Smart Car Controller Detection System

Muhammad Furqan Wakeel | **Hassan Ali Brohi** | **M Haseeb Bin Fayyaz**

051-21-134215 | **051-21-134006** | **051-21-134209**

Miss Maria Hanif

A Final Year Project Report is

Submitted in Partial Fulfilment of the

Requirements for the Degree of

Bachelor of Science in Software Engineering

Department of Computing & Technology

Iqra University

2025

Certificate

We hereby accept the work contained in this report titled: ***Smart Car Controller Detection System***, as a confirmation to the required standards for the partial fulfillment of the degree of Bachelors of Science in Computer Science.

Internal Examiner

External Examiner

Project Supervisor

Head of Department

Declaration

We, solemnly declare that research work presented in our undergraduate project, as the case maybe, titled "Smart Car Controller Detection System" is solely our research work with no significant contribution from any other person or Generative AI Tools. Small contribution / help wherever taken has been duly acknowledged/cited and that complete project/thesis has been written by us in accordance with the latest plagiarism policy declared by HEC and our respective university in-line with the policy for use of Generative AI Tools.

We understand the zero-tolerance policy of the HEC and Iqra University Islamabad towards plagiarism. Therefore, we as an Author of the above titled project/thesis declare that no portion of our project/thesis has been plagiarized, and any material used as reference is properly referred/cited. we undertake that if we are found guilty of any plagiarism in the above titled project even after award of undergraduate degree, the University reserves the right to withdraw/revoke our degree and that HEC and the University has the right to publish our name on the HEC/University Website on which names of students are placed who submitted plagiarized project.

Muhammad Furqan Wakeel 34215

Hassan Ali Brohi 34006

M Haseeb Bin Fayyaz 34209

Dedication

Our sincere thanks are sent to God, who has provided everything required to finish both this project and the programmed it was started for. We dedicate this initiative to my parents, who have taught us the principles of perseverance and hard work, and who have always supported me. They are the most important people in my life. Our colleagues and classmates, who have made this trip joyful and unforgettable and who have assisted us in overcoming the obstacles and hurdles along the way, as well as our supervisor, who has been a mentor and a friend to us and who has graciously shared his expertise and experience with us. They are the ones who have inspired me to pursue this project with passion and curiosity, and they are the ones who have made this project possible.

Acknowledgements

First and foremost, we are grateful to Allah for providing me with the chance and fortitude to successfully finish this job. We also want to thank and honor the Prophet Muhammad (PBUH), who serves as an example for all Muslims.

We would like to thank Miss Maria Hanif, our supervisor, for his guidance and assistance during this project. She has been helpful and encouraging. She has consistently been available for discussion and advice, and she has given us insightful criticism and recommendations. Lastly, we would like to express our gratitude to our friends and family for their unwavering inspiration and support. They have always supported us throughout both happy and sad moments.

Abstract

The smart car system represents a comprehensive solution designed to enhance road safety and elevate the driving experience through the integration of advanced safety features and remote-control capabilities. Equipped with cutting-edge technologies such as fog detection, GPS navigation, and blind spot detection, the system aims to mitigate potential hazards associated with adverse weather conditions and improve overall driving awareness. By seamlessly incorporating these functionalities, the system not only enhances safety for drivers and pedestrians but also provides unparalleled convenience and control. Additionally, the accompanying mobile application further enhances user accessibility by offering real-time tracking functionalities and remote-control capabilities, thereby revolutionizing the way users interact with their vehicles.

With a strong emphasis on software quality attributes, including reliability, efficiency, maintainability, and user-friendliness, the smart car system sets a new standard in automotive innovation. Through a systematic methodology encompassing design, development, testing, and implementation phases, the system is poised to deliver a seamless and reliable driving experience in various environments and conditions. By addressing key problem areas such as the lack of advanced safety features in traditional vehicles and the absence of comprehensive remote control and tracking functionalities, the smart car system emerges as a transformative solution with wide-ranging applications across personal transportation, fleet management, emergency services, and logistics. Ultimately, it represents a significant step towards safer, more efficient, and more convenient transportation solutions, with far-reaching implications for individuals, businesses, and society.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The "Overall Description" chapter gives a thorough rundown of the smart car system project [1], emphasizing its goals, the fundamental issue, and the creative solution used to solve it. This section explores important topics such the project's objectives, methodology, target market, product scope, and operating environment. Presenting the smart car system as a clever solution highlights its cutting-edge safety features and remote-control capabilities, which are intended to improve driving and encourage road safety.

Readers will learn more about the project's goals, the environment in which it functions, and the target user groups. To ensure a clear knowledge of how the solution connects with its surroundings, the chapter also describes the operating environment of the system. It offers a comprehensive perspective of the system's design and operation by carefully examining external interface requirements, functional and non-functional needs, and system features.

Additionally covered are scenarios and use cases [2], which provide real-world illustrations of the system's functionality and advantages. This chapter lays the groundwork for the following sections, which go deeper into the creation, execution, and evaluation of the project. Readers are ready to comprehend the project's architecture, goal, and expected impact thanks to this thorough introduction.

1.1. Overall Description

The smart car system is a comprehensive solution that incorporates advanced safety features and remote-control capabilities. It is designed to improve road safety and enhance the driving experience. It is equipped with advanced features like fog detection, GPS navigation, and blind spot detection system. The integration of these technologies aims to enhance driving safety and provide convenience to the users.

1.1.1. Objectives

- Design and put into use a cutting-edge smart automobile system that improves driving enjoyment and road safety.
- Incorporate cutting-edge features such as:
- **Fog Detection:** Reduce the risks of low visibility during inclement weather.
- **GPS Navigation:** Makes it easier to navigate through new places and provide smooth route direction.
- **Blind Spot Detection:** To lower the likelihood of crashes, make sure there is increased awareness of blind areas.
- Offer a complete solution that puts pedestrian and car occupant safety first while enhancing the entire driving experience through automation and cutting-edge assistance.
- Provide an additional mobile application that functions as a remote control interface for the smart automobile system.
- Give users the ability to utilize the mobile application to remotely check the position and condition of the car.

1.1.2. Problem Description

Traditional cars lack advanced safety features that can mitigate risks associated with adverse weather conditions like fog, navigation challenges [3], and blind spots. Additionally, existing car control systems do not offer comprehensive remote control and real-time tracking functionalities.

1.1.3. Methodology

Our project will follow a systematic approach involving design, development, testing, and implementation phases. The team will collaborate to integrate fog detection sensors, GPS modules, and blind spot detection systems into the car's framework. Concurrently, the mobile application will be developed to enable remote control and location tracking functionalities.

Our methodology can be divided into two key areas.

- Hardware Based Approach
- Application Based Approach

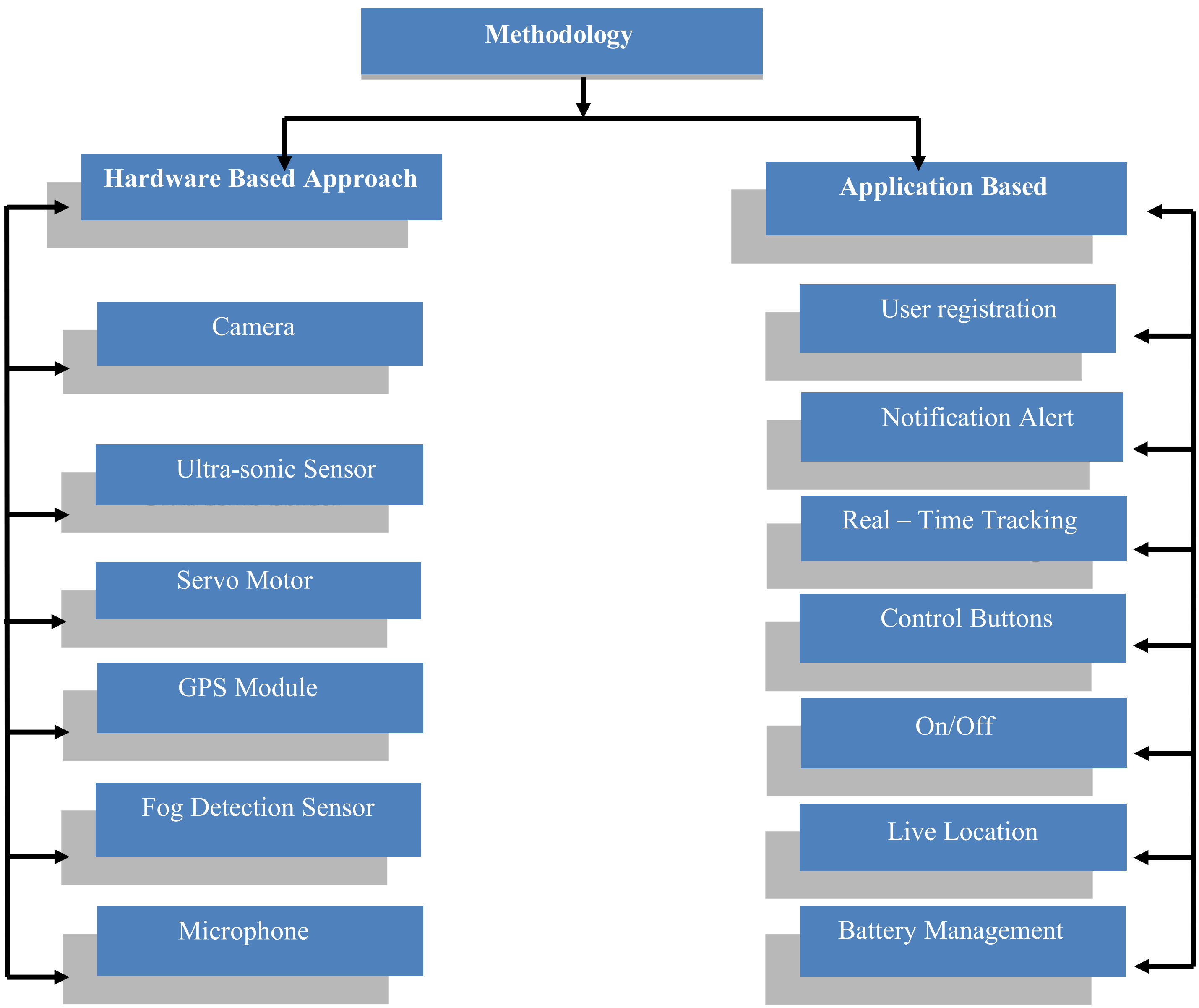

Figure 1. 1: Methodology

Product Scope

While our primary focus is on conflict management, the developed vehicle has the potential to benefit various sectors, including transport, and medical assistance. By creating an Android application, we aim to make the hardware vehicle accessible to a wider audience, empowering users to assess the vehicle conveniently.

1.1.4. Business Context

The proposed smart car system has wide application areas including personal transportation, fleet management, emergency services, and logistics. It can benefit individuals, businesses, and organizations by providing safer, more efficient, and convenient transportation solutions. By providing an intuitive tool for early detection, this project aligns with broader healthcare goals, aiming to improve delivery services and reduce crucial disparities.

1.1.5. User Classes and Characteristics

The primary users of the smart car system are individuals who drive regularly and businesses that manage vehicle fleets. These users value safety, efficiency, and convenience in their transportation solutions.

1.1.6. Operating Environment

The smart car system is designed to operate in a variety of environments, including urban and rural settings. It is equipped to handle different weather conditions and navigate various types of terrain.

1.2. Functional Requirements

The functional requirements for the smart car system are designed to ensure comprehensive capabilities that address various aspects of safety, navigation, and user interaction. Firstly, the system must possess the ability to detect fog and autonomously adjust vehicle operations to suit the prevailing weather conditions.

Secondly, leveraging GPS technology, the smart car system must facilitate efficient navigation by accurately determining the vehicle's position and providing route guidance to the driver.

Thirdly, to mitigate the risks associated with blind spots, the system should incorporate sensors and alerts to detect the presence of nearby vehicles or obstacles that may not be visible to the driver.

Moreover, the smart car system must offer remote control functionality via a dedicated mobile application, empowering users to interact with the vehicle from afar. Through the application, users should be able to perform essential tasks such as locking/unlocking doors, starting/stopping the engine, adjusting climate control settings, and activating security features.

1.3. External Interface Requirements

1.3.1. User Interfaces

The system will have a user-friendly interface on the mobile application for remote control and tracking.

1.3.2. Hardware Interfaces

The system will interface with fog detection sensors, GPS modules, Fire detection, Fuel level monitoring, Battery level indicator and microcontrollers installed in the car.

1.3.3. Software Interfaces

The system will interface with the mobile application developed using Android Studio and Flutter.

Operating System: Compatible with Android OS (version 7.0 and above)

1.3.4. Communications Interfaces

The smart car system integrates several key communication interfaces to facilitate seamless interaction between the vehicle and the user. Firstly, equipped with front and back view cameras, the system provides live visual feeds, enabling drivers to monitor their surroundings effectively. This feature enhances situational awareness and aids in safe maneuvering, particularly during parking and low-visibility conditions. Additionally, a robust tracking system is employed to continuously monitor the vehicle's real-time location. By leveraging GPS technology, this interface enables accurate positioning and navigation, allowing users to track the vehicle's movements remotely and plan routes effectively.

1.4. System Features

1.4.1. System Feature 1

Blind spot detection and fog detection.

1.4.1.1 Description and Priority

High priority feature that alerts the driver of potential disasters like fog or obstacles in the blind spot.

1.4.1.2 Stimulus/Response Sequences

Stimulus: The system detects a potential disaster and sends an alert to the driver.

Response:

- System processes input data.
- Alert will be displayed to the user.

1.4.1.3 Functional Requirements

REQ-1: The system must accurately detect potential disasters and promptly alert the driver.

REQ-2: In the event of invalid inputs, the system must provide appropriate error messages.

1.4.2. System Feature 2

User Registration.

1.4.2.1 Description and Priority

Medium priority feature that allows users to register and access the remote control and tracking functionalities.

1.4.2.2 Stimulus/Response Sequences

- Stimulus: The user inputs their information in the mobile application to register.
- ***Response:***

  User profile is created upon successful registration.

  The system verifies the information and grants the user access.

1.4.2.3 ***Functional Requirements***

REQ-3: The system must provide a secure registration process.

REQ-4: Users must input necessary information during the registration.

REQ-5: The system must securely store user information and verify it accurately.

1.5. Nonfunctional Requirements

1.5.1. Performance Requirements

The system must respond quickly and accurately to user inputs and environmental stimuli.

1.5.2. Safety Requirements

The system must prioritize the safety of the user and operate within all legal and ethical guidelines.

1.5.3. Security Requirements

The system must protect user information and prevent unauthorized access. App will ask for permission for location and storage access. The personal information will never be shared with anyone for security purpose.

1.5.4. Software Quality Attributes

The smart car system is designed with a focus on software quality attributes to ensure reliability, efficiency, maintainability, and user-friendliness. Reliability is paramount, ensuring that the system consistently performs its intended functions accurately and without failure. Through rigorous testing and robust error-handling mechanisms, the system maintains a high level of dependability, instilling confidence in users regarding its performance under various conditions.

1.6. Scenarios

Scenario 1: A user is driving in foggy conditions. The system detects the fog and alerts the user.

Scenario 2: A user wants to track their vehicle's location. They open the mobile application and view the real-time location of the vehicle.

1.7. Report Structure

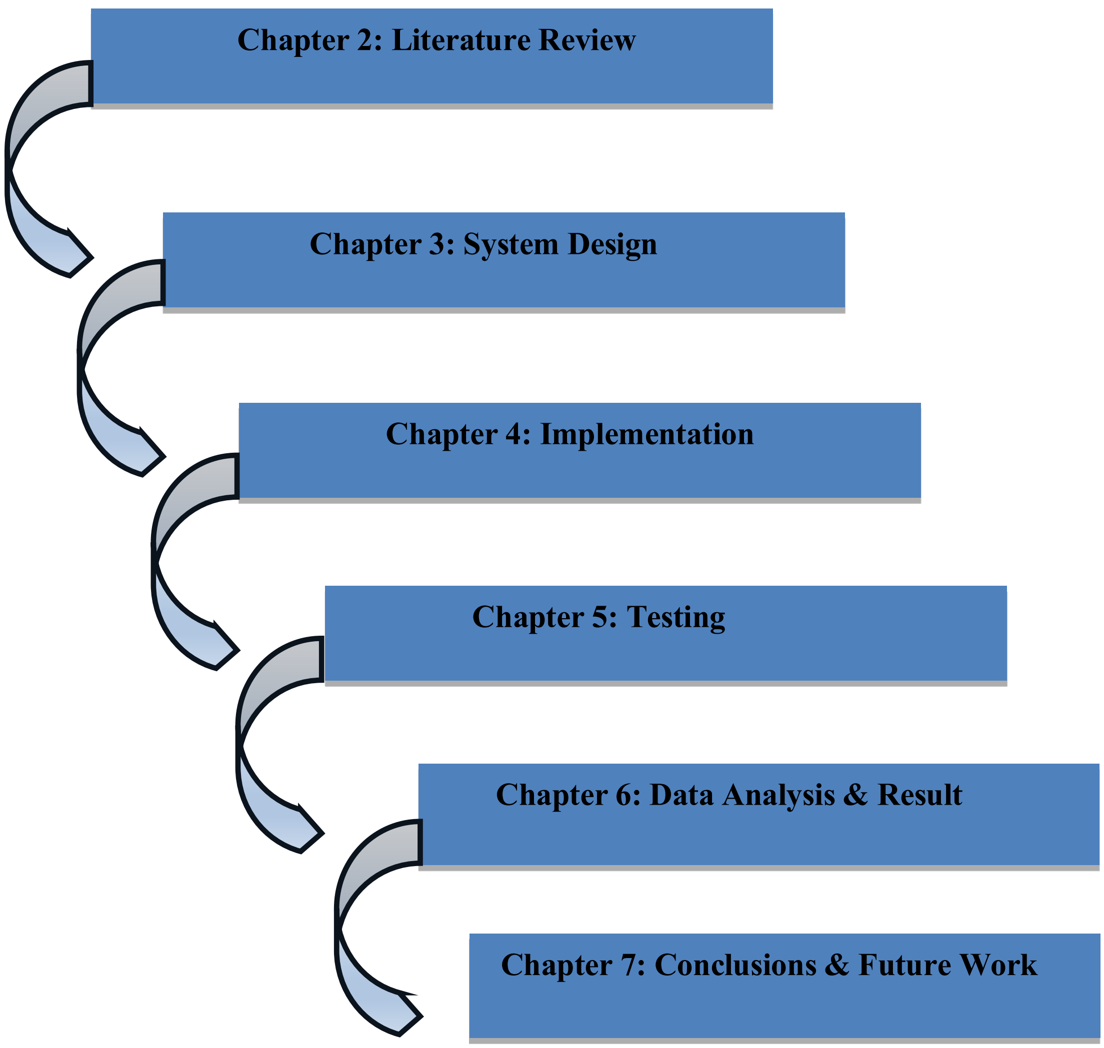

Figure 1. 2: Report Structure

Chapter 2 Literature Review

2.1. Introduction

This innovative car transforms the driving experience by fusing cutting-edge technology with simple design. Our prototype delivers unmatched safety, convenience, and connectivity with features including GPS navigation, blind spot monitoring, fog detection, object detection and seamless app integration. You may easily operate your automobile remotely with the help of our companion mobile app. Additionally, get real-time updates on the whereabouts of your automobile to keep informed and in control and to travel with peace of mind.

2.2. Related Works

1- The existing "**Smart Car System**" [4] focuses on integrating essential features for security and basic functionality, our project takes a leap forward by incorporating state-of-the-art technologies to revolutionize the driving experience.

2- The existing "**Car Health and Security Systems**" [5] predominantly utilize GPS and GSM modules, along with sensors to monitor oil levels and trigger basic security functions like door locking. These systems provide rudimentary protection against theft and help in locating lost vehicles, but they lack advanced functionalities and seamless integration with modern digital platforms.

3- The existing project on a "**Vehicle Blind Spot Sensors**" [6] with a blind spot detection system represents a step towards enhancing vehicle safety through advanced sensor technology. However, this project is limited in scope, focusing solely on the integration of a blind spot detection system.

4- A step towards improving vehicle safety with cutting-edge sensor technology is the ongoing project on a "**System for Vehicular Collision Detection and Alert**" [7] with a pre-collision system. But the scope of this project is restricted to the integration of a pre-collision system.

2.2.1. Categorization of Existing Techniques/Works/Research

The existing project, which is centered on a smart automobile equipped with a blind spot detection system, falls into the category of sensor-based vehicle safety enhancement. This category includes the smart car's infrastructure that uses radar or other sensor technologies to identify vehicles in blind spots. Even though the current research is limited to blind spot detection, it falls under the category of sensor fusion, which is the process of combining information from many sensors to give a more complete picture of the environment around the vehicle.

The existing "Smart Car System" aims to integrate essential features into vehicles to enhance security and functionality. Utilizing Arduino hardware and an Android-based application, the system establishes connectivity via a Bluetooth module, enabling users to control various aspects of their car remotely.

The existing project, which is centered on “Car health and Security system” techniques can be categorized into basic car security systems utilizing GPS and GSM modules for location tracking, and sensor-based systems for monitoring oil levels and detecting obstacles. These systems primarily focus on reactive measures rather than proactive safety features.

2.2.2. Limitations/Gaps within Existing Techniques/Works

After a detailed comparison and research, the limitations of existing “Smart Car” are as follows:

- **Limited Safety Coverage:**
  The primary limitation of the existing project is its narrow focus on blind spot detection alone. While this feature enhances safety by addressing a specific area of concern, it does not provide comprehensive coverage of all potential safety hazards on the road.
- **Single-Point Solution:**
  Focusing solely on blind spot detection results in a single-point solution that may overlook other critical safety aspects such as fog detection and GPS monitoring.
- **Lack of Integration:**

The absence of integration with other safety features for example an application by which we control our car or systems in the car is another drawback.

After a detailed comparison and research, the limitations of existing "Smart Car System" are as follows:

- **Limited Functionality:**
  While it provides essential features, it lacks advanced capabilities such as real-time monitoring and comprehensive connectivity.
- **Compatibility issues:**
  The system's reliance on specific hardware and software components may limit its compatibility with a wide range of vehicles and devices.
- **Security concerns:**
  The use of Bluetooth connectivity may pose security risks, potentially exposing the system to vulnerabilities.

After a detailed comparison and research, the limitations of existing "Car Health and Security System" as follows:

- **Limited functionality:**
  Current systems offer basic security measures and location tracking but lack advanced features such as blind spot monitoring and fog detection.
- **Lack of integration:**
  Existing systems often operate as standalone units with limited connectivity to other devices or digital platforms.
- **Reliance on outdated technology:**
  Many car security systems still rely on GSM modules, which may not offer robust data transmission or compatibility with modern smartphones.
- **Limited user control:**
  Users have minimal control over their vehicles remotely and may not receive real-time updates on their car's status.

2.3. Proposed Improvements in Existing Works

With the help of our state-of-the-art Smart Car Controller Detection System, enter the world of automotive innovation. This cutting-edge car, which aims to reinvent driving, has an extensive feature set that improves connectivity, safety, and ease of use.

Our Intelligent Car Controller Detection System goes above and beyond, in contrast to other projects that only concentrate on blind spot detection. We're changing the way you connect with your car with features like blind spot monitoring, GPS navigation, integrated fog detection, and an intuitive smartphone app. Incorporate fog detection capabilities into the existing sensor suite to enhance visibility and safety in adverse weather conditions. Expand the functionality of the mobile app to allow for remote control of the vehicle, including starting and stopping the engine.

Enhanced functionality:

By integrating advanced features such as GPS navigation and real-time monitoring, our system offers an unparalleled driving experience.

Universal compatibility:

Our prototype is designed to be compatible with various vehicles and mobile devices, ensuring widespread adoption and usability.

Robust security measures:

We implement robust encryption protocols and authentication mechanisms to safeguard against potential cyber threats.

2.4. Comparative Analysis

Table 2. 1: Comparative Analysis

Features	Smart Car System	Car Health & Security System	Vehicle Blind Spot Sensors	Vehicular Collision Detection & Alert	Proposed System
Fog Detection	No	No	No	No	Yes
Car Lock/Unlock	No	Yes	No	No	No
Pre-Collision	No	No	No	Yes	Yes
GPS Monitoring	No	Yes	No	No	Yes
Blind Spot Detection	No	No	Yes	No	Yes
Safety Features	Yes	Yes	Yes	Yes	Yes
Oil Level Monitoring	No	Yes	No	No	Yes

Overview of Comparison Table

The existing "**Smart Car System**" focuses on integrating essential features for security and basic functionality, existing "**Car Health and Security Systems**" predominantly utilize GPS and GSM modules, along with sensors to monitor oil levels and trigger basic security functions like door locking, the existing project on a "**Vehicle Blind Spot Sensors**" with a blind spot detection system represents a step towards enhancing vehicle safety through advanced sensor technology and step towards improving vehicle safety with cutting-edge sensor technology is the ongoing project on a "**System for Vehicular Collision Detection and Alert**" with a pre-collision system but these projects lack advanced functionalities and safety features which our proposed system provides like fog detection, pre-collision and blind spot detection features and more in low cost budget.

2.5. Summary

The Smart Car Controller Detection System introduces a transformative approach to driving, integrating cutting-edge technology seamlessly. key technologies including GPS navigation, blind spot monitoring, fog detection, object detection security features and app integration. Our prototype aims to revolutionize the driving experience. Unlike existing projects, our system offers comprehensive coverage and enhanced functionality, addressing limitations such as limited safety coverage, lack of integration, and reliance on outdated technology. With universal compatibility and robust security measures, our prototype sets new standards in automotive innovation, promising unmatched safety, convenience and peace of mind for users.

Chapter 3 System Design

3.1. Introduction

We are working on a project called "Smart car controller detection," which is a smart car system with cutting-edge capabilities like fog detection, GPS navigation, and blind spot identification that is specifically made to increase road safety and improve the driving experience. We are developing an Internet of Things (IoT)-based smart automobile system that will allow us to operate it via a mobile application. Since the IOT model is secure, dependable, and affordable, we will connect several sensors to check the reading data and communicate with each key component. This model will assist individuals in ensuring that life is smartly made easier using electronics. We will talk about the model's nomenclature and the kinds of new features it adds to our project. Every part of the smart car model, including the Arduino Pro with ESP32, fog sensor, blind spot sensor, ultra-sonic sensor, buzzer, camera, 12v adapter, wires, and microphone, will be shown. The sensors are linked to a microcontroller that is installed in the smart automobile gadget. Depending on its intended usage, the user can effortlessly control the features to switch it on or off. To provide real-time readings and to activate the buzzer in case of an emergency, the smart car is also linked to a smartphone application.

3.1.1. Purpose

This system's primary goal is to improve driving safety and enjoyment. Modern technology including fog detection, GPS navigation, and blind spot detection techniques will all be incorporated into the system. Users will be able to communicate with the smart automobile system from their smartphones thanks to this application, which will work as a remote-control interface. Enabling remote control and location tracking features will be facilitated by our project. Therefore, to increase the degree of safety and security, we suggest a safety system called "Smart Car Controller detection."

3.1.2. System Overview

The "Smart Car Controller detection" allows the user to control devices that are only connected to the smart car. The smart automobile system is a complete solution that includes remote control capabilities and cutting-edge safety features. It is intended to increase driving enjoyment and road safety. It has sophisticated features like a blind spot detection system, GPS navigation, and fog detection. The goal of integrating these technologies is to improve user ease and driving safety. Flutter will be used to build the mobile application, which we will develop using Android Studio/Visual Studio.

3.1.3. Design Map

The system's structure is determined by our project's design map.

Design Artifacts:

- **Use Case Diagram**: This illustrates how administrator and user access interact with one another by include the functionalities we have implemented in our system.
- **Sequence Diagram:** This illustrates our system's general operation and process.

3.2. System Architecture

This displays a summary of the system and how it functions. Our system is working on related hardware and software.

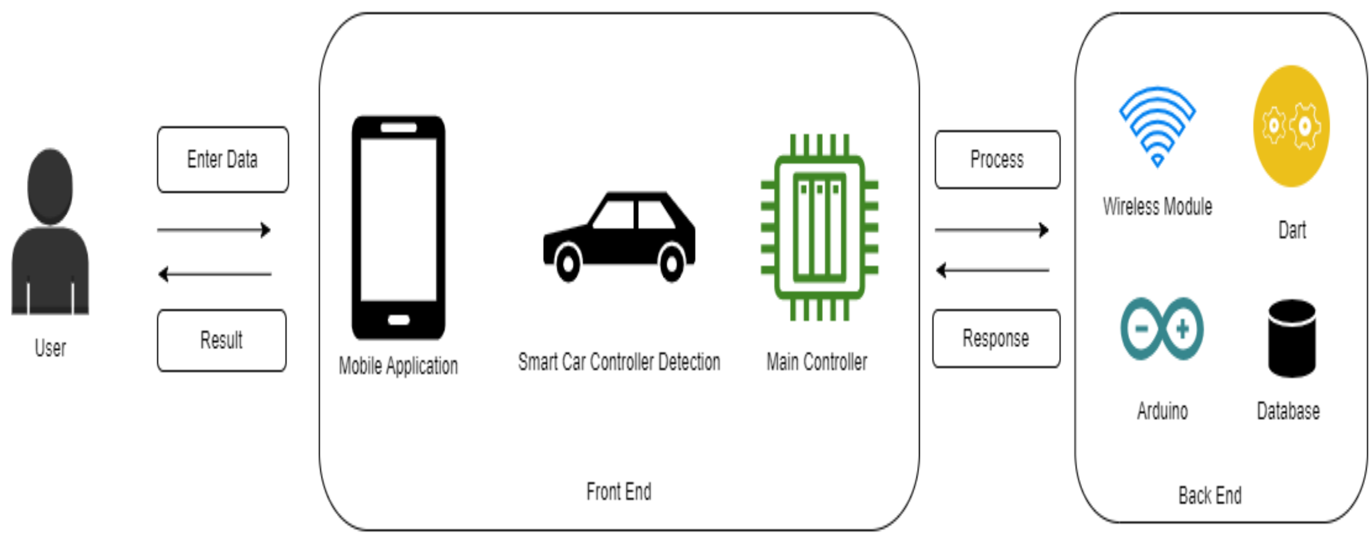

Figure 3. 1: System Architecture

3.3. Design Considerations

The following presumptions and limitations may be included in the design considerations:

3.3.1. Assumptions

- Assuming we have an Arduino board, we connect the different components we wish to control.
- The mobile system can be controlled by several commands via its interface.
- To connect this model to a smartphone app, we presume that it has a built-in wireless Arduino board connection.

3.3.2. Constraints

- The application should only be written on Flutter (Dart).
- The correctness of our project is dependent on the real-time value, which is indicated by the smart car controller's real-time position GPS system and the feature that allows it to communicate with a mobile application.

3.4. Context Diagram

Due to its lack of system detail, these graphic displays high-level diagrams. Rather, it only provides a user-friendly map of the complete system. In essence, it provides a summary of the entire system and its operations.

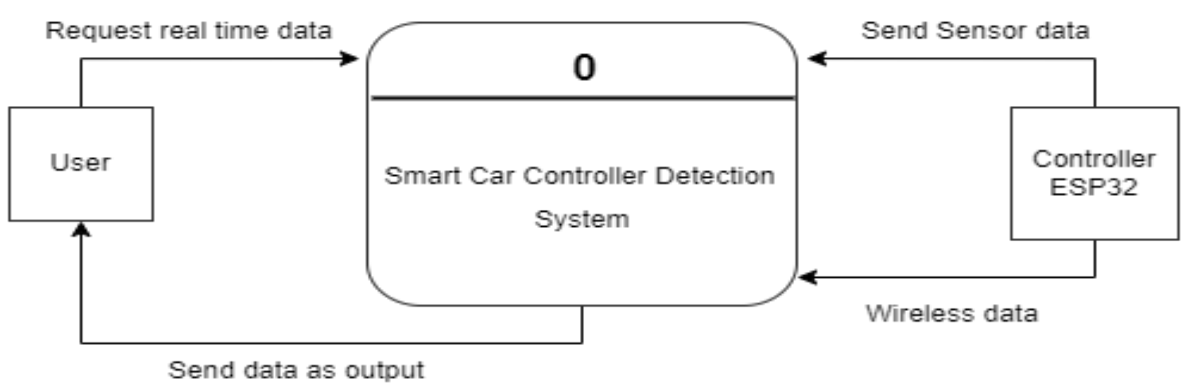

Figure 3. 2: Level 0 Context Diagram

3.5. Data Flow Diagram

A data flow diagram illustrates the system's communication with other components. the information flow and process step sequence utilizing a system's graphical representation.

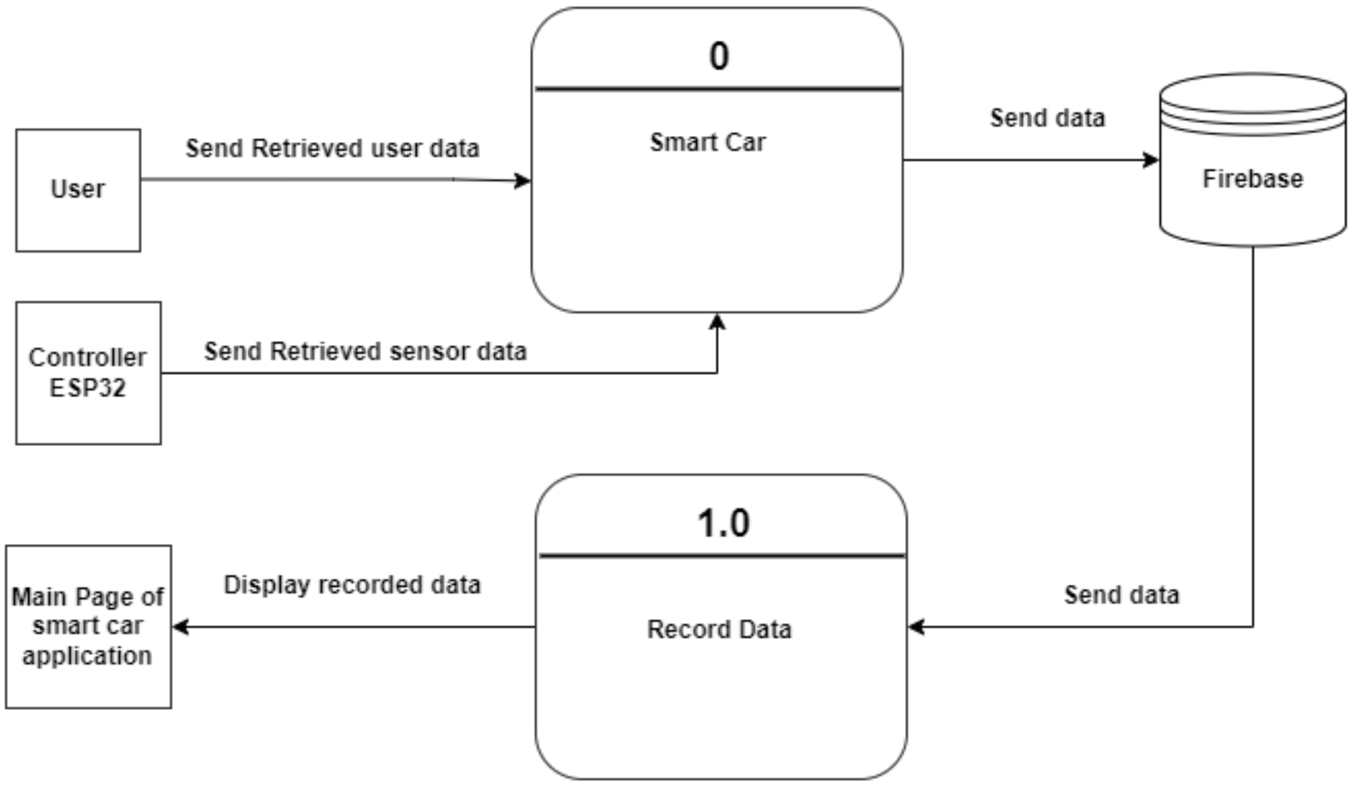

Figure 3. 3: Level 1 Data Flow Diagram

3.6. Use Case Model

According to the use case model, our system can be controlled by both the administrator and the user.

3.6.1. Use Case Diagram

The system's use case model illustrates the administrator's and user's authority.

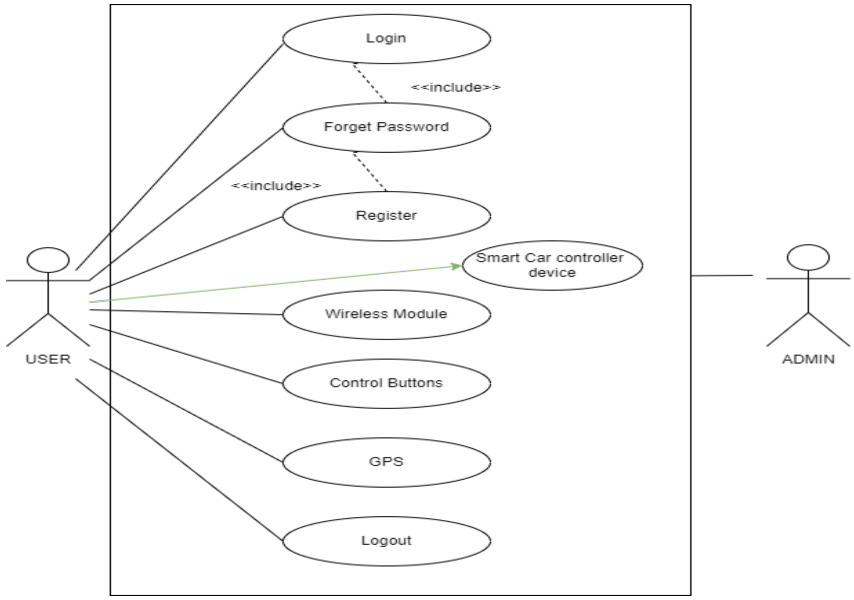

Figure 3. 4: User Use case

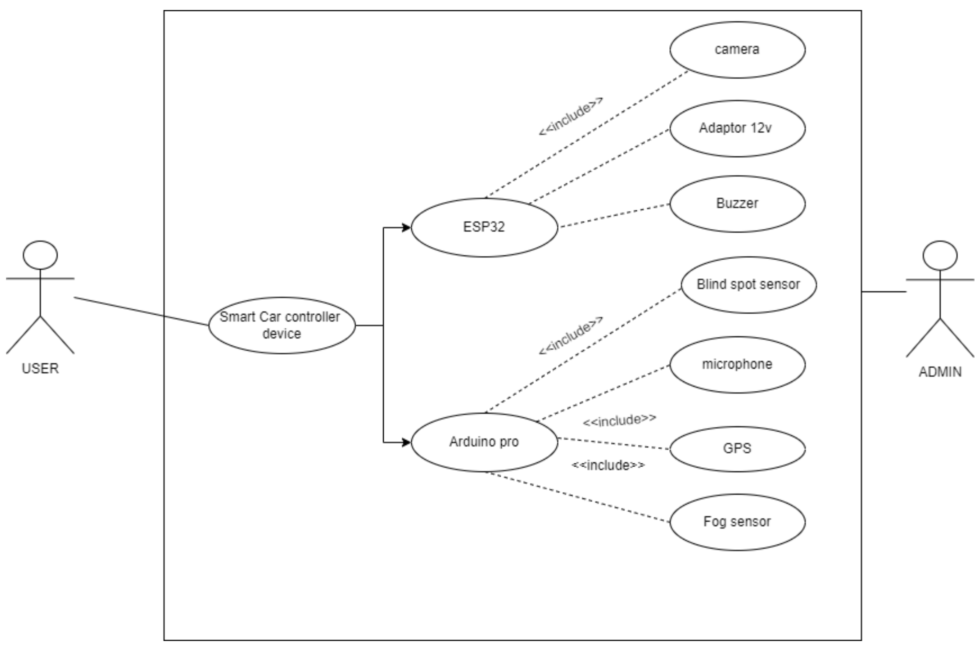

Figure 3. 5: Hardware Use case

3.7. Fully Dressed Use Case

A complete use case explains the system's intricate structure, including its actors, administrators, scenarios, and conditions. All the project's use cases are designed to help people complete the work and get better results.

Login

Table 3. 1: Login

Use Case ID	C1	
Use Case	Login	
Primary Actor	User	
Stakeholders	Users for using the system.	
Preconditions	The user has a system registration.	
Guarantee	To gain additional access to the system, the user is logged in.	
Scenario	Actor Action	System Responsibility
	1. The application is launched by the user. 2. The user taps the button for login. 4. The user inputs their login credentials. 5. The login button is tapped by the user.	3. Opens the home screen.
Extensions	The password is forgotten by the user. 1. Press the button labelled "forgot password." 2. The questions part of the system prompts the user. 3. The user inputs the questions' responses. 4. The system verifies that the responses are accurate. 5. If the responses are accurate, the user moves on to step 7 and is prompted to provide their email address so that the password can be supplied. 6. If not, the user is sent to the login page and receives an error notice. 7. The user inputs their email address. 8. The system sends the password to the written email.	
Requirements	- There should be text areas for entering the login credentials. - A database should be accessible so that user data can be accessed. - The user's credentials ought to be protected.	

Register

Table 3. 2: Register

Use Case ID	C2	
Use Case	Register	
Primary Actor	User	
Stakeholders	User for gaining access to all the system's capabilities.	
Preconditions	To continue using the system, the user must first register.	
Guarantee	To continue using the system, the user must first register.	
Scenario	Actor Action	System Responsibility
	1. The application is launched by the user. 3. The user taps the button for registration. 4. The user fills out the field. 5. The register button is tapped by the user.	2. Shows the screen for registration.
Extensions	The input entered by the user is incorrect. 1. The password is entered by the user. 2. The system verifies that the password complies with all rules. 3. If the password was entered incorrectly, the system displays an error message. 4. The user is requested to give it another go.	

Requirements	- There should be text areas where users can enter their login details. - To access user data, a database must be accessible. - It is important to protect the user's credentials.

Wireless Module

Table 3. 3: Wireless Module

Use Case ID	C3	
Use Case	Wireless Module	
Primary Actor	User	
Stakeholders	User for gaining access to all the system's capabilities.	
Preconditions	The device is linked to the user.	
Guarantee	To continue using the system, the user must log in.	
Scenario	Actor Action	System Responsibility
	1. The application is launched by the user. 2. The connectivity button is tapped by users. 4. The user fills out the field. 5. Users press the Wi-Fi icon.	3. Makes the connection visible on the screen.

Extensions	The person wasn't using Bluetooth. 1. The user inputs incorrect data. 2. The system determines if there is a connection. 3. If the connection did not adhere to the protocols, the system displays an error message. It asks the user to try again.
Requirements	- There should be text areas for entering the login credentials. - A database should be accessible so that user data can be accessed. - The user's credentials ought to be protected.

Smart Car Module Device

Table 3. 4: Smart Car Module Device

Use Case ID	C4	
Use Case	Smart Car Module Device	
Primary Actor	User	
Stakeholders	Administrators and users can access additional system functions.	
Preconditions	The user has logged into the program.	
Guarantee	To continue using the system, the user must first register.	
Scenario	Actor Action	System Responsibility
	1. The system is accessed by the user. 3. The user taps the selected option.	2. The user sees the homepage.
Extensions	The homepage is accessed by the user. 1. The details dialogue is opened by the user tapping on the System buttons. 2. The user verifies connectivity. 3. If the user is asking for connectivity, the system verifies it. 4. The system displays the relevant notice.	
Requirements	- The homepage should provide buttons for operating the system. - When the user clicks on the controlling System Buttons, a new activity page needs to open. - When the right input is entered, the system ought to be strong enough to display an error message.	

Main

Table 3. 5: Main

Use Case ID	C5	
Use Case	Main Controller	
Primary Actor	User	
Stakeholders	Users for viewing the system's overall commands and control.	
Preconditions	The user has logged into the program.	
Guarantee	The output is reflected once the user verified the movement of the hand motion in real time.	
Scenario	Actor Action	System Responsibility
	1. The user enters their login redentials. 2. They press the control button.	2. The user was presented with buttons.
Extensions	The homepage is accessed by the user. 1. The details dialogue is opened by the system when users tap on the System buttons. 2. The user makes sure the buttons are functioning correctly. 3. If the user is asking for the real-time buttons, the system looks for them. 4. The system displays the relevant notice.	
Requirements	- The homepage should provide buttons for operating the system. - When the user clicks on the controlling System Buttons, a new activity page needs to open. - When the right input is entered, the system ought to be strong enough to display an error message.	

Logout

Table 3. 6: Logout

Use Case ID	C6	
Use Case	Logout	
Primary Actor	User	
Stakeholder	After using the application, users have the option to log out.	
Preconditions	The user has logged into the program.	
Success Guarantee	The user is informed with most recent information.	
Main Success Scenario	Actor Action	System Responsibility
	1. The user opens the application and logs in. 4. From the menu list, the user selects the logout button.	2. The user sees the homepage. 3. The menu items are displayed by the system. 5. The system logs the user off successfully.
Extensions	A system is in operation. 1. The log-off button is clicked by the user. 2. The system alerts the user that a system is operating. 3. A dialogue box verifying the logout is displayed by the system. 4. The user presses the "Confirm" button. The user is logged off by the system.	
Special Requirements	- Every screen of the system should offer a log-out option. - No active system should be terminated by the system. - It should be possible for users to either cancel or keep using the system.	

3.8. Sequence Diagram

A sequence diagram illustrates our system's relationships and overall structure.

User

What the user can do in our system is controlled by the user side. This essentially illustrates the system's flow and overview.

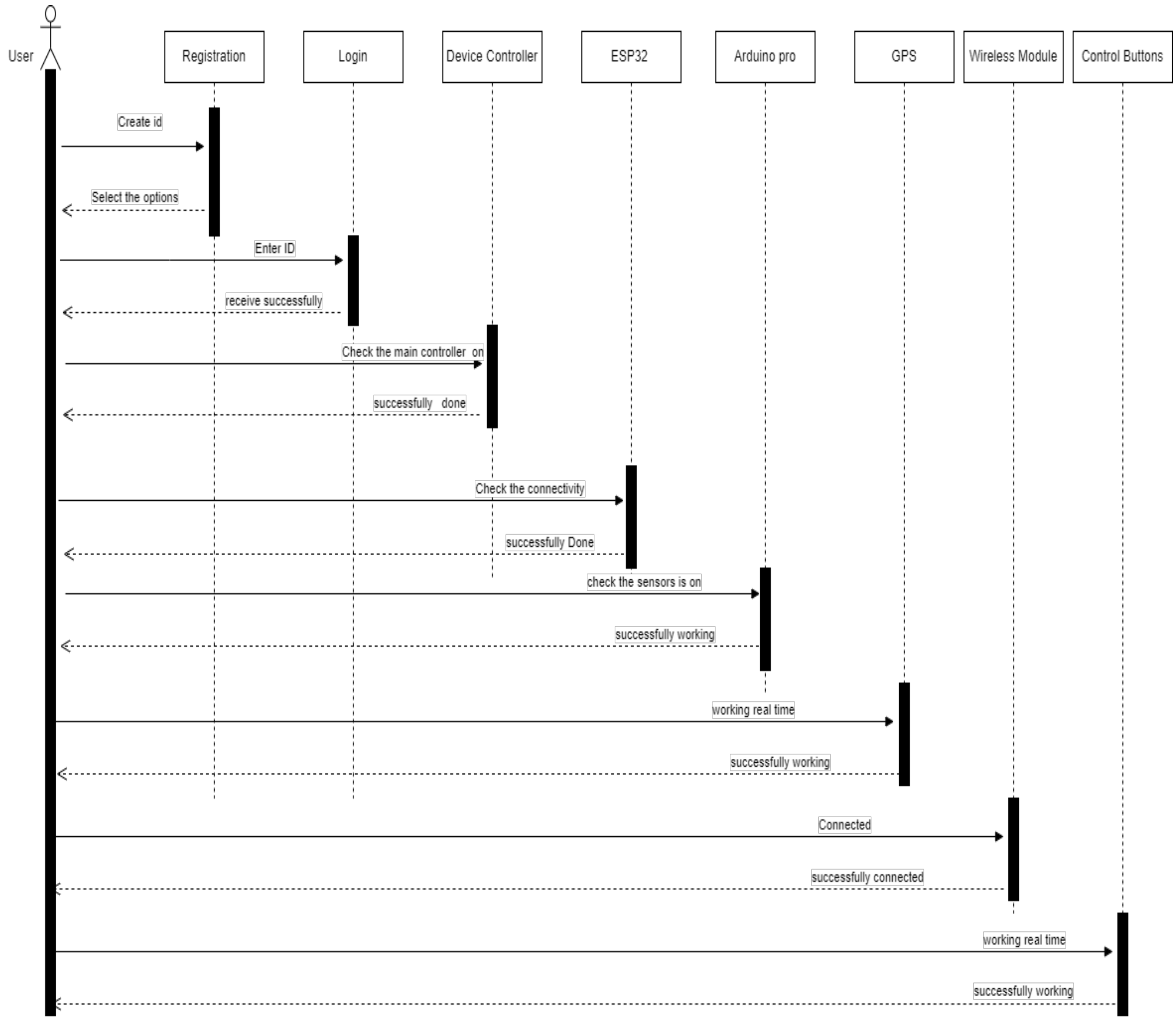

Figure 3. 6: User Sequence Diagram

3.9. Activity Diagram

Our system's activity diagram displays the system's competitive structure. The beginning, middle, and end of it.

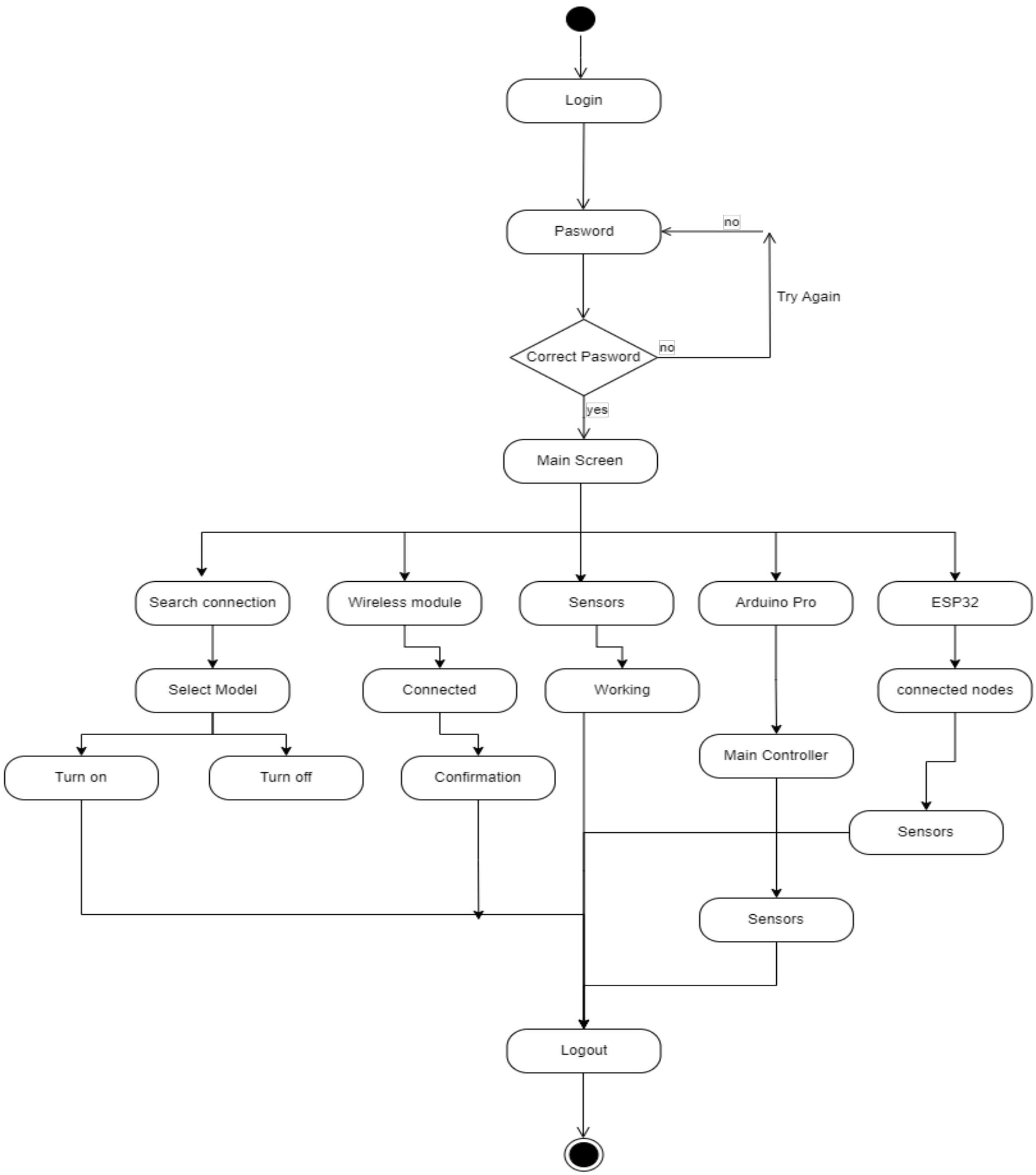

Figure 3. 7: Activity Diagram

3.10. User Interface Design

Flutter will be used to implement our system's user interface. Our framework's user interface designed to be as easy to use as a mobile application. Our framework's point of interaction is th the smart controller car, which manages the system's operations, can identify blind spots or fog to provide a safer trip with a lower chance of damage and alert the user via a buzzer if something goes wrong.

3.11. Model Architecture Overview

The following elements make up our module's design. Buzzer, GPS, camera, ESP32, Arduino Pro, fog sensor, blind spot detection module, and smart car framework.

3.11.1. Humidity Sensor

Humidity sensor, the primary component of this project, are used to identify fog through humidity sensor in the surrounding area and provide alerts or triggers when visibility drops below predetermined thresholds.

Figure 3. 8: Humidity Sensor

3.11.2. ESP32

Our system's primary controller, the ESP32, regulates the operation of additional hardware parts like the GPS, buzzer, blind spot sensor, and fog sensor. This signals the recipient to take the following action.

Figure 3. 9: ESP32

3.11.3. GPS Module

The GPS system we use for our projects allows us to track the car's present location. On our project's primary controller, the GPS Module is also linked to a mobile application. Turn on the application when it recognizes your present location.

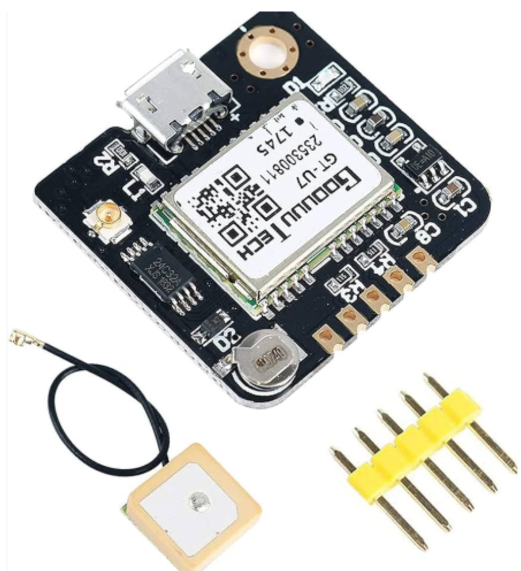

Figure 3. 10: GPS Module

3.11.4. Buzzer

We are employing a buzzer system in our project so that the user can be guided in the event of an emergency when the alarming buzzer is triggered.

Figure 3. 11: Buzzer

3.11.5. Arduino nano / Esp-32

Our idea makes use of an Arduino nano [8] to interface with additional hardware components. Our smart automobile system makes use of this.

Figure 3. 12: Arduino nano

3.11.6. Blind spot detector

Our device uses a blind spot detector sensor [9] to identify signals coming from outside the smart car system.

Figure 3. 13: Blind Spot Detector

3.12. Summary

This chapter covers the features we incorporate into our gadget as well as the project's system design. We talk about the language we use, the tools we use, and how it relates to it. We also talk about the newest technologies that will improve the accuracy and safety of our endeavor.

Chapter 4 Implementation

4.1. Discussion

The initial phase of implementation involved setting up the hardware components. This included installing the ultrasonic sensors for obstacle detection, the infrared sensors for line following, the fire sensor to detect fire in car and the motor driver circuits for controlling the car's movement. The primary challenge during this phase was ensuring all components were properly connected and communicated effectively with the ESP-32 microcontroller.

Software implementation was the next critical step. The design necessitated writing code to process data from the sensors and make real-time decisions to control the car. One significant difficulty faced during this phase was optimizing the code to run efficiently on the limited processing power of the ESP-32. This was resolved by breaking down the tasks into smaller functions and using efficient data handling techniques to minimize processing delays.

Another challenge was ensuring the reliability and accuracy of the sensor data. Initial tests showed inconsistent readings from the ultrasonic sensors due to environmental noise and obstacles' varying shapes and materials. To address this, the code was adjusted to include multiple readings and averaging techniques to filter out noise and improve accuracy.

4.2. Development Methodologies

The development of the smart car system followed an iterative and incremental methodology. This approach allowed for continuous testing and refinement of both hardware and software components. Key methodologies included:

Prototyping: Early prototypes of the car were developed to test basic functionalities. This helped in identifying issues early in the development process.

Modular Development: The system was divided into smaller modules, such as sensor modules, motor control, and communication modules. Each module was developed and tested independently before integrating them into the final system.

Agile Practices: Regular reviews and testing cycles were conducted to ensure that the development stayed on track and any issues were promptly addressed. This iterative process allowed for flexibility and adaptability in the development cycle.

4.3. Implementation Tools and Technologies

For the implementation we use different tools & technologies such that are:

- **Technologies:**
  - Ultra-Sonic Sensors.
  - Fuel Level Sensor.
  - Motor Driver Circuits.
  - GPS Module.
  - LCD.
  - Motor Driver Module.
  - Flame Sensor.
  - Battery.
  - Wi-Fi 2.4Ghz Module
- **Tools**
  - Arduino nano/ Esp32 Microcontroller
  - Firebase

• Android Studio

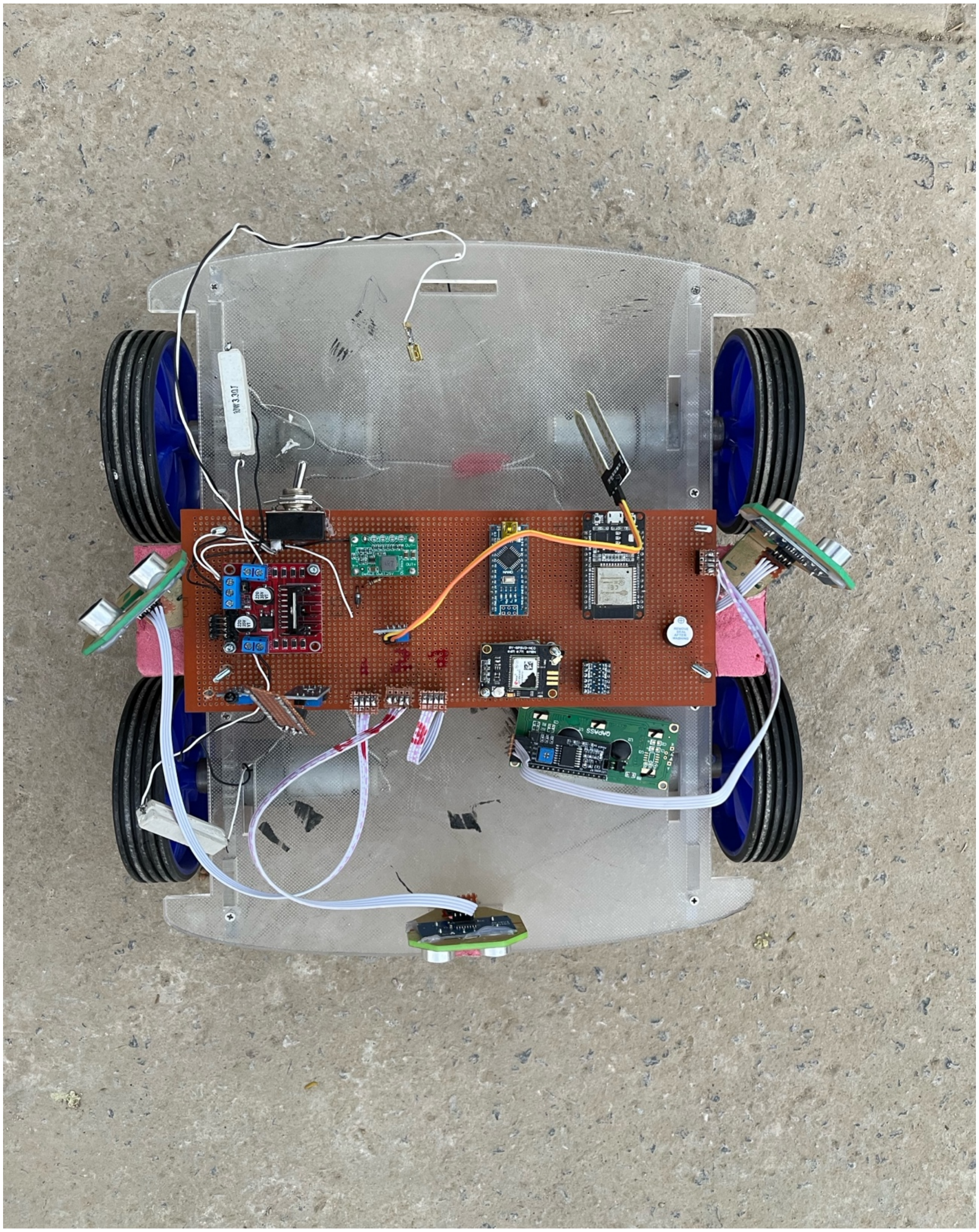

Figure 4. 1: Hardware of Car

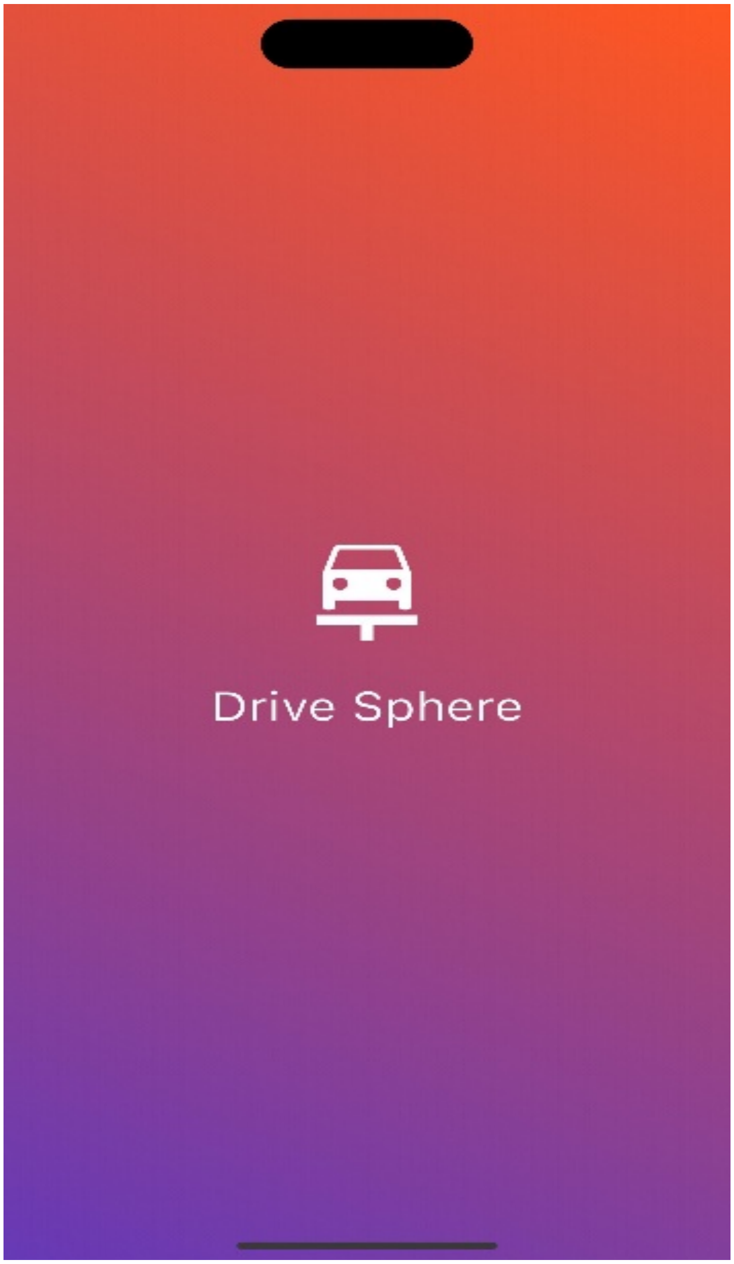

Figure 4. 2: Splash Screen

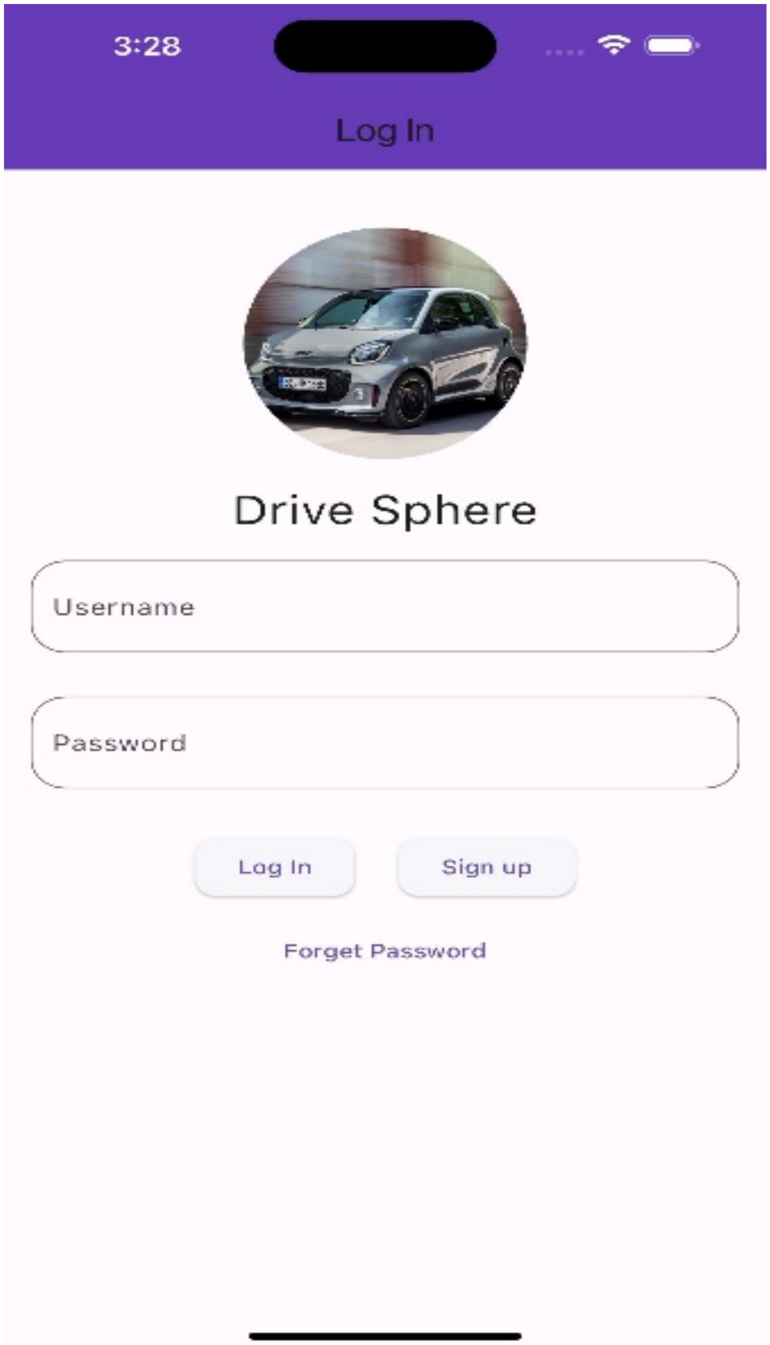

Figure 4. 3: Login Screen

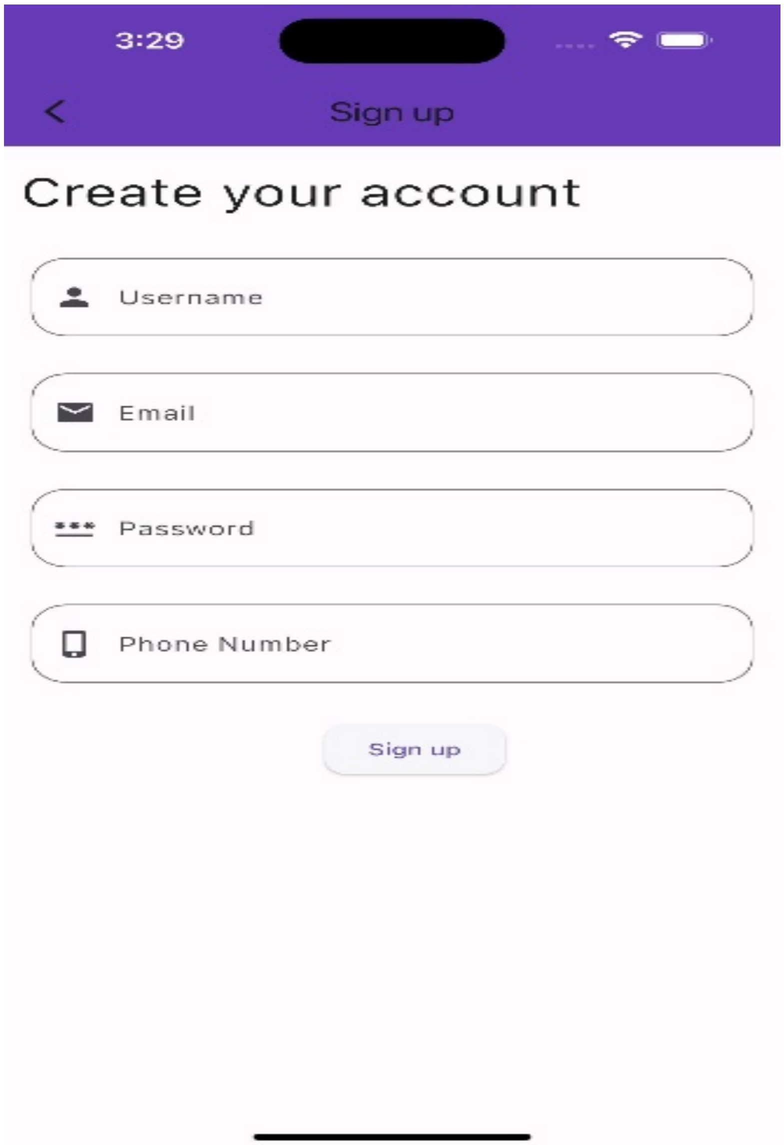

Figure 4. 4: Signup Screen

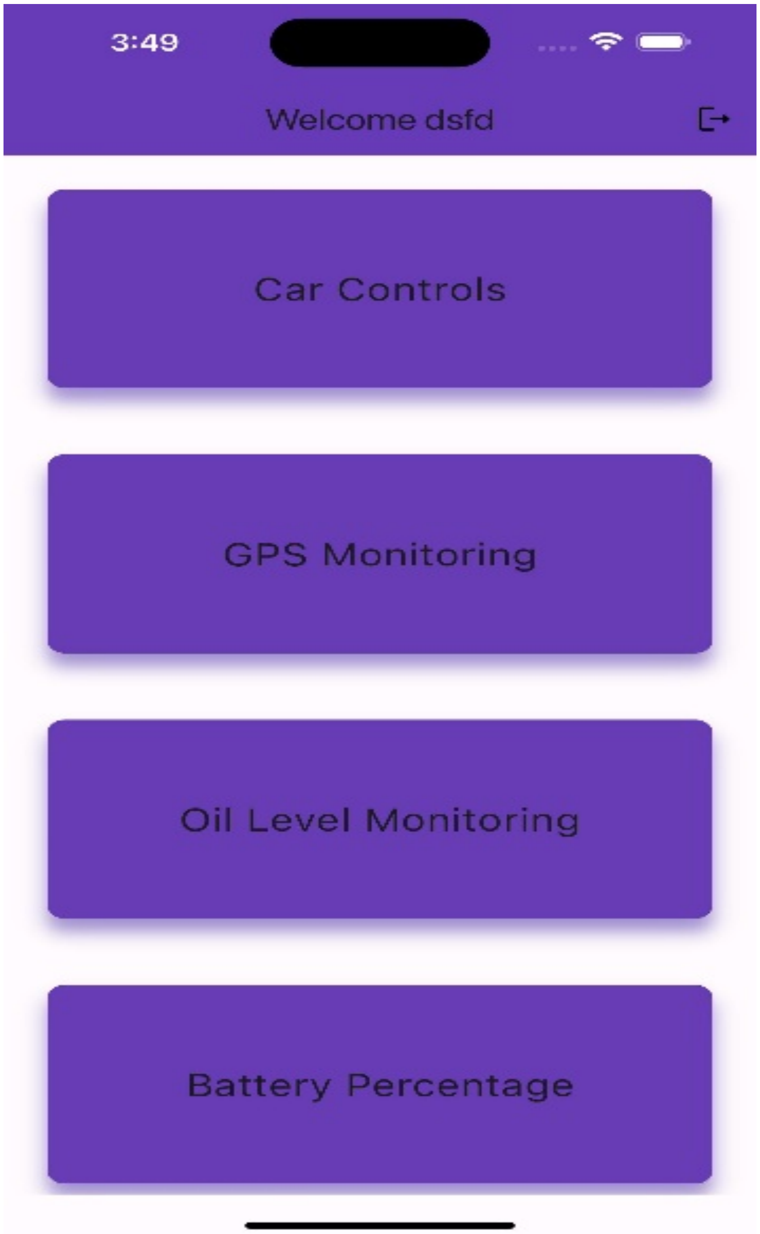

Figure 4. 5: Main Screen

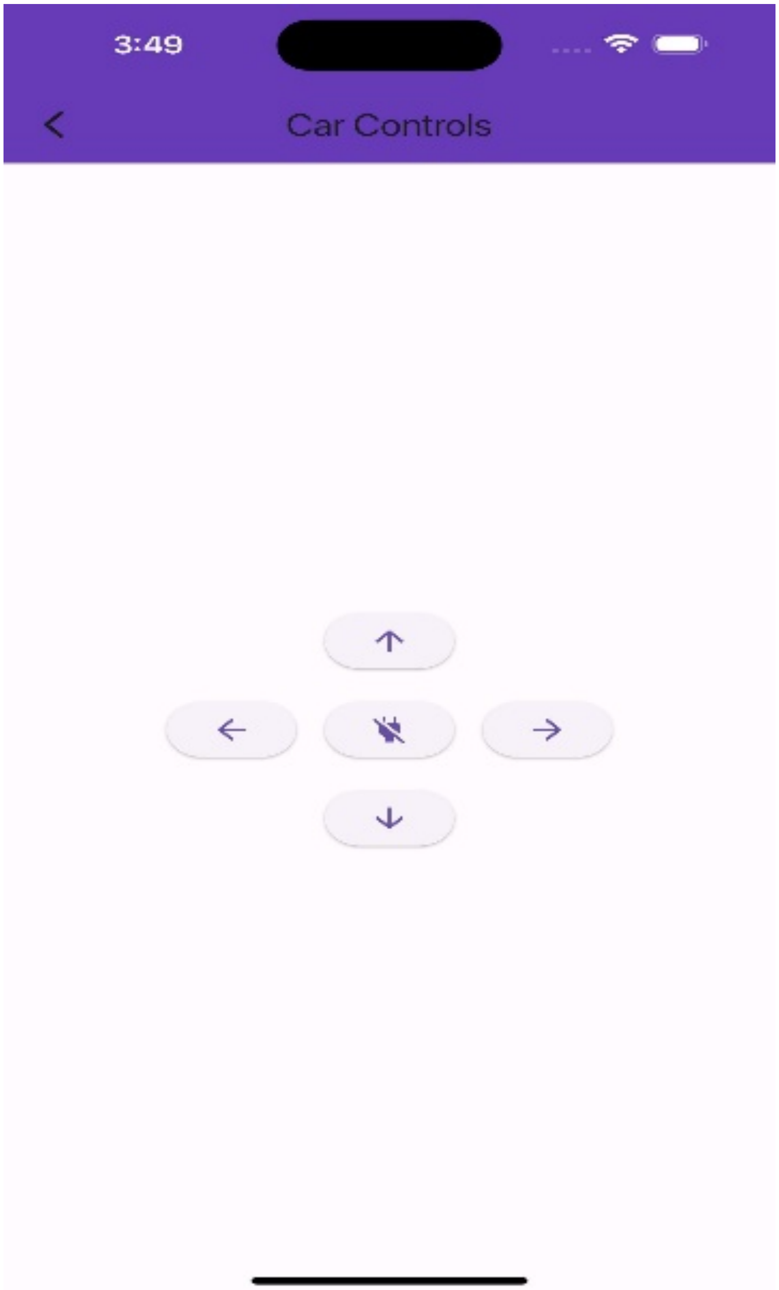

Figure 4. 6: Car Controls Screen

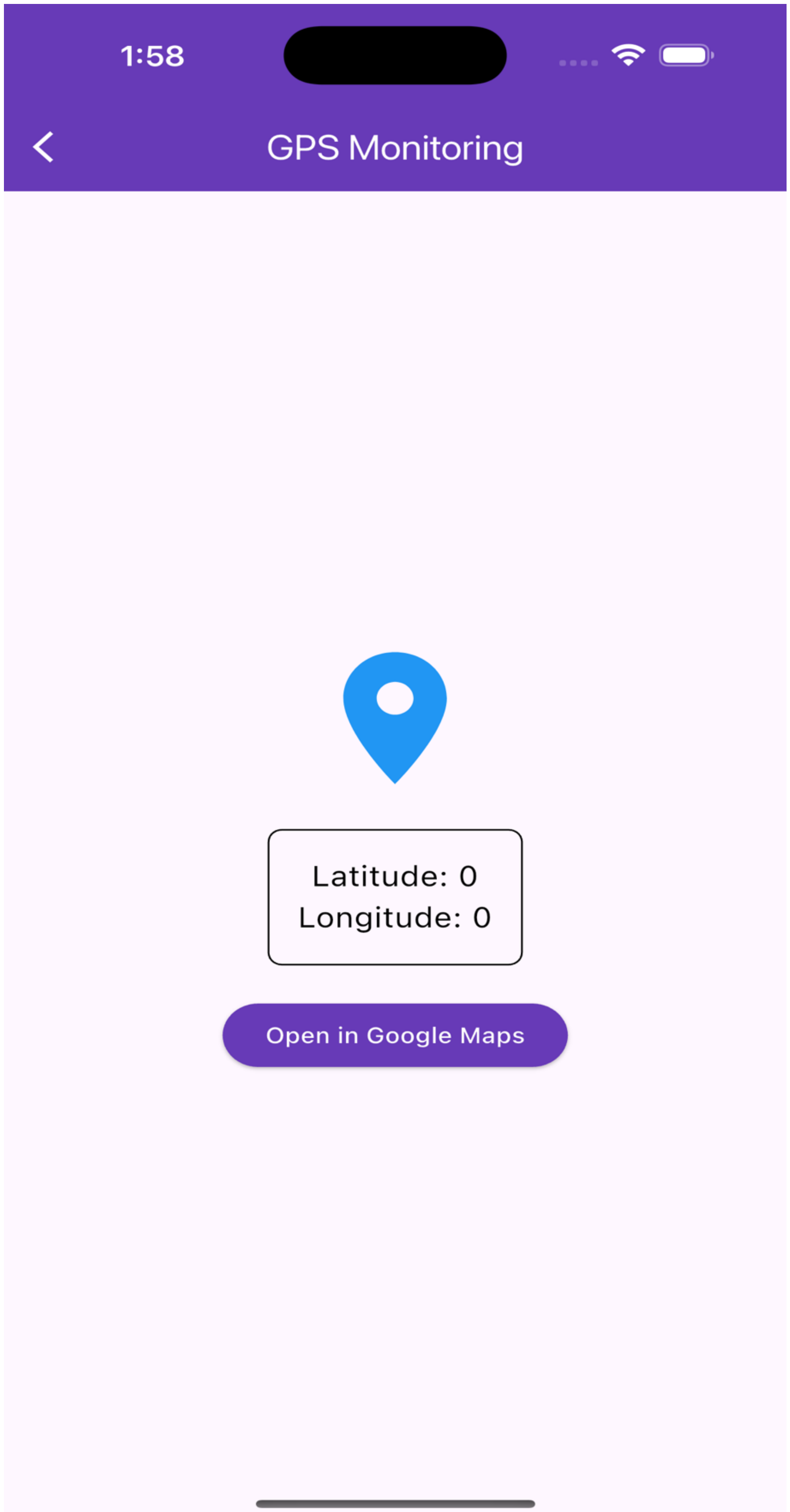

Figure 4. 7: GPS Monitoring Screen

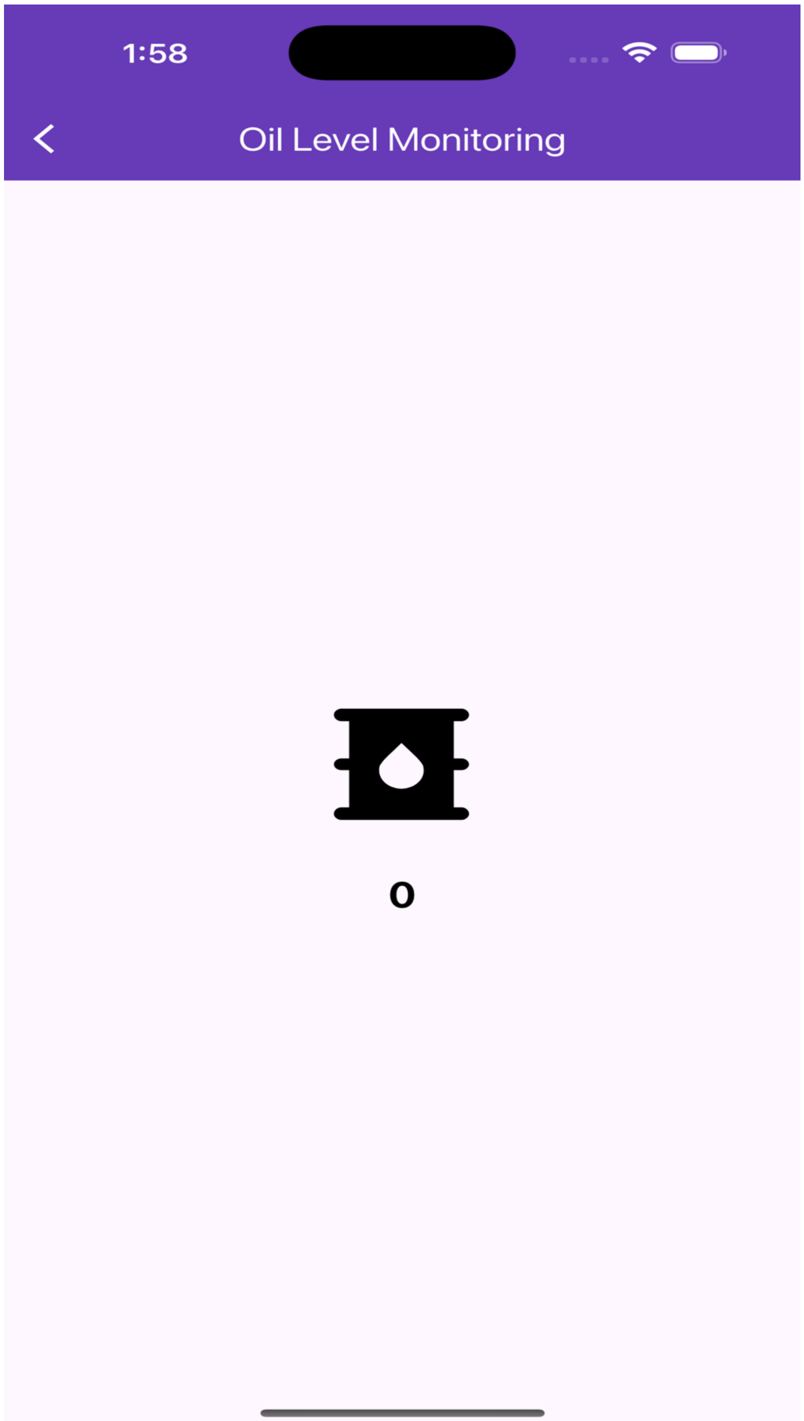

Figure 4. 8: Oil Level Monitoring Screen

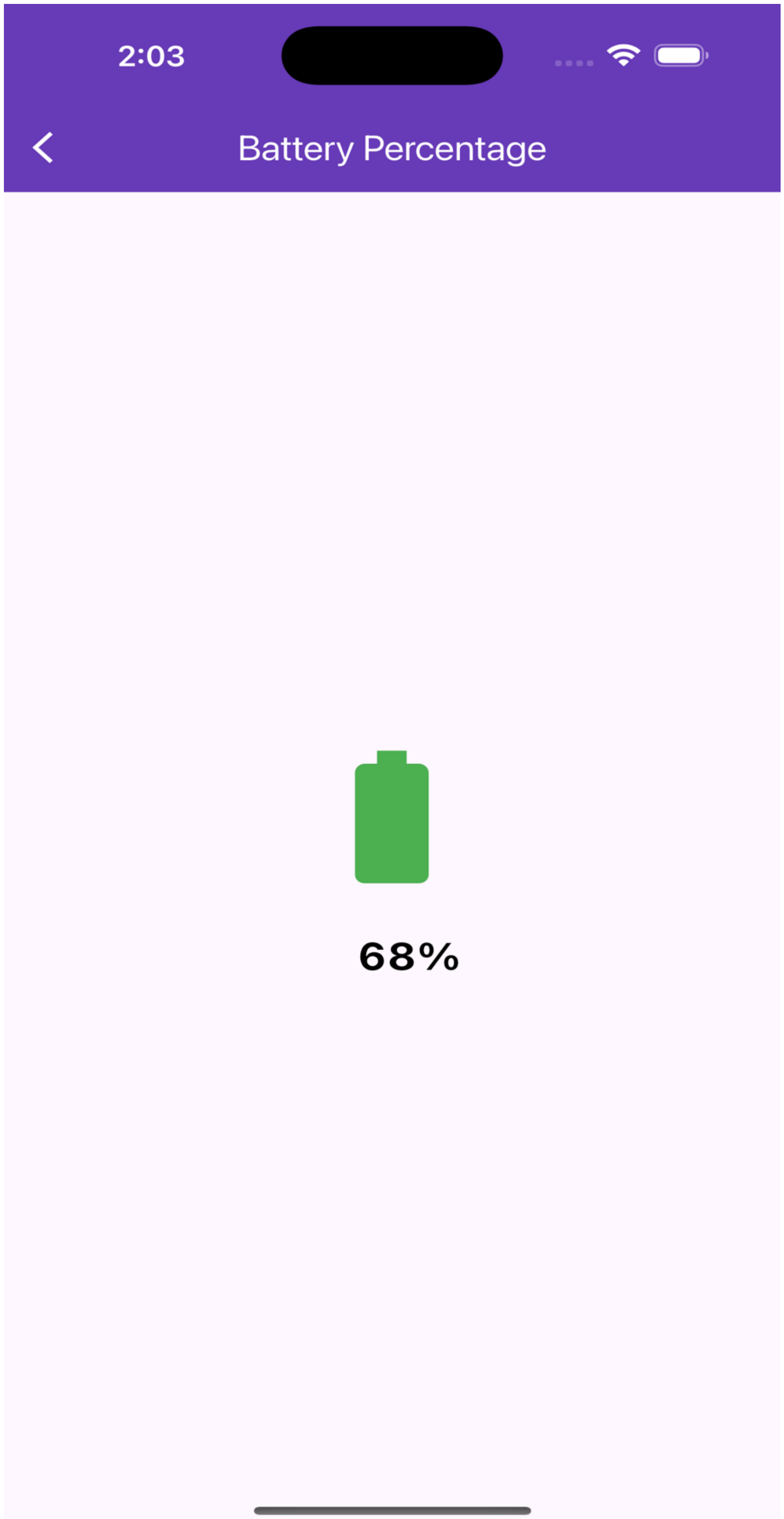

Figure 4. 9: Battery Percentage Screen

4.4. Summary

The implementation of the smart car system involved a detailed and methodical approach, addressing various challenges in hardware setup, software optimization, and system integration. The iterative development methodology and the use of specific tools and technologies were crucial in overcoming difficulties and achieving a functional autonomous vehicle. The process underscored the importance of thorough testing, modular design, and adaptability in developing complex system.

Chapter 5: Testing

5.1 Testing Techniques Employed for This Project

The following testing methods [10] were employed to guarantee functionality and dependability to make sure the Smart Car Controller Detection System Vehicle satisfies its functional and non-functional requirements:

- **Unit Testing:** To confirm their operation, separate modules, including blind spot sensors, GPS, and fog detection, were tested separately.
- **Integration Testing:** To make sure the combined software and hardware systems operated as a unit, they were tested.
- **System Testing**: To replicate real-world situations, the entire system was tested in a variety of settings.
- **Performance Testing:** Aimed at assessing the system's dependability and response under pressure.
- **Usability Testing:** Confirmed that the mobile application and system's interface were simple to use and intuitive.

5.2 Test Cases

Here is the test case represented in a table format:

Table 5. 1: Test Cases

Test Case ID	Test Objective	Preconditions	Steps	Expected Outcome
TC01	Check the fog detection system.	The fog detection sensor is set up and linked.	1. Create the illusion of fog. 2. Keep an eye on system reaction and sensor readings.	When the system detects fog, a buzzer and a mobile application sound a warning.
TC02	Check the operation of the GPS module.	The GPS module is operational and connected to the app.	1. Drive the vehicle to several places. 2. Keep an eye on the mobile application's location changes.	Accurate real-time location updates are shown on the app.
TC03	Inspect for blind spots	The blind spot sensors are working.	1. Place items in the blind area of the vehicle. 2. Track the system's reaction.	The system notifies the user when it detects things in blind regions.
TC04	Test the login for a mobile application.	The user's credentials have been registered.	1. Launch the smartphone application. 2. Type in your authentic login information. 3. Press "Login."	The user accesses the application after successfully logging in.
TC05	Check the system's functionality.	Every module is operational.	1. Run the system in typical circumstances. 2. Model stressful situations (such network outages).	The system continues to operate with few hiccups or delays.

5.3 Test Results

- **Fog Detection**: In every tested scenario, the system successfully identified fog and sent out timely alerts to guarantee safety.
- **GPS Module**: The vehicle's position was precisely and quickly tracked, providing dependable navigation assistance.
- **Blind Spot Sensor**: This feature improved situational awareness for safer driving by smoothly identifying nearby cars and obstructions.
- **Mobile Application**: The app was able to establish a successful connection with the hardware, allowing for real-time tracking and seamless remote control.
- **Performance Metrics**: Despite difficult circumstances including poor visibility and network outages, the system continued to operate consistently.

5.4 Summary

The testing procedure verified that the smart car system's primary features—such as GPS tracking, remote control via a mobile app, and sophisticated safety measures like fog and blind spot detection are carried out efficiently. After extensive testing, the system proved to be reliable, user-friendly, and extremely effective, making driving safer and more convenient. Its main goals were achieved, and every test case that was specified succeeded.

Chapter 6: Conclusion & Future Work

6.1 Contributions

With its modern technology, the Smart Car Controller Detection System marks a significant advancement in both user convenience and vehicular safety. The following are some ways that the effort advances the domain.

6.1.1 Contribution 1:

The system offers a complete safety solution by integrating blind spot identification, GPS tracking, and fog detection. It greatly increases pedestrian and driving safety by resolving important flaws in conventional vehicle systems.

6.1.2 Contribution 2:

Development of an intuitive smartphone app that permits remote control, real-time tracking, and vehicle monitoring. The app transforms how people interact with their cars by offering ease and control. With its sophisticated features and user-friendly interface, its primary goal is to improve the user experience. This is how it is unique:

- **Remote Control:** From smartphones, users may easily adjust climate and vehicle settings. This provides unparalleled convenience and is particularly helpful in emergency situations or bad weather.

- **Real-Time Tracking:** Users may track the whereabouts of their vehicle in real-time with the app's live GPS tracking feature. This is crucial for fleet management, thwarting theft, and guaranteeing loved ones' safety when travelling.

- **Safety Alerts:** Users receive immediate messages when important events like fog detection, blind spot warnings, a car fire, or unauthorized access occur. Users can react swiftly and keep one step ahead of possible dangers thanks to these notifications.

- **Personalized Experience:** By allowing users to customize settings and user profiles, the app allows users to customize the interface and features to suit their own tastes.

• **IoT Integration:** Serving as a central hub, the app easily integrates with smart auto system components that are IoT-enabled, guaranteeing flawless hardware-software connection.

• **Cross-Platform Accessibility:** The application is inclusive and accessible to a wide range of users, making it user-friendly and adaptable. It is made to function on a variety of devices and operating systems.

6.2 Future Work

6.2.1 Improvements to the existing system:

1. Sensor Precision and Calibration:

- Improve hardware integration and sophisticated calibration to raise the accuracy of fog and blind spot detecting sensors. This will improve performance in real-world situations and lower false alarms.

2. Expanded Mobile Application Features:

- **Voice Commands:** To improve user convenience, implement hands-free voice control for select features.
- **Predictive Maintenance:** Include functions that use sensor data analysis to forecast maintenance requirements, including monitoring tire pressure, oil levels, or battery health, and alert users before issues occur.
- **Geofencing**: This feature lets users establish geographic limits and sends out notifications when the vehicle crosses them.
- **Enhanced Security:** To ensure safe data transfer between the app and the vehicle, use multi-factor authentication, biometric app access, and cutting-edge encryption.

6.2.2 Future System Designs:

Future System Designs

1. Autonomous Driving Features:

- Support semi-autonomous or fully autonomous driving by integrating features including adaptive cruise control, automatic emergency braking, and lane assistance.

2. Vehicle-to-Everything (V2X) Communication:

- This allows the car to interact with pedestrians, other cars, and road infrastructure to avoid collisions, get real-time traffic updates, and facilitate cooperative driving.

3. Augmented Reality (AR) Dashboard:

- To improve situational awareness, place AR overlays on the windscreen for navigation, hazard alerts, and other real-time data.

4. Modular System Design:

- Modular System Design: Create an adaptable system that can be scaled and cost-effective for a range of applications by allowing users to add or remove components as needed.

Empirical Studies

1. Driver Behavior Analysis:

- Examine how users engage with the system to find areas where usability and user experience can be enhanced.

2. Performance Comparisons:

- To emphasize the system's advantages and pinpoint areas for development, compare it to other comparable technologies.

3. Impact on Road Safety:

- Gather empirical data to evaluate how well the system works to lower collision rates and encourage safer driving.

Sustainability and Green Technology

1. Energy Efficiency:

- To reduce power consumption and make the system compatible with electric vehicles, provide hardware and software that uses less energy.

2. Renewable Energy Integration:

- To power parts like sensors and cameras, investigate incorporating solar panels or other renewable energy sources.

3. Eco-Driving Assistance:

- To lessen the impact on the environment, provide tools that help drivers adopt energy-saving or fuel-efficient driving practices.

Expansion into New Domains

1. Fleet and Logistics Management:

- Modify the system for business usage, adding functions like load monitoring, real-time tracking, and fleet vehicle predictive maintenance.

2. Emergency and Medical Services:

- Integrate priority routing and real-time patient or vehicle status updates into the system for ambulances and rescue vehicles.

3. Smart City Integration:

- Work with smart city initiatives to incorporate the system into urban infrastructure, which will improve public transport and allow for more intelligent traffic control.

References:

[1] T. L. M. T. a. O. T. Darko Hercog, "Design and Implementation of ESP32-Based IoT Devices," *Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška Cesta 46, 2000 Maribor, Slovenia,* vol. , no. , p. , 2023.

[2] D. S. S. R. Reza Fauzan, "A Different Approach on Automated Use Case Diagram Semantic Assessment," *International Journal of Intelligent Engineering & Systems,* vol. 14, no. 1, 2021.

[3] K. K. D. A. D. M. N. S. V. A. a. J. K. B. Goyal, "Empowering Assets and Vehicles with Cutting-Edge ESP32 Real-Time Tracking System," *11th International Conference on Computing for Sustainable Global Development ,* pp. 504-509, 2024.

[4] B. Students, Smart Car, vol. 1, Islamabad: Iqra University, 2018.

[5] B. Students, Car Health and Security Systems, Islamabad: Iqra University, 2020.

[6] M. A. J. M. M. D. a. A. Z. A. S. K. Shnain, "Developing Vehicles Blind Spot Sensors Using Arduino to Avoid Parking Accidents," *2024 35th Conference of Open Innovations Association (FRUCT),* vol. 1, no. , pp. 683-691, 2024.

[7] C. C. Joe Essien, "Ultrasonic Sensor-Based Embedded System for Vehicular Collision Detection and Alert," *Journal of Computer and Communications,* vol. 11, no. 8, 2023.

[8] A. Kurniawan, Arduino Nano A Hands-On Guide for Beginner, PE Press, 2019.

[9] "Amazon," Carbest, 2024. [Online]. Available: https://www.amazon.com/CarBest-Ultrasonic-Detection-Monitoring-Assistant/dp/B07BCZ68LL.

[10] A. Z. Shamsu Abdullahi, "Software Testing: Review on Tools, Techniques and Challenges," *International Journal of Advanced Research in Technology and Innovation,* vol. 2, no. 2, pp. 11-18, 2020.