VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY
THE INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING
IT076IU

FINAL REPORT

Topic: Web portal for motor servicing at home

By Group 03 – Member List

No.	Full name – Student name	Student ID
1	Trịnh Thuỷ Tiên	ITITIU21328
2	Trần Bách Tùng	ITITIU21340
3	Đặng Nguyễn Trường Huy	ITITIU21010
4	Nguyễn Hoàng Minh Khôi	ITITIU21229
5	Bùi Phương Thanh	ITITIU21311
6	Nguyễn Trần Gia Huy	ITITIU21054
7	Vũ Văn Đô	ITITIU21125
8	Bạch Huy Hoàng	ITITWE21113
9	Nguyễn Trần Quốc Thịnh	ITITIU20312

Instructor: Assoc. Prof. Nguyen Thi Thuy Loan

May 2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Background

In a world characterized by time constraints and demanding schedules, accessing reliable and convenient motor services often poses a significant obstacle. Our project seeks to alleviate these burdens by bringing professional automotive expertise directly to the doorstep of our clientele.

Doorstep motor servicing allows individuals to schedule vehicle maintenance and repair services online, which are then carried out at their home or another specified location. This type of service utilizes the convenience of the internet and on-site delivery, making it easier for vehicle owners to maintain their vehicles without having to visit a garage or service center.

1.2 Problem Statement

Cognizant of modern homeowners' time constraints, the location of service use and the available resources, service availability may be limited in real-world scenarios that still need to be addressed. Factors such as terrain conditions and weather might necessitate taking the vehicle to the nearest service station for timely support for assurance regarding service provider competence.

1.3 Scope and Objectives

1.3.1 Scope

- **User Authentication and Profiles:** Register and login and manage customer profiles (name, phone, e-mail, address, preferred vehicles…).
- **Service Booking:** Reservation system and calendar integration to schedule appointments.
- **Service Catalog:** Display a list of available motor services such as oil changes, brake repairs, battery replacements, etc.
- **Technician Management:** Automatically assign and dispatch list the nearest available technicians to the customer’s location.
- **On-Site Service:** Professional technicians arrive at the specified location with the necessary tools and parts to perform the service.
- **Payment Integration:** Support for multiple payment methods (credit/debit cards, digital wallets, bank transfers, etc.) and invoicing and receipt generation for transactions.

1.3.2 Objectives

This project aims to revolutionize the motor servicing industry, making it more efficient, customer-centric, and accessible. The objectives include:

- **Enhance Convenience:** The primary objective is to provide customers with the convenience of scheduling motor servicing appointments without the need to visit a physical service center.

- **Accessibility:** Ensure the web portal is accessible to a wide range of users, including those with different devices and internet connections from anywhere at any time.
- **Transparency:** Provide transparent pricing and detailed descriptions of the services offered, including what each service entails and any additional charges that may apply.
- **Increase Customer Satisfaction:** Prioritize customer satisfaction by offering personalized service options and addressing customer inquiries. Keep users informed about the status of their service requests with real-time notifications.
- **Efficiency:** Streamline the booking and service delivery process to minimize waiting times and maximize efficiency.
- **Improve Service Quality:** Keep users informed about the status of their service requests with real-time notifications.

1.4 Customer Requirements

The development of this web portal aligns with the requirements outlined by our industry partners, who emphasize the need for the convenient and integration benefits of doorstep cars service system that can operate in diverse real-world scenarios.

1.5 Structure report

Title Page:
- University Name
- School name
- Course name, Course ID
- Title of the Report
- Topic's Name
- Group Information
- Lecturer's Name
- Date.

Table of Contents:
List of Sections and Subsections with Page Numbers.
List of Figures:
List of Figures with Page Numbers.
List of Tables:
List of Tables with Page Numbers.

This report is organized into seven chapters.
Chapter I. Introduction:
- Background
- Problem Statement
- Scope & Objectives
- Customer Requirements
- Structure report

Chapter II. Project Timeline and In-Charge table:
- Project Timeline
- In-charge Table

Chapter III. Methodology:
- Overview
- Requirement analysis

        - Requirement Register
        - Resource Register
        - Risk Register
    - Requirement Specifications
        - Use Case Diagram & Use case Descriptions
    - System Design
        - Entity Relationship Diagram
        - Class Diagram
        - Sequence Diagram
        - Database design
        - User Interface design

Chapter IV. Implementation & Results:

- Implement
    - Technologies
    - Programming languages
    - Frameworks
    - Tools
- Results of System's Architecture
    - Backend Overview
    - Frontend Overview

Chapter V. Discussion, Testing & Evaluation:

- Discussion
- Testing
    - Testing methodologies
        - Unit tests
        - Integration tests
        - System tests
    - Test cases
- Comparison
- Evaluation

Chapter VI. User Documentation:

- User manuals
- Technical guides

Chapter VII. Conclusion & Future work:

- Conclusion
- Future work

References

CHAPTER II

PROJECT TIMELINE AND IN-CHARGE TABLE

2.1 Project Timeline

2.1.1 Project timeline & milestones

Table 2. 1 Project timeline & milestones

Timeline	Task Title	Activities
Phase 1: Planning and Requirements		
Week 1	Project Planning	- Choose project topic. - Identify the general information of project: project goal and scope, milestone schedule, team member.
Week 2		- Identify key stakeholders. - Write proposal report. - Present the proposal to the project advisor.
Week 3	Requirements Analysis	- Identify features and functionalities. - Identify requirement Register (functional and non-functional requirements). - Identify resource requirements. - Identify risks. - Refine requirements based on feedback.
Week 4		Make requirement specifications: - Create Use case Diagram and Descriptions. - Finalize project plan with adjusted timelines. - Submit proposal report.
Phase 2: Design		
Week 5	System Design	- Create Database design. - User Interface design. - Software Development Methodology. - Design ERD, Class Diagram, Sequence Diagram.
Week 6		- Choose technologies, programming languages, frameworks and tools to use. - Build system architecture.
Phase 3: Implementation		
Week 7	Development	- Write midterm report.

Week 8		Frontend Development - Implement user interface following the requirement specifications. - Integrate libraries, frameworks with backend APIs for data.
Week 9		
Week 10		- Submit Midterm report. Backend Development - Develop server and APIs. - Implement database functionality. - Write crafting final report.
Week 11		
Week12		
Phase 4: Testing		
Week 13	Test cases & Quality Assurance	- Different testing techniques are employed to ensure the software meets the requirements. - Provide test case for each function. - Debugging and quality assurance. - Write final report.
Phase 5: Deployment & Maintenance		
Week 14	Deployment & Maintenance	- Launch the book appointment website to make it live. - Ensure continuous performance monitoring. - Address any issues that come up promptly. - Gather valuable feedback from stakeholders. - Submit final report, code, database, powerpoint.
Week 15	Presentation & Demo	- Present all parts of the project to the lecturer. - Q&A and receive feedback for improvements.
Week 16		
Week 17		

2.1.2 Sprint backlog

(Work Breakdown Structure)

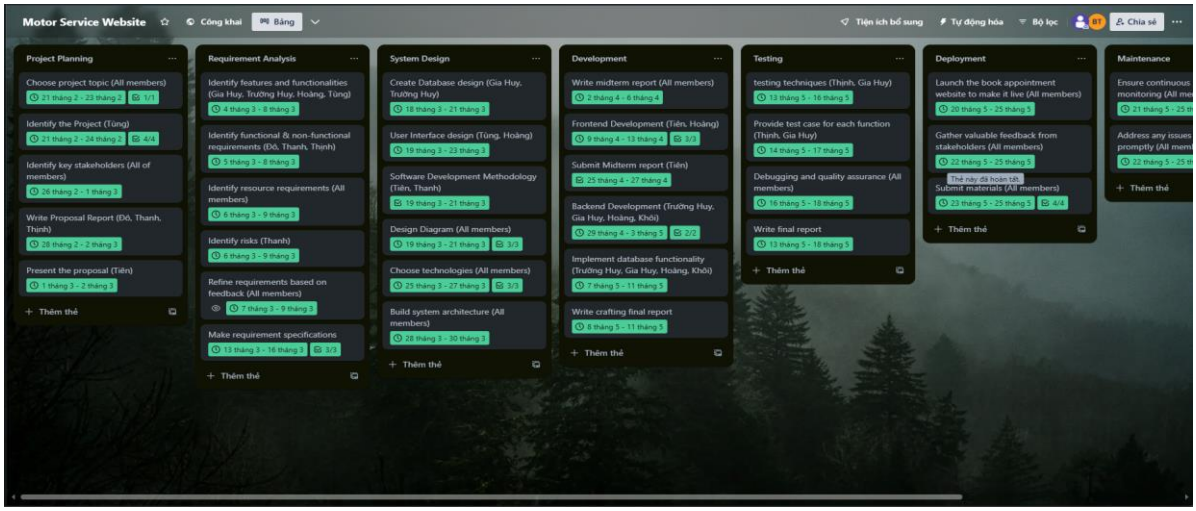

Figure 2. 1 Sprint backlog

2.2 In-charge Table

Table 2. 2 Roles and Responsibilities

Full name – Student name	Student ID	Roles
Trịnh Thuỷ Tiên	ITITIU21328	- Project manager - Front-end member - Midterm and Final report
Trần Bách Tùng	ITITIU21340	- Main Web Designer - Figma - Midterm and Final report
Đặng Nguyễn Trường Huy	ITITIU21010	- Back-end Leader - Midterm report
Nguyễn Hoàng Minh Khôi	ITITIU21229	- Back-end member - Midterm report
Bùi Phương Thanh	ITITIU21311	- Report Leader (Proposal ,Midterm and final report)
Nguyễn Trần Gia Huy	ITITIU21054	- Back-end member - Midterm and Final report
Vũ Văn Đô	ITITIU21125	- Report member (Proposal, Midterm and Final report)
Bạch Huy Hoàng	ITITWE21113	- Front-end Leader - Back-end member - Midterm and Final report
Nguyễn Trần Quốc Thịnh	ITITIU20312	- Report member (Proposal, Midterm and Final report)

CHAPTER III

METHODOLOGY

3.1 Overview

The methodology for developing a web portal for motor servicing at home begins with a thorough user requirement analysis. This phase involves identifying both functional requirements and non-functional requirements. A resource register is compiled to list necessary personnel, tools, and budgetary needs, while a risk register identifies potential technical, operational, and market risks.

Following this, requirement specifications are created using use case diagrams and detailed use case descriptions. These diagrams outline key interactions between users, technicians, and the system.

The system design phase with the creation of an Entity-Relationship Diagram (ERD) defines the database structure by illustrating entities such as User, Technician, Service, Booking, Payment, and Feedback, and their relationships. The Class Diagram details the system's object-oriented design, specifying classes, attributes, and methods. Sequence Diagrams map out the flow of operations for scenarios like service booking, from user action to system response.

Database design is then elaborated, creating tables for Users, Technicians, Services, Bookings, Payments, and Feedbacks, each with relevant fields. Finally, User Interface Design (UID) is conducted, producing wireframes and prototypes for key screens to ensure an intuitive and responsive user experience across devices. This comprehensive methodology ensures a well-structured, user-centric web portal that meets both business objectives and user needs efficiently.

3.2 User requirement analysis

3.2.1 Requirement Register

Table 3. 1 Functional and Non-functional requirements

Req.ID	Requirement Name	Detailed Description	Type
001	User Registration and Login	Users can create accounts using email or social media. Secure login with password recovery options. If the user doesn't have an account, then the user will be asked to register.	Functional requirement
002	Service Catalog	Display a comprehensive list of services (oil changes, brake repairs, battery replacements, etc). Detailed descriptions and pricing for each service.	Functional requirement
003	Appointment Scheduling	Users can choose time slots for the selected services.	Functional requirement
004	Service Cart	Customers can select and add desired services to a cart. Option to view and modify the cart before checkout.	Functional requirement
005	Order Confirmation	Users receive order confirmation and details via email/SMS.	Functional requirement
006	Real-Time Tracking	Users receive updates on the status of their service requests.	Functional requirement
007	Payment Integration	Support for multiple payment methods (credit/debit cards, digital wallets, bank transfers). Secure and encrypted payment gateway.	Functional requirement
008	Technician Management	Automated assignment and dispatch of the nearest available technician.	Functional requirement
009	Feedback System	Users can rate and review services and technicians. Option to provide comments and suggestions.	Functional requirement
010	Customer Support	In-app chat for customer support. FAQs and support documentation.	Functional requirement

Req.ID	Requirement Name	Detailed Description	Type
011	Scalability	System must support a growing number of users and service requests.	Non-functional requirement
012	Performance	Fast loading times and responsive interactions. Efficient processing of bookings and payments.	Non-functional requirement
013	Security	Secure handling and storage of user data and payment information. Use of encryption for all sensitive data transmissions. Compliance with data protection regulations (e.g., GDPR).	Non-functional requirement
014	Usability	Intuitive and easy-to-navigate interface for both web and mobile platforms. The interface is easy to learn and navigate; buttons, headings, and help/error messages are simple to understand	Non-functional requirement
015	Reliability	High availability with minimal downtime. Robust error handling and recovery mechanisms.	Non-functional requirement
016	Maintainability	Modular code structure for easy updates and maintenance. Comprehensive documentation for developers and users.	Non-functional requirement
017	Environmental Responsibility	Promote digital transactions to reduce paper usage. Encourage eco-friendly practices during vehicle servicing.	Non-functional requirement

3.2.2 Resource Register

Table 3. 2 Hardware resource

Items	Quantity	Description	Purpose	Price	Total Cost	Date needed
Dell Inspiron 5515	1	Mobile PC	Programming & Writing report	19.500.000 VND	19.500.000 VND	25/03/2024
Acer Aspire 7	1	Mobile PC	Backend Programming & Writing Report	21.000.000 VND	21.000.000 VND	28/04/2024
Asus Vivobook	1	Mobile PC	Backend Programming & Writing Report	20.000.000 VND	20.000.000 VND	28/04/2024
Dell Vostro 3400	1	Mobile PC	Writing Report	18.000.000 VND	18.000.000 VND	25/03/2024
Total Cost: 78.500.000 VND						

Table 3. 3 Software resource

Application	Quantity	Description	Purpose	Price	Total Cost	Date needed
Visual Studio Code	9	Code Editor	Programming Environment	Free	Free	25/03/2024
MongoDB	1	NoSQL Database as a service	Project Database	Free	Free	28/04/2024
Node.js	1	JavaScript runtime environment	Backend Programming	Free	Free	28/04/2024
Vue.js	1	Frontend JavaScript framework	Frontend Programming	Free	Free	25/03/2024
GitHub	1	Software Management Platform	Project storage & management	Free	Free	25/03/2024

3.2.3 Risk register

Table 3. 4 Risk Register

Risk ID	Risk Category	Risk Title	Affect	Probability	Impact	Risk Response Plan
001	HR	Team Cooperation Issues	Can slow project progress and produce unreliable results from uncooperative work.	Medium	Low	Organize team-building activities and resolve conflicts.
002		Key Personnel Loss	Loss of critical team members can disrupt project continuity.	Low	High	Cross-train team members and have a succession plan in place.
003	Requirements	Addition of New Requirements	Affects project progress and can lead to scope creep.	High	Medium	Hold a meeting to update the schedule and plan accordingly.
004	Technology	Technical Challenges	Unanticipated technical issues can delay the project and increase costs.	Medium	High	Conduct thorough technical risk assessments and have backup plans.
005	Schedule	Delayed Task Completion	Delays in individual tasks can cause the overall project timeline to slip.	Medium	High	Implement strict timeline monitoring and milestone tracking.

Risk ID	Risk Category	Risk Title	Affect	Probability	Impact	Risk Response Plan
006	Quality	Inadequate Testing	Poor testing can lead to a buggy and unreliable product.	Medium	High	Allocate sufficient time for thorough testing and quality assurance.
007		Poor User Experience Design	Can result in low user adoption and satisfaction.	Medium	Medium	Involve end-users in the design process and conduct usability testing.
008	Scope	Undefined Project Scope	Can lead to misaligned objectives and wasted resources.	Medium	High	Clearly define project scope and obtain stakeholder agreement.

3.3 Requirement Specifications

3.3.1 Use Case Diagram

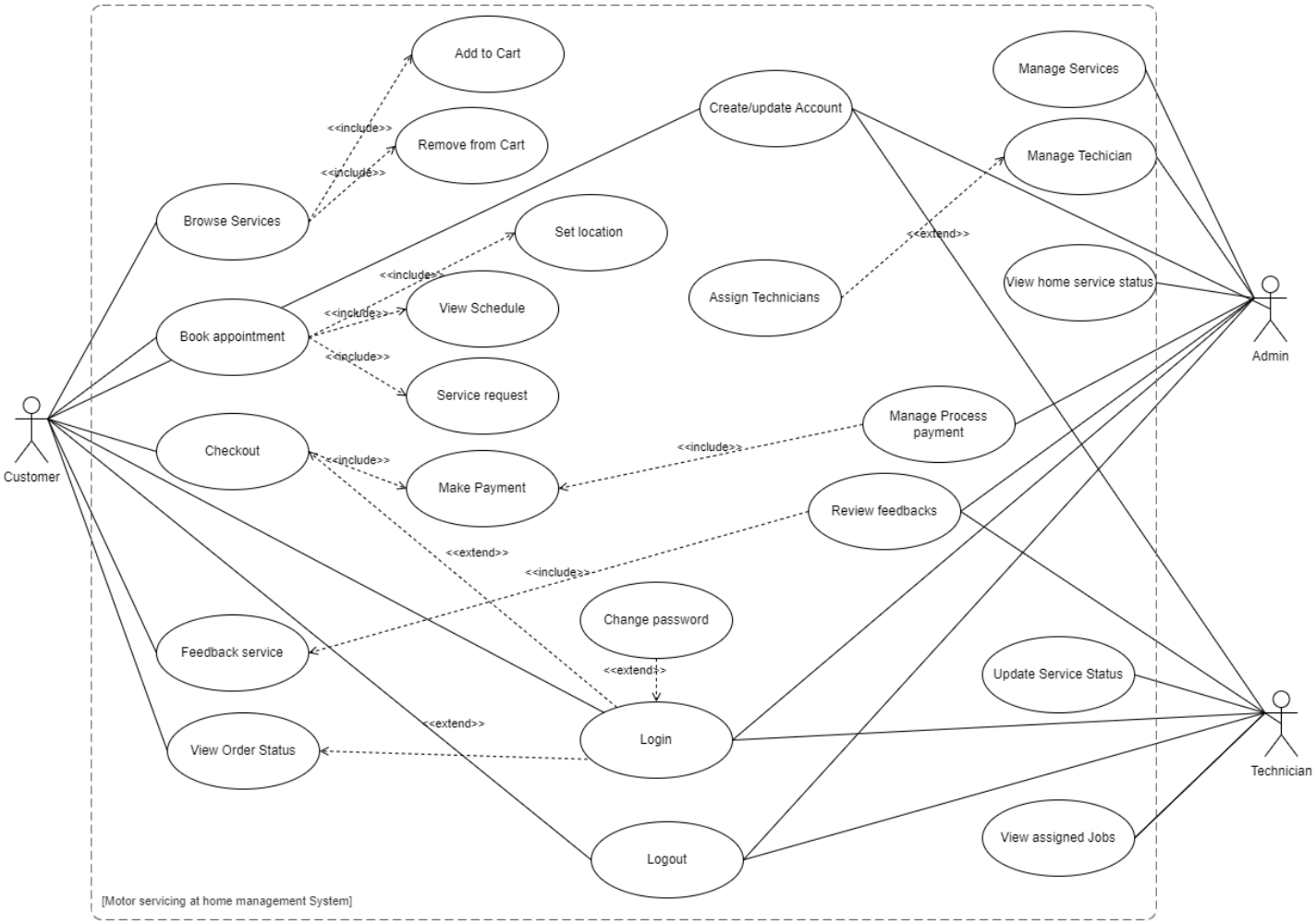

Figure 2. 2 Use Case Diagram

3.3.2 Use Case Descriptions

01. Use Case 1: Create/Update Account

Name: Create/Update Account

Identifier: UC1

Inputs:
1. Personal Information (Name, Address, Phone Number, etc.)
2. Account Information (Username, Password, Email)

Outputs:
1. Confirmation message [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 5 UC1 Main Flow

Actor User (Customer/Admin/Technician)	System
1. Access the account creation/update page.	
2. Enter the required personal and account information	2. Validate the information
3. Submit	3. Save the information and confirm [if successful]

Preconditions:
1. The user is not logged in (for account creation).
2. The user is logged in (for account update).

Postconditions:
1. The user's account is created or updated in the system.

02. Use Case 2: Browse Services

Name: Browse Services

Identifier: UC2

Inputs:

1. Customer's search/filter criteria (optional).

Outputs:

1. List of services matching the criteria.
2. Service details (Name, Description, Price, etc.).

Main Flow:

Table 3. 6 UC2 Main Flow

Actor User (Customer)	System
1. Navigate to the services page.	1. Display the services page.
2. Add or Remove from services page to cart	
3. View details of the selected services	3. Display a list of services matching the criteria.

Preconditions:

1. The customer is on the services page.

Postconditions:

1. The customer has viewed the available services.

03. Use Case 3: Book Appointment

Name: Book Appointment

Identifier: UC3

Inputs:

1. Selected service.
2. Preferred date and time.
3. Location.

Outputs:

1. Appointment confirmation. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 7 UC3 Main Flow

Actor User (Customer)	System
1. Select a or many services and navigate to the booking page.	
2. Enter the preferred date, time, and location.	
3. Submit	3.1 Check availability. 3.2 Confirm the booking and provide details.

Preconditions:

1. The customer is on the services page.

Postconditions:

1. The customer has viewed the available services.

04. Use Case 4: Checkout

Name: Checkout

Identifier: UC4

Inputs:

1. Payment Information
2. Billing Address.

Outputs:

1. Payment confirmation. [If successful]
2. Receipt.
3. Error message. [If fail]

Main Flow:

Table 3. 8 UC4 Main Flow

Actor User (Customer)	System
1. Review the booked services and total cost.	1. Display a process the payment
2. Enter payment and billing information.	2. Confirm the payment and generate a receipt.
3. Payment	

Preconditions:

1. The customer has booked one or many services and is ready to pay.

Postconditions:

1. The payment is processed, and the booking is confirmed.

05. Use Case 5: Feedback Service

Name: Feedback Service

Identifier: UC5

Inputs:

1. Service ID
2. Feedback text
3. Rating

Outputs:

1. Confirmation message [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 9 UC5 Main Flow

Actor User (Customer)	System
1. Navigate to the feedback page	
2. Select the service to provide feedback on	
3. Enter their feedback and rating	3. Save the feedback
4. Submit	4. Confirm the submission

Preconditions:

1. The customer has completed a service.

Postconditions:

1. The feedback is recorded in the system.

06. Use Case 6: View Order Status

Name: View Order Status

Identifier: UC6

Inputs:

1. Order ID (optional, for searching specific orders)

Outputs:

1. List of orders with their status
2. Detailed view of a selected order

Main Flow:

Table 3. 10 UC6 Main Flow

Actor User (Customer)	System
1. Navigate to the order status page	1. Display the order status page
2. View the list of their orders and their current status.	
3. Enter their feedback and rating	
4. Click on an order to view detailed information.	

Preconditions:

1. The customer has completed a service.

Postconditions:

1. The feedback is recorded in the system.

07. Use Case 7: Login

Name: Login

Identifier: UC7

Inputs:

1. Username
2. Password

Outputs:

1. The home page with the user's authorization [If successful]
2. The login page. [If fail]

Main Flow:

Table 3. 11 UC7 Main Flow

Actor User (Customer/Admin/Technician)	System
1. Open the login page	1. Display the order status page
2. Enter username and password	
3. Submit	3.1 Check the user's info 3.2 If success, return the home page 3.3 Else return the login page

Preconditions:

1. The user has a registered account.

Postconditions:

1. The user is logged into the system.

08. Use Case 8: Logout

Name: Logout

Identifier: UC8

Inputs:

1. None

Outputs:

1. Come back to the login page.

Main Flow:

Table 3. 12 UC8 Main Flow

Actor User (Customer/Admin/Technician)	System
1. Click the logout button	1. End the user's session 1.1 Redirect to the login page

Preconditions:

1. The user is logged in.

Postconditions:

1. The user is logged out of the system.

09. Use Case 9: Change Password

Name: Change Password

Identifier: UC9

Inputs:

1. Current password
2. New password
3. Confirm new password

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 13 UC9 Main Flow

Actor User (Customer/Admin/Technician)	System
1. Navigate to the change password page	
2. Enter the current password, new password, and confirms the new password	
3. Submit	3. Validate the current password and check if the new passwords match. 3.1 If valid, the system updates the password and confirms the change. 3.2 If invalid, error message.

Preconditions:

1. The user is logged in.

Postconditions:

1. The user's password is updated in the system.

10. Use Case 10: Set Location

Name: Set Location

Identifier: UC10

Inputs:

1. Address details of customer

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 14 UC10 Main Flow

Actor User (Customer)	System
1. Navigate to the set appointment page	1. Validate the address.
2. Enter customer’s address details.	2. Update the customer's location and confirm the update.
3. Submit	

Preconditions:

1. The customer chooses service and fills in information.

Postconditions:

1. The customer's location is updated in the system.

11. Use Case 11: View Schedule

Name: View Schedule

Identifier: UC11

Inputs:

1. Customer ID

Outputs:

1. List of scheduled services with details (Date, Time, Service, Technician)

Main Flow:

Table 3. 15 UC11 Main Flow

Actor User (Customer)	System
1. Navigate to the set appointment page	
2. View the details of each scheduled service.	2. Retrieve and display the list of scheduled services.
3. Submit	

Preconditions:

1. The customer chooses service and fills in information.

Postconditions:

1. The customer views your upcoming service schedule.

12. Use Case 12: Service Request

Name: Service Request

Identifier: UC12

Inputs:

1. Service details (Service type, Preferred date and time, Location)

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 16 UC12 Main Flow

Actor User (Customer)	System
1. Navigate to the set appointment page	1. Validate the request details.
2. Enter the request service details.	
3. Submit	3. Confirm the request

Preconditions:

1. The customer chooses service and fills in information.

Postconditions:

1. The service request is saved and scheduled in the system.

13. Use Case 13: Manage Services

Name: Manage Services

Identifier: UC13

Inputs:

1. Service details (Name, Description, Price, etc.)

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 17 UC13 Main Flow

Actor User (Customer)	System
1. Navigate to the manage services page	
2. Confirm user's services selected.	2. Validate and apply the change.
3. Submit	3. Confirm the request

Preconditions:

1. The admin is logged in.

Postconditions:

1. The service information is updated in the system.

14. Use Case 14: Manage Technicians

Name: Manage Technicians

Identifier: UC14

Inputs:

1. Technician details (Name, Contact Info, Skills, etc.)

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 18 UC14 Main Flow

Actor User (Customer)	System
1. Navigate to the manage technicians page	
2. Confirm the user's technician selected. 2.1 Enter or modify the technician details.	2. Validate and apply the change.
3. Submit	3. Confirm the request

Preconditions:

1. The admin is logged in.

Postconditions:

1. The technician information is updated in the system.

15. Use Case 15: Assign Technicians

Name: Assign Technicians

Identifier: UC15

Inputs:

1. Service request ID
2. Technician ID

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 19 UC15 Main Flow

Actor User (Customer)	System
1. Navigate to the service requests page.	
2. Select a service request and assign a technician following the user's request. 2. 1 If a technician busy, add technician other.	2. Update the service request with the assigned technician
3. Submit	3. Confirm the assignment

Preconditions:

1. The admin is logged in.
2. There are pending service requests.

Postconditions:

1. The service request is updated with the assigned technician.

16. Use Case 16: Manage Process Payment

Name: Manage Process Payment

Identifier: UC16

Inputs:

1. Payment details (Customer ID, Amount, Payment method)

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 20 UC16 Main Flow

Actor User (Customer)	System
1. Navigate to the payment processing page.	
2. Enter the payment details	2. Validate the payment details and process the payment.
3. Submit	3. Confirm the payment and update the order status

Preconditions:

1. The admin is logged in.

Postconditions:

1. The payment is processed, and the order status is updated in the system.

17. Use Case 17: View Assigned Jobs

Name: View Assigned Jobs

Identifier: UC17

Inputs:

1. Technician ID (implicit through login)

Outputs:

1. List of assigned jobs with details

Main Flow:

Table 3. 21 UC17 Main Flow

Actor User (Tẹchnician)	System
1. Navigate to the assigned jobs page.	
2. View the details of each job.	2. Display a list of jobs assigned to the technician.
3. Confirm	3. Update for status of user.

Preconditions:

1. The technician is logged in.

Postconditions:

1. The technician views the jobs assigned to them.

18. Use Case 18: Update Service Status

Name: Update Service Status

Identifier: UC18

Inputs:

1. Service request ID
2. New status (e.g., In Progress, Completed)

Outputs:

1. Confirmation message. [If successful]
2. Error message. [If fail]

Main Flow:

Table 3. 22 UC18 Main Flow

Actor User (Technician)	System
1. Navigate to the assigned jobs page.	
2. Select a service request and Update its status.	2. Display a list of jobs updated.
3. Submit	3. Update the status and confirm the update.

Preconditions:

1. The technician is logged in.
2. The technician has an assigned service request.

Postconditions:

1. The service request status is updated in the system.

19. Use Case 19: Review Feedback

Name: Review Feedback

Identifier: UC19

Inputs:

1. Feedback details (Customer ID, Service ID, Feedback text, Rating)

Outputs:

1. List of feedback entries
2. Detailed view of selected feedback

Main Flow:

Table 3. 23 UC19 Main Flow

Actor User (Admin/Technician)	System
1. Navigate to the feedback review page.	
2. View the details of selected feedback.	2. Display a list of feedback entries.

Preconditions:

1. The admin or technician is logged in.

Postconditions:

1. The service request status is updated in the system.

3.4 System Design

3.4.1 Entity Relationship Diagram

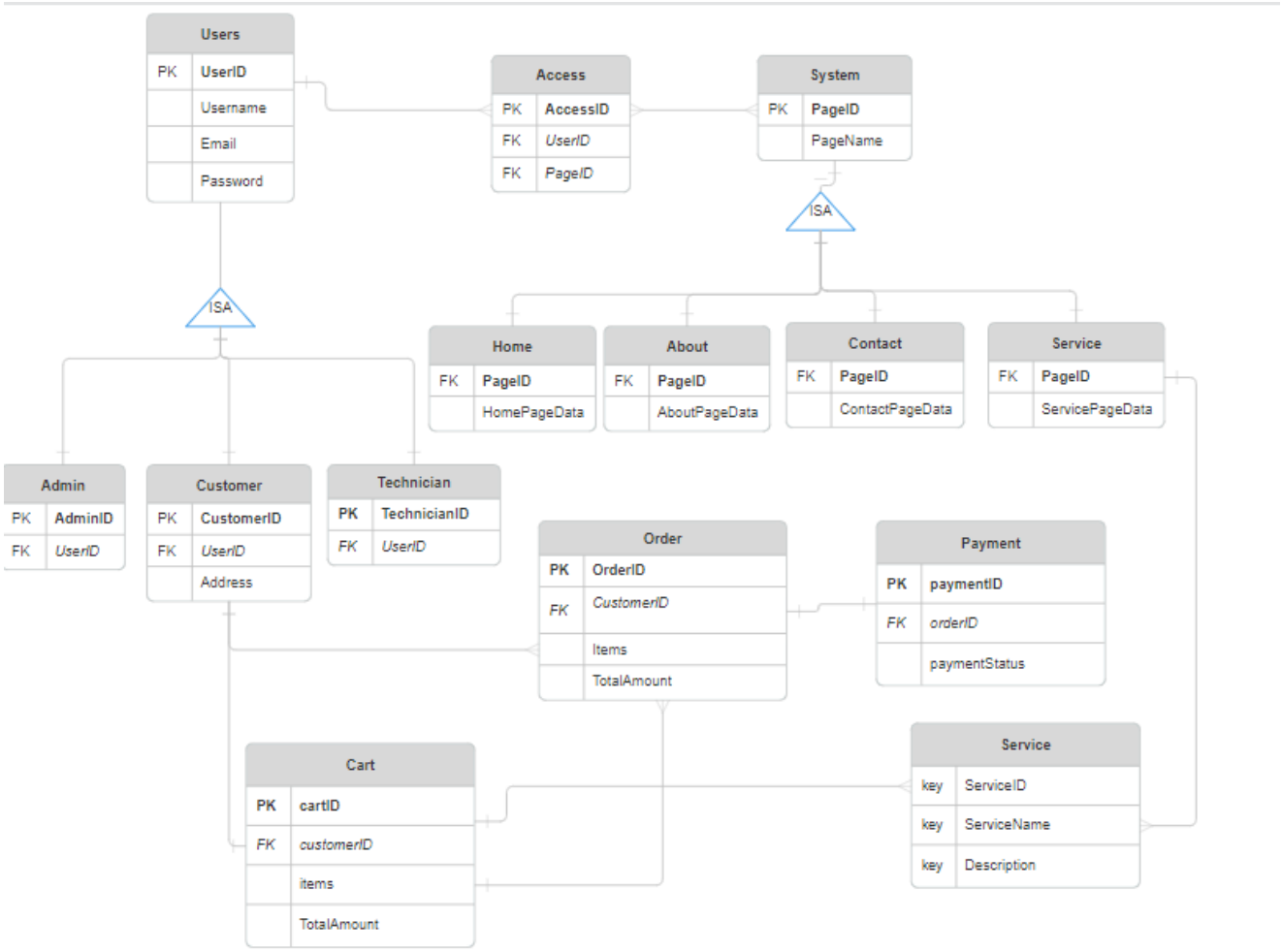

Figure 3. 1 ER Diagram

a. Explanation:

+ A user will have attributes: UserID, FirstName, LastName, Username, Password, UserType.
+ There are 3 types of Users: Admin, Customer and Technician
+ Admin will have AdminID to identify.
+ Customer will have CustomerID.
+ There are 4 types of Page: Home, About, Contact, Service, each with their own data.
+ An access entity with UserID and PageID will help the user access the pages.
+ Cart will have cart ID to identify, along with Items (list datatype) and TotalAmount (integer).
+ Order will have order ID to identify, as well as items and TotalAmount attribute.
+ Payment will have payment ID, order ID and payment status (to show whether the payment is successful or not).
+ Service will have ServiceID (to identify each service) and ServiceName and Description.

b. Relationships:

+ Admin has a one-to-one relationship with User (Each admin corresponds to exactly one user).
+ Customer has a one-to-one relationship with User (Each customer corresponds to exactly one user).
+ Technician has a one-to-one relationship with User (Each admin corresponds to exactly one user).
+ Users have a one-to-many relationship with Access (One user can have access to multiple pages).
+ Customer has one-to-one relationship with cart (each customer only have 1 cart).
+ Customer has many-to-one relationship with order (each customer can have many orders).
+ Cart has one-to-many relationship with Order (each cart contains many order).
+ Payment has one-to-one relationship with Order (each order means 1 payment)
+ System has a many-to-many relationship with User through Access (A page can be accessed by multiple users, and a user can access multiple pages).
+ Each specific page (Home, About, Contact, Service) is related to the System entity.

3.4.2 Class Diagram

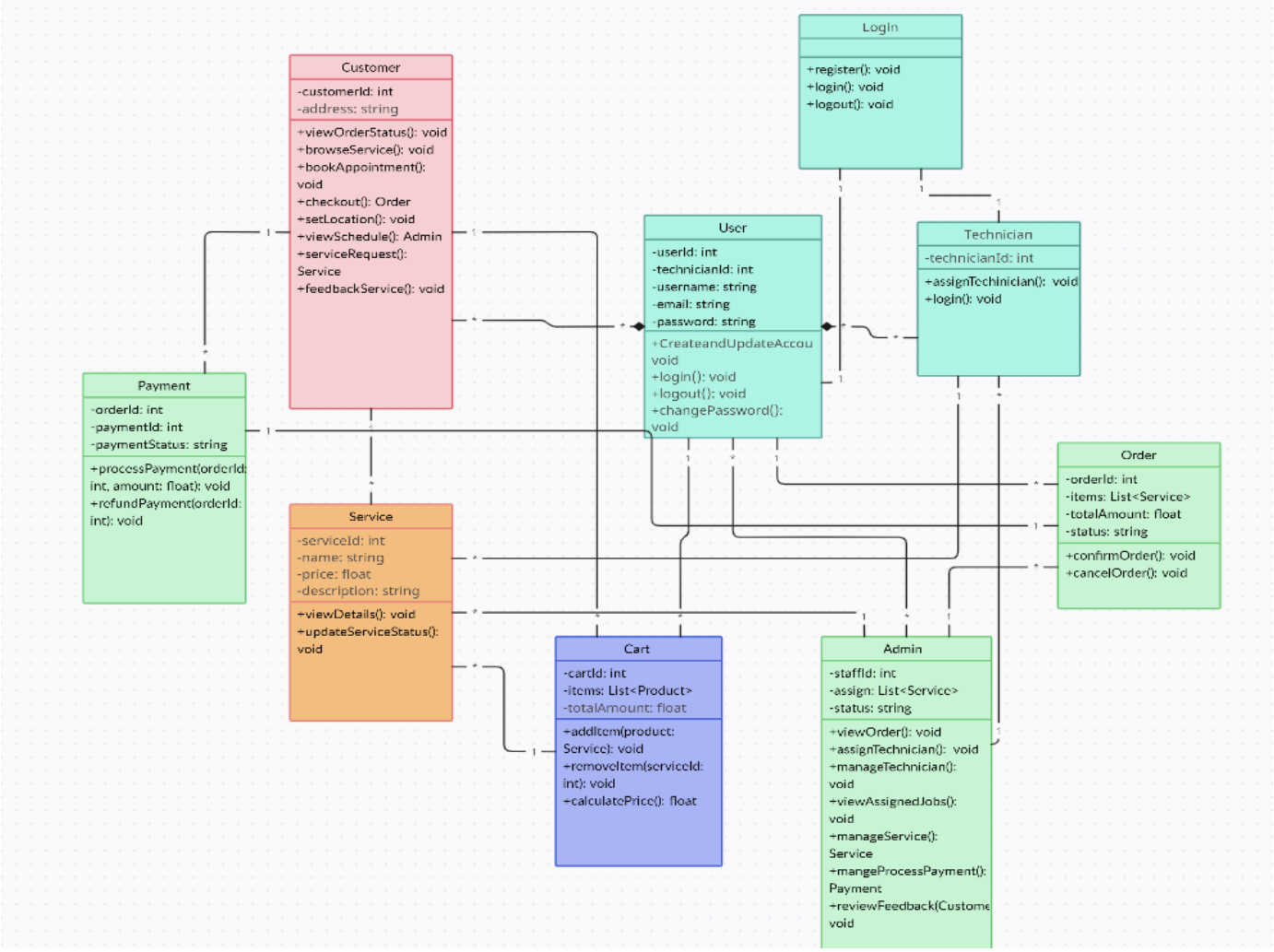

Figure 3. 2 Class Diagram

a. Explanation of Relationships:

+ User (Generalization): Customer, and Staff (Technician and Admin) inherit from User.
+ Staff and Customer: Everyone must log in to become a Customer or Staff (Technician - Admin).
+ Customer and Order (Association): Customers place Orders.
+ Order and Payment (Association): Each Order is linked to a Payment.
+ Order and Services (Composition): Orders consist of multiple Services.
+ Customer and Cart (Aggregation): Each Customer has one Cart containing Services.
+ Admin and Order (Association): Staff can view and be assigned to Orders.

3.4.3 Sequence Diagram

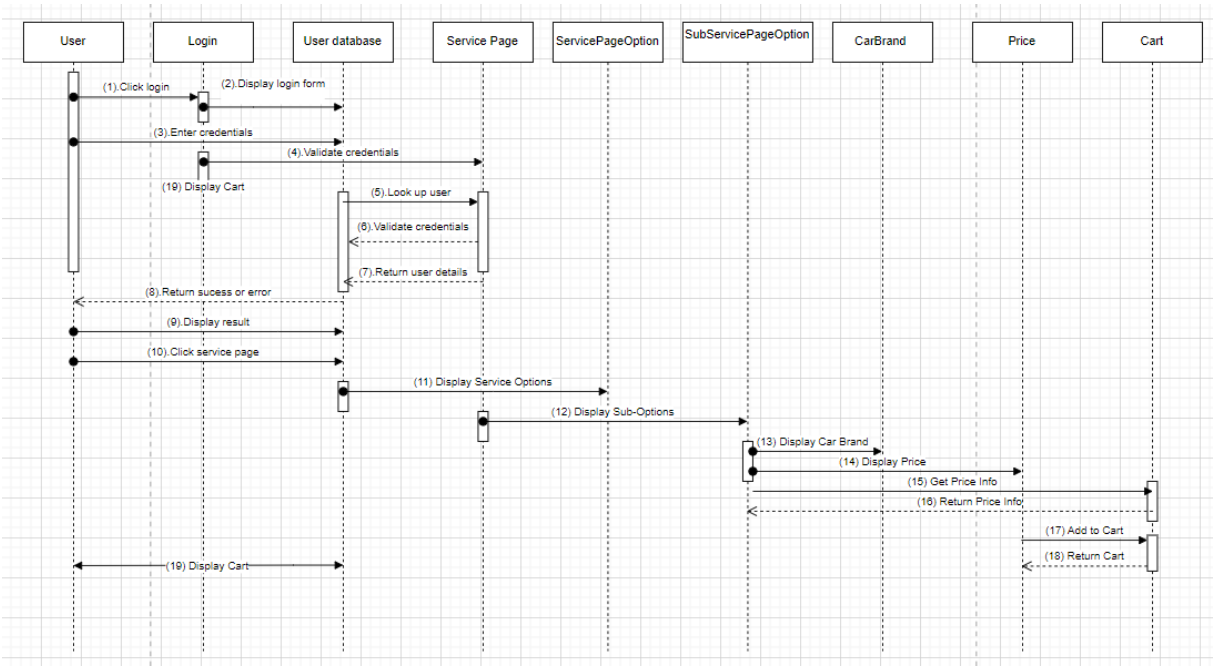

Figure 3. 3 Sequence Diagram

a. Explanation:

+ User clicks on the Login button: The User initiates the login process by clicking on the Login button on the Login Page.
+ Login Page displays the login form: The Login Page responds by displaying the login form where the user can enter their credentials.
+ User enters their credentials (username and password): The User enters their username and password into the login form.
+ Login Page validates the credentials: The Login Page sends the entered credentials to the User Database for validation.
+ User Database looks up the user: The User Database searches for a user with the provided username.
+ User Database validates the credentials: The User Database checks if the provided password matches the stored password for the user.
+ User Database returns user details: If the credentials are valid, the User Database returns the user details to the Login Page.
+ Login Page returns success or error: The Login Page processes the response and returns a success or error message to the user.
+ Login Page displays the result: The Login Page displays the result of the login attempt to the user, indicating whether the login was successful or not.
+ User clicks on the Service Page: The User navigates to the Service Page.
+ Service Page displays service options: The Service Page responds by displaying the available service options (All services, Car services, Tire services).
+ User selects a specific service option (e.g., Car services): The User selects the Car services option.
+ Service Page displays sub-options for the selected service (e.g., Brake): The Service Page responds by displaying the sub-options for Car services (Brake, Oil Change, etc.).
+ Service Page displays car brands for the selected sub-service: The Service Page displays the available car brands for the Brake service.
+ User selects a car brand (e.g., Toyota): The User selects Toyota as the car brand.
+ Service Page retrieves and displays the price for the selected service and car brand: The Service Page fetches and displays the price for the Brake service for a Toyota car.
+ User adds the service to the cart: The User adds the Brake service for Toyota to their cart.
+ Cart displays the updated cart with the added service: The Cart displays the updated cart with the added service.

This sequence diagram provides a comprehensive view of the interactions between the user and the system for both logging in and using a service from the service page.

3.4.4 Database design

a. Overview of Database for Customer Login and Authentication:
   - This section provides a detailed explanation of the account database functionality for customer login and authentication in the provided system. The system is built using Node.js, Express, Mongoose, and bcryptjs for password hashing:
     - The User.js file defines the Mongoose schema and model for the User. This schema includes fields for username, password, and role, along with timestamps for record-keeping.
     - The server.js file sets up the Express server, connects to the MongoDB database, and includes middleware for JSON parsing and CORS. It also integrates the authentication routes defined in auth.js.
     - The auth.js file handles user registration and login. It uses bcryptjs to hash passwords and jsonwebtoken for generating JWT tokens.
     - The axios.js file configures an axios instance with a base URL for making API requests to the server.
   - Functionality Overview:
     - User Registration:
       - Users can register by providing a username, password, and role. The password must be confirmed.
       - Passwords are hashed using bcryptjs before being stored in the database.
       - A new user is created in the MongoDB database if the username does not already exist.
     - User Login:
       - Users can log in using their username and password.
       - The system verifies the credentials by comparing the hashed password with the stored hash.
       - If the credentials are valid, a JWT token is generated and returned to the user.
     - Database Integration:
       - The application connects to a MongoDB database using Mongoose.
       - User information is stored and retrieved from the MongoDB database.
       - Express middleware is used for JSON parsing and handling CORS.

Field Name	Data Type	Description
`_id`	ObjectId	Unique identifier for the account
`username`	String	Username of the account holder
`fullname`	String	Full name of the account holder
`email`	String	Email address of the account holder
`password`	String	Hashed password of the account holder
`isAdmin`	Boolean	Indicates if the account holder is an admin
`phoneNumber`	Number (Integer)	Phone number of the account holder
`dateOfBirth`	Date	Date of birth of the account holder
`createdAt`	Date	Timestamp when the account was created
`updatedAt`	Date	Timestamp when the account was last updated
`__v`	Number (Integer)	Version key for internal use by Mongoose

Figure 3. 4 Sample of our database

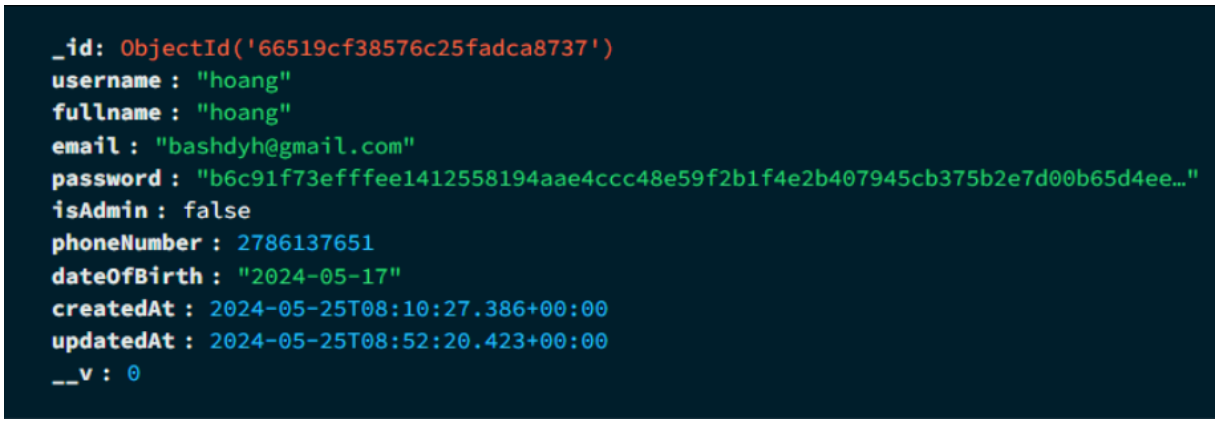

Figure 3. 5 Sample of account main

b. Overview of Services Database:
   - The services database is structured to provide detailed information about various vehicle maintenance services, including descriptions, categories, ratings, and pricing for different brands. Each service entry includes a default image path to visually represent the service offered.
   - Data Structure:
     - id: A unique identifier for the service (not present for all services).
     - label: The name of the service.
     - description: A detailed description of the service.
     - category: The category of the service (e.g., car, tire).
     - rating: The customer rating for the service (e.g., 5).
     - prices: An array of objects, each containing the brand and the corresponding price for that service.
     - image: The path to the default image representing the service.

```
const defaultImagePath = '/src/assets/img/brake.jpg'
servicesData.forEach((service) => {
  service.image = defaultImagePath
})

export default servicesData
```

Figure 3. 6 Path of image representing the service

- A default image path is provided for each service to visually represent the service offered and the service data is exported for use in other parts of the application

➢ This database structure allows customers to easily compare prices across different brands for each service, ensuring they can make informed decisions based on their specific vehicle needs.

3.4.5 User Interface Design

a. Home Page:

- This section will get to what you see first when you download and log in to our link. The starting page of our e-commerce Car Services as the entry point for users to access their accounts. This report outlines the design and functionality of the login page, implemented using VS Code and Vue.js library. Implementing a dark mode functionality enhances user experience by offering a visually comfortable browsing option, especially in low-light environments.

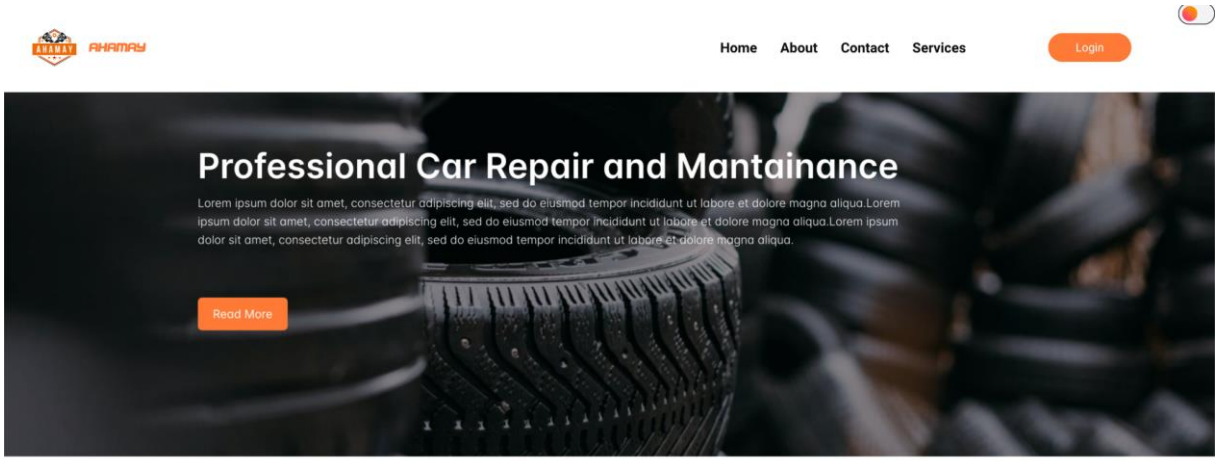

Figure 3. 7 Interface of Home page at light mode

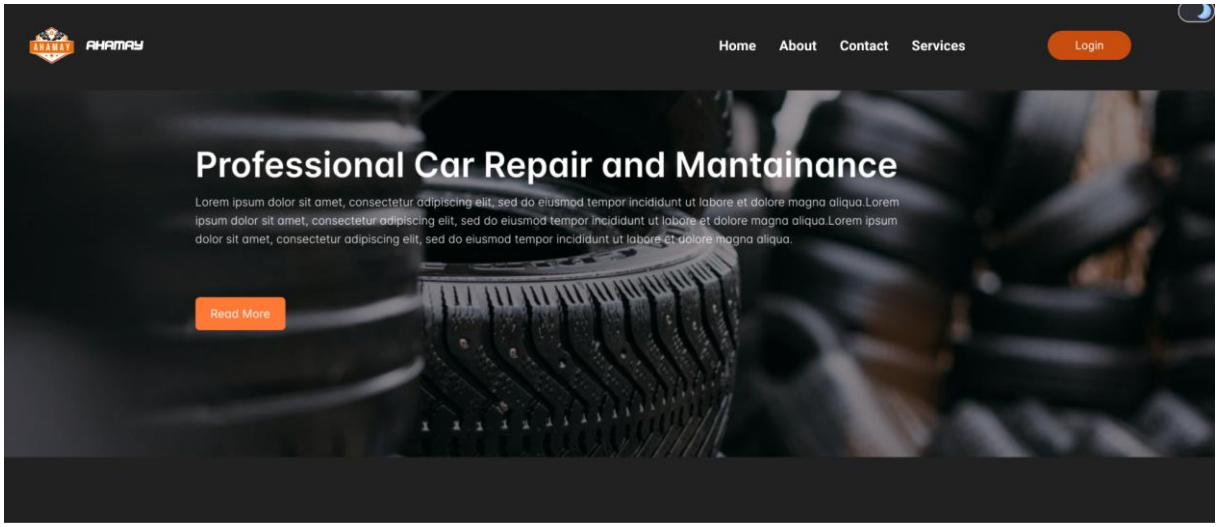

Figure 3. 8 Interface of Home page at dark mode

b. Login Interface:

- The login interface is a critical component of the starting page, allowing users to sign in with their respective credentials. The interface consists of two input fields:
  - Username: Users are required to enter their email address associated with their accounts.
  - Password: A secure password input field ensures the confidentiality of user credentials.
  - The login interface is designed for ease of use, providing a clear and concise layout for users to input their information.
  - Role-based Authentication:
  - The system supports two distinct roles: Admin and Customer. Upon entering their credentials and pressing the “Login” button, users are redirected to role-specific pages based on their account type.

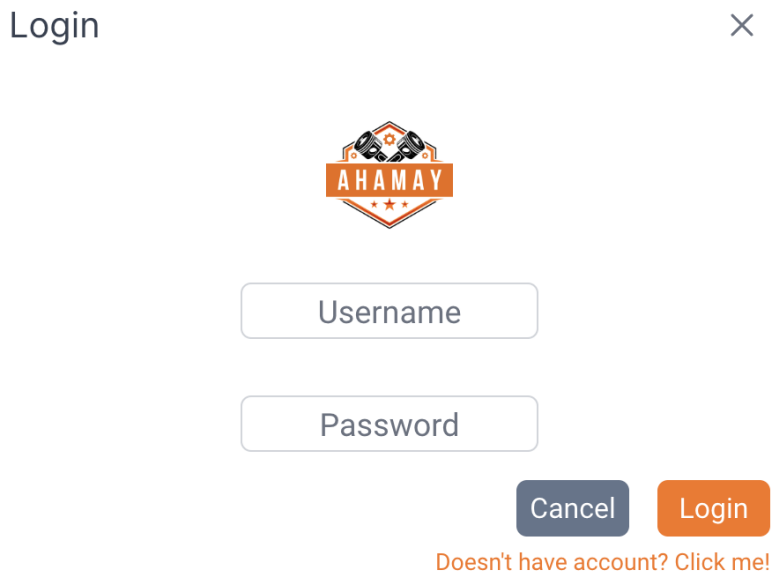

Figure 3. 9 Login Interface

c. About Us Page:

- The "About Us" page is a crucial component of the website. It serves multiple purposes, from building trust with potential customers to providing essential information about the company. Here are the detailed purpose of the page:
  - Build Trust: Provides transparency about the company’s mission, vision, history, and values, which helps in building trust with visitors.
  - Humanize the Brand: Showcases the people behind the company, making the brand more relatable and trustworthy.
  - SEO Benefits: Improves search engine rankings by including relevant keywords and providing fresh content.
  - Customer Engagement: Encourages customer engagement by sharing the company's story and philosophy.

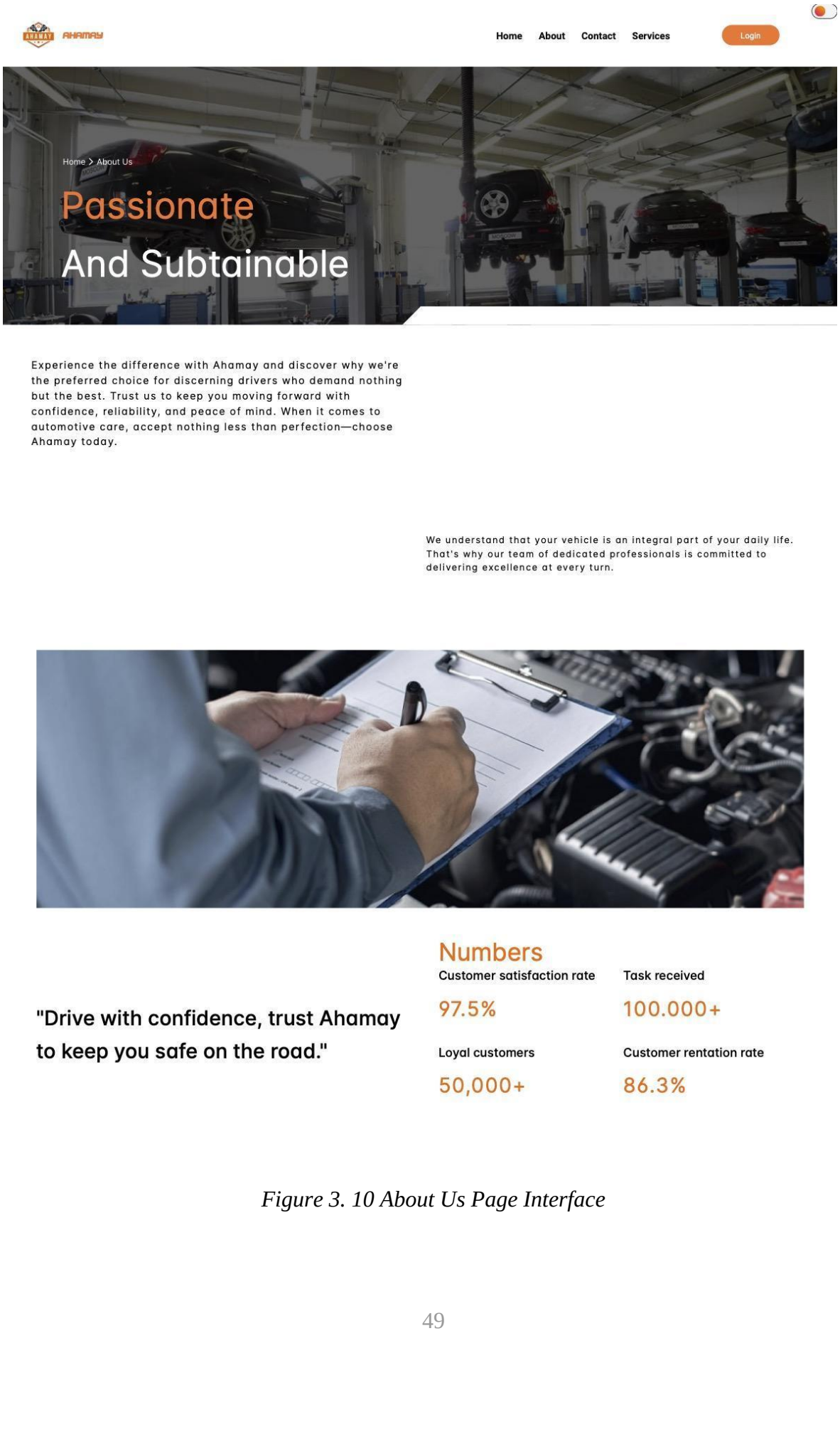

Figure 3. 10 About Us Page Interface

d. Services Page:

- The "Services" page on an e-commerce website is designed to provide customers with detailed information about the services offered, allow them to order these services directly, and schedule appointments for further advice or consultation. Here are the detailed purpose of the page:
  - Showcase Services: Provide detailed descriptions of the services offered, including features, benefits, and pricing.
  - Service Ordering: Enable customers to select and order services directly from the page.
  - Appointment Scheduling: Allow customers to set appointments for consultations or additional advice.
  - User Engagement: Engage users with easy-to-navigate service options and streamlined ordering processes.

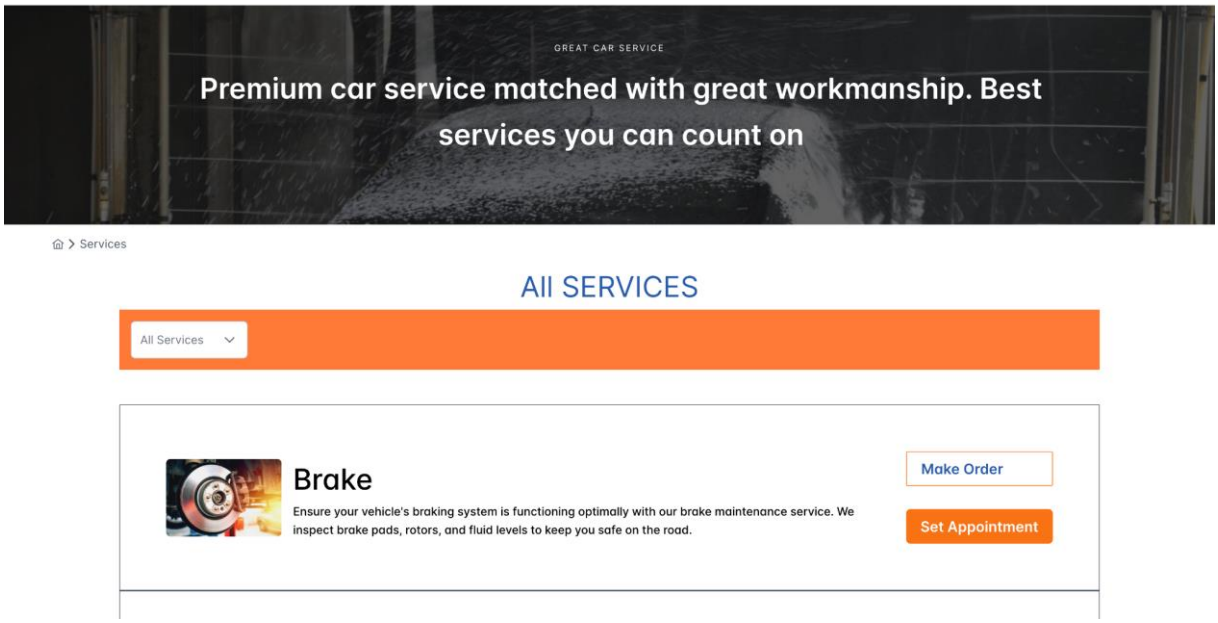

Figure 3. 11 Services Page Interface

e. Account Information Page:

- The Account Information page is a crucial part of an e-commerce website, providing users with access to their personal information and order history. It consists of two main components: "Account Details" and "My Orders," along with a logout button. Purpose of the page:
  - Account management: Allows users to view and update their personal information, including basic and contact details.
  - Order Management: Provides a list of orders added to the cart, with the ability to review, modify, and proceed to checkout.
  - User Authentication: Includes a logout button for users to securely end their sessions.
  - Enhanced User Experience: Streamlines the management of personal and order information, enhancing overall user satisfaction.

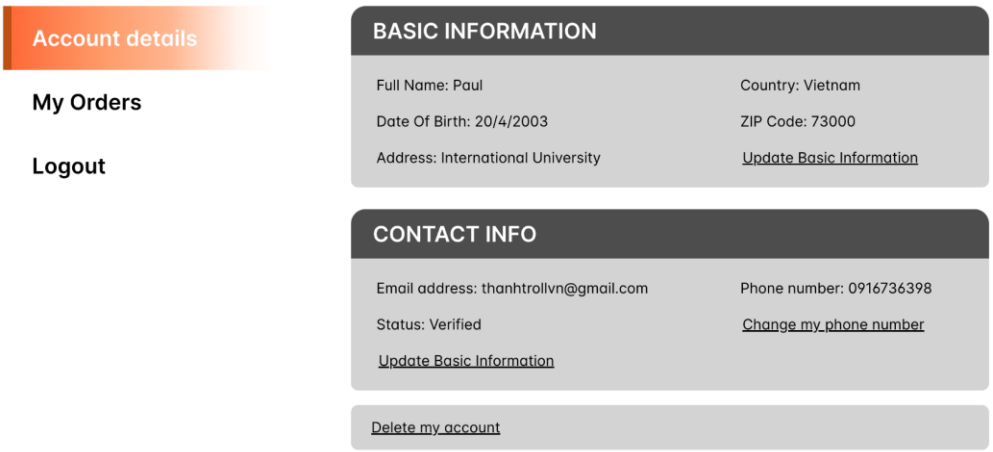

Figure 3. 12 Account details Page Interface

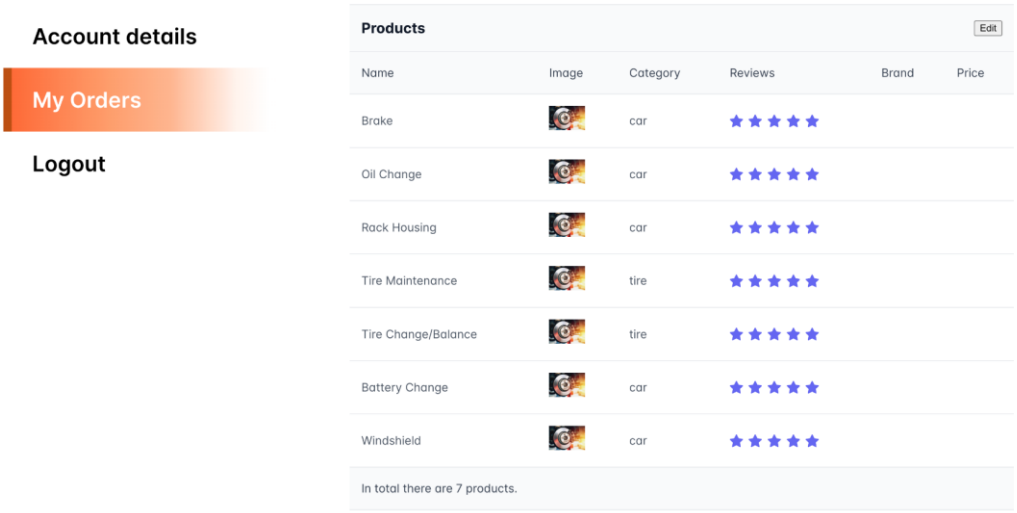

Figure 3. 13 My Orders Page Interface

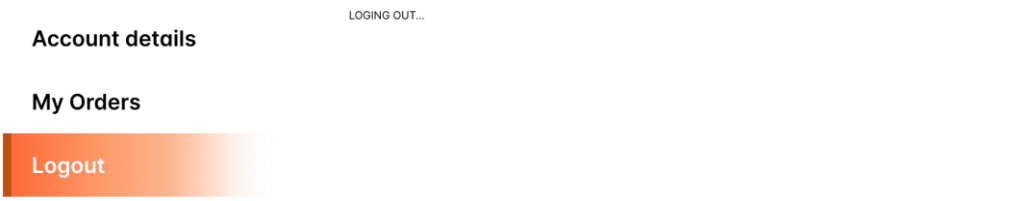

Figure 3. 14 Logout Interface

f. Checkout Page:

- The Checkout Page is a critical component of an e-commerce website, serving as the final step in the purchasing process where users review their orders, provide payment information, and confirm their purchases. Purpose of the page:
  - Order Review: Display a summary of items or services the user intends to purchase, including quantities, prices, and total cost.
  - User Information: Collect necessary shipping and billing information from the user.
  - Payment Processing: Provide a secure method for users to enter and process their payment details.
  - Order Confirmation: Confirm the order details and provide a final review before the user completes the purchase.
  - User Experience: Ensure a smooth, user-friendly process that minimizes friction and reduces cart abandonment rates.

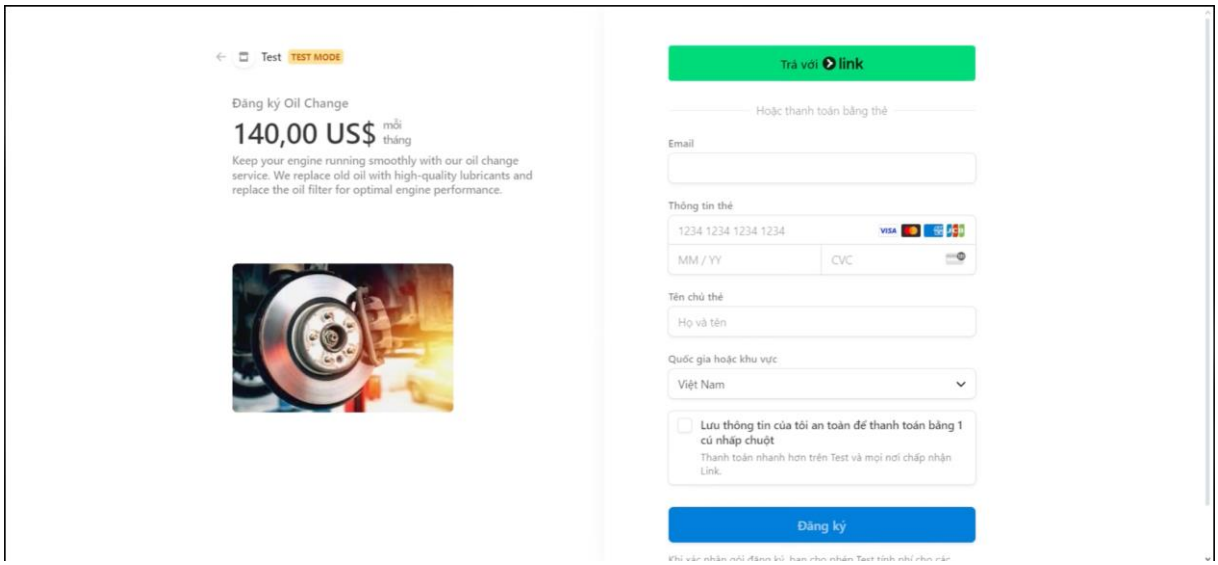

Figure 3. 15 Checkout Interface

CHAPTER IV

IMPLEMENT & RESULTS

4.1 Implement

4.1.1 Front-end

a. Technologies & Tools Used:

- Programming Languages: JavaScript, HTML, CSS
- Frameworks and Libraries:
  - Vue.js: For building the user interface.
  - Vue Router: For handling routing.
  - PrimeVue: For ready-to-use UI components.
  - PrimeIcon: For iconography.
  - TailwindCSS: For utility-first CSS.
  - PrimeFlex: For layout utilities.
- Build Tools:
  - Vite: For bundling and running the development server.
  - Linters: ESLint: For code quality and consistency.
- Software Used:
  - VSCode: code editor software that redefined and optimized for building and debugging modern web applications.
  - Figma: Collaborative application for interface design.

b. Algorithms & Models Used:

- Search and Filter Algorithms:
  - Faceted Search: Enabling users to filter services based on various criteria like service type, price, ratings, and location.
- User Behavior Analysis:
  - Web Analytics: Tracking user interactions on the website to understand navigation patterns and optimize the user experience.
  - Heatmaps: Visualizing areas of the website that receive the most attention to improve UI/UX design.
  - A/B Testing: Experimenting with different versions of the website to determine which design or feature works best.
- Logistics and Inventory Management:
  - Inventory Optimization Models: Ensuring the right balance of parts and materials are available.
  - Route Optimization Algorithms: For any services involving vehicle pick-up/drop-off, ensuring the most efficient routes are taken.
- User Interface Personalization:
  - Adaptive UI/UX Models: Changing the interface based on user behavior and preferences.
  - Responsive Design Algorithms: Ensuring the website works well across different devices and screen sizes.

4.2 Results of System's Architecture

4.2.1 Backend Overview

The implementation of the Doorstep Motor Repair website involved setting up the development environment with Node.js, Express, and MongoDB, and using Git for version control. API development included user authentication with JWT, service booking functionalities, service provider management, and payment integration with gateways like Stripe. MongoDB schemas were designed for users, services, bookings, and providers, with data validation handled by Mongoose. Middleware for authentication, authorization, and error handling was integrated, and testing included unit and integration tests using frameworks like Mocha or Jest. Continuous Integration/Continuous Deployment (CI/CD) pipelines were established using tools like GitHub Actions to automate testing and deployment, and the application was deployed to cloud platforms such as AWS, ensuring secure and scalable operations.

Several challenges were encountered during the development of the Doorstep Motor Repair website, which required careful consideration and strategic solutions:

- Scalability Issues: Managing increased load and avoiding performance bottlenecks.
- Data Consistency and Integrity: Ensuring accurate and consistent data during concurrent transactions.
- Security Concerns: Protecting sensitive user data and securing transactions.
- Integration with Third-Party Services: Seamless integration with payment gateways, geolocation, and notification services.
- Real-time Updates and Notifications: Providing real-time updates about booking status and service provider location.
- Managing Development and Deployment Environments: Keeping environments consistent and managing configurations.
- User Experience and Feedback: Meeting user expectations and improving the application based on feedback.

4.2.2 Frontend Overview

The implementation of the front-end for an e-commerce website using Vue.js and associated technologies involves careful planning, a good understanding of the tools, and efficient management of state and UI components. While challenges such as performance optimization and responsive design are inevitable, the use of modern frameworks and libraries like Vue.js, Vite, TailwindCSS, and PrimeVue can significantly streamline the development process and result in a robust and user-friendly e-commerce platform.

Challenges Encountered:

a. **Performance Optimization:** Ensuring the site loads quickly and efficiently was a significant challenge. This involved optimizing images, leveraging lazy loading for routes, and ensuring efficient state management.
b. **Responsive Design:** TailwindCSS and PrimeFlex made it easier to create a responsive design, but it required careful planning and testing across different devices and screen sizes.

c. **Component Management:** Managing a large number of components and ensuring they were reusable and maintainable was complex. This was addressed by adopting best practices in component design and Vue's composition API.
d. **Routing Complexity:** Implementing dynamic routes and nested routes for a seamless user experience required a deep understanding of Vue Router. Integration with Backend: Integrating with the backend APIs for features like user authentication, product management, and order processing required careful handling of asynchronous data fetching and state management.
e. **Linting and Code Standards:** Maintaining consistent code quality with ESLint required setting up proper rules and integrating it into the development workflow.

CHAPTER V

DISCUSSION, TESTING & EVALUATION

5.1 Discussion

Before diving into testing and evaluation we had a detailed discussion to outline the objectives, scope, requirements, and priorities of the project. This will ensure that the testing efforts are aligned with the overall goals of the web portal for motor servicing at home.

5.2 Testing methodologies employed

5.2.1 Unit tests

Table 5. 1 Unit test applications

Feature	Description
Service Booking Component	- Check if the booking function correctly adds a new booking entry. - Validate input fields for the booking form (e.g., date, time, service type).
User Authentication	- Check if the user login function validates credentials correctly. - Ensure the registration function handles user data properly and stores it securely.
Payment Processing	- Test the calculation of total cost based on selected services. - Verify that the payment gateway integration works correctly with mock responses.

5.2.2 Integration tests

Table 5. 2 Integration test applications

Feature	Description
Booking and Notification	- Ensure that after a booking is made, a notification email or message is sent to the user. - Test the interaction between the booking module and the notification system.
User Profile and Booking History	- Verify that a user's booking history is correctly updated and displayed in the user profile. - Check the integration between the user profile service and the booking service.
Service Availability and Booking	- Check if the booking system correctly checks the availability of service slots before confirming a booking. - Ensure the integration between the availability service and the booking module.

5.2.3 System tests

Table 5. 3 System test applications

Feature	Description
Functional Testing (End-to-End Booking Process)	- Simulate a user navigating the portal, selecting a service, booking it, and receiving confirmation. - Ensure all steps (login, service selection, booking, payment, confirmation) work smoothly.
Performance Testing:	- Test the performance of the portal under different load conditions (e.g., multiple simultaneous bookings). - Ensure the system can handle peak traffic without degradation of performance.
Security Testing:	- Conduct tests to protect user data (e.g., test for SQL injection, cross-site scripting). - Verify that secure payment processing is enforced.

5.3 Test cases

a. Login/Registration

Table 5. 4 Test Case 1

Test Case #: 1	**Test Case Name:** Login
System: Login/Registration	**Subsystem:** Login
Designed by: Tran Bach Tung	**Design Date:** 08/04/2024
Executed by: Tran Bach Tung	**Execution Date:** 13/05/2024
Short Description: Test the login function of the Login page	
System Testing: Functional testing	

Pre-conditions:

1. The user already has a Google/Github account.
2. The account is valid.
3. The system displays the login page.

Step	Action	Expected System Response	Pass/ Fail	Comments
1	Navigate to the login page.	The login page is displayed.	Pass	
2	Enter an invalid username and password.	The system displays an error message indicating invalid credentials.	Pass	
3	Click on the "Login" button	The user is authenticated and remains on the login page.	Pass	
4	Login in successfully	The system returns to the homepage and displays the user’s information	Pass	

Post-conditions:
User authenticated with Google/GitHub

b. About Us Page

Table 5. 5 Test Case 2

Test Case #: 2	**Test Case Name:** Us Page
System: About Us Page	**Subsystem:** None
Designed by: Bach Huy Hoang	**Design Date:** 12/04/2024
Executed by: Trinh Thuy Tien	**Execution Date:** 15/05/2024
Short Description: Serves multiple purposes, from building trust with potential customers.	
System Testing: Usability testing	

Pre-conditions:

1. Provides transparency about the company's mission, vision, history, and values.
2. Making the brand more relatable and trustworthy.

Step	Action	Expected System Response	Pass/ Fail	Comments
1	At home page	The home page is displayed.	Pass	
2	Proceed to navigate "About"	The information on "Passionate and Sustainable" is displayed	Pass	

Post-conditions:

Providing essential information about the company.

c. Services Page

Table 5. 6 Test Case 3

Test Case #: 3	**Test Case Name:** Services Page
System: Services Page	**Subsystem:** None
Designed by: Trinh Thuy Tien	**Design Date:** 20/04/2024
Executed by: Bach Huy Hoang	**Execution Date:** 20/05/2024
Short Description: Provide customers with detailed information about the services offered.	
System Testing: Functional testing	

Pre-conditions:
1. Order these services directly.
2. Schedule appointments for further advice or consultation.

Step	Action	Expected System Response	Pass/ Fail	Comments
1	At home page.	The home page is displayed.	Pass	
2	Proceed to connect with Services.	All the services will be displayed.	Pass	
3	Click “Make order”	Order the schedule.	Pass	
4	Click “Set Appointment”	The appointment will be saved.	Pass	

Post-conditions:
Help users with easy-to-navigate service options and streamlined ordering processes.

d. Account Information Page & Logout

Table 5. 7 Test Case 4

Test Case #: 4	**Test Case Name:** Account Info Page
System: Account Info & My Orders	**Subsystem:** None
Designed by: Bach Huy Hoang	**Design Date:** 29/04/2024
Executed by: Bach Huy Hoang	**Execution Date:** 20/05/2024
Short Description: Providing users with access to their personal information and order history.	
System Testing: Functional testing	

Pre-conditions:

1. Allows users to view and update their personal information.
2. Provides a list of orders added to the cart.

Step	Action	Expected System Response	Pass/ Fail	Comments
1	Navigate to "Account details".	Basic information and contact info will be displayed.	Pass	
2	Move to "My orders"	The information of the history customers booked and reviewed	Pass	
3	Click "Log out"	To end the session.	Pass	

Post-conditions:

Access to successfully their personal information and order history.

5.4 Comparison

With 4 test cases above all of the cases were satisfied and showed successful results of the actions taken by our team.

5.5 Evaluation

Each member tries to complete their tasks and complete them on time. Furthermore, we use Google Drive and GitHub as our primary workspace as well as a place to store files and documents.

CHAPTER VI

USER DOCUMENT

6.1 User manuals

Welcome to the Web Portal for Motor Servicing at Home. This user manual is designed to guide you through the process of using the portal, from registering an account to booking a motor servicing appointment. Follow the instructions carefully to make the most out of our convenient, doorstep motor servicing.

1.1 User Registration and Login

Step 1: Access the Portal

- Open your web browser and navigate to the web portal URL.

Step 2: Register a New Account

- Click on the "Login" button on the toolbar of the homepage.
- Fill in the required fields: Name, Email, Phone Number, Address, and Password.
- Click "Submit" to create your account.

Step 3: Login to Your Account

- Click on the "Login" button on the homepage.
- Enter your registered email and password.
- Click "Login" to access your account dashboard.

1.2 Booking a Service

Step 1: Browse Services

- Once logged in, navigate to the “Services” page.
- Browse through the list of available services such as oil changes, brake repairs, and battery replacements.

Step 2: Select a Service

- Click on the service you wish to book.
- Read the detailed description and pricing.

Step 3: Schedule an Appointment or make an Order

- Click on the “Set appointment” or “Make Order”.
- Add any special instructions or notes for the technician.

Step 4: Add to Cart

- After selecting the date and time, click “Add to Cart”.
- You can continue to add more services or proceed to checkout.

Step 5: Checkout

- Review your selected services in the cart.
- Click “Checkout”.
- Enter your payment details and confirm the booking.

1.3 Managing Your Account/Orders

Step 1: View Account/Orders

- Navigate to the “Settings” section in your account dashboard.
- View the account status or order details of your upcoming and past appointments.

Step 2: Reschedule or Cancel

- If you need to reschedule or cancel an appointment, click on the “My Orders”.
- Click the “Edit” button for change or cancel option.
- Follow the prompts to confirm your changes.

1.4 Payment and Feedback

Step 1: Payment Methods

- During checkout, you can choose from multiple payment methods such as credit/debit cards, digital wallets, and bank transfers.

Step 2: Feedback

- After your service is completed, navigate to the "My Orders" section.
- Click on the completed service and provide a rating and feedback for the technician and service quality.

6.2 Technical guides

- Organization:
  - + Categories: Routine Maintenance, Troubleshooting, Specific Models, Advanced Repairs.
- Models, Advanced Repairs:
  - + Sub-categories: Oil Changes, Brake Repairs, Engine Overhauls, etc.
- Guide format:
  - + Text Instructions: Clear, concise steps with bullet points.
  - + Visuals: High-quality images and diagrams.
  - + Videos: Demonstrative videos for complex tasks.
  - + Downloadable Content: PDFs for offline use.

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Throughout this project, our group will be able to acquire and experience new information and abilities, as well as apply software engineering theory in practice. This project also allows us to strengthen our soft skills, such as teamwork, critical thinking, communication, and planning. To boost efficiency, group members establish a single working direction and work toward it incrementally. Because each person has unique abilities, we work together to better comprehend and apply what we've learned to the project. In the initial part of the project, we discussed and divided the work equitably and explicitly, making it easier for everyone to understand the group's work and perform their tasks.

7.2 Future work

a. Advanced Booking System
- Online Scheduling: Implement a more sophisticated booking system that allows users to book multiple services for different time slots.
- Real-Time Availability: Integrate real-time availability to show users the available slots for each service.

b. Loyalty Programs and Discounts
- Loyalty Points: Introduce a loyalty points system where customers earn points for each purchase, which can be redeemed for discounts or free services.
- Seasonal Discounts: Offer seasonal promotions and discounts to attract new customers and retain existing ones.

c. Service Customization
- Custom Packages: Allow customers to create custom service packages tailored to their specific needs.
- Personalized Recommendations: Use machine learning algorithms to analyze user behavior and recommend personalized services.

d. Mobile Application
- Native Mobile App: Develop a native mobile application for both iOS and Android to provide a seamless and accessible user experience on mobile devices.
- Push Notifications: Implement push notifications to remind users about upcoming appointments, special offers, and maintenance reminders.

e. Enhanced User Profile Management
- Service History: Provide detailed service history and records within the user profile for easy reference.
- Document Upload: Allow users to upload important documents related to their vehicles, such as registration and insurance papers.

f. Integration with Vehicle Diagnostic Tools
- IoT Integration: Integrate with IoT devices and vehicle diagnostic tools to provide real-time data on vehicle health and maintenance needs.
- Predictive Maintenance: Use predictive analytics to alert users about potential issues before they become serious problems.

g. Customer Support and Chatbots

- Live Chat Support: Integrate live chat support to assist customers with their queries in real-time.
- AI Chatbots: Implement AI-powered chatbots to provide 24/7 customer support and handle routine inquiries.

h. Comprehensive Payment Options

- Multiple Payment Methods: Offer a variety of payment options, including credit/debit cards, digital wallets, and installment plans.
- Secure Payment Gateway: Ensure the integration of a secure and reliable payment gateway to protect user data.

i. Reviews and Ratings System

- Service Ratings: Allow customers to rate and review the services they receive, providing valuable feedback for improvement.
- Technician Reviews: Include reviews and ratings for individual technicians to build trust and transparency.

j. Augmented Reality (AR) for Service Visualization

- AR Integration: Use augmented reality to help users visualize service processes or see a 3D model of their vehicle with highlighted service areas.
- Virtual Consultations: Provide virtual consultations where users can interact with technicians through AR.

k. Expanded Content and Resources

- Blog and Articles: Create a blog with articles on car maintenance tips, industry news, and how-to guides.
- Video Tutorials: Offer video tutorials for basic maintenance tasks that users can perform at home.

l. Community Engagement

- User Forums: Create a community forum where users can discuss their experiences, share tips, and ask questions.
- Events and Workshops: Host virtual or in-person events and workshops on car maintenance and safety.

REFERENCES

1. Overarching, C. service Vietnam- (no date) *Michelin Car Service Vietnam: Tire and car maintenance service, MICHELIN CAR SERVICE Vietnam*. Available at: https://carservice.michelin.vn/ (Accessed: 22 May 2024).
2. *F8 - Learn programming to get a job* (no date) *F8 - Learn programming to get a job*. Available at: https://fullstack.edu.vn/ (Accessed: 22 May 2024).
3. *Vue.js* (no date) *Vue.js - The Progressive JavaScript Framework | Vue.js*. Available at: https://vuejs.org/ (Accessed: 22 May 2024).
4. Aminatou05 (no date) *Aminatou05/Capstone-2024, GitHub*. Available at: https://github.com/Aminatou05/Capstone-2024 (Accessed: 22 May 2024).
5. Product Focus (2022) *Scrum: What it is and how it works, Product Focus*. Available at: https://www.productfocus.com/product-management-resources/journal-articles/scrum-what-it-is-and-how-it-works/ (Accessed: 22 May 2024).
6. Siu, E. and Written by Eric Siu is a seasoned entrepreneur and CEO of the digital marketing agency Single Grain (2024) *PPC funnel: Tactics for every stage of the journey, Single Grain*. Available at: https://www.singlegrain.com/blog/x/ppc-funnel/ (Accessed: 22 May 2024).
7. Atlassian (no date) *Comprehensive kanban tutorial with jira, Atlassian*. Available at: https://www.atlassian.com/agile/tutorials/how-to-do-kanban-with-jira (Accessed: 22 May 2024).
8. *Develop stakeholder requirements using user story* (no date) *BAC TRAINING & CONSULTANCY VN*. Available at: https://www.bacs.vn/en/develop-stakeholder-requirements-using-user-story-20876.html (Accessed: 22 May 2024).