (2018). Real- Time Vehicle Tracking System. 98.

APPENDICES

Appendix 1

Some of the form used manually vehicle-tracking system




The format for posting the schedule for the customer weekly
Wolkite University
ወልቂጤ ዩኒቨርሲቲ