

**Department of Computer Science**

**COMP4300-Graduation Project**

**Spring 2024/2025**



**MotorCarCare**

**Group Members with IDs:**

|  |  |
| --- | --- |
| Alaa Dar Yaqoub | 1202007 |
| Ayham Jameel | 1190935 |

**Supervisor:** Mr. Fadi Khalil

Section – B

Title of Project: MotorCarCare

Project No: Section (30)

Supervisor: Mr. Fadi Khalil

Key Areas:

* Website and application for booking appointments online.
* Track repair progress in real-time.
* Manage appointments easily for workshop managers
* Firebase framework for backend
* Built website with PHP, HTML, and CSS.
* Built an App with Flutter and Dart.

**Section – C**

**Student Signature:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Date Submitted: [ /07/2025 ]**

**First Supervisor Name: Mr. Fadi Khalil**

**First Supervisor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date Approved: \_\_\_ / \_\_\_\_ / \_\_\_\_\_**

**Acknowledgments**

Praise and glory be to Almighty God, whose mercy and guidance granted us the patience, strength, and determination to complete this project.We would like to express our sincere gratitude to our esteemed professor, Mr. Fadi Khalil, for his continuous support, valuable guidance, and encouragement throughout the development of this project. His supervision and insights were fundamental to our success.Our appreciation also goes to all faculty members and respected professors of the Faculty of Computer Science at Birzeit University, whose dedication and knowledge shaped our academic journey.We extend heartfelt thanks to our families, friends, and colleagues for their constant support, encouragement, and understanding. A special thanks to our dear parents, who have been a source of strength and who supported us in every possible way throughout the years—we are forever grateful for your patience and sacrifices.Lastly, we are thankful to everyone who contributed positively, in any form, to the completion of this work.

**Abstract**

Many car owners face difficulties when trying to book car maintenance services. Traditional methods, such as calling or visiting workshops, are time-consuming and often lead to confusion. Workshop managers also struggle with organizing appointments, tracking services, and managing customer information.

To solve this problem, we developed **MotorCarCare**, a web and mobile application that allows car owners to book services online, track their car’s repair status, and receive automatic service reminders. Workshop managers can use the system to manage appointments, view service history, and organize daily operations more efficiently.

The system was built using PHP, HTML, CSS for the web interface, and Flutter with Firebase for the mobile version. These technologies ensured that the platform is fast, user-friendly, and accessible across devices.

Our testing showed that **MotorCarCare** improves service efficiency, reduces manual errors, and provides a better experience for both customers and workshop staff.

Table of Contents

[Chapter 1: Introduction 11](#_Toc202141833)

[1.1 Overview 11](#_Toc202141834)

[1.2 Goals & Objective 11](#_Toc202141835)

[1.3 Overview of the technical area 12](#_Toc202141836)

[1.4 Overview of the report 13](#_Toc202141837)

[Chapter 2: Related Work 15](#_Toc202141838)

[2.1 Background 15](#_Toc202141839)

[2.2 Review of past/reported work 15](#_Toc202141840)

[2.2.1 Online Car Maintenance Systems 15](#_Toc202141841)

[2.2.2 Strengths and Limitations of Existing Systems 15](#_Toc202141842)

[2.3 Review of similar apps 16](#_Toc202141843)

[2.4 Algorithm 20](#_Toc202141844)

[2.5 Tools and Technology 21](#_Toc202141845)

[2.5.1 Flutter 21](#_Toc202141846)

[2.5.2 PHP. 21](#_Toc202141847)

[2.5.3 Firebase 21](#_Toc202141848)

[2.5.4 HTML and CSS. 22](#_Toc202141849)

[2.6 Real-Time Updates and Automation 22](#_Toc202141850)

[Chapter 3: System Analysis & Design 22](#_Toc202141851)

[3.1 System Model and System Architecture 22](#_Toc202141852)

[3.2 Product Description 24](#_Toc202141853)

[3.2.1 System Objectives 24](#_Toc202141854)

[3.2.2 System Main Features 24](#_Toc202141855)

[3.2.3 Operating Environments 25](#_Toc202141856)

[3.2.4 Constraints 25](#_Toc202141857)

[3.2.5 Functional Requirements 25](#_Toc202141858)

[3.2.6 Non-Functional Requirements 27](#_Toc202141859)

[3.3.1 Actors (actor list and description of their roles) 28](#_Toc202141860)

[3.3.2 Use Cases 29](#_Toc202141861)

[*Use case Booking Car Service.* 30](#_Toc202141862)

[*Use case Manage Repair Car* 31](#_Toc202141863)

[Use case Manage Insurance 32](#_Toc202141864)

[3.3.3 Use Cases Diagram 32](#_Toc202141865)

[3.4 System Models 34](#_Toc202141866)

[3.4.1 ClassDiagram 34](#_Toc202141867)

[3.4.2 Sequence Diagram 35](#_Toc202141868)

[3.4.3 Activity Diagram 36](#_Toc202141869)

[3.4.4 State Chart Diagram 37](#_Toc202141870)

[3.5 System Architecture 37](#_Toc202141871)

[3.5.1 Sub-System (descriptions of the sub-systems and their services) 37](#_Toc202141872)

[3.5.2SoftwareArchitecture 39](#_Toc202141873)

[3.5.3 DeploymentDiagram 40](#_Toc202141874)

[3.6 Data Management and Data Models 41](#_Toc202141875)

[Chapter 4: Implementation 42](#_Toc202141876)

[4.1 Chapter overview 42](#_Toc202141877)

[4.2 Screen shots of your actual application 42](#_Toc202141878)

[4.3 Website screens: 47](#_Toc202141879)

[Chapter 5: System Testing 65](#_Toc202141880)

[5.1 Overview 65](#_Toc202141881)

[5.2 List of features to be tested 65](#_Toc202141882)

[5.3 List of test cases 66](#_Toc202141883)

[5.4 End-user testing (Acceptance Testing), if applicable. 80](#_Toc202141884)

[Chapter 6: Conclusion and Future Works 81](#_Toc202141885)

[6.1 Review of the project 81](#_Toc202141886)

[6.2 Future Works 81](#_Toc202141887)

[5. References and Bibliography 82](#_Toc202141888)

**List of Table**

[Table1 APP Comparison 20](#_Toc187716122)

[Table 2 Principal Actors and Their Roles 31](#_Toc187716123)

[Table 3Secondary Actors 31](#_Toc187716124)

[Table4 Use Case Registration 32](#_Toc187716125)

[Table5 Use case Booking Car Service 34](#_Toc187716126)

[Table6 Use case Manage Repair Car 35](#_Toc187716127)

[Table 7 Use case Manage Insurance 35](#_Toc187716128)

[Table 8 CaseStud1 Booking appointment 54](#_Toc187716129)

[Table 9 CaseStudy2 Viewing Service History 56](#_Toc187716130)

[Table 10 CaseStudy3 Updating User Profile 58](#_Toc187716131)

[Table 11CaseStudy4 Sending Notification 60](#_Toc187716132)

[Table 12 CaseStudy5 Sign Up 61](#_Toc187716133)

[Table 13 CaseStudy6 Login 62](#_Toc187716134)

[Table 14 CaseStudy7 Forgot password 63](#_Toc187716135)

[Table 15 CaseStudy8 Payment System 65](#_Toc187716136)

[Table 16 CaseStudy9 Login and Booking History Verification 66](#_Toc187716137)

[Table 17 test register 67](#_Toc187716138)

**List of Figures**

[Figure1 add your car (Carfax) 18](#_Toc187742282)

[Figure2 Service History (Carfax) 18](#_Toc187742283)

[Figure 3Fuel Tracker (Carfax) 18](#_Toc187742284)

[Figure 4cost log (fuel) 19](#_Toc187742285)

[Figure5 information of fuel (fuel) 19](#_Toc187742286)

[Figure 6Sign in (fuel) 19](#_Toc187742287)

[Figure7 Drivvo app (imagDrivo) 20](#_Toc187742288)

[Figure8 simulation Software 26](#_Toc187742294)

[Figure9 Use Cases Diagram 37](#_Toc187742295)

[Figure10 Class Diagram 38](#_Toc187742296)

[Figure11 Sequence Diagram 39](#_Toc187742297)

[Figure12 Activity Diagram 40](#_Toc187742298)

[Figure13 State Chart Diagram 41](#_Toc187742299)

[Figure14 Software Architecture 43](#_Toc187742300)

[Figure15 Deployment Diagram 44](#_Toc187742301)

[Figure16 Data Management and Data Models 45](#_Toc187742302)

[Figure17 welcompage 46](#_Toc187742303)

[Figure18 user type 46](#_Toc187742304)

[Figure19 signup 47](#_Toc187742305)

[Figure20 login customer 47](#_Toc187742306)

[Figure21 appointment 48](#_Toc187742307)

[Figure22 My Appointment 48](#_Toc187742308)

[Figure23 services history 49](#_Toc187742309)

[Figure24 information of services history 49](#_Toc187742310)

[Figure25 status of appointment 49](#_Toc187742311)

[Figure27 notification 50](#_Toc187742312)

[Figure28 completed 50](#_Toc187742313)

[Figure29 rejection 50](#_Toc187742314)

[Figure30 payment stamen 51](#_Toc187742315)

[Figure31 forget password 52](#_Toc187742316)

[Figure32 email to reset password 52](#_Toc187742317)

[Figure33 new password 52](#_Toc187742318)

[Figure34 car care 53](#_Toc187742319)

[Figure35 company or register 54](#_Toc187742320)

[Figure36 create account 55](#_Toc187742321)

[Figure37 Login 56](#_Toc187742322)

[Figure38 My Appointment on website 57](#_Toc187742323)

[Figure39 services Booking 58](#_Toc187742324)

[Figure40 our value 59](#_Toc187742325)

[Figure41 Contact Us 60](#_Toc187742326)

[Figure42 Update Profile 61](#_Toc187742327)

[Figure43 Sign up 62](#_Toc187742328)

[Figure44 company login 63](#_Toc187742329)

[Figure45 Dashboard 64](#_Toc187742330)

[Figure46 Website Appointment screen 65](#_Toc187742331)

[Figure47 Update status 66](#_Toc187742332)

[Figure48 Add New services 67](#_Toc187742333)

[Figure49 manage services 68](#_Toc187742334)

[Figure50 Update services 69](#_Toc187742335)

# Chapter 1: Introduction

## 1.1 Overview

Online car maintenance booking systems like " MotorCarCare" have become essential tools in the modern technological era, significantly enhancing customer experience and streamlining maintenance management processes. The system enables customers to conveniently and efficiently book maintenance appointments online, saving both time and effort.

The core benefits of " MotorCarCare" include providing a user-friendly interface that allows users to access maintenance information, select desired services, and schedule appointments based on their availability. It also allows customers to track the status of their maintenance in real time, receiving timely updates regarding their vehicle's condition.

Moreover, the system enhances the efficiency of maintenance workshops by organizing appointments and automating processes, thereby reducing human errors and increasing operational accuracy. As a result, " MotorCarCare" serves as an ideal solution for both customers and service providers, improving the overall service quality and contributing to higher customer satisfaction.

## 1.2 Goals & Objective

The main goal of the **MotorCarCare** system is to simplify and organize car maintenance for both car owners and workshop managers. The system saves time, improves service quality, and reduces errors caused by manual processes.

**Specific objectives:**

* Allow car owners to book and manage maintenance appointments easily online.
* Provide real-time tracking of service status and send automated reminders.
* Help workshop managers manage appointments, service history, and customer details efficiently.
* Reduce the need for paperwork by storing service records digitally.
* Ensure access to the platform via both mobile app and website.
* Improve the communication between customers and workshops through clear notifications and confirmations.
* Support secure login and avoid duplicate registration (e.g., same email can’t be registered twice).

#### **For Car Owners:**

* **Book Easily**: Car owners can book their maintenance appointments online without needing to visit the workshop in person.
* **Save Time**: Online booking reduces waiting time and avoids delays.
* **Track Services**: Users can follow up on their car’s repair status and view their full service history.
* **Get Reminders**: The system sends automatic reminders for services like oil changes or checkups.
* **Share Feedback**: Customers can write reviews and feedback after their appointments to help improve service.

#### **For Workshop Managers:**

* **Organize Appointments**: Managers can schedule services and assign tasks more efficiently.
* **Follow Service History**: The system keeps track of all past services for each car.
* **Manage Inventory**: It helps keep an eye on available parts and supplies to avoid delays or shortages.

## 1.3 Overview of the technical area

To build MotorCarCare, we used modern technologies to create a fast, reliable, and easy-to-use system. Here's a breakdown of the tools we used:

**Web Application:**

* **Frontend:** HTML, CSS, and Bootstrap were used to design the website for a clean and responsive interface.
* **Backend:** PHP handles server-side logic, including user registration and appointment booking.
* **Database:** Firebase, utilizing JavaScript, is used to store and manage real-time data.

**Mobile Application:**

* **Frontend:** Flutter and Dart were used to build the mobile App, allowing it to work on both Android and iOS devices.
* **Backend:** Firebase also powers the mobile App's backend, handling user data, notifications, and real-time updates.

**Why These Technologies?**

* **Flutter**: A powerful tool for building cross-platform apps with one codebase, saving time and providing a consistent experience across devices.
* **PHP**: A reliable language for server-side tasks, perfect for handling user requests and processing data.
* **Firebase**: A cloud-based platform for real-time data storage, user authentication, and notifications, ensuring the App works smoothly even when offline.

## 1.4 Overview of the report

In this report, we examined the challenges faced by car owners and workshop managers in accessing car maintenance and repair services. We identified common issues, including long waiting times, poor communication, and difficulties managing appointments and services. To address these issues, we developed MotorCarCare, a web and mobile application that simplifies the process of scheduling and managing car maintenance.

The MotorCarCare system has key features such as:

* **Appointment Booking**: Customers can easily book and manage their car service appointments online, reducing the need for phone calls or in-person visits.
* **Service Tracking**: Customers can check the status of their car repairs and view their service history, while workshop managers can manage repair jobs and inventory.
* **Notifications and Reminders**: The system automatically sends reminders before scheduled appointments and provides real-time updates about service status, helping customers stay informed.

To design the system, we studied existing car maintenance applications and identified areas where they failed to meet customer expectations. This analysis helped us build a more user-friendly and efficient platform.

We created different diagrams to represent the system structure and behavior, including **use case diagrams, class diagrams, sequence diagrams, and activity diagrams**, to clarify how the system functions and how the different parts are connected.

We also outlined both **functional** and **non-functional requirements**.

* **Functional requirements** include booking appointments, viewing service history, and managing workshop schedules.
* **Non-functional requirements** focus on the system's **performance, reliability, usability**, and **compatibility across devices** such as smartphones, tablets, and computers.

Furthermore, we discussed some **system limitations**, such as the need for an internet connection and a rule that appointments cannot be canceled less than 24 hours in advance.

We also considered **ethical aspects**, particularly protecting customer data and respecting user privacy.

After finalizing the system design, we conducted several tests to ensure it meets the needs of both car owners and workshop managers. This included testing the booking process, service tracking, and notification features to ensure everything works smoothly and reliably.

# Chapter 2: Related Work

## 2.1 Background

The **MotorCarCare** project aims to improve the car maintenance experience by leveraging modern technology. The web and mobile applications enable car owners to book appointments online, receive real-time notifications about their service status, and view their complete service history. At the same time, it helps workshop owners manage bookings, organize services, and keep records, making the entire process more efficient for both customers and businesses.

According to a recent study, mobile car service applications have significantly reduced waiting times and improved customer satisfaction by simplifying the booking process and service tracking [1]. This supports our system’s goal to provide a smooth and efficient experience for all users.

## 2.2 Review of past/reported work

## ****2.2.1 Online Car Maintenance Systems****

Several applications have been developed to support car maintenance scheduling and tracking.One notable example is **CarFax Car Care**, which allows users to view their vehicle’s service history and receive maintenance reminders. It focuses on keeping detailed records and improving vehicle performance by ensuring timely servicing. Another app is **Fuelio**, which tracks fuel consumption and vehicle-related expenses. It’s widely used by individuals and small businesses to manage car costs and maintenance schedules efficiently. **Drivvo** also provides similar features, offering financial insights into fuel, maintenance, and insurance costs.

## ****2.2.2 Strengths and Limitations of Existing Systems****

While these apps offer helpful features like reminders and cost tracking, many lack a direct booking system integrated with workshops. Most require manual entry of service details and don’t support appointment confirmation or workshop-side management.

According to research, existing car maintenance platforms often fail to integrate workshop management, causing delays and lack of service updates for users [2].

MotorCarCare was designed to fill this gap. It provides a complete solution that allows customers to:

* Schedule services online.
* Track service status.
* View service history.
* Receive reminders.

At the same time, workshop owners can:

* Manage appointments.
* Update service status.
* View and organize service records.

## 2.3 Review of similar apps

Booking a service appointment on the web and mobile application makes it much easier for clients, as they can easily view available appointments and quickly determine which options are most suitable for their vehicle needs. Additionally, they can access their profile page to view their recent appointment dates and the services provided to their vehicle. These are some of the web and mobile applications used in these areas:

**1- CARFAX Car Care:** Carfax is a comprehensive web and mobile application dedicated to providing vehicle owners with detailed maintenance histories and reminders for upcoming service needs. Catering to both web-based and mobile app users, Carfax offers a range of features, including maintenance tracking, service record logging, and reminders for scheduled maintenance tasks. Users can easily access their vehicle's maintenance history reports, making it a valuable tool for those interested in maintaining their vehicle's resale value and staying up-to-date with essential service requirements. The application also provides real-time notifications and insights, ensuring users never miss important service deadlines, ultimately helping extend the life of their vehicles and optimize performance. (CarFax)

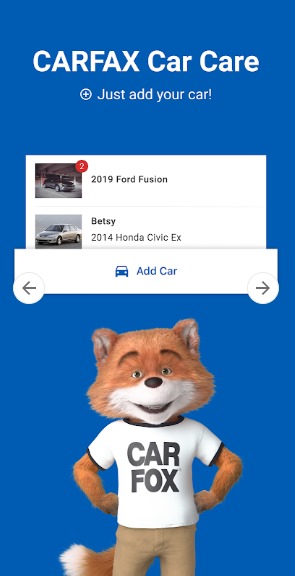
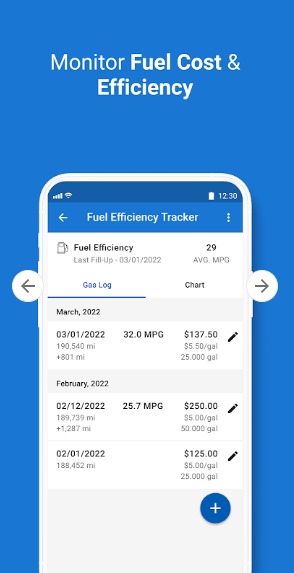
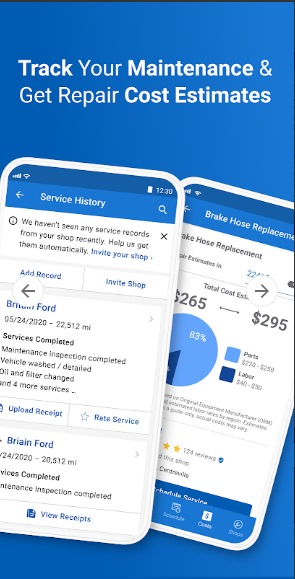


Figure 1 add your car

Figure2 Service History (Carfax)

Figure3 Fuel Tracker (Carfax)

Figure1 add your car (Carfax)

**2- Fuelio:** Fuelio is a

versatile mobile app that seamlessly combines fuel tracking with cost management to offer users detailed insights into their vehicle's operational expenses. Designed for individual drivers and small business owners alike, Fuelio provides essential features such as fuel consumption tracking, cost analysis, and mileage statistics. Users can efficiently monitor their fuel expenses, create budgets, and generate comprehensive reports to aid in budgeting and financial planning. With its user-friendly interface and robust functionality, Fuelio is an invaluable tool for anyone looking to manage their vehicle expenses effectively.

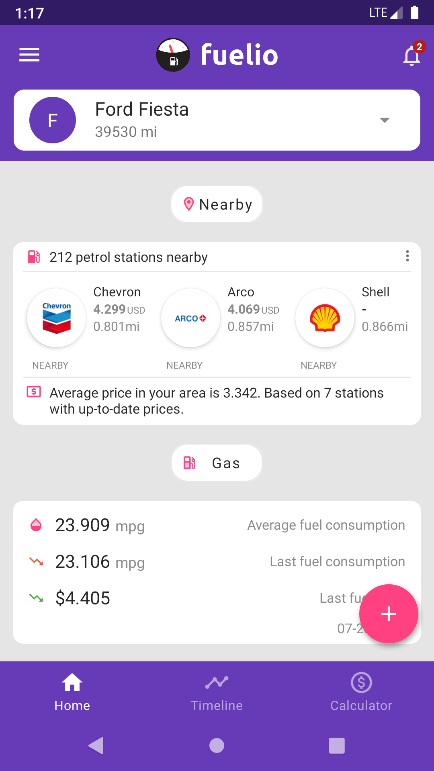
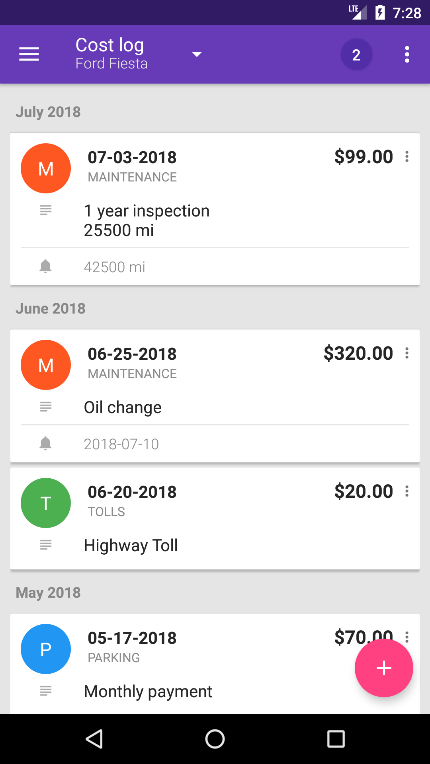
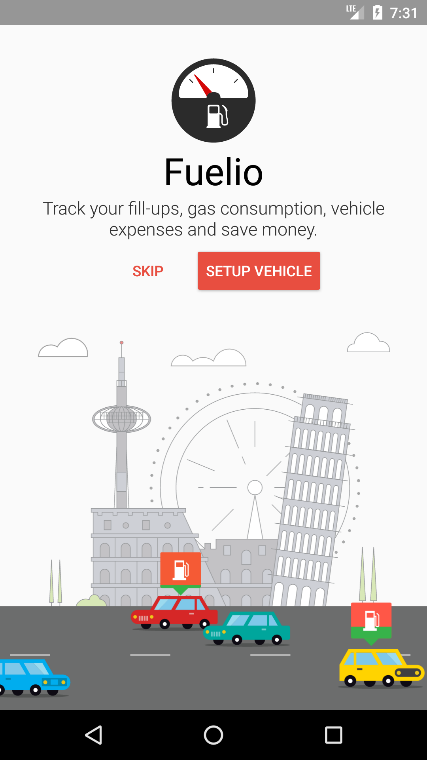
****

Figure 4cost log (fuel)

Figure 5 information of fuel (fuel)

Figure 6Sign in (fuel)

**3- Drivvo:**  Drivvo is a versatile mobile app designed to meet the needs of both personal vehicle owners and commercial fleet operators, offering comprehensive expense management for all aspects of vehicle operation. With a focus on expense tracking, Drivvo enables users to monitor costs related to fuel, maintenance, insurance, and more. The App empowers users to generate detailed reports on their spending habits, facilitating better financial planning and decision-making. Additionally, Drivvo allows for the recording and categorization of vehicle expenses, providing a clear overview of expenditures. Its intuitive budget management tools further assist users in effectively managing their vehicle-related expenses, making Drivvo an essential companion for anyone seeking to maintain control over their vehicle finances. (drivoo)

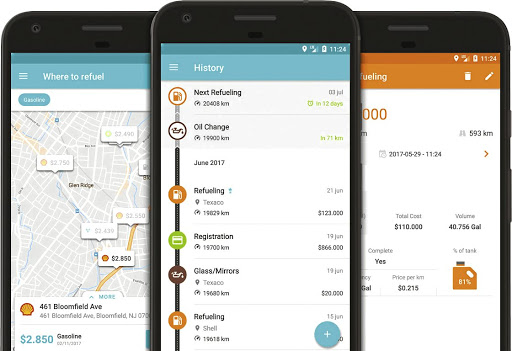


Figure7 Drivvo app (imagDrivo)

**4- Fixd:** Fixd is an innovative mobile app designed to empower both car owners and mechanics with real-time automotive diagnostics and maintenance solutions. Tailored for use by car owners and mechanics alike, Fixd offers a range of features, including engine diagnostics and maintenance alerts, ensuring timely attention to vehicle issues. The App also provides invaluable repair guidance, tracks maintenance needs, and offers practical solutions. With its intuitive interface serving as a communication bridge between technicians and car owners, Fixd revolutionizes the way vehicle maintenance is approached, ensuring peace of mind and optimal performance for all users. (fixed)

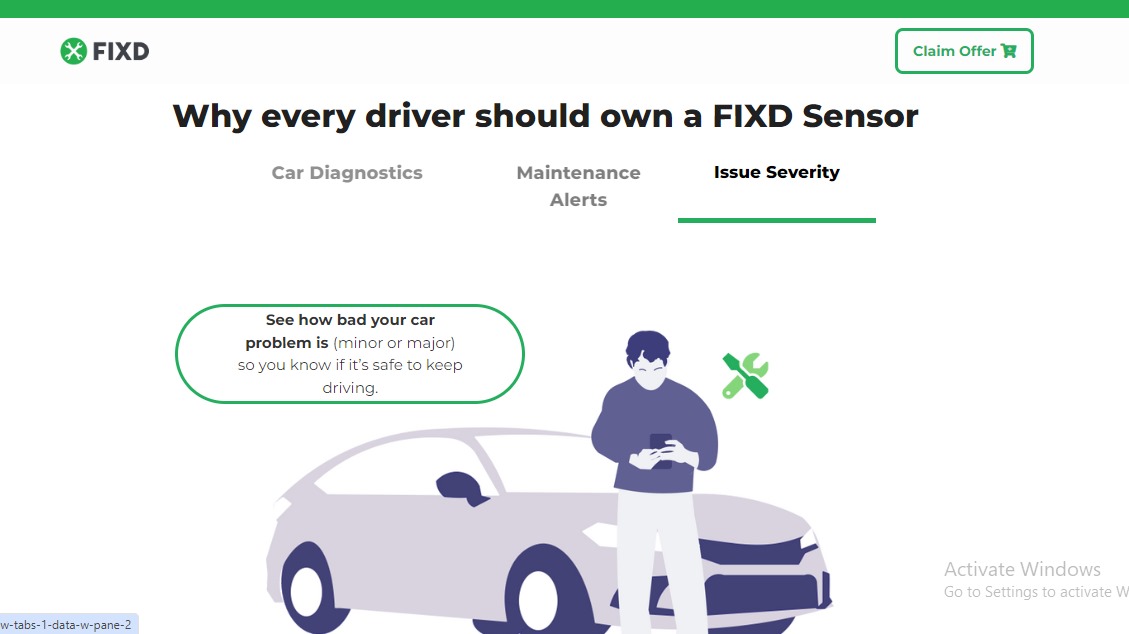


Figure8 Fixed App (fix image)

2.2.3 App Comparison

The table below explains the comparison between the mentioned web/mobile application MotorCarCare web/mobile application based on some features as shown below:

Table 1 APP Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Features** | **MyCarFax** | **Fuelio** | **Drivvo** | **Fixed** | **MotorCarCare** |
| Online Booking | ✓ | ✓ | ✓ | 🗶 | ✓ |
| Client Profile file | 🗶 | ✓ | ✓ | ✓ | ✓ |
| Notifications and Reminders | ✓ | 🗶 | 🗶 | ✓ | ✓ |
| communication with garage | ✓ | 🗶 | 🗶 | ✓ | ✓ |

## 2.4 Algorithm

In the MotorCarCare system, we used simple and clear algorithms to ensure that all actions such as booking, updating appointments, and managing service records are handled correctly and efficiently. These algorithms are not complex but are essential for the system to work smoothly.

* **Appointment Booking Validation:**  
  When a customer chooses a date and time, the system checks if the time slot is still available before confirming the booking. If the time is already taken, the user is asked to choose another slot.
* **Service Status Update Logic:**  
  Technicians and workshop owners can change the status of any ongoing maintenance. Once the status is updated (e.g., “In Progress” or “Completed”), it is saved to the database and shown to the customer immediately.
* **Appointment Cancellation Rule:**  
  The system allows cancellation of appointments only if it’s more than 24 hours before the scheduled time. If it’s less than 24 hours, the cancel option is disabled for the customer.
* **Reminder Notification Scheduling:**  
  Before every appointment, the system automatically sends a reminder notification to the customer using a scheduled background task that checks upcoming bookings and sends alerts.

## 2.5 Tools and Technology

To successfully build our project, we will use a range of technologies that offer essential services to bring our vision to life. These technologies are briefly traversed in the following

### 2.5.1 Flutter

Flutter is an open-source toolkit by Google for developing user interfaces. It is used to create applications and websites that work across different platforms, ensuring they look beautiful and consistent.

Why Flutter: Flutter is a perfect choice because it collects excellent performance, ease of development, and flexibility in scalability, making it suitable for building an app like MotorCarCare that provides a seamless experience for users across platforms.

### 2.5.2 PHP.

PHP is an open-source, general-purpose scripting language that is especially suited to web development. It is fast and flexible. It is used for the backend of a website. Additionally, many hash functions are available in PHP to encrypt users' data, making PHP a secure and reliable server-side scripting language.

### 2.5.3 Firebase

Firebase, a foundation that developed backend software to solve complex app development challenges using Firebase makes development faster and more secure with its out-of-the-box tools, support for instant updates, and ability to scale with the project (firebase)

Why Firebase: It is a great choice for an application like MotorCarCare, which relies on real-time performance and a seamless user experience. Using Firebase Cloud Messaging (FCM), we can send customized notifications to users, such as maintenance reminders, new appointment booking notifications, and service status update notifications. real-time sync support so users receive the latest updates as they happen, such as maintenance status updates or appointments. Sending an Email when resetting a password should be easy to use.

### 2.5.4 HTML and CSS.

HTML is a markup language used to build web applications. CSS is the language used to style an HTML document. (HTML).

Why HTML and CSS: We use HTML to build the basic structure of the application and CSS to format the design, making it attractive and responsive. Together, they provide a strong foundation for developing a web application that delivers an excellent and engaging user experience.

## 2.6 Real-Time Updates and Automation

The MotorCarCare system focuses on improving user experience through real-time synchronization and automated features. By leveraging Firebase, the application ensures that data such as service status updates, appointment confirmations, and notifications are instantly reflected across both web and mobile platforms.

This real-time functionality helps customers stay updated about their car maintenance progress without delays, while workshop owners can efficiently manage appointments and service records. Automated notifications, such as service reminders or booking confirmations, reduce manual tasks and improve overall communication between users and service providers.

# Chapter 3: System Analysis & Design

## 3.1 System Model and System Architecture

**Main work:**

The MotorCarCare system focuses on simplifying the management of car maintenance services for clients, workshop Employees, and administrators. It simplifies Booking, enhances communication, and provides detailed records of past and ongoing services. The primary objective is to provide a Smooth user experience on both web and mobile applications.

**Solution approach:**

The system is designed to handle everyday challenges in car service management, such as:

* Efficient Booking: Allowing users to book, cancel, or reschedule appointments easily.
* Communication**:** Providing notifications and alerts for upcoming bookings or changes.
* Data Management: Storing service history and managing workshop appointment schedules.
* Multi-Platform Accessibility: Ensuring the system is accessible across various platforms, including mobile devices and web browsers.

**Theory:**

The system works by:

* Making sure that the website or App communicates quickly and reliably with the database that stores all the information.
* Keeping all the data in one place (Firebase) makes it easy to update and access instantly.

**Simulation software:**

We used simulation to test the MotorCarCare app before launching it, ensuring everything worked properly. We tested the appointment booking system to see how the App behaved when different people booked. I also tested the notifications to make sure they were delivered to users on time. We used tools like Firebase Emulator to test storage and updates without affecting real data, and we ensured the UI was easy and convenient by simulating the user experience. We had different people test it to make sure it was understandable and easy to use.

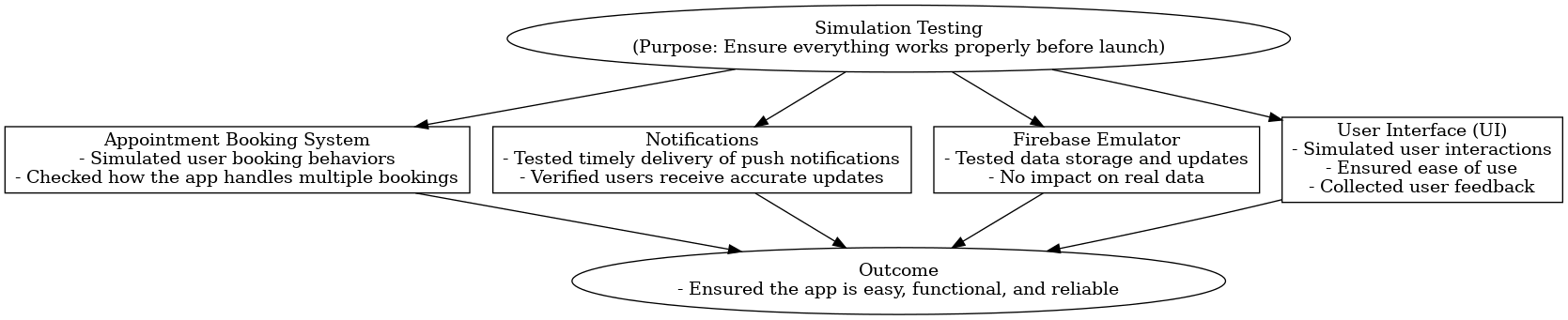


Figure 8 simulation Software

## 

## 3.2 Product Description

### 3.2.1 System Objectives

* The system streamlines the booking process for clients, ensuring quick and easy confirmation of their appointments.
* It provides detailed information about available services, including descriptions and prices, enabling clients to compare options and choose the best fit for their needs.
* It helps workshop owners effectively showcase, organize, and manage their services and bookings.
* It creates a user-friendly platform (both website and mobile application) to attract attention and retain clients by providing an excellent user experience.

### 3.2.2 System Main Features

The website provides the following main features:

* **Home Page**: Offers an overview of services, promotions, and news.
* **User Authentication**: Includes login, logout, and sign-up functionalities.
* **User Profiles**: Allows users to manage their personal information.
* **Contact and About Us Pages**: Provide information on how to contact support and company details.
* **Service Search and Filters**: Offers search and filtering options to find specific services and workshops.
* **Service Booking**: Allows users to book appointments for services.
* **Service Details and Pricing**: Displays detailed descriptions of available services along with their prices.
* **Notifications and Alerts**: Sends system-generated notifications about appointment status and updates.

**Owner** **garage main feature:**

* **Appointment Management**: View, confirm, or reject client bookings.
* **Service History Review**: Access past maintenance and repair details for each customer.
* **Service Scheduling**: Organize and manage the workshop's schedule efficiently to ensure optimal workflow.

### 3.2.3 Operating Environments

The MotorCarCare Application is designed to be responsive, allowing users (Clients and workshop Owners) to access it through smartphones, computers, and laptops via internet browsers. We used PHP for server-side logic, HTML, CSS, and Bootstrap for the web design, and Firebase with JavaScript for the backend. For the mobile application, Flutter and Dart were used for the design, with Firebase to store data. The system enables users to manage appointments, track repairs, and communicate effectively, regardless of their device type. This project helps minimize waiting times, improve communication, and make managing car services more efficient.

## 3.2.4 Constraints

* An internet connection is necessary to use the MotorCarCare website and mobile application.
* An email account and a phone number are required to register on the platform.
* Clients will not be able to cancel their Booking if it has been 24 hours since the appointment was scheduled.
* The workshop will be able to cancel Bookings if the selected time slot is not available, ensuring better scheduling management.

### 3.2.5 Functional Requirements

**UR1**: Customers can select the right dates and times for their maintenance appointments through the App or website.

* **SR1.1**: The system displays available dates and times for booking.
* **SR1.2**: The customer selects a suitable appointment and confirms it.

**UR2**: The system displays detailed information about upcoming appointments.

* **SR2.1**: The system retrieves and displays appointment details, including date, time, and service type.
* **SR2.2**: The user can edit or cancel the appointment (under allowed conditions).
* **SR2.3**: After deletion, the appointment is removed from both the website and app.

**UR3**: Customers can access a history of all past services and repairs performed on their vehicle.

* **SR3.1**: The system enables users to view past services and repairs.
* **SR3.2**: The system stores detailed records of each service, including date, type, and cost.
* **SR3.3**: The system creates a new profile for first-time clients.
* **SR3.4**: Detect if a user is new based on email or phone number.

**UR4**: Technicians can view a list of their assigned appointments for maintenance or repair tasks.

* **SR4.1**: The system displays all assigned appointments.
* **SR4.2**: Technicians can update the list in real-time.
* **SR4.3**: Technicians can delete or edit appointments.

**UR5**: Technicians update the status of repairs and notify customers of progress.

* **SR5.1**: The system allows status updates via a dedicated interface.
* **SR5.2**: Technicians can edit or reject a status.
* **SR5.3**: The client can view the final status.

**UR6**: The system confirms the appointment booking.

* **SR6.1**: The system includes appointment details in the confirmation message.

**UR7**: The system allows users to search for specific maintenance services, available technicians, or service records.

* **SR7.1**: The system implements a search functionality for services and bookings.

**UR8**: The system sends automated reminders to customers before their scheduled appointments.

* **SR8.1**: The system schedules and sends reminders.
* **SR8.2**: The user can choose to enable or disable reminders.

**UR9**: The system provides a workshop summary of services scheduled for a specific day, including total income collected.

* **SR9.1**: The system shows services performed per day and their payment details.

**UR10**: The workshop can manage services by adding, updating, or deleting them.

**UR11**: Customers can provide feedback on their maintenance experience.

* **SR11.1**: The system enables customers to write feedback.
* **SR11.2**: Feedback is displayed with the customer’s username.

## 3.2.6 Non-Functional Requirements

**1. Time:**  
1.1 The system ensures fast response times to maintain efficiency and provide a smooth user experience.

**2. Usability:**  
2.1 The client can easily understand and use the system’s features.  
2.2 Training time for workshop owners is estimated to be less than one hour.  
2.3 Customers can complete a service booking within 5–10 minutes.

**3. Reliability:**  
3.1 The booking system is stable and reliable, ensuring continuous availability so that users can book appointments at any time.

**4. Security:**The system protects user information by using secure login and encrypted data storage. For example, passwords are stored in a secure way so that no one can read them. We also use Firebase Authentication, which provides a trusted and safe login process. Only authorized users can access or change data. This helps keep personal information, appointment details, and payment records private and secure.

* **4.1** The system uses Firebase Authentication to allow only authorized users to log in.
* **4.2** User data (like passwords and emails) is stored in an encrypted format.
* **4.3** Only admins or owners can edit or delete sensitive information.
* **4.4** The system checks if an email is already used before allowing registration. This prevents fake accounts or duplicate users from using the same email.

**5. Portability:**  
5.1 The website is compatible with multiple platforms, including iOS, Android, Windows, and can be accessed from phones, tablets, and computers using major web browsers like Chrome, Firefox, and Safari.

3.3 Functional Decomposition (Use Case Diagram)

**Principle Actors:**

|  |  |
| --- | --- |
| Actors | Description |
| |  | | --- | | **Workshop Owner** |  |  | | --- | |  | | |  | | --- | | Manages the workshop profile, confirms/rejects bookings, updates service status, and responds to client inquiries. |  |  | | --- | |  | |
| Customer | People like you and me use the system to book car services, pay bills, and ask questions about our vehicles. |
| Technician | |  | | --- | | Handles maintenance tasks, updates repair status, and manages parts usage. |  |  | | --- | |  | |

Table 2 Principal Actors and Their Roles

**Secondary Actors:**

|  |  |
| --- | --- |
| Actors | Description |
| System | Sends automated notifications, stores and retrieves data from Firebase, handles user authentication. |

Table 3Secondary Actors

### 3.3.1 Actors (actor list and description of their roles)

**Car Owner**

* + Registers a personal account.
  + Books a service appointment.
  + Views service history and current status.
  + Receives notifications.
  + Reviews and pays for completed services.

**Workshop**

* Manages workshop profile and services.
* Handles appointment scheduling and confirmations.
* Updates service status and progress.
* Manages services parts.

### 3.3.2 Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and name | ID#1.  Registration | | |
| Created By: | Alaa DarYauop | Date Created: | 2 June 2025 |
| Primary Actors:  (external entity  to the system) | Customer |  |  |
| Trigger: | Customer initiates the registration process. | | |
| Description: | This use case describes the process of registering a new customer in the Car Service Center Management System. | | |
| Pre-conditions: | The customer must have access to the registration functionality. | | |
| Postconditions: | The customer's information is successfully stored in the system, and they receive confirmation of registration. | | |
| Normal Flow: | **1**. Customer accesses the registration page: The customer goes to the registration section on the website or App.  2. System presents the registration form: The system displays a form for the customer to enter their personal information, including name, email address, phone number, address, username, password, and vehicle details.  3. Customer fills in their information: The customer enters their details into the form.  4. Customer submits the form: The customer clicks 'submit' to send their info.  5. System verifies and creates the account  - System checks:  - Email format and send a verification email.  - Enter your phone number and receive a verification code.  - Checks username availability.  - Ensures password meets security requirements.  - Verifies vehicle information.  - The system sends a confirmation message to the customer. | | |
| Alternative flow: | If the customer's information is incomplete or invalid, the system prompts the customer to correct the errors and resubmit the form. | | |

Table 4 Use Case Registration

## Use case Booking Car Service.

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and name | ID#2.  Booking Car Service | | |
| Created By: | Alaa DarYauop | Date Created: | 6 June 2025 |
| Primary Actors: | Customer | Secondary Actors: | Workshop Owner |
| Trigger: | The customer wishes to schedule a service for their vehicle. | | |
| Description: | This use case describes how a customer can book a service for their car through the Car Service Center Management System. | | |
| Pre-conditions: | 1- **Customer Registration and Login:**   * The customer must be registered in the Car Service Center Management System. * The customer must be logged into their account.   2- **Access to Vehicle Management Section:**   * The system must provide a section or functionality designed explicitly for managing vehicles.   3-**Authorization to Add a Car:**   * The customer must have the necessary permissions or authorization to add a new car to their profile.   4- **Complete Car Details:**   * The customer must have all relevant information about the new car ready to enter into the system, including:   + Make   + Model   + Year   + VIN (Vehicle Identification Number) or equivalent identification details   + Any other required information as specified by the system.   5- **Stable System Availability:**   * The system must be stable and available for the customer to access and complete the process without interruptions or technical issues. | | |
| Postconditions: | The customer successfully books a service appointment for their car. | | |
| Normal Flow: | * The customer logs into the system. * The customer navigates to the booking section. * The system prompts the customer to select the car for which they want to book the service. * The customer selects the desired car. * The system displays available service options (e.g., oil change, tire rotation). * The customer selects the type of service they require. * The system prompts the customer to choose a preferred date and time for the service appointment. * The customer selects a suitable date and time from the available options by timetable. * The system asks the customer to select an alternative date or a different workshop location. * The workshop owner receives notification of the Booking. | | |
| Alternative flow: | If the customer's preferred time slot is unavailable, the system prompts the customer to choose an alternative time slot. If the workshop is fully booked, the system informs the customer and suggests alternative workshop locations or dates. | | |

Table 5 Use case Booking Car Service

## Use case Manage Repair Car

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and name | ID#3.  Manage Repair Car | | |
| Created By: | Alaa DarYauop | Date Created: | 6 Jun 2025 |
| Primary Actors: | Mechanic | Secondary Actors: | Customer |
| Trigger: | Mechanic receives a car for repair. | | |
| Description: | This use case describes how a mechanic manages the repair process for a car in the Car Service Center Management System. | | |
| Pre-conditions: | * The mechanic must be logged into the system. * A car must be assigned to the mechanic for repair. * The customer's car details and repair requirements must be accessible in the system. | | |
| Postconditions: | * The car repair process has been successfully completed, and the customer has been notified. | | |
| Normal Flow: | 1. The mechanic logs into the system.  2. The system assigns a car for repair to the mechanic. This assignment can be done automatically using predefined algorithms that prioritize workload distribution or manually by a supervisor. If assigned manually, the supervisor will be the actor.  3. The mechanic reviews the repair requirements and car details in the system. These details are initially inspected and added by an authorized supervisor or inspector.  4. The mechanic begins the repair process, following the specified repair instructions.  5. As work progresses, the mechanic updates the repair status in the system.  6. If the mechanic encounters difficulties during the repair process, they can consult with other mechanics or escalate the issue to a supervisor for assistance.  7. If additional issues are discovered during the repair, the mechanic will add them to the repair record and inform the customer.  8. If the required parts are unavailable, the mechanic can request them through the system and proceed with the repair once the parts are obtained.  9. Once the repair is completed, the mechanic updates the repair status to indicate completion.  10. The system automatically notifies the customer that their car repair is finished. | | |
| Alternative flow: | - If the mechanic encounters difficulties during the repair process, they can consult with other mechanics or escalate the issue to a supervisor for assistance.  - If the required parts are unavailable, the mechanic can request them through the system and proceed with the repair once the parts are obtained.- If additional issues are discovered during the repair, the mechanic records them in the repair document and informs the customer. | | |

Table 6 Use case Manage Repair Car

## Use case Manage Insurance

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and name | ID#4.  Manage Insurance | | |
| Created By: | Alaa DarYauop | Date Created: | 10 June 2025 |
| Primary Actors: | Admin | Secondary Actors: | Customer |
| Trigger: | Super Admin initiates the management of insurance policies. | | |
| Description: | This use case describes how the admin manages insurance policies for cars in the Car Service Center Management System. | | |
| Pre-conditions: | The admin must be logged in to the system.  - Insurance policy details must be accessible in the system.  - Cars must be registered in the system with associated insurance policies. | | |
| Postconditions: | The insurance policies are successfully managed and updated in the system. | | |
| Normal Flow: | 1. The admin logs into the system.  2. The admin navigates to the insurance management section.  3. The admin selects the option to add a new insurance policy.  4. The admin enters the details of the new insurance policy.  5. The admin reviews and confirms the addition of the new policy.  6. The system updates with the new insurance policy information.  7. The system notifies relevant stakeholders (e.g., customers) about the new insurance policy. | | |
| Alternative flow: | - If there are any discrepancies or issues with insurance policy information, the admin can investigate further, contact insurance providers, or consult with other authorized personnel to resolve the issues.  - If new insurance policies need to be added but are not available in the system, the Super Admin can initiate the process to obtain and add them to the system. | | |

Table 7 Use case Manage Insurance.

## 3.3.3 Use Cases Diagram

The diagram illustrates how different users interact with the car service application.

* **Customers** can log in, book appointments for car services, track their service history, and add new cars to their profiles. They also receive notifications about their bookings or service updates.
* **Workshop managers** can view customer details and manage appointments to ensure smooth workflow and efficient service handling.
* **The system administrator** maintains the system's functionality, ensuring all components work correctly and data is managed securely.

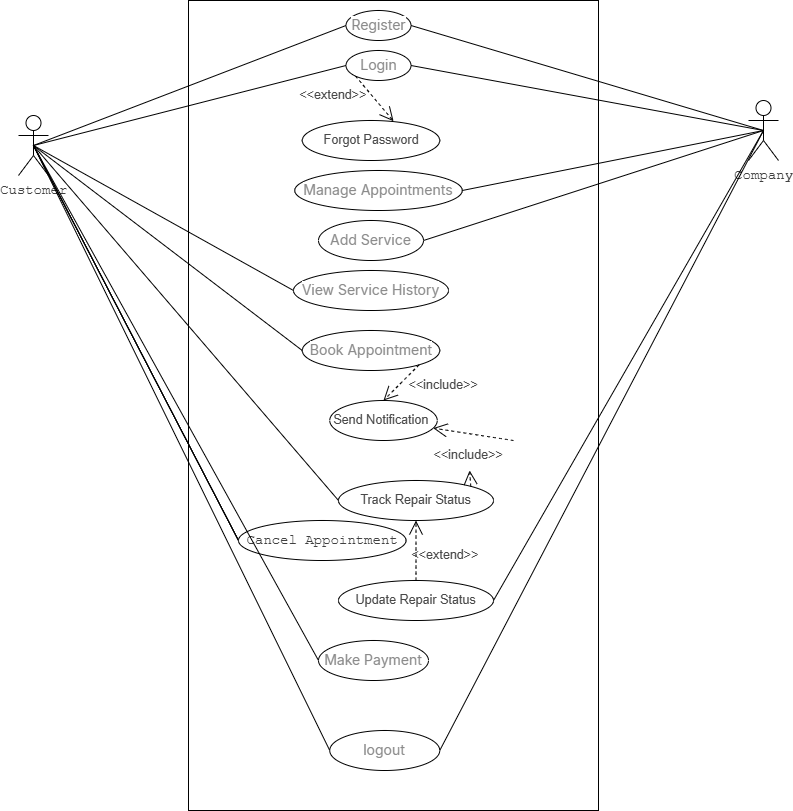


Figure 9 Use Cases Diagram

## 3.4 System Models

### **3.4.1 ClassDiagram**

A class diagram describes the structure of a system by showing the system's classes, their attributes, and the relationships among objects, as well as how they interact with each other.

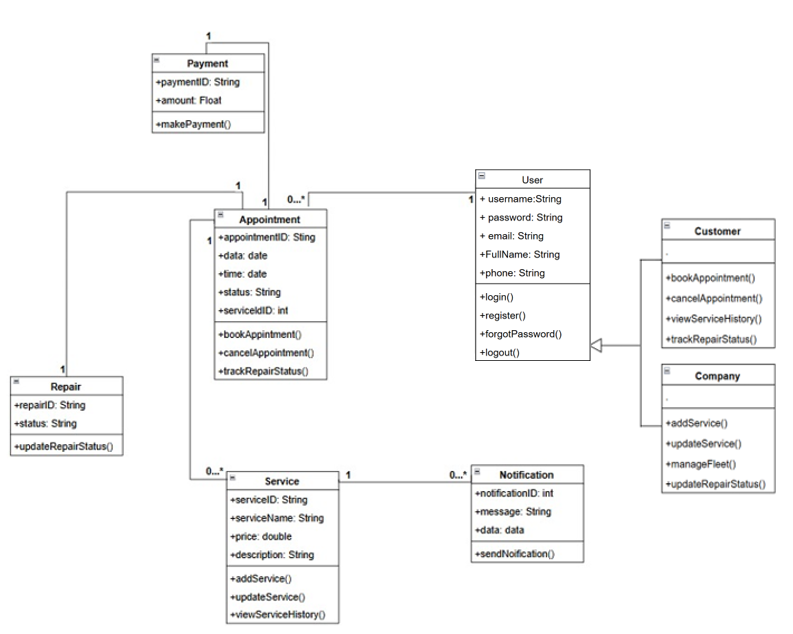


Figure 10 Class Diagram

### 3.4.2 Sequence Diagram

A sequence diagram that represents interactions among objects or components in a system and the order in which these interactions occur. It is used to model the flow of messages, events, and actions between objects in a system.

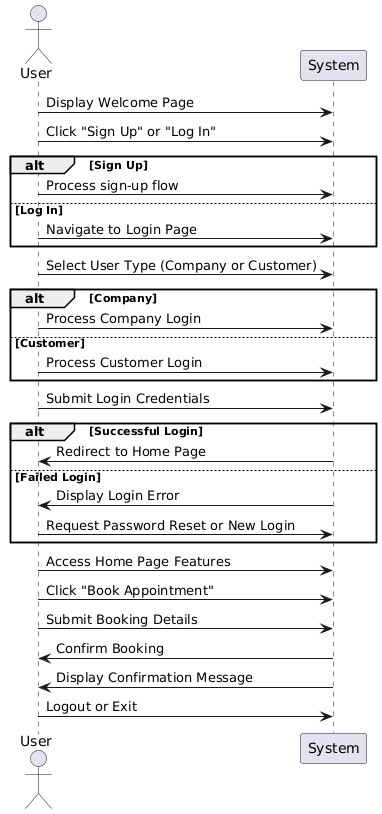


Figure 11 Sequence Diagram

## 3.4.3 Activity Diagram

An activity diagram is a type of UML (Unified Modeling Language) diagram that used to model the workflow of a system or a process and to represent the control flow from one activity to another

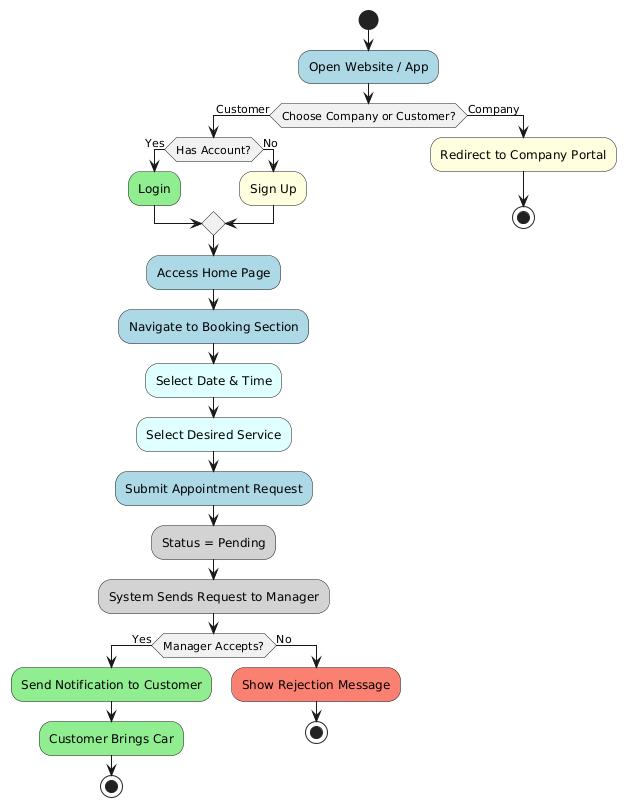


Figure 12 Activity Diagram

## 3.4.4 State Chart Diagram

A state chart diagram describes the behavior of an object as a series of states and describes the transitions among these states.

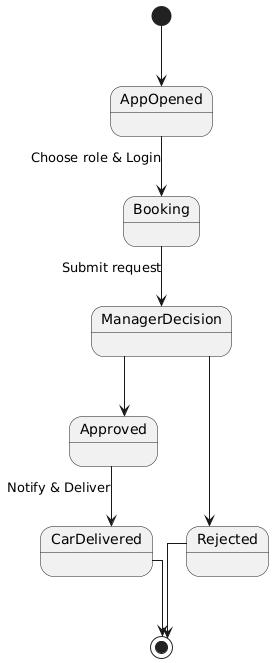


Figure13 State Chart Diagram

## 3.5 System Architecture

### 3.5.1 Sub-System (descriptions of the sub-systems and their services)

#### **External Access:**

This sub-system represents the external devices that users use to access the application.

#### **Components:**

* **Phone:** Mobile devices for accessing services via apps or the web.
* **PC End:** Desktop or laptop computers for accessing services through a web interface.

#### **Application/PC Access:**

This sub-system handles the user-facing functionalities and services of the application.

**Services:**

* **Login:** Enables users to authenticate and securely access the system.
* **Sign-up:** Allows new users to register their accounts.
* **View Service History:** Displays a record of the user’s previous bookings and maintenance activities.
* **Services:** Provides access to core features offered by the platform.
* **Book Appointment:** Allows users to schedule appointments with workshops.

#### **Data Service:**

This sub-system manages data storage, retrieval, and processing for the application.

**Components:**

* **Centralized Data Storage:** Represents the database responsible for storing all application data.
* **Automation / Backend Logic:** Handles business logic, request processing, and background operations.

#### **Flow Description:**

Users access the application through the internet using mobile or desktop devices. Their requests are directed to the **Application/PC Access** sub-system, where appropriate services are executed. Any data-related operations are then forwarded to the **Data Service** sub-system, which ensures reliable data handling and system performance.

### 3.5.2SoftwareArchitecture

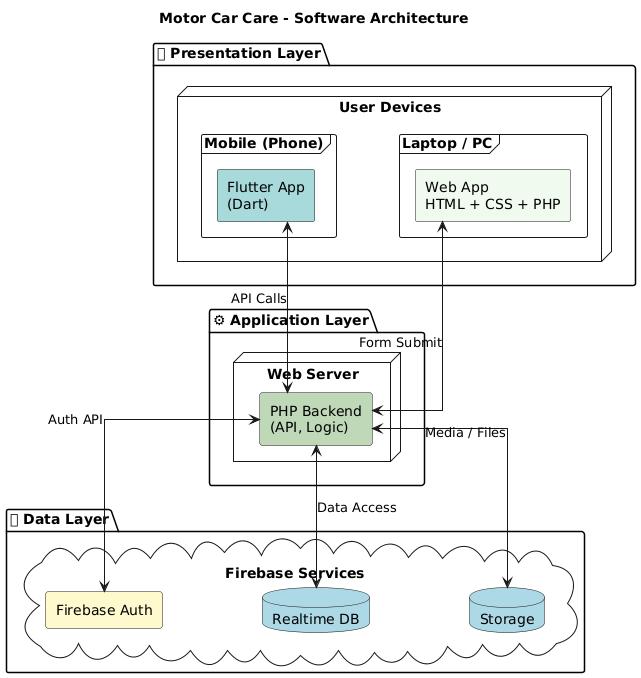


Figure 14 Software Architecture

### 3.5.3 DeploymentDiagram

Deployment diagrams are used to model the physical components of a system

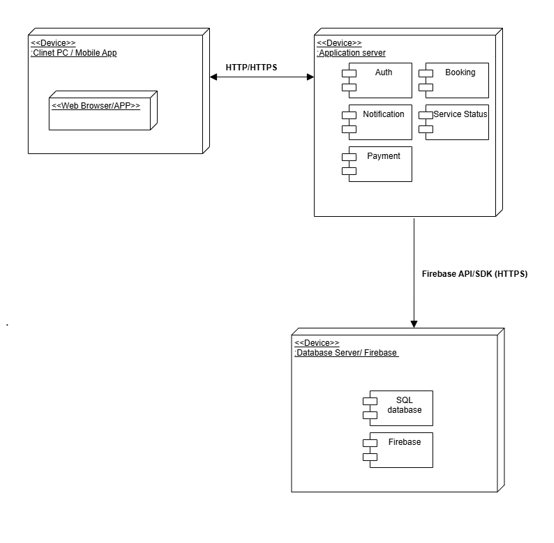


Figure 15 Deployment Diagram

## 3.6 Data Management and Data Models

Entity Relationship (ER) Diagram is a type of flowchart that shows how "entities" in a system, such as people, objects, or concepts, interact with one another.

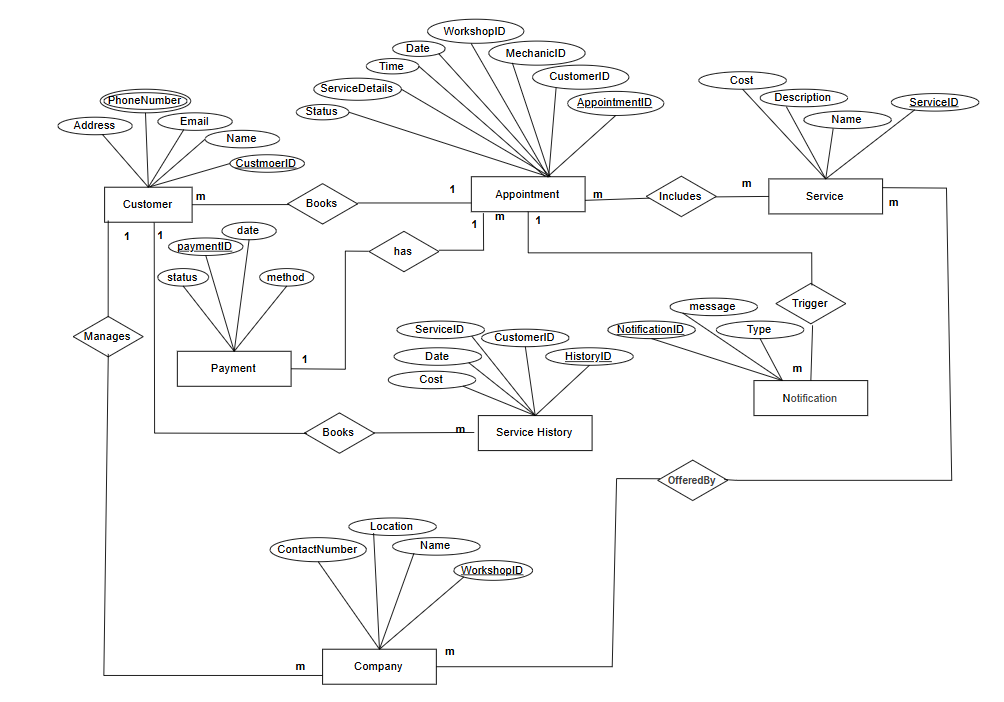


Figure 16 Data Management and Data Models

# Chapter 4: Implementation

## 4.1 Chapter overview

The previous chapter presented the software analysis of the proposed system. This chapter describes how our system works and how each user and workshop interacts with the system and uses it.

## 4.2 Screen shots of your actual application

**Welcome page:**

This screen aims to let the user choose whether they are a customer or a company to register to the system

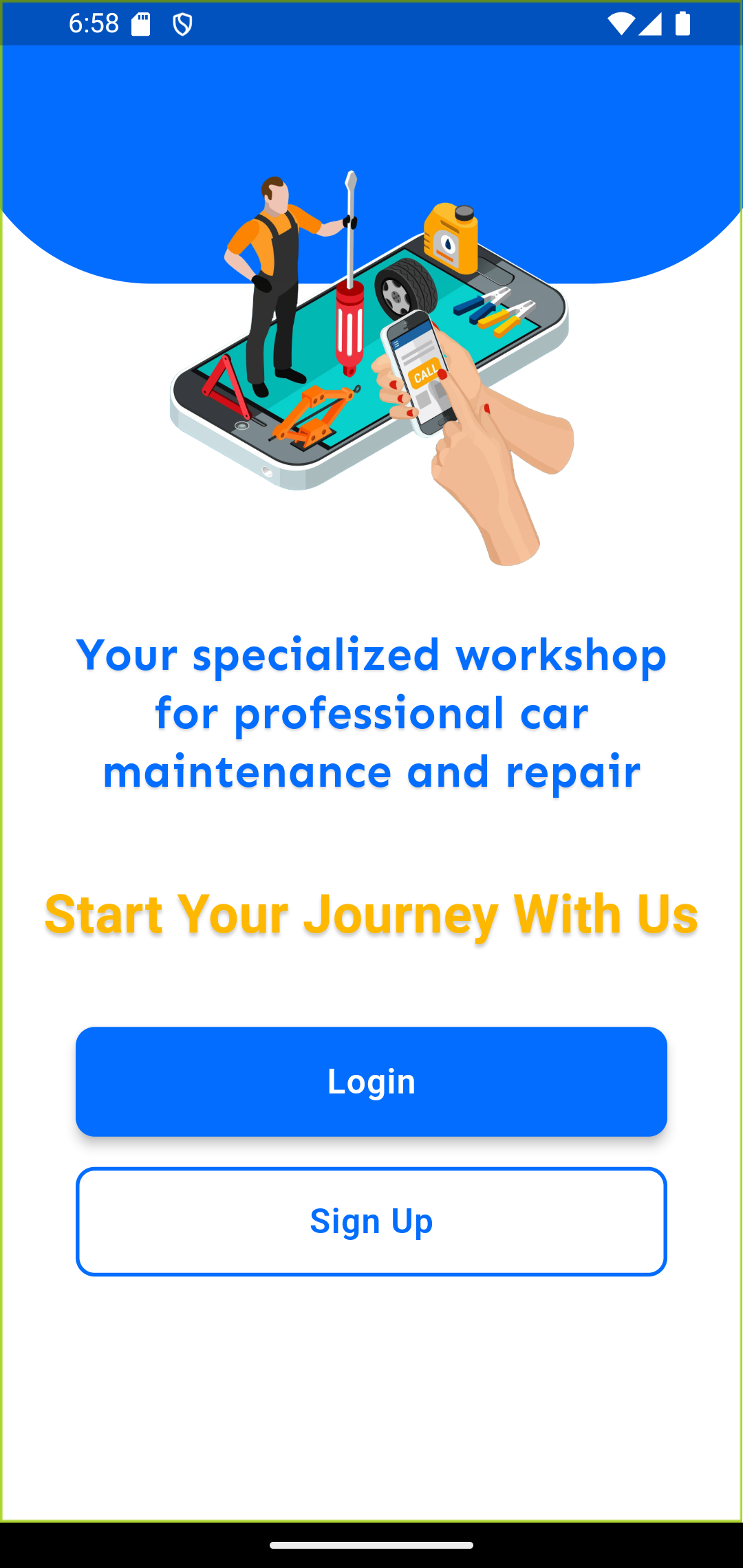
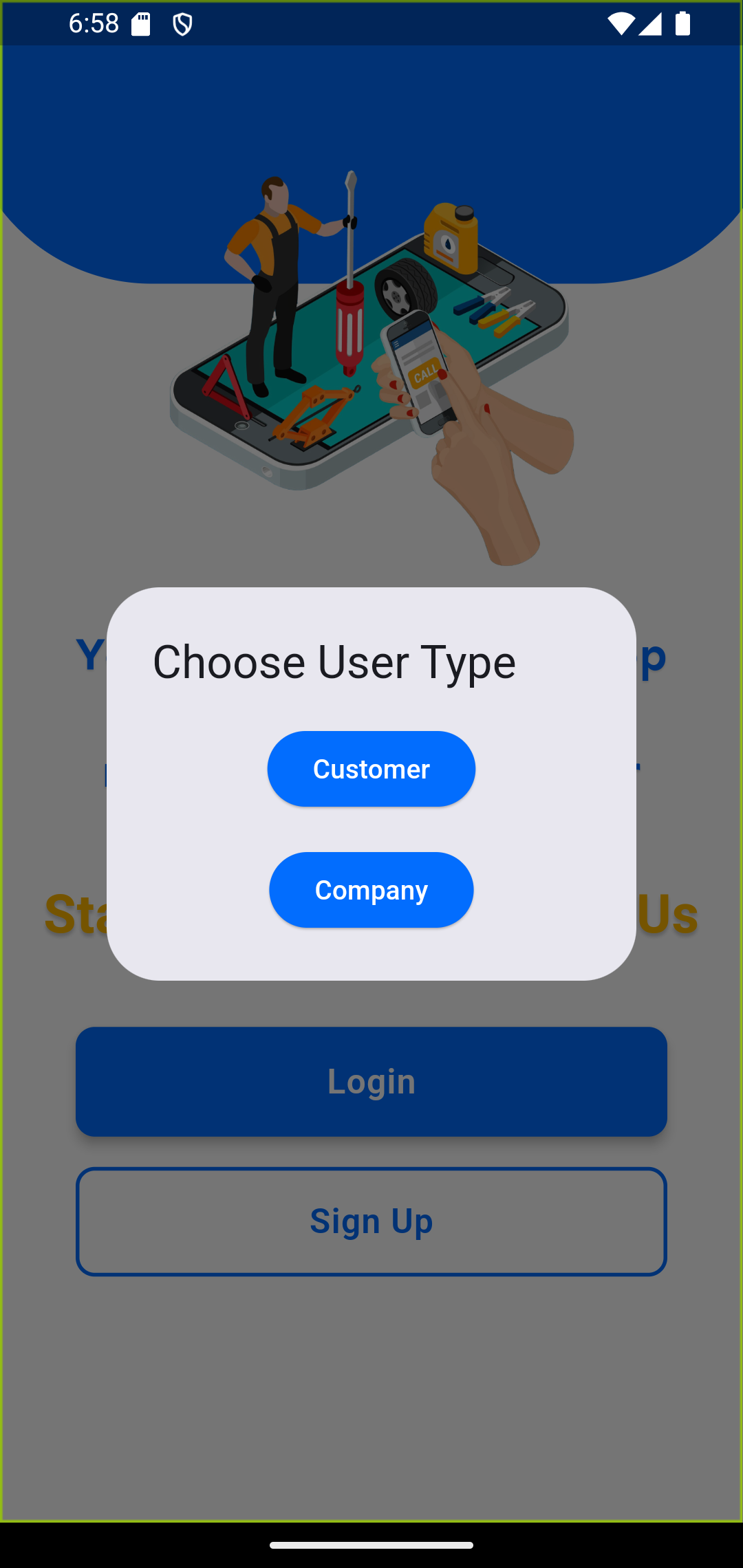


Figure18 user type

Figure17 welcompage

**Register page for customer:** These screens allow the user to signup for App if they don't have an account and provide information like (Username, Full Name, Email, Phone, Password, and confirm password). After creating the account, the user can access to the App.

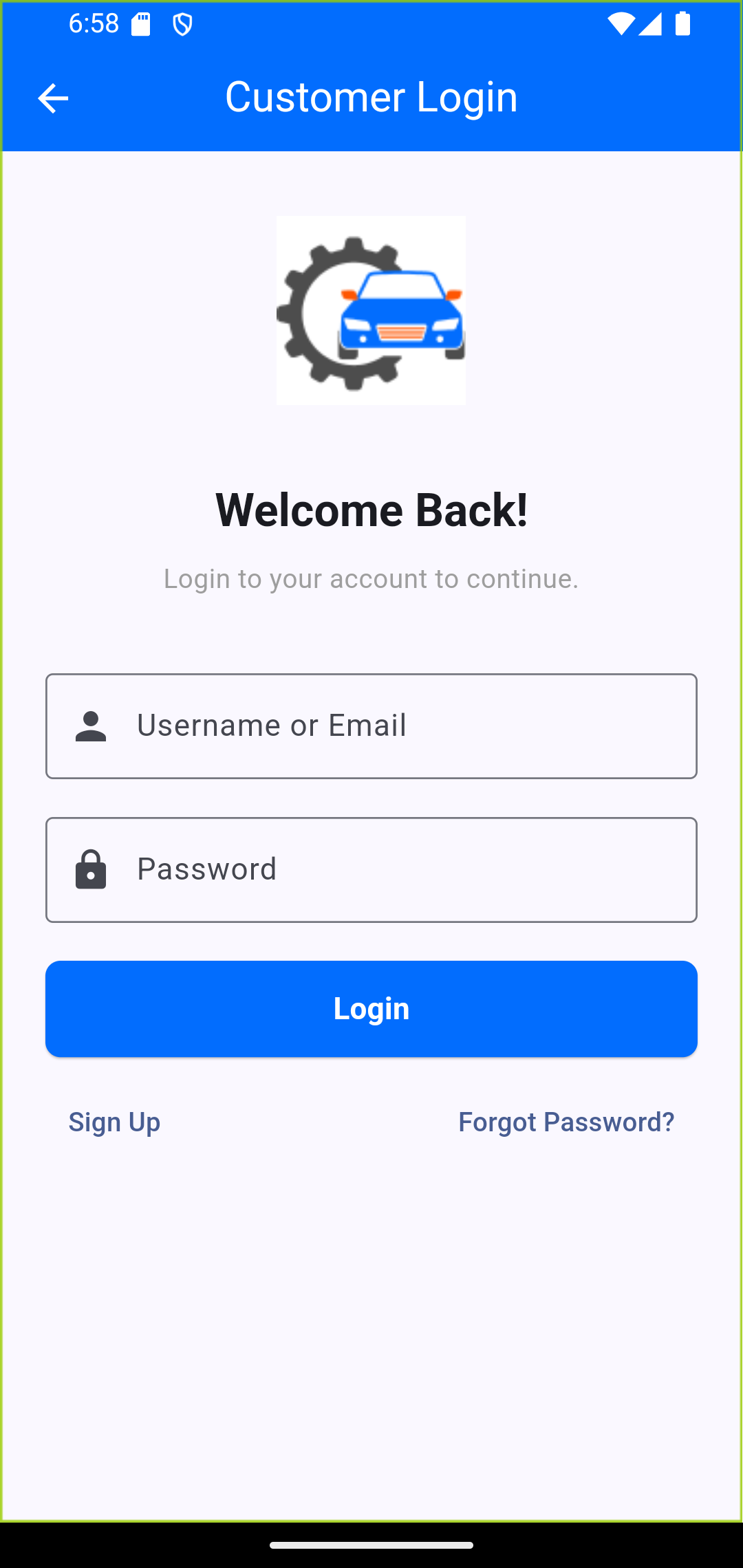
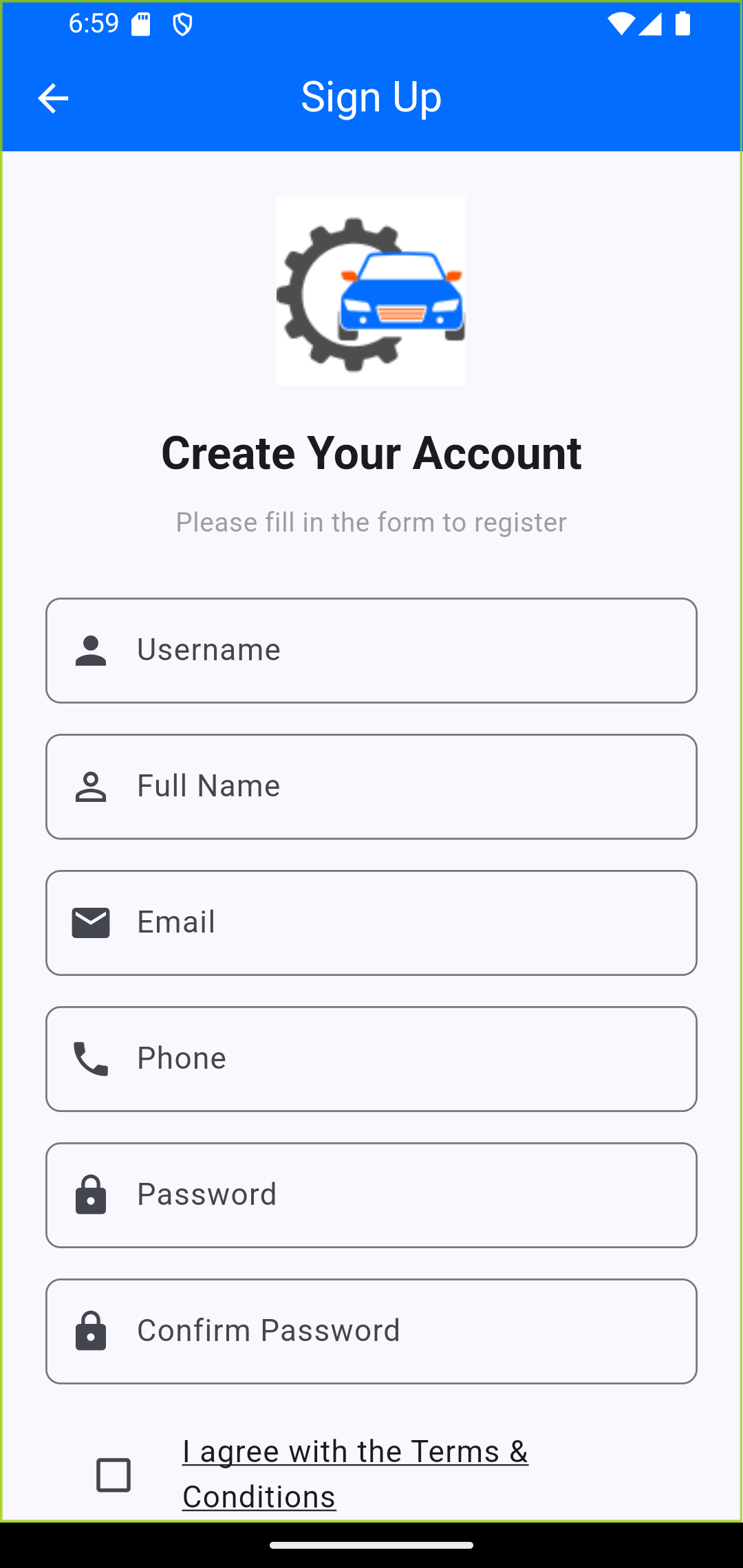


Figure19 signup

Figure20 login customer

**Booking Appointment:** This page allows the customer to make an appointment by adding information about his car, such as (Name, Phone, Card Number, Choose Date, Choose Time). When making a booking, the details are displayed on the My Appointment screen.

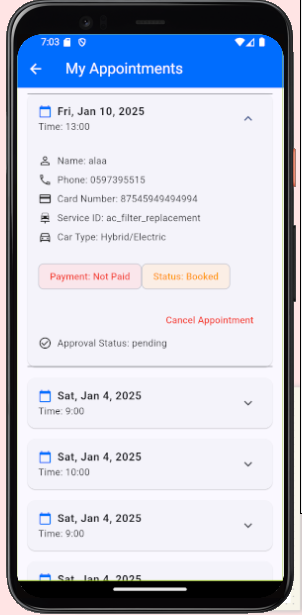
****

Figure21 appointment

Figure 22 My Appointment

**Viewing Service History:** These screens allow the user to access their service history and know the status of their car. For example, when the color of the icon changes from orange to blue, the status means it is in progress. When the icon color changes to green, the service is completed.

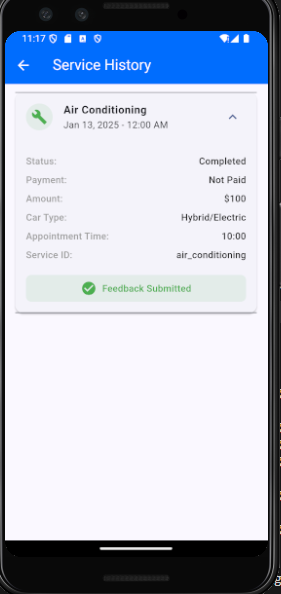
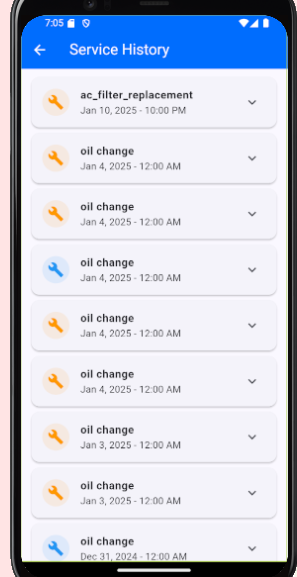


Figure25 status of appointment



Figure 23 services history

Figure 24 information of services history

**Sending Notification:** These screens allow the user to send a notification when the workshop updates status, rejects their appointment, or when their appointment is scheduled for tomorrow.

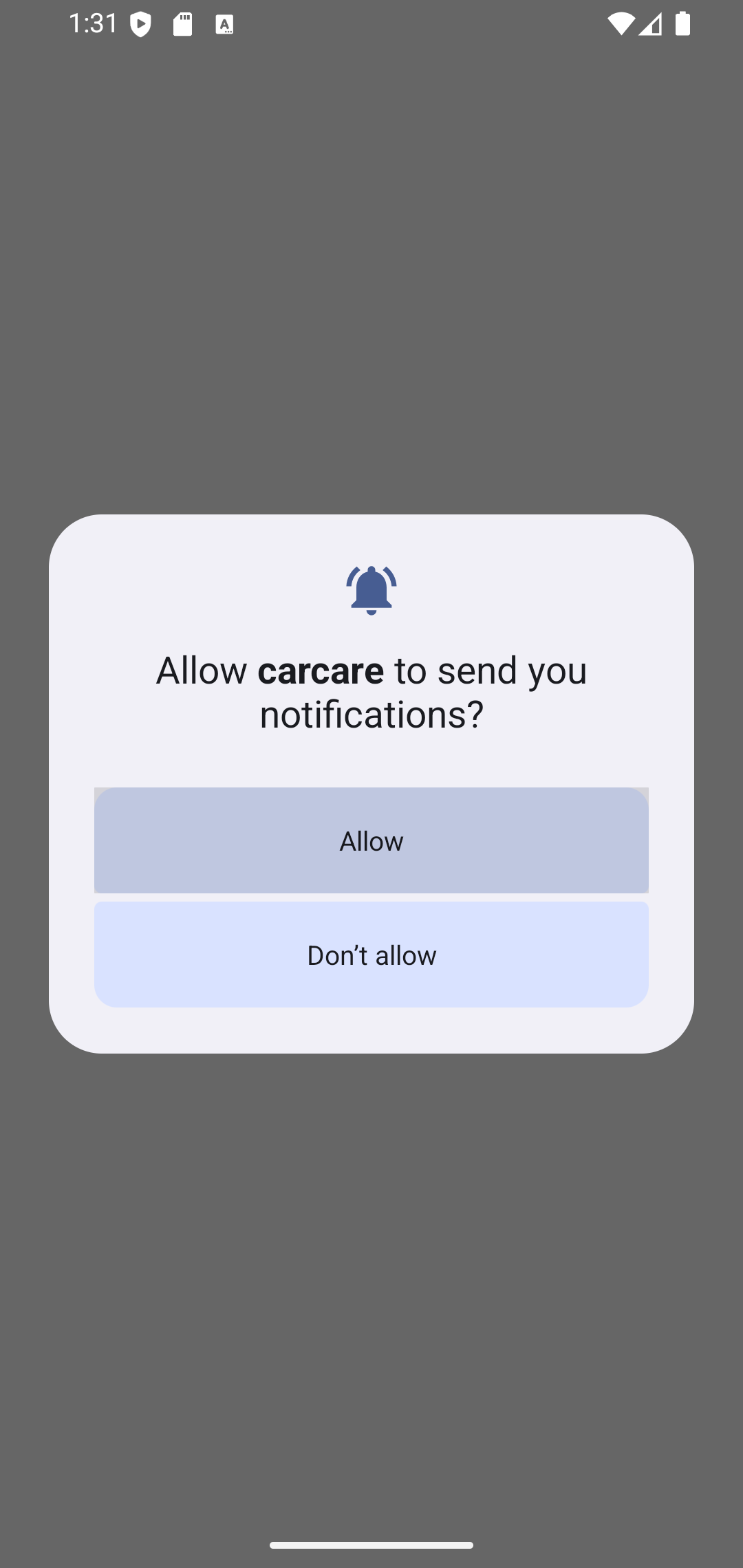


Figure26 notification

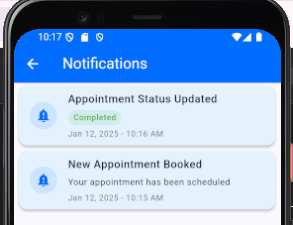


Figure27 completed

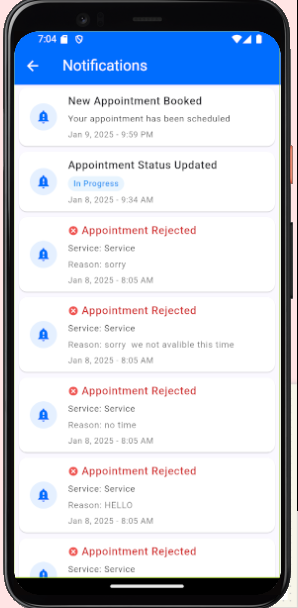


Figure28 rejection

**Payment system**: These screens allow the workshop to know the total of payment collected on a

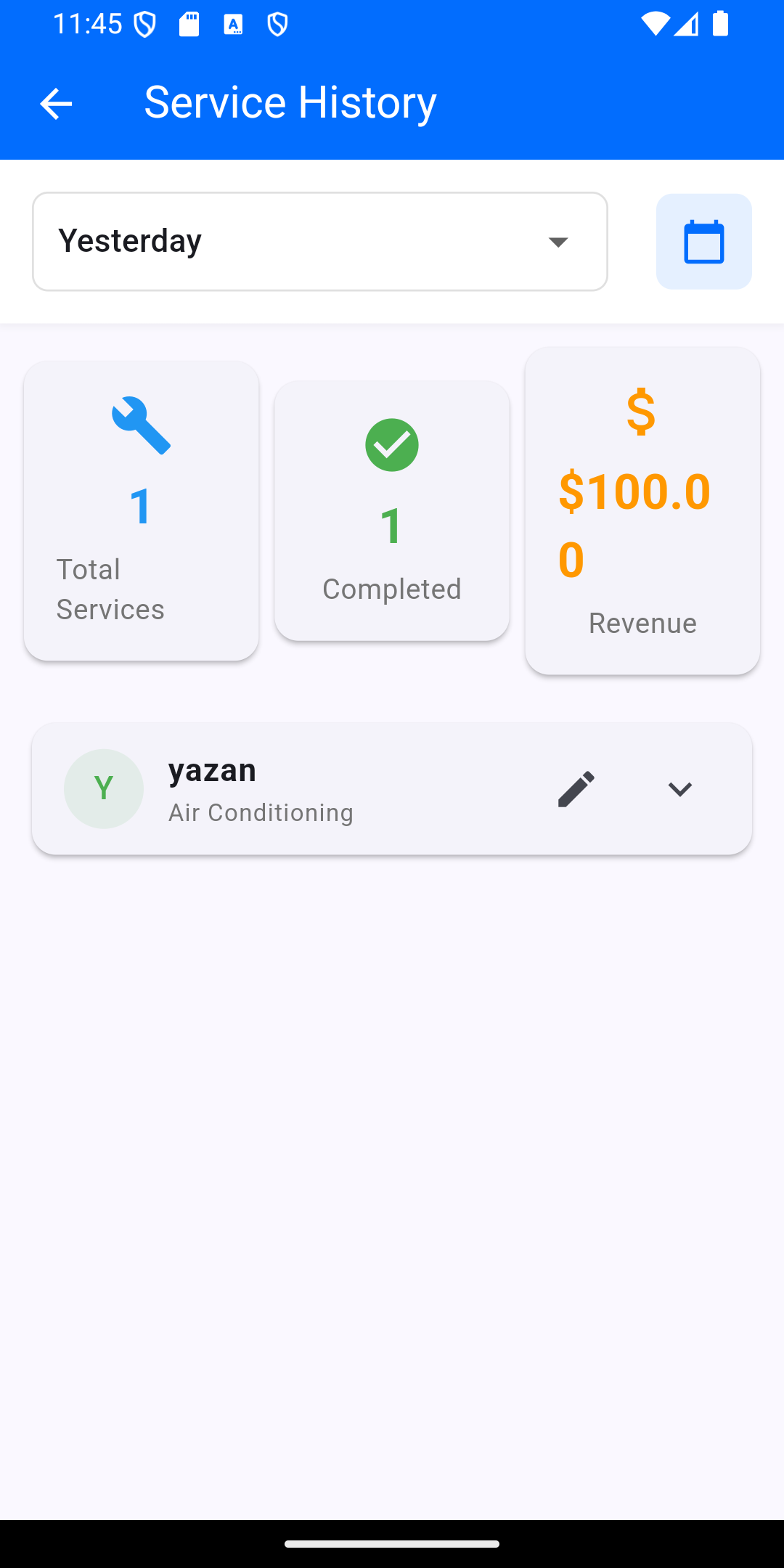
chosen day, week, or month

Figure29 payment stamen

**Forget Password:** This screen allows the user to reset the password when they forget and send the link for the actual email who logs in with it

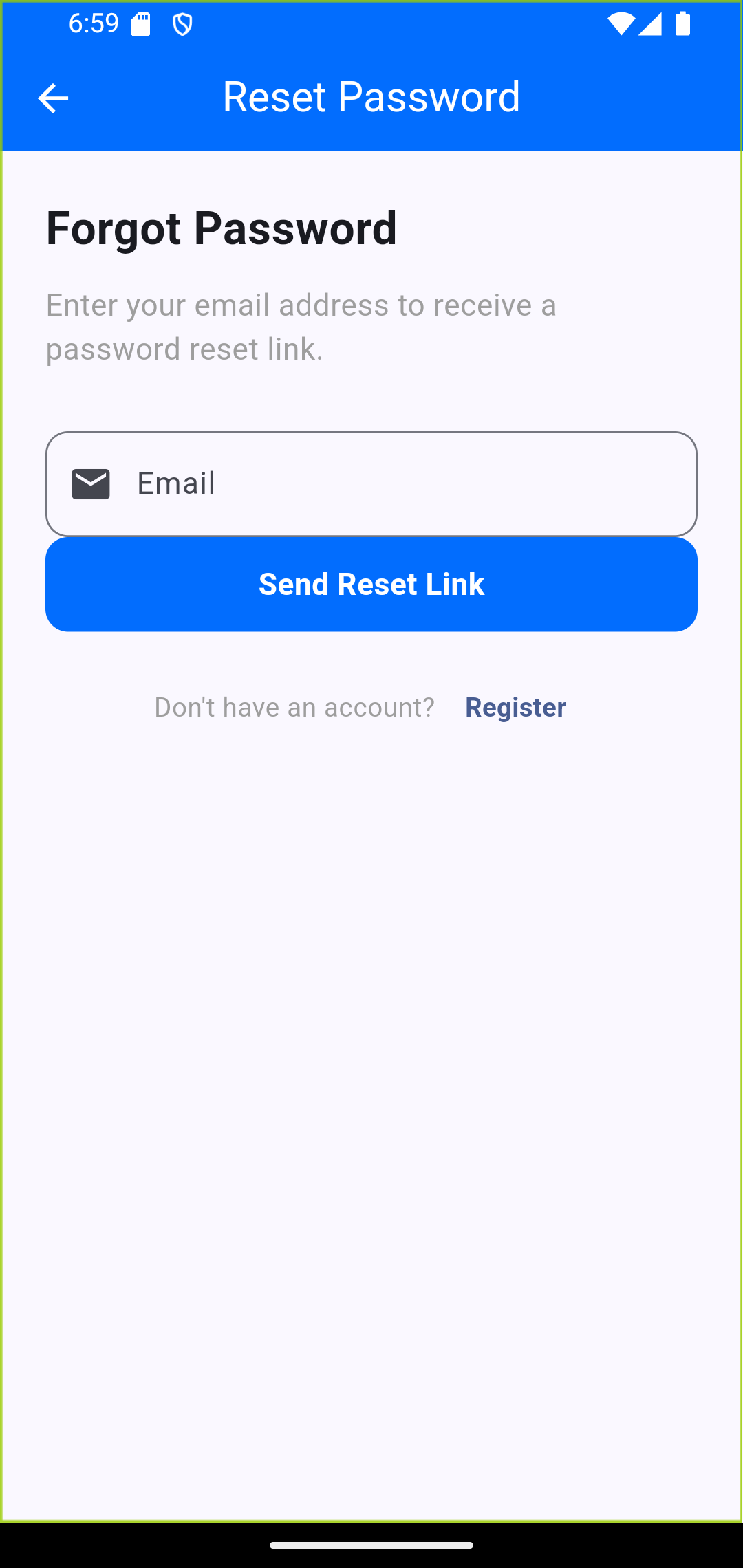
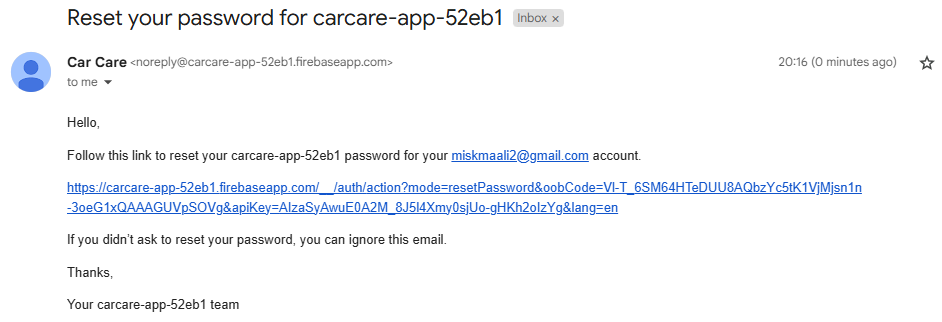


Figure 30 forget password

Figure 31 email to reset password

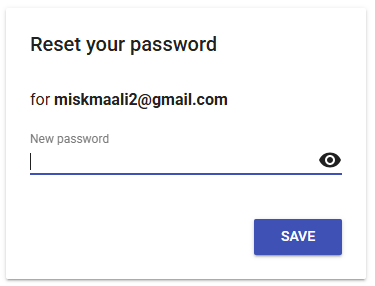


Figure 32 new password

## 4.3 Website screens:

**"MotorCarCare" homepage:** The "MotorCarCare" homepage guides users through a simple process: book a car service online, drop off the vehicle, and let the team handle the rest while you enjoy your day. It also features a chat option for quick assistance.

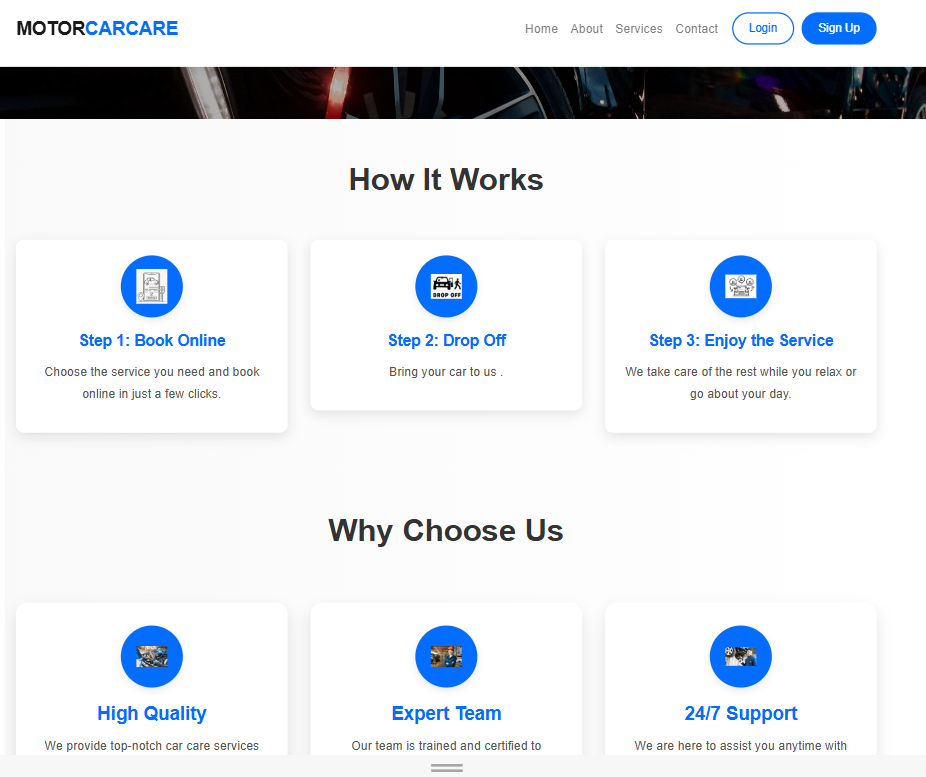


Figure 33 car care

This is the homepage of Motor Car Care. It asks users to choose between 'Customer' or 'Company' to access services. Customers can book and track car maintenance while companies manage appointments. The page also offers links to Home, About, Services, Contact, and Login/Sign Up, as well as Customer Support for assistance.

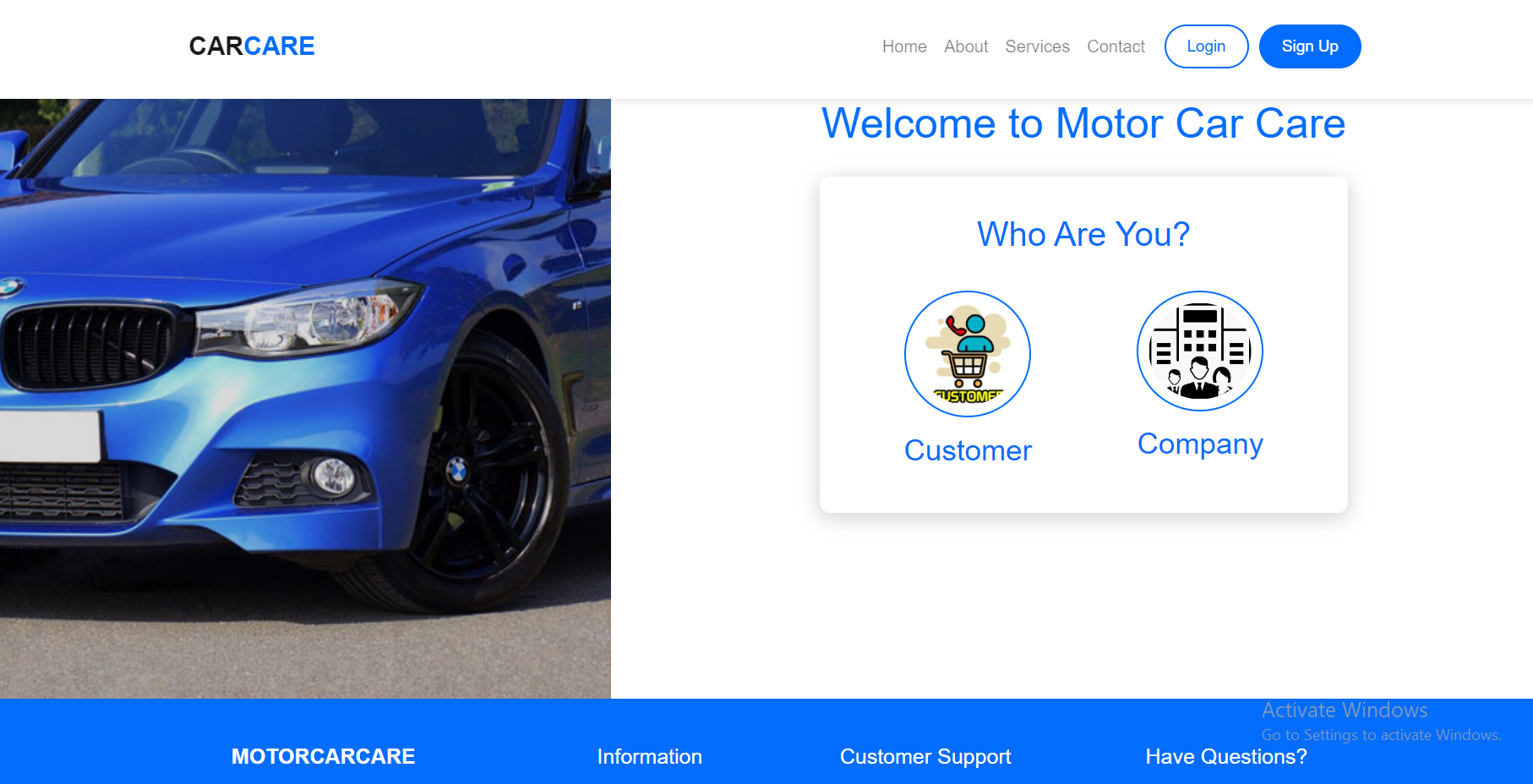


Figure 34 company or register

**Sign Up:** This is the sign-up page of the Motor Car Care web application. It allows new users to create an account by entering their Username, Email, Phone Number (in the format: 050xxxxxxx), and password (minimum 6 characters), with a confirmation step. Existing users can Sign In

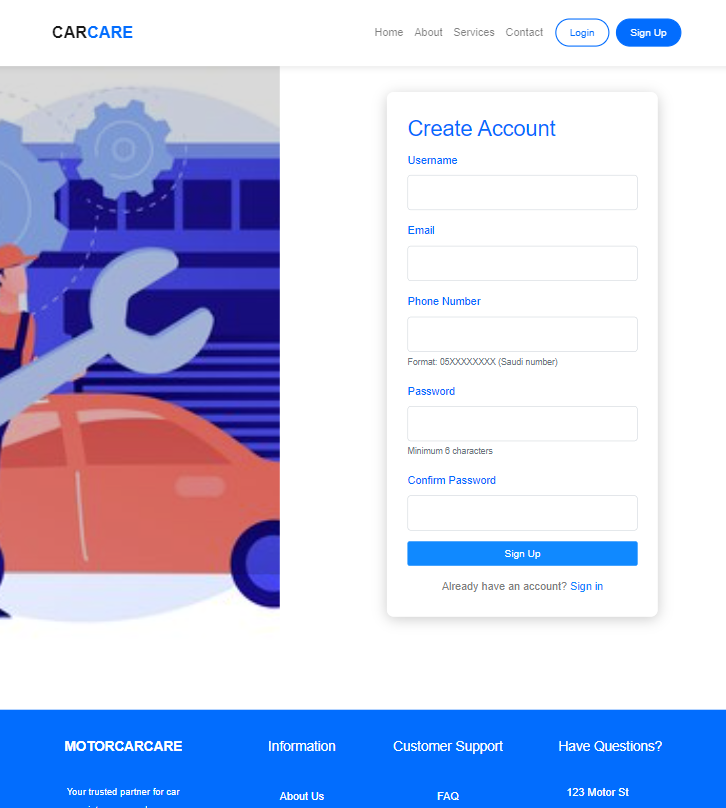


Figure35 create account

**Login page:** This is the Login page of the Motor Car Care web application. It allows users to log in to their accounts. If users don't have an account, they can click "Sign Up" to create one.

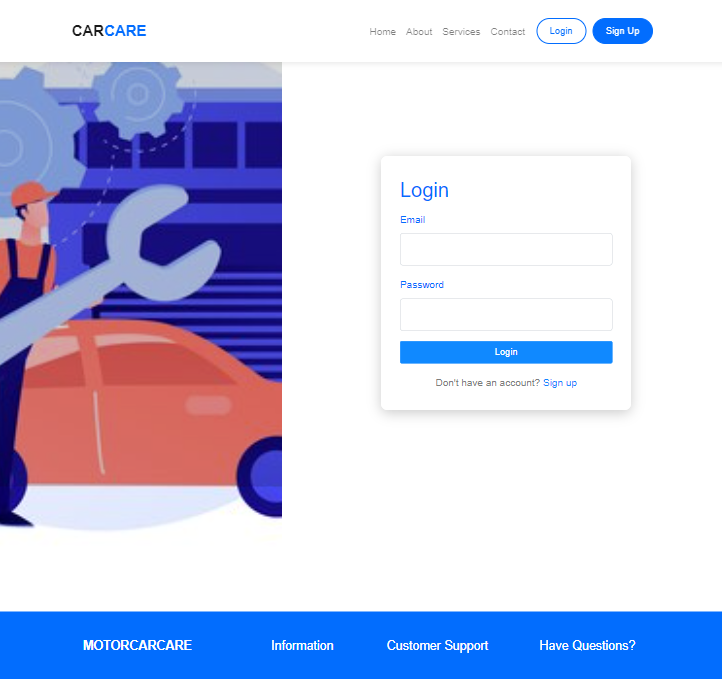


Figure36 Login

**My Appointments**: This page is part of the My Appointments section in the **Motor Car Care** web application. It displays the user's appointments, including details such as name, Service, Date, Time, Phone, **Status**, and **Payment** status.

* The first appointment is **Pending** with a status of **N/A** and **Not Paid**.
* The second appointment is **Booked** with a status of **Paid**.

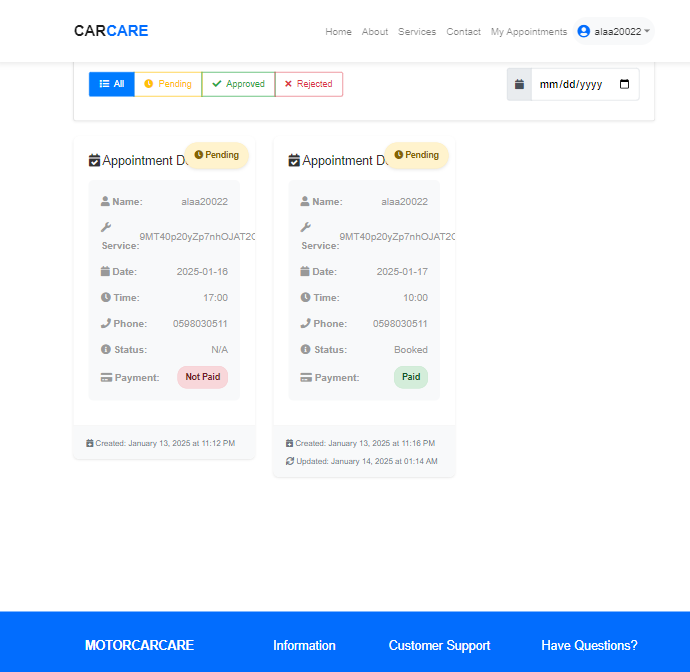


Figure37 My Appointment on website

**Services section:**This page is the Services section of the Motor Car Care web application. It lists various car maintenance and repair services available for Booking. Each service includes details such as Description, Duration, and Price. Users can click Book Now to schedule a service.

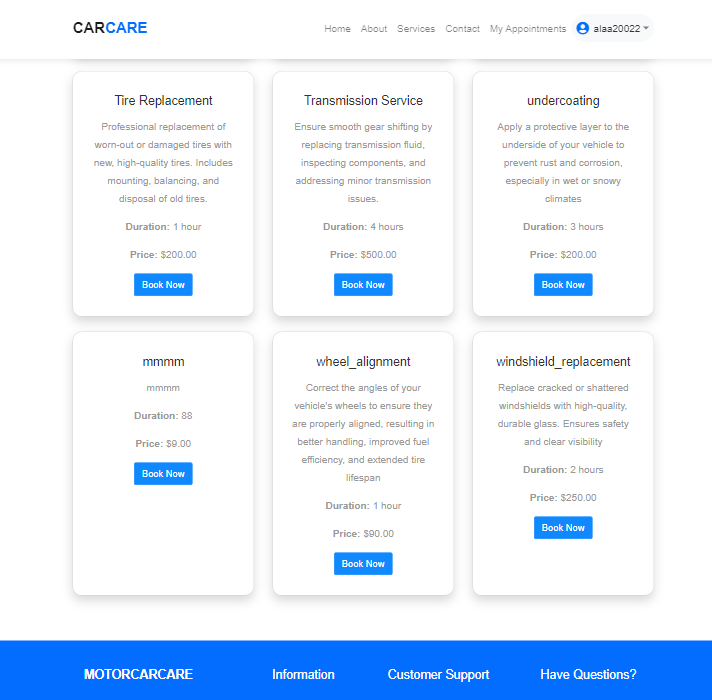


Figure38 services Booking

**About Us:** This page is the About Us section of the MotorCarCare web application. It highlights the company's commitment to providing high-quality automotive services.

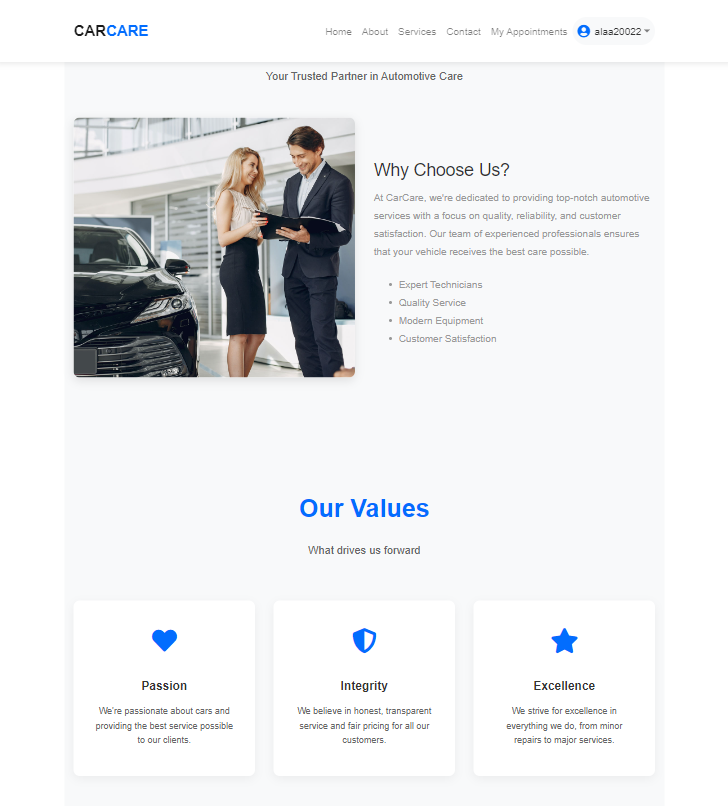


Figure 39 our value

**Contact Us:** This page is the Contact Us section of the Motor Car Care web application. It allows users to contact the company by filling out a form with their Name, Phone Number, Email Address, Subject, and Message.

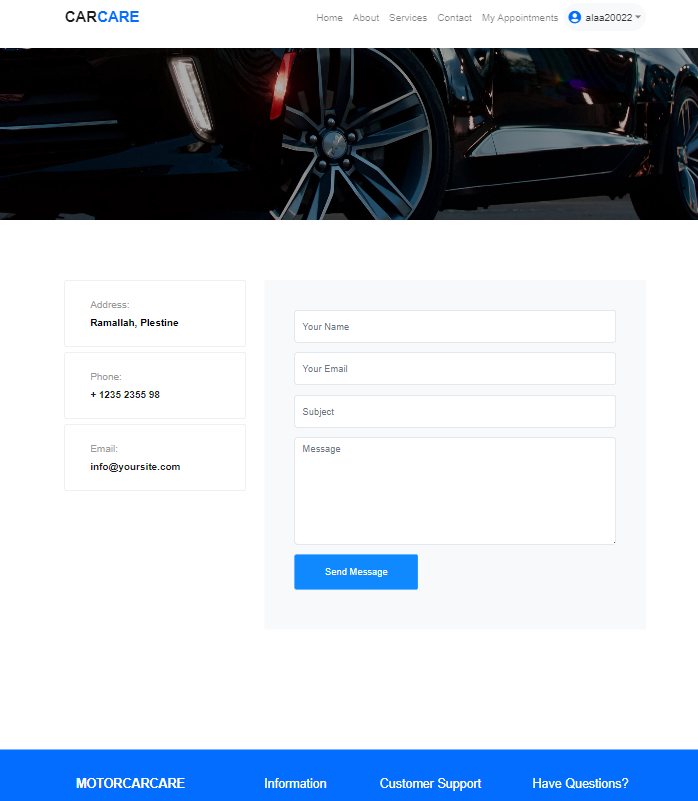


Figure 40 Contact Us

**Update Profile**: This page is part of the Update Profile section in the Motor Car Care web application. It allows users to update their profile information. The note "Final cannot be changed for security reasons" suggests that specific details (like username or ID) are fixed for security purposes. Users can modify other personal information as needed.

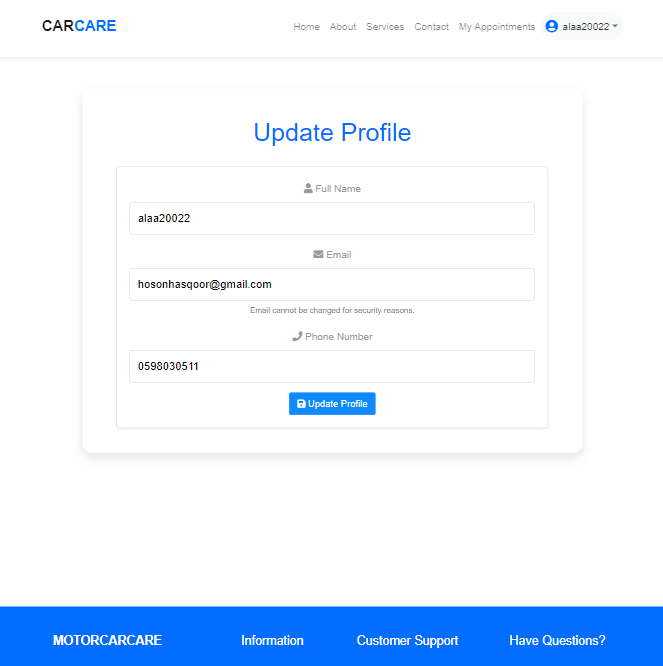


Figure 41 Update Profile

**Company Signup**: This page is the Company Signup section of the Motor Car Care web application. It allows companies to create an account by entering their Company Name, Username, Email, Password, and Confirm password. Companies must agree to the Terms & Conditions before signing up. If a company already has an account, they can click "Sign in " to log in.

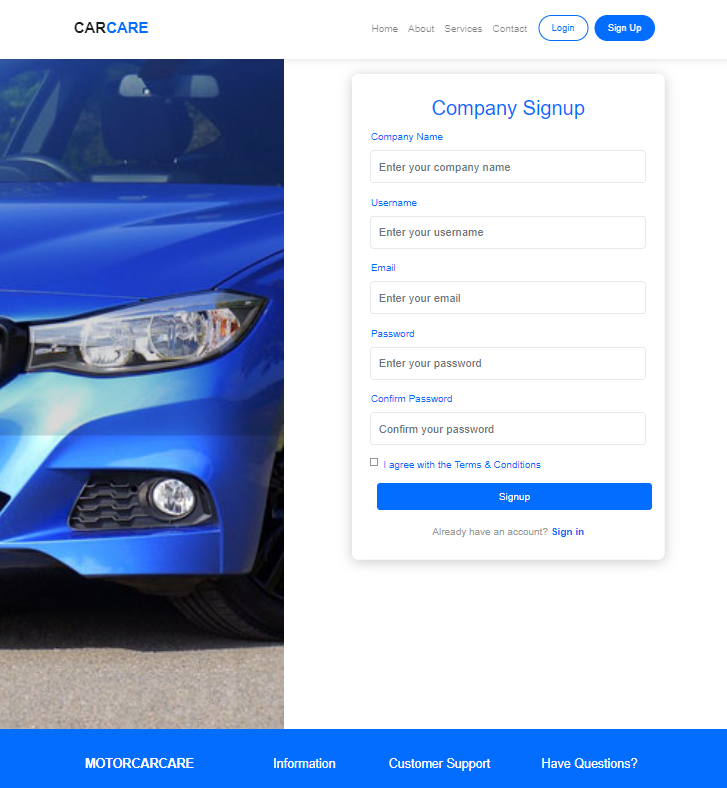


Figure42 Sign up

**Company Login**: This page is the Company Login section of the Motor Car Care web application. It allows companies to log in by entering their Email and Password. If a company does not have an account, they can click "Signup Now " to create one.

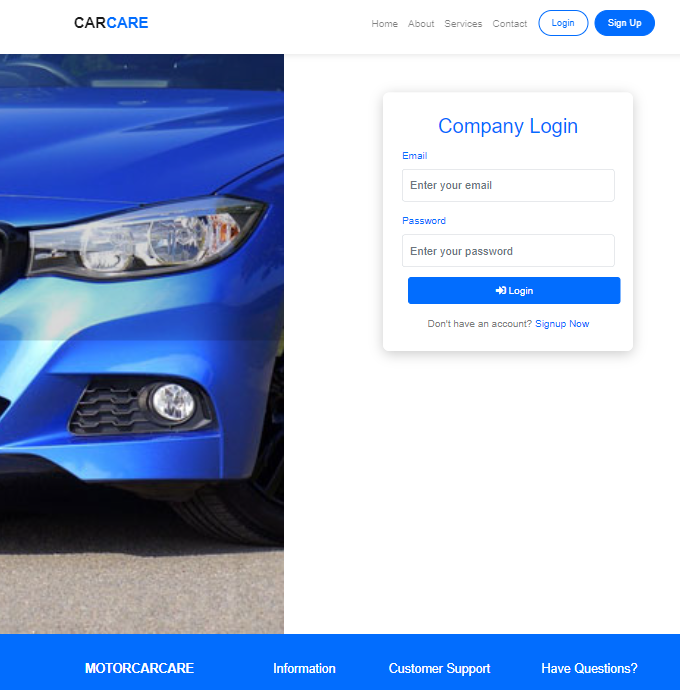


Figure 43 company login

**Dashboard**: This page is the Dashboard of the Motor Car Care web application, designed for workshop managers. It provides an overview of key metrics like Total Appointments, Today's Appointments, Active Services, and Total Customers. The Recent Appointments section lists details such as customer names, services, dates, times, and statuses, with options to manage appointments.

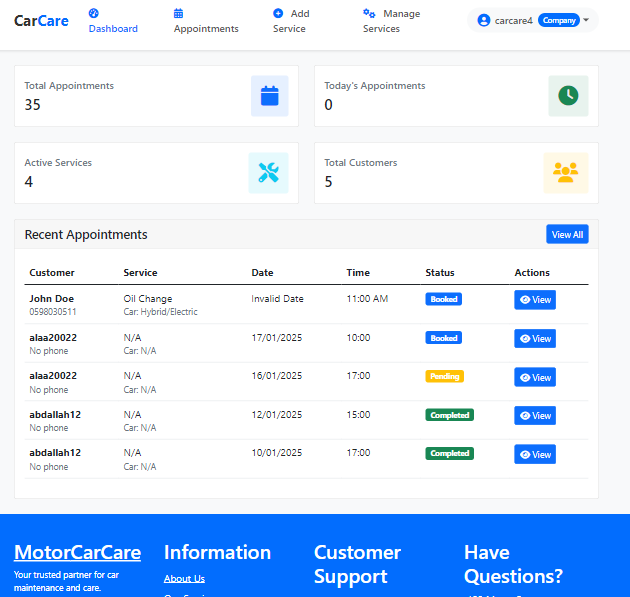


Figure44 Dashboard

**Appointments:** This page is the Appointments section of the Motor Car Care web application, designed for workshop managers. It displays detailed information about customer appointments, including service type, Date & time, car details, status, and payment. Managers can update or manage appointments and filter them by service status, payment status, and dates. The page also allows adding or managing services, ensuring efficient tracking and smooth operations.

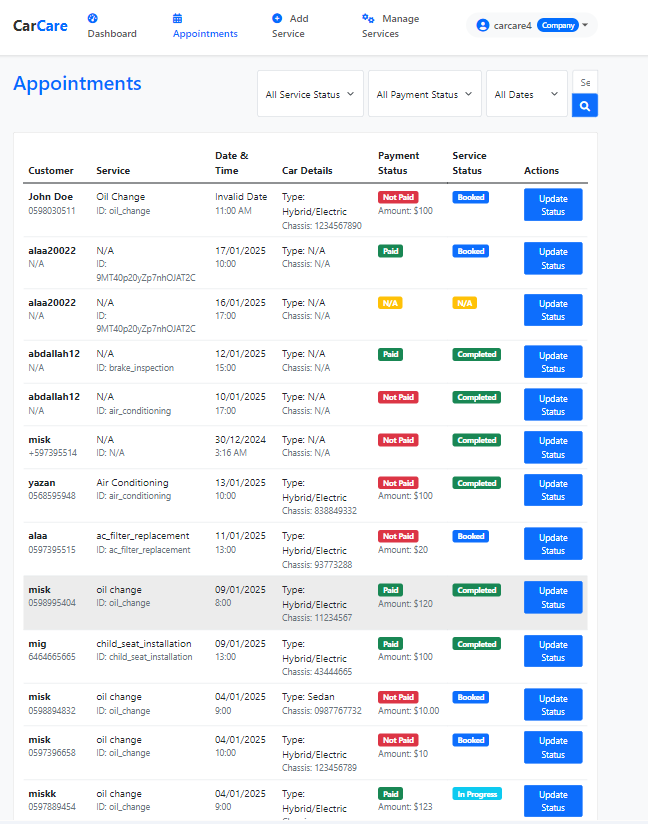


Figure45 Website Appointment screen

**Update Status:** This page is the Update Status section of the Motor Car Care web application, designed for workshop managers. It allows managers to update the status of customer appointments, including service type, payment details, and progress (e.g., In Progress, Completed). Managers can filter appointments by service status and payment status to efficiently track and manage services.

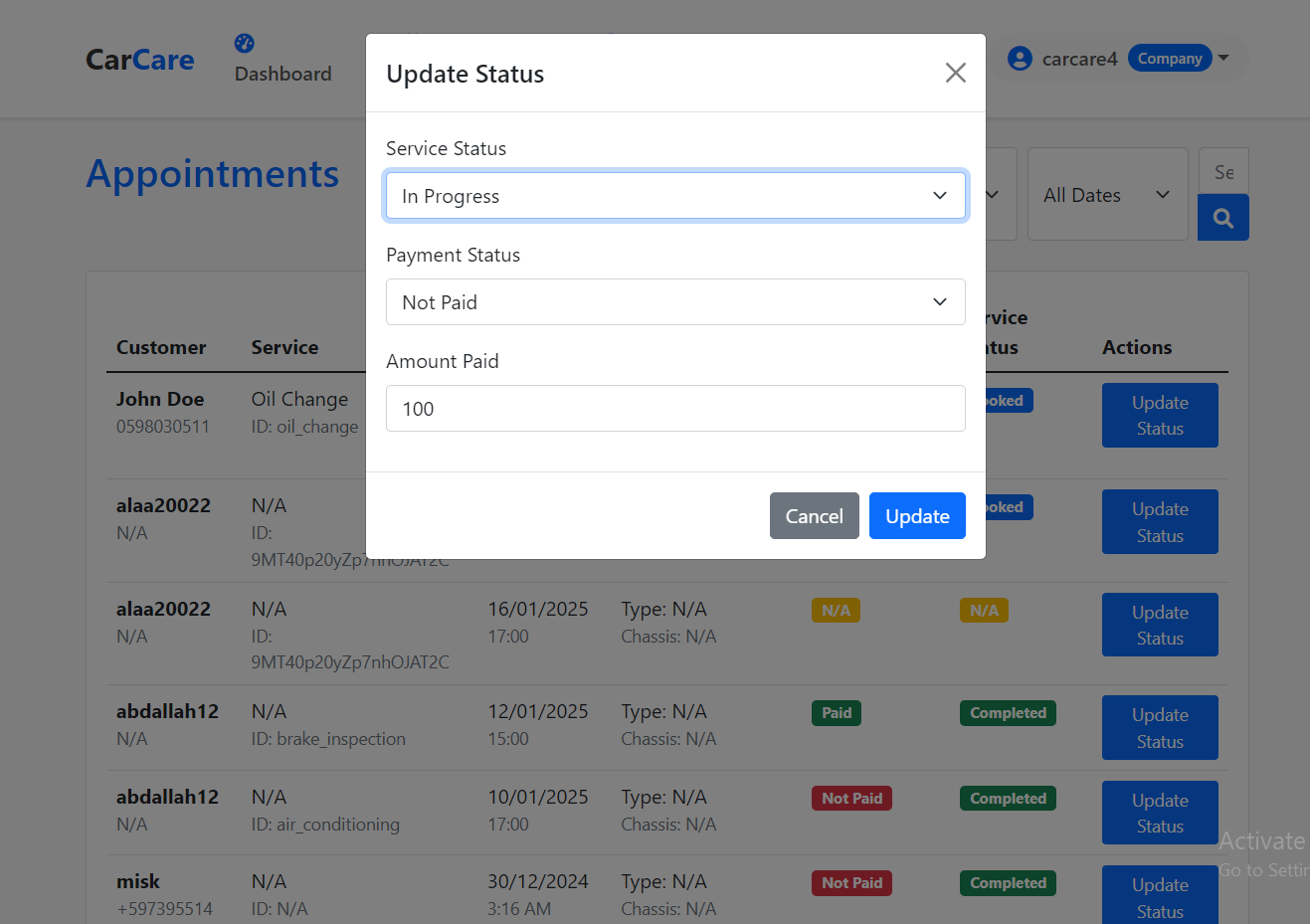


Figure46 Update status

**Add Service:** This page is part of the Add Service section in the Motor Car Care web application, designed for workshop managers. It allows adding new services by entering the Service Name, Description, Duration, and Price and uploading an Image (max 5MB, JPG/PNG). Managers can click "Add Service " to save the new service, thereby expanding the range of options available to customers.

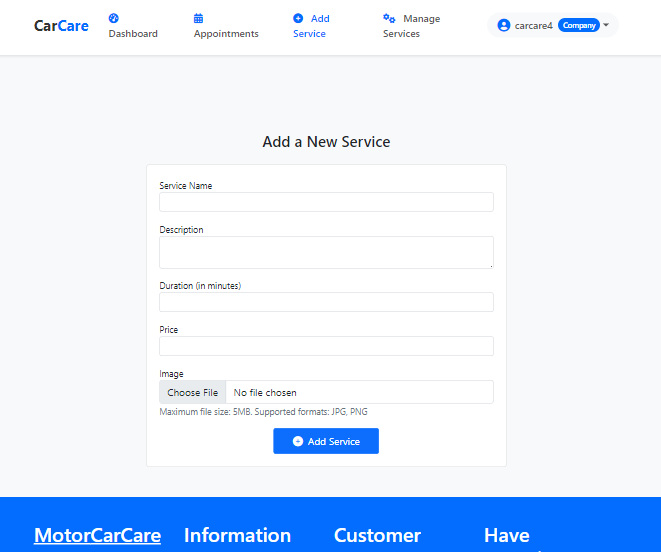


Figure47 Add New services.

**Manage Services**: This page is part of the Manage Services section in the Motor Car Care web application, designed for workshop managers. It lists all services, including name, description, duration, price, status, and last updated Date. Managers can edit service details or add new services. The page helps efficiently manage and update the range of services offered.

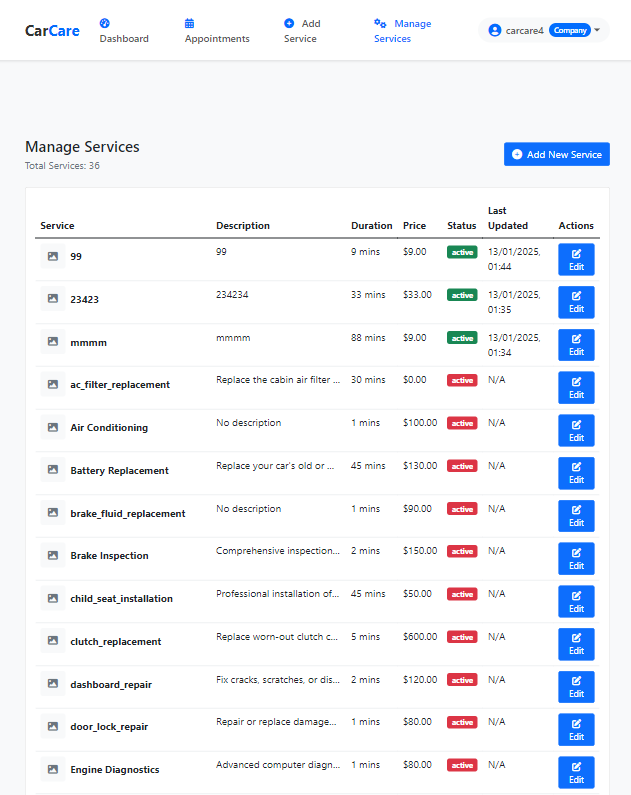


Figure48 manage services

**Update Service**: This page is the Update Service section of the Motor Car Care web application, designed for workshop managers. It allows managers to edit service details, including Service Name, Description, Duration, Price, and Status. Managers can also upload a new image (max 5MB, JPG/PNG) for the service. Options to Activate, Cancel, or Save Changes are provided. The page helps managers maintain accurate and up-to-date service information.

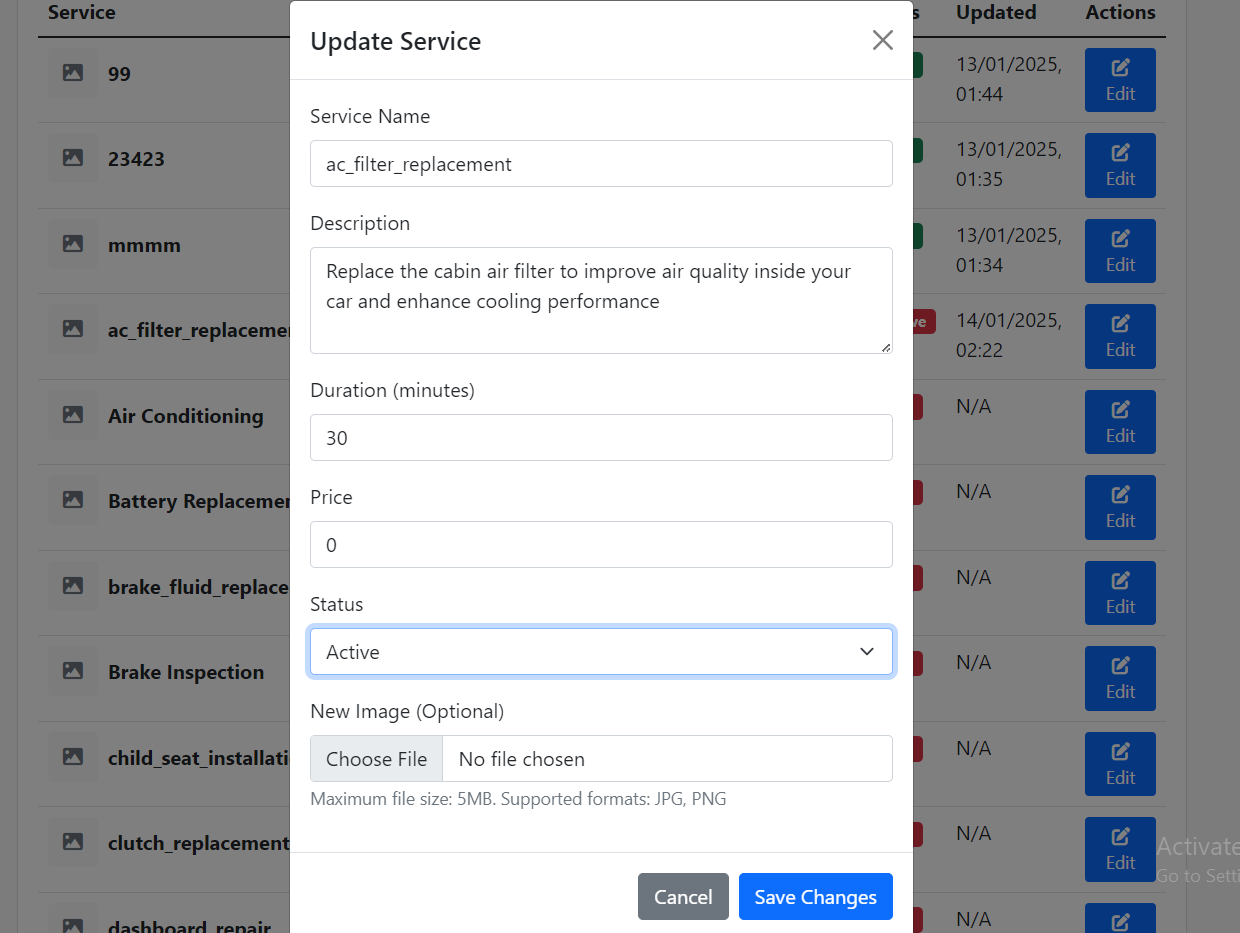


Figure49 Update services

# Chapter 5: System Testing

## 5.1 Overview

MotorCarCare is an integrated solution that aims to simplify the car maintenance and repair process for both car owners and workshop managers. The project allows customers to book appointments online with ease, track the progress of the repair process, and receive automatic alerts. As for workshops, the system helps them manage appointments and inventory with high efficiency. The project is designed to be easy to use for everyone, as it works like a familiar application, with instant updates that ensure seamless communication between customers and workshops. The project is also scalable and can accommodate an increasing number of users as the business expands, making car care easy and hassle-free for everyone.

## 5.2 List of features to be tested

* Booking a Car Service
* Viewing Service History
* Updating User Profile
* Sending Notification
* Sign Up
* Login
* Login and Booking History Verification
* Forger password
* Payment system

## 5.3 List of test cases

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | |  | | |  | | | **TEST DATE** | | | | |
| Booking a Car Service | | | | | | | Jane 12, 2025 | | | | | |
| TEST SCENARIO | | | | |  | | | TEST DESIGNED BY | | | | | |  |
| Booking a Car Service | | | | | | | Alaa Mohammad Dar Yaqoub | | | | | |
|  | | | | | | | | | | | | |
| TEST DESCRIPTION | | |  | | | | | | PRE-Conditions | | |  |
| The system should enable a logged-in user to book a car service appointment by providing the required details, including card number, chassis number, vehicle type, service type, appointment date, and time. | | | | | | | | | The user must be logged in.  The booking page (appointment form) must be accessible. | | | |
|  | | | | | | | | | | | | |
| STEP ID | STEP DESCRIPTION | |  | EXPECTED RESULTS | | ACTUAL RESULTS | | | | PASS / FALL | ADDITIONAL NOTES | |
| 1 | Log in to the system using valid **credentials**. | The user is successfully logged in and redirected to the user dashboard | | | | The user logged in successfully and was redirected to the Dashboard. | | | | PASS |  | |
| 2 | Navigate to the "Book Appointment" page. | The booking form is displayed with all required fields. | | | | The booking form loaded successfully with all fields available. | | | | PASS |  | |
| 3 | Fill in the "Card Number" field. | The card number is entered correctly. | | | | The card number was entered correctly. | | | | PASS | Used test data: "1234567890". | |
| 4 | Fill in the "Chassis Number" field. | The chassis number is entered correctly. | | | | The chassis number was entered correctly. | | | | PASS | Used test data: "CH123456789". | |
| 5 | Select a "Vehicle Type" from the dropdown. | The vehicle type is selected correctly. | | | | The vehicle type was selected correctly. | | | | PASS | Selected "Sedan" from the dropdown. | |
| 6 | Select a "Service Type" from the dropdown. | The service type is selected, and the service details (time and price) are updated. | | | | The service type was selected, and the service details were  updated  correctly. | | | | PASS | Selected "Change the engine oil" (Time: 30 mins, price: $20). | |
|  |  |  | | | |  | | | |  |  | |
| 7 | Select an "Appointment Date" from the calendar. | The appointment date is selected correctly. | | | | The appointment date was selected correctly. | | | | PASS | Selected Date: "2025-01-15". | |
| 8 | Select an "Appointment Time" from the buttons. | The appointment time is selected correctly. | | | | The appointment time was selected correctly. | | | | PASS | Selected time: "10:00 AM". | |
| 9 | Fill in the "Service Description" field. | The service description is entered correctly. | | | | The service description was entered correctly. | | | | PASS | Used test data: "Regular maintenance service for Sedan." | |
| 10 | Click the "Book Appointment" button. | The appointment is booked successfully, and the user is redirected to the Dashboard. | | | | The appointment was booked successfully, and the user was redirected to the Dashboard. | | | | PASS | A success message appeared: "Appointment booked successfully!" | |
| 11 | Verify the booked appointment in the "Appointments History" table. | The booked appointment is displayed correctly in the appointment's history. | | | | The booked appointment was displayed correctly in the appointment's history. | | | | PASS |  | |
| 12 | Attempt to book an appointment with missing required fields. | An error message is displayed, and the Booking is prevented. | | | | The system allowed the Booking despite missing required fields. | | | | FALL | Left the "Card Number" field empty. Expected an error message but the Booking succeeded. | |
| 13 |  |  | | | |  | | | |  |  | |
|  | Attempt to book an appointment with an invalid date (past Date). | An error message is displayed, and the Booking is prevented. | | | | An error message appeared: "Please select a valid date." | | | | PASS | Selected a past date: "2024-01-01". | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | | | |  | |  | **TEST DATE** | | | | | | |
| Viewing Service History | | | | | June 12, 2025 | | | | | |
| **TEST SCENARIO** | | | | | |  | **TEST DESIGNED BY** | | | | | |  |
| Ensuring users can view their service history (appointments) correctly. | | | | | | Alaa Mohammad Dar Yaqoub | | | | | |
|  | | | | | | | | | | | |
| **TEST DESCRIPTION** | | |  | | | **PRE-Conditions** | | | |  | |
| The system should allow a logged-in user to view their service history (appointments) in a table format, including details such as service type, appointment date, time, price, and status. | | | | | | * The user must be logged in. * The user must have at least one booked appointment. * The user dashboard (user) must be accessible. | | | | | |
|  | | | | | | | | | | | |
| STEP ID | STEP DESCRIPTION |  | | EXPECTED RESULTS | | ACTUAL RESULTS | | PASS / FALL | ADDITIONAL NOTES | | |
| 1 | Log in to the system using valid credentials. | | | The user is successfully logged in and redirected to the user dashboard. | | The user logged in successfully and was redirected to the Dashboard. | | PASS |  | | |
| 2 | Navigate to the "Appointments History" section. | | | The appointments history table is displayed with all booked appointments. | | The appointments history table loaded successfully with all booked appointments. | | PASS |  | | |
| 3 | Verify the "Service Type" column. | | | The service type for each appointment is displayed correctly. | | The service type was **not displayed correctly** for some appointments. | | FALL | Example: Expected "Change the engine oil" but displayed "N/A". | | |
| 4 | Verify the "Appointment Date" column. | | | The appointment date for each appointment is displayed correctly. | | The appointment date was displayed correctly for all appointments. | | PASS | Example: "2025-01-15". | | |
| 5 | Verify the "Appointment Time" column. | | | The appointment time for each appointment is displayed correctly. | | The appointment time was displayed correctly for all appointments. | | PASS | Example: "10:00 AM". | | |
| 6 | Verify the "Service Time" column. | | | The service time for each appointment is displayed correctly. | | The service time was displayed correctly for all appointments. | | PASS | Example: "30 mins". | | |
| 7 | Verify the "Service Price" column. | | | The service price for each appointment is displayed correctly. | | The service price was displayed correctly for all appointments. | | PASS | Example: "$20.00". | | |
| 8 | Verify the "Service Description" column. | | | The service description for each appointment is displayed correctly. | | The service description was displayed correctly for all appointments. | | PASS | Example: "Regular maintenance service for Sedan." | | |
| 9 | Verify the "Payment Status" column. | | | The payment status for each appointment is displayed correctly. | | The payment status was displayed correctly for all appointments. | | PASS | Example: "Paid" (green badge) or "Not Paid" (red badge). | | |
| 10 | Verify the "Service Status" column. | | | The service status for each appointment is displayed correctly. | | The service status was displayed correctly for all appointments. | | PASS | Example: "Completed" (green badge), "In Progress" (yellow badge), or "Booked" (blue badge). | | |
| 11 | Verify the "Service Status1" column. | | | The secondary service status for each appointment is displayed correctly. | | The secondary service status was displayed correctly for all appointments. | | PASS | Example: "Approved" (green badge), "Disapproved" (red badge), or "No Status" (gray badge). | | |
|  |  | | |  | |  | |  |  | | |
| 12 | Attempt to view service history with no appointments. | | | A message or empty state is displayed indicating no appointments found. | | A message was displayed: "No appointments found." | | PASS | This step requires a user with no booked appointments. | | |
|  |  | | |  | |  | |  |  | | |

Table 8 CaseStud1 Booking appointment

Table 9 CaseStudy2 Viewing Service History

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | |  | | | | |  | **TEST DATE** | | | |
| Updating User Profile | | | | | | June 10, 2025 | | | | | |
| **TEST SCENARIO** | | | | | | |  | **TEST DESIGNED BY** | | | | |  |
| Ensuring users can update their profile information. | | | | | | Alaa Mohammad Dar Yaqoub | | | | | |
|  | | | | | | | | | | | |
| **TEST DESCRIPTION** | | | |  | | **PRE-Conditions** | | | | |  |
| The system should allow a logged-in user to update their profile information (Full Name, Email, Phone) and ensure the changes are saved correctly. | | | | | | * The user must be logged in. * The update profile page (update\_profile) must be accessible. | | | | | |
|  | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** | |  | | **EXPECTED RESULTS** | **ACTUAL RESULTS** | | | **PASS / FALL** | **ADDITIONAL NOTES** | |
| 1 | Open the registration page. | Registration page is displayed. | | | | The registration page loaded successfully. | | | PASS |  | |
| 2 | Fill in the registration form with valid data. | The data is entered correctly into the form fields. | | | | The form fields were filled with valid data. | | | PASS | Used test data: Full Name = "Alaa DarYaqoub", Email = "[alaadaryaqoub@gmail.com](mailto:alaadaryaqoub@gmail.com)", Phone = "0598030523". | |
| 3 | Submit the registration form. | Account is created, and confirmation is displayed. | | | | The account was created, and the user was redirected to the login page. | | | PASS | A confirmation message appeared: "Account created successfully." | |
| 4 | Log in using the newly created account. | The user is successfully logged in and redirected to the user dashboard (user.php) | | | | The user logged in successfully and was redirected to the Dashboard. | | | PASS |  | |
| 5 | Navigate to the "Update Profile" page. | The update profile form is displayed with the current user's information pre-filled. | | | | The update profile form loaded with the correct user data pre-filled. | | | PASS |  | |
| 6 | Modify the "Full Name", "Email", and "Phone". | The form fields are updated with the new data. | | | | The form fields were updated with the new data. | | | PASS | Updated data: Full Name = "Alaa Mohammad", Email = "[alaa@gmail.com](mailto:alaa@gmail.com)", Phone = "0598030547". | |
| 7 | Click the "Update Profile" button. | Click the "Update Profile" button. | | | | The profile was updated, and the user was redirected to the Dashboard. | | | PASS | A success message appeared: "Profile updated successfully!" | |
| 8 | Verify the updated profile information. | The updated information (Full Name, Email, Phone) is displayed correctly. | | | | The updated information was displayed correctly on the Dashboard. | | | PASS |  | |
| 9 | Attempt to update with invalid email format. | An error message is displayed, and the update is prevented. | | | | An error message appeared: "Invalid email format." | | | PASS | Used invalid email: "invalid-email". | |
| 10 | Attempt to update with invalid email format. | An error message is displayed, and the update is prevented. | | | | The system allowed the update despite the invalid email format. | | | FALL | Used invalid email: "invalid-email". Expected an error message but the update succeeded. | |
|  |  |  | | | |  | | |  |  | |
| 11 | Log out and log in again. | The updated profile information persists and is displayed correctly. | | | | The updated information was still displayed after logging back in. | | | PASS |  | |

Table 10 CaseStudy3 Updating User Profile

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | | | |  | |  | **TEST DATE** | | | | |
| Sending Notification | | | | | June 10, 2025 | | | | | | |
| **TEST SCENARIO** | | | | | |  | **TEST DESIGNED BY** | | | | | |  |
| Sending notifications to users for booking-related status. | | | | | Alaa Dar Yaqoub | | | | | | |
|  | | | | | | | | | | | |
| **TEST DESCRIPTION** | | |  | | **PRE-Conditions** | | | | |  | |
| The test ensures that notifications are sent to the correct users with right details for booking confirmation, reminders, rejection, or status updates. | | | | | * The user has allowed the system to send notifications. * A booking event (e.g., appointment booked, canceled, or updated) has been initiated in the system | | | | | | |
|  | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** |  | | **EXPECTED RESULTS** | **ACTUAL RESULTS** | | | **PASS / FALL** | **ADDITIONAL NOTES** | |
| 1 | |  | | --- | | Install the App and open it for the first time. |  |  | | --- | |  | | | | |  | | --- | | The notification permission request is displayed. |  |  | | --- | |  | | The permission request shown | | | PASS | |  | | --- | | Ensure the App is in a fresh state. |  |  | | --- | |  | | |
| 2 | |  | | --- | | Click "Allow" on the permission request. |  |  | | --- | |  | | | | |  | | --- | | Notifications are enabled. |  |  | | --- | |  | | The user will have a notification when make a booking | | | PASS | |  | | --- | | Verify this through system logs. |  |  | | --- | |  | | |
| 3 | |  | | --- | | Reopen the App after closing it. |  |  | | --- | |  | | | | |  | | --- | | The permission request does not appear again. |  |  | | --- | |  | | The permission request dost not shown | | | PASS | |  | | --- | | Ensure it only appears on the first launch. |  |  | | --- | |  | | |
| 4 | |  | | --- | | Click "Don't Allow" on the permission request. |  |  | | --- | |  | | | | |  | | --- | | Notifications stay disabled, and no notifications are sent. |  |  | | --- | |  | | no notification shown | | | PASS | Verify this through system logs. | |
| 5 | The workshop rejects the booking request with a specific reason. | | | The user receives a notification if the Booking was rejected, with the reason for rejection. | The rejection of appointment with reason is show on notification page | | | PASS | Verify notification details are accurate. | |
| 6 | The workshop updates the status of your Booking | | | The user receives a notification if the Booking was update. | The update of status of appointment with reason is show on notification page | | | PASS | Verify notification details are accurate. | |

Table 11CaseStudy4 Sending notification

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | | | |  | | |  | **TEST DATE** | | | | | |
| Sign Up | | | | | June 11, 2025 | | | | | | | | |
| **TEST SCENARIO** | | | | | | |  | **TEST DESIGNED BY** | | | | |  | |
| Check that new users can sign up properly, and the system shows errors if required details are missing or wrong. | | | | | Ayham Jameel | | | | | | | | |
|  | | | | | | | | | | | | | |
| **TEST DESCRIPTION** | | |  | | | **PRE-Conditions** | | | | | |  | |
| This test checks if a new user can successfully create an account. It ensures the welcome page provides the "Sign Up" option and verifies that the system properly validates the signup form. If a user forgets to fill in a required field or enters invalid information, the system should display an error message. The test also confirms that valid inputs lead to successful account creation. | | | | | | * The App or website must show a welcome page when opened for the first time. * The welcome page must have an option for "Sign Up." * The system must ask if the user is a "Customer" or a "Company." * The form must validate all required fields and show errors for missing or incorrect inputs. | | | | | | | |
|  | | | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** |  | | **EXPECTED RESULTS** | | | | | **ACTUAL RESULTS** | **PASS / FALL** | **ADDITIONAL NOTES** | | |
| 1 | |  | | --- | | Open the App or website. |  |  | | --- | |  | | | | |  | | --- | | A welcome page is displayed with "Sign Up" and "Login" options. |  |  | | --- | |  | | | | | | The welcome page appeared as expected. | pass | Ensure the design and layout of the welcome page are working successfully | | |
| 2 | |  | | --- | | Click on Sign Up. |  |  | | --- | |  | | | | |  | | --- | | The system displays options to select Customer or Company. |  |  | | --- | |  | | | | | | The options for Customer and Company" were displayed. | pass | Check both "Customer" and "Company." | | |
| 3 | |  | | --- | | Select customer and click Login. |  |  | | --- | |  | | | | |  | | --- | | The system displays the "Customer Signup" page. |  |  | | --- | |  | | | | | | Errors appeared for the missing fields. | pass | Test for invalid inputs | | |
| 4 | |  | | --- | | Enter valid details and submit the form. |  |  | | --- | |  | | | | |  | | --- | | The account is created, and a confirmation message is displayed. |  |  | | --- | |  | | | | | | The account was successfully created. | pass | |  | | --- | |  |  |  | | --- | | Check the database to ensure data is saved. | | | |

Table 12 CaseStudy5 Sign Up

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | | | |  | |  | **TEST DATE** | | | | | | | | |
| Login | | | | | June 11, 2025 | | | | | | | | | | |
| **TEST SCENARIO** | | | | | |  | **TEST DESIGNED BY** | | | | | | | | | |  |
| Check that new users can Login properly, and the system shows errors if required details are missing or wrong. | | | | | Ayham Jameel | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| **TEST DESCRIPTION** | | |  | | | | | | **PRE-Conditions** | | | | |  | |
|  | | |  | | | | | |  | | | | |  | |
| This test checks if a new user can successfully create an account. It ensures the welcome page provides the "Sign Up" option and verifies that the system properly validates the signup form. If a user forgets to fill in a required field or enters invalid information, the system should display an error message. The test also confirms that valid inputs lead to successful account creation. | | | | | | | | | The App or website must show a welcome page when opened for the first time.  ·The system must validate the entered credentials and show errors for incorrect inputs. | | | | | | |
|  | | | | | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** |  | | **EXPECTED RESULTS** | | | | **ACTUAL RESULTS** | | | **PASS / FALL** | | **ADDITIONAL NOTES** | |
| 1 | |  | | --- | | Open the App or website. |  |  | | --- | |  | | | | |  | | --- | | A welcome page is displayed with and "Login" options. |  |  | | --- | |  | | | | | The welcome page appeared as expected. | | pass | | Ensure the design and layout of the welcome page are working successfully | | | |
| 2 | |  | | --- | | Click on Login. |  |  | | --- | |  | | | | |  | | --- | | The system displays options to select Customer or Company. |  |  | | --- | |  | | | | | The options for Customer and Company" were displayed. | | pass | | Check both "Customer" and "Company." | | | |
| 3 | |  | | --- | | Select Customer |  |  | | --- | |  | | | | |  | | --- | | The system displays the Customer Login page. |  |  | | --- | |  | | | | | The login page loaded without problems. | | pass | | Confirm all fields are visible. | | | |
| 4 | |  | | --- | | Enter incorrect information and submit. |  |  | | --- | |  | | | | |  | | --- | | An error message like "Invalid email or password" is shown. |  |  | | --- | |  | | | | | Errors appeared for wrong login details. | | pass | | |  | | --- | |  |  |  | | --- | | Test multiple invalid | | | | |
| 5 | Enter correct information and submit. | | | The user is logged in, and they are redirected to their Dashboard. | | | | Login was successful, and the user reached the Dashboard. | |  | | Verify Login through system logs. | | | |

Table 13 CaseStudy6 Login

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | | | |  | |  | | **TEST DATE** | | | |
| Forgot password | | | | | January 11, 2025 | | | | | | |
| **TEST SCENARIO** | | | | | |  | | **TEST DESIGNED BY** | | | |  |
| Sending a request to reset code to the user's email for the "Forgot Password" process. | | | | | Misk Maali | | | | | | |
|  | | | | | | | | | | | |
| **TEST DESCRIPTION** | | |  | | **PRE-Conditions** | | | | | |  |
| The test ensures that users can reset their password by receiving a link via email to rest it. | | | | | * The user has account already with valid email. * The "Forgot Password" including is active in the system. | | | | | | |
|  | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** |  | | **EXPECTED RESULTS** | | | **ACTUAL RESULTS** | | **PASS / FALL** | **ADDITIONAL NOTES** | |
| 1 | |  |  |  | | --- | --- | --- | | |  | | --- | | Open the login page and click on "Forgot password." |  |  | | --- | |  | |  |  | | --- | |  | | | | |  |  |  | | --- | --- | --- | | |  | | --- | | The system views a forget password page |  |  | | --- | |  | |  |  | | --- | |  | | | | The page of forget screen | | PASS | The system displays a place to enter the email address. | |
| 2 | |  | | --- | | Enter a email address and submit. | |  |  |  | | --- | |  | | | | |  | | --- | | The system sends a link provided email when tap on it go to reset password screen. |  |  | | --- | |  | | | | Having email from the system including link to reset password | | PASS | Verify the email if have a message and check the spam | |
| 3 | |  |  |  | | --- | --- | --- | | |  | | --- | | Enter the new password on the "Reset Password" page. |  |  | | --- | |  | |  |  | | --- | |  | | | | |  | | --- | | The system validates the new password and allows the user to proceed to reset the password. |  |  | | --- | |  | | | | The user reset the password and can open the system | | PASS | Keep remember the password | |
| 4 | Login with the new password | | | |  | | --- | | The system successfully updates the password and logs the user in. |  |  | | --- | |  | | | | Enables the user to log in to the system and access the home screen. | | PASS | Make sure your password is strong, unpredictable, and something you can remember. | |

Table 14 CaseStudy7 Forgot password

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | | | |  | |  | **TEST DATE** | | | | | |
| Payment System | | | | | January 11, 2025 | | | | | | | |
| **TEST SCENARIO** | | | | | |  | **TEST DESIGNED BY** | | | | |  | |
| Tolal of payment of each day | | | | | Misk Maali | | | | | | | |
|  | | | | | | | | | | | | |
| **TEST DESCRIPTION** | | |  | | **PRE-Conditions** | | | | | |  | |
| This test ensures that the payment system exactly records the payments for services during Booking, displays the amount paid on the customer's service history page, allows workshops to update the payment status, and provides financial summaries for the company based on daily or monthly earnings. | | | | | * The user has an account in the system. * The user books a service throw the App or website. * The workshop has access to their services history dashboards. | | | | | | | |
|  | | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** |  | | **EXPECTED RESULTS** | | | | **ACTUAL RESULTS** | **PASS / FALL** | **ADDITIONAL NOTES** | | |
| 1 | |  |  |  | | --- | --- | --- | | |  | | --- | | The customer books a service and proceeds to the services history |  |  | | --- | |  | |  |  | | --- | |  | | | | |  |  |  | | --- | --- | --- | | |  | | --- | | The system displays the total cost of the services. |  |  | | --- | |  | |  |  | | --- | |  | | | | | The customer know how much he paid for the services | PASS | |  | | --- | | Ensure service cost is correct. |  |  | | --- | |  | | | |
| 2 | |  |  |  | | --- | --- | --- | | |  | | --- | | Check the customer's service history page after payment. |  |  | | --- | |  | | |  |  |  | | --- | |  | | | | |  | | --- | | The service history page displays the amount paid, Date, and service name. |  |  | | --- | |  | | | | | The user has information about the services provided to him. | PASS | |  | | --- | |  |  |  | | --- | | Ensure information is accurate. | | | |
| 3 | |  |  |  | | --- | --- | --- | | |  | | --- | | The workshop marks the service as "paid" in their Dashboard. |  |  | | --- | |  | |  |  | | --- | |  | | | | |  | | --- | | The system updates the payment status for the Booking. |  |  | | --- | |  | | | | | The status on the user screen has been changed. | PASS | Ensure status is update. | | |
| 4 | The workshop updates the price of the service performed by the user. | | | |  | | --- | | The system successfully updates the price. |  |  | | --- | |  | | | | | The update shown on customer screen and workshop screen | PASS | |  | | --- | | Verify the update process is seamless. |  |  | | --- | |  | | | |
|  |  | | |  | | | |  |  |  | | |
| 5 | The workshop views the financial summary for the day, month. | | | The company dashboard shows total earnings for the day, month or specific day | | | | The earning is shown on services history page | PASS | Ensure correct collecting of payments. | | |

Table 15 CaseStudy8 Payment System

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE TITLE** | | | |  | |  | **TEST DATE** | | | | | |
| Login and Booking History Verification | | | | | January 11, 2025 | | | | | | | |
| **TEST SCENARIO** | | | | | |  | **TEST DESIGNED BY** | | | | |  | |
| Verify that users can successfully log in, and the system displays a history of past car bookings and fixes. | | | | | Ayham Jameel | | | | | | | |
|  | | | | | | | | | | | | |
| **TEST DESCRIPTION** | | |  | | **PRE-Conditions** | | | | | |  | |
| This test checks if a user can log in to their account successfully and ensures that the system retrieves and displays accurate booking history, including details of past car bookings and fixes. It validates proper storage and retrieval of user data, ensuring that the history reflects the correct status and details. | | | | | * + The system must have a registered user account.   + The system must store historical booking and repair records.   + The login page must validate credentials and allow only valid users.   + The "Booking History" section must be accessible after a successful login. | | | | | | | |
|  | | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** |  | | **EXPECTED RESULTS** | | | | **ACTUAL RESULTS** | **PASS / FALL** | **ADDITIONAL NOTES** | | |
| 1 | |  |  |  | | --- | --- | --- | | |  | | --- | | Open the App or website. |  |  | | --- | |  | |  |  | | --- | |  | | | | |  |  |  | | --- | --- | --- | | |  | | --- | | The login page is displayed with fields for email/username and password. |  |  | | --- | |  | |  |  | | --- | |  | | | | | The login page appeared as expected. | PASS | |  | | --- | | Ensure the design and layout of the login page. |  |  | | --- | |  | | | |
| 2 | |  |  |  | | --- | --- | --- | | |  | | --- | | Enter valid login credentials and submit. |  |  | | --- | |  | | |  |  |  | | --- | |  | | | | |  | | --- | | The system logs the user in and redirects to the Dashboard. |  |  | | --- | |  | | | | | User logged in successfully. | PASS | |  | | --- | |  |  |  | | --- | | Verify the user logged in | | | |
| 3 | |  |  |  | | --- | --- | --- | | |  | | --- | | Navigate to Booking History from the Dashboard. |  |  | | --- | |  | |  |  | | --- | |  | | | | |  | | --- | | The system displays a list of past car bookings and fixes. |  |  | | --- | |  | | | | | The booking history appeared correctly. | PASS | Ensure the history includes all relevant details | | |
| 4 | Verify details of a specific past booking. | | | |  | | --- | | The details match the records stored in the database. |  |  | | --- | |  | | | | | All booking details matched successfully | PASS | |  | | --- | | Confirm that repair status and costs are shown accurately. |  |  | | --- | |  | | | |
|  |  | | |  | | | |  |  |  | | |

Table 16 CaseStudy9 Login and Booking History Verification

## 5.4 End-user testing (Acceptance Testing), if applicable.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  | |  | |
| **TEST DESCRIPTION** | | | |  | | **TEST DESCRIPTION** | |
| Testing the system to ensure that new user can register successfully and receive confirmation | | There must be a registration from available on the web or App | | | | | | | | |
| User must have a valid email address | | **Test control:** manual | | | | | | | | |
|  | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** | |  | | **EXPECTED RESULTS** | | **ACTUAL RESULTS** | | **PASS / FALL** | **ADDITIONAL NOTES** |
| 1 | |  |  | | --- | --- | | |  | | --- | | Open the website or application and sign  up | | | The screen of sign up should be shown | | | | | signup screen shown successfully | | PASS | Ensure valid email are used |
| 2 | |  | | --- | | Enter user details (Email, password, username, phone, etc.), and submit if not access then appear error message | |  |  |  | | --- | |  | | |  | | --- | | User details are correct and submit successfully |  |  | | --- | |  | | | | | | submit successfully | | PASS | Ensure all fields are filed |
|  |  |  | | | | |  | |  |  |
| 3 | Login with the registered email and password | User is logged in successfully | | | | | User is logged in successfully | | PASS | Check if access to services history |

Table 17 test register

# Chapter 6: Conclusion and Future Works

## 6.1 Review of the project

The MotorCarCare project aimed to simplify car maintenance and repair services by providing an efficient and user-friendly platform for car owners and workshop managers. The system allowed car owners to book appointments online, track the progress of their maintenance, and receive automated reminders for scheduled maintenance.

For Workshop managers, tools benefited them in organizing and managing appointments, as well as Tracking inventory for better operational efficiency.

The project successfully utilized modern technologies like:

- Flutter for a cross-platform mobile application.

- PHP, HTML, and CSS for a responsive web interface.

- Firebase for real-time updates, user authentication, and data management.

The testing and simulations ensured that the App delivered a smooth, reliable, and scalable experience. The project achieved its goal of creating a suitable digital solution for car care services.

## 6.2 Future Works

* Add a payment gateway for online payments.
* Add support for Arabic language to serve a diverse audience.
* Integrate GPS and Google Maps to show nearby workshops and estimate arrival time.
* Enable push notifications for service reminders and updates.
* Add rating and feedback system for users to review services and workshops.
* Include a live chat feature for customer support or communication with the workshop.
* Develop an admin dashboard for managing users, bookings, and system settings.
* Improve security by implementing two-factor authentication (2FA).
* Add support for uploading and storing vehicle-related documents (e.g., registration, insurance).
* Extend compatibility to iOS devices through Flutter deployment.

# 5. References and Bibliography

[1] Kumar, S., & Ramesh, R. (2021). Design and Development of Car Maintenance System Using Mobile Technology. International Journal of Computer Applications, 174(7), 1–5.

[2] Ahmed, H., & Suleiman, A. (2022). Evaluation of Mobile-Based Auto Repair Systems. International Journal of Information Systems, 14(2), 33–40.

[3] Carfax, “Carfax Car Care,” Apple App Store. [Online]. Available: <https://apps.apple.com/us/app/carfax-car-care/id552472249>

[4] Drivvo, “Drivvo: Mileage & Fuel Manager,” Drivvo. [Online]. Available: <https://www.drivvo.com/en>

[5] Monta, “Best EV Charging Apps,” Monta Blog. [Online]. Available: <https://monta.com/uk/blog/best-ev-charging-apps>

[6] FIXD, “FIXD Car Health Monitor,” FIXD. [Online]. Available: [https://www.fixd.com](https://www.fixd.com/)

[7] “Analysis of Car Maintenance Apps,” International Journal of Creative Research Thought, vol. 11, no. 3, 2023. [Online]. Available: <https://ijcrt.org/papers/IJCRT2312308.pdf>

[8] University of Helsinki, “Evaluating Car Service Platforms,” Helda – Digital Repository. [Online]. Available: <https://helda.helsinki.fi/server/api/core/bitstreams/6a852459-531e-4c1c-943c-711453d8f6bb/content>

[9] Wikipedia contributors, “PHP,” Wikipedia: The Free Encyclopedia. [Online]. Available: <https://en.wikipedia.org/wiki/PHP>

[10] Amazon Web Services, “What is MySQL?,” AWS Documentation. [Online]. Available: <https://aws.amazon.com/rds/mysql/what-is-mysql/>

[11] Flutter, “Flutter Documentation,” Flutter Dev. [Online]. Available: <https://flutter.dev/docs>

[12] Fuel.io, “Fuel: Fuel Price App,” Fuel IO. [Online]. Available: [https://fuel.io](https://fuel.io/)