Unique Rent A Car

Session 2019-2023

By

Sadeeq Ur Rahman
Atif Shahzad

Bachelor of Science in Software Engineering

Department of Computer Science
City University of Science & Information Technology
Peshawar, Pakistan
August, 2023

Unique Rent A Car

Session 2019-2023

By

Sadeeq Ur Rahman
Atif Shahzad

Supervised by

Mr. Adnan Hussain

Department of Computer Science
City University of Science & Information Technology
Peshawar, Pakistan
August , 2023

Unique Rent A Car

By

Sadeeq Ur Rahman (11028)
Atif Shahzad (11029)

CERTIFICATE

A THESIS SUBMITTED IN THE PARTIAL FULFILMENT OF THE REQUIRMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

We accept this dissertation as conforming to the required standards

(Supervisor) Mr. Adnan Hussain	(Internal Examiner)
(External Examiner)	(Head of the Department)
(Coordinator FYP)	(Approved Date)

Department of Computer Science
City University of Science & Information Technology
Peshawar, Pakistan
August , 2023

Dedication

This project is dedicated to our loving parents respected teachers, friends, and to all those who have played a crucial role in our life.

Declaration

We consequently certify that this thesis has only been written by us. Except when otherwise noted, the work included in this dissertation is the outcome of our own study.

Sadeeq Ur Rahman
Atif Shahzad
August, 2023

Acknowledgment

I want to sincerely thank my supervisor, [Mr.Adnan Hussain], for his guidance, encouragement, and assistance during my FYP. His expertise feedback were invaluable in directing my research and supporting me to develop my skills as a developer.

I also want to express appreciation to my friends and classmates for their support and encouragement during my academic career. Their insights and feedback were incredibly helpful in shaping my ideas and pushing me to achieve my goals.

I'm extremely thankful to my family for their unflagging affection and assistance. Their encouragement and belief in me were the driving force behind my success in this project.
I would like to acknowledge the [Unique Rent A Car] for their financial support, which allowed me to carry out this development project.

I would also like to thank the technical staff of [City University of Science and Information Technology] for their assistance in providing the resources and facilities needed for my project.

Finally, I'd want to thank everyone who took part in my study and contributed to it for their willingness to do so. The completion of this project would not have been feasible without their involvement.

Abstract

The car rental industry faces several challenges in meeting the increasing demands of customers while maintaining operational efficiency. In response to these challenges, this thesis presents the development of a web-based vehicle rental system that aims to streamline the rental process and improve overall customer experience. Usability and scalability were all taken into consideration during the system's design and development. To assure the system's dependability and efficacy, extensive user testing and functional testing were carried out. The testing strategy includes in-depth analyses of data management, security protocols, and user interfaces. This thesis presents the design, development, and implementation of a web-based vehicle rental system for a car rental company. The system was developed using modern web technologies and tools such as Flutter and Firebase. The system's primary goal is to offer a practical and user-friendly platform so that clients may hire cars and the auto rental firm can keep track of its inventory of vehicles. [1].
Customers may explore available vehicles, check availability, and make bookings using the system's user-friendly interface. For the car rental firm to handle their fleet of vehicles, reservations, and client information, the system also features an admin panel.
The development of the system involved a thorough analysis of the requirements of the car rental company and its customers.The platform was created to be scalable and adaptive to the firm's future growth and changes.
The system was tested extensively for functionality, usability, and security. The results of the testing showed that the system met all the requirements and provided an efficient and reliable solution for the car rental company and its customers.
This thesis demonstrates the successful development and implementation of a vehicle rental system that meets the needs of a car rental company and its customers. The system provides a user-friendly and efficient platform for vehicle rental and management, and is expected to improve the overall customer experience and streamline the operations of the car rental company.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CRS	Car Rental System
FR	Functional Requirements
OCRS	Online Car Rental System
WBS	Web Base System.
MCRS	Manual Car Rental System
NFR	Non-Functional Requirements
SRS	Software Requirements Specification

Chapter 1

Introduction

1.1 Overview

Although Pakistan's vehicle rental market has seen tremendous expansion in recent years, issues with effective operations have reduced client satisfaction and negatively impacted company performance. This thesis suggests a web-based approach to transform the nation's experience with automobile rentals. Although the sector is essential for offering local and international travellers mobility options, it struggles with inefficiencies in reservation processes, vehicle management, and client relations. To simplify processes and enhance the user experience for both consumers and automobile rental firms, sophisticated software is required.A country like Pakistan, where the majority of people are from middle-class backgrounds. A car is usually out of their price range. People today are forced to travel to other locations for employment, business meetings, and tourism since they have less time and more job to do.To design a rental vehicle management system that is only focused on the requirements of the customer, our team, growth, innovation, and efficiency [1]. All of these components will work to propel us forward towards success. The process takes a long time if the customer database is searched manually and there are thousands of customers. Staff workers also have to manually input reservations, and it might be difficult to improve monthly or annual reports. In order to guarantee uninterrupted operation, vehicles are inspected and maintained in good condition. Using an online auto rental system, one may reserve or hire a car [2].These days, life is becoming increasingly simpler for almost everyone. It is only practical due of technology. Numerous internet-based platforms considerably increase people's comfort. In only a few clicks, we can fulfil all of our home-based desires. We are accustomed to online banking and shopping [3].A similar web-based application called The Car Rental System (CRS) enables consumers to book cars with only a few clicks. Some people just cannot afford to own a car, therefore this approach is really helpful to them. This system features a range of automobiles that may be purchased and delivered based on the customer's comfort level and location within the region. Only online services are available for long-distance reservations. Users may easily search for and rent cars for the needed length because to its user-friendly design. Depending on the type of vehicle the customer requires, the

user will be able to make bookings. Customers may now use the internet at any time to rent a car [1]. Using the vehicle rental system makes the reservation straightforward. Work and time are saved. The programme will ask the user for information such as the destination, the departure date and time, the kind of vehicle, etc. It will also need a special identification number. The tool will let the customer use these details to reserve a car for the trip. Simply put, a vehicle rental company's website-based system (CRS) is what it stands for. This system enables the business to record information about its services and make them accessible to the public online [4].

1.2 Background

Your company will go out of business if you don't have an online presence" – Bill Gates, Founder of Microsoft (September 2011)Numerous Internet apps is being created every day 1 to improve the quality of our lives. The online car rental system is one of these systems are utilized and perform functions more like to those of personal computers as a result of the emergence and evolution of technology [5]. These days, we may use our mobile devices to enjoy the flavors of many applications, and these flavors affect every part of our everyday life. These flavors may contain various software programmes used for entertainment activities including listening to music, viewing movies, surfing the internet, playing games, and other similar activities. To handle such complex application, mobile phones always require a separate operating system [6].

1.3 Motivation

The intent of this project is to enhance customer car rental experiences and business operational effectiveness. A web-based user-centric solution may grow the company's client base, boost repeat business, and enhance its standing as a dependable service provider. The system uses cutting-edge technology to cut down on paperwork, physical labour, and prices, which improves resource allocation.In recent years, both public access to computers and the number of people utilising computers and cellphones have dramatically expanded. However, the noticeable problem in currently Vehicles Company which lacks the concept to book online vehicles system. As a result to book a more comfortable vehicle on a low budget in less time-consuming. This reason has motivated us to develop a modern software (application and website), which can compete with other business solutions present in the market for small businesses with the addition of the concept of productivity so that in addition to providing productivity clients should also save burden and hassle an interactive environment.

1.4 Problem Statement

In Pakistan, this initiative attempts to provide a user-friendly, effective vehicle rental system, addressing problems such lengthy booking procedures, restricted vehicle visibility, and subpar customer service. With a seamless link between clients and vehicle rental services, the web-based solution promises to increase customer satisfaction and profitability for the car rental industry.The bulk of auto systems are manual, which is uncomfortable for cars and intended to make users spend more time operating them. Additionally, the system doesn't have the capability to save user booking information that would enable the owner of the business to easily identify their consumers. As was already indicated, these systems prevent learnability and offer the sole satisfaction. The primary issue is that customers who wish to easily reserve a rental car and complete the task faster. Only during regular business hours is service offered via the Manual car rental system. Customers only have a limited window of time to reserve vehicles.

1.5 Proposed Solution

To provide a smooth user experience across several platforms, the suggested solution integrates Dart and Flutter. Real-time data synchronisation, dynamic vehicle availability updates, and user authentication are all made possible through Firebase integration. A reliable and scalable car rental system is created with the help of Android Studio and Visual Studio Code.A solution that utilizes the development of modern software apps and web-based car rental services is suggested, keeping in mind the issue raised in the problem description. This is good accessible on an Android smartphone. With the help of this programme, the client may effortlessly hire a car from any location. However, this will also offer a notion of adaptability,productivity, and agility. Client satisfaction will occur as a consequence. This project encompasses a wide range of fields, from business principles to computer science, thus it necessitates the conclusion of several research in order to meet the project's aims.

1.6 Scope and Objective

The Scope and objective(s) of this study are:

1.6.1 Scope

The web-based auto rental system provides users with a number of capabilities, such as a simple booking procedure, real-time tracking of vehicle availability, and safe payment

methods. Companies that rent cars have access to a sizable client database, performance statistics, and vehicle management system. On-road support services and actual vehicle monitoring are not included in the project.The goal is to create a web-based information system that will assist in addressing ongoing problems with the manual information system and assist in making jobs that appear to be challenging for both the personnel of the vehicle rental firm and those renting the vehicles (customers) easier to do.The objective of the project is to establish a web-based system that will aid in managing the automobile rental industry. Users will be able to quickly access the company's website to search for and reserve their preferred vehicles online. To assist in promoting a company's vehicle rental services through the system's availability online [7].

1.6.2 Objectives

In an effort to save clients time contacting and waiting for a car, the project's goal is to automate vehicle rental and reservations.

1.7 Required Tools Techniques

- Flutter
- Dart
- Firebase
- Android Studio
- Visual Studio Code

.

1.8 Thesis Outline

The discussion of the software project plan and issues related to project management, such as defining project deliverables, project planning, and risk management, are the main topics of chapter 2. The focus of Chapter Three is on the requirements that must be met and the characteristics that the product must have. I also made an effort to provide a thorough explanation of the behaviours of the application that would be created. Both functional and non-functional needs are included. The design of my programme, its interfaces, workflow diagram, and flow graphs are covered in chapter four. The test plans for testing the application are shown in Chapter 5. This chapter also includes a

thorough set of tests that will aid in the timely, bug-free completion of the product. In this chapter, I organised the testing requirements into categories, described the testing methodology to be applied, listed the test cases, and listed the expected outcomes for each test. The whole project's conclusion is demonstrated in Chapter 6.

Chapter 2

Software Project Planing

This chapter's goal is to give a thorough overview of the software project planning stage for a car-rental system. The chapter provides an overview of the project's setting, describes its goals, and emphasises the need of good project planning. It also outlines the essential elements of project planning, such as managing stakeholders, allocation of resources, work breakdown structure, and risk evaluation. As the chapter comes to a close, it is highlighted how important a well-structured project plan is to ensuring the creation and implementation of the rent-a-car system.

2.1 Feasibility Study

The Feasibility study assesses the likelihood of a web-based system for car rentals being successful and identifies its advantages and disadvantages. The study draws attention to the system's existing state and areas for development.The web-based car rental service may draw proficient in technology clients looking for simplicity and flexibility in their car hires. Tourists, business travellers, and local residents are all drawn to its user-friendly interface. It competes in the market thanks to simple online booking and real-time information on car availability. The system's capacity to scale and adapt allows it to evolve to accommodate future market expansion and manage bigger traffic volumes, maintaining efficiency and responsiveness even during busy times or periods of increased demand.The feasibility study discovered a number of flaws,mapping services. Inconsistencies or service interruptions might result from this. The system's reliance on internet access may be a flaw in places with patchy network coverage, which might result in lost income and clients. Sensitive user data must be protected with strong security measures since flaws in the system's design might leave it vulnerable to cyber-attacks and data breaches, weakening consumer confidence and harming the company's brand.

2.2 Stakeholders Identification

The success of a project depends heavily on its key stakeholders since they determine its goals, demands, and expectations. Even from senior management, securing buy-in

and support from these stakeholders is crucial for project success. A comprehensive understanding of their views, needs, and possible effects may be achieved by including key stakeholders in the development of a vehicle rental system. This makes it possible to collaborate, communicate, and make decisions effectively over the course of a project. In order to create a supportive atmosphere and increase the chances of project success, it is important to address the goals and concerns of these stakeholders.

2.2.1 Direct Stakeholders

Those who are actively involved in the project or who will be significantly influenced by its execution are considered direct stakeholders. These stakeholders may include the management team of the car rental firm, staff members, and system users. Providing guidance, assigning resources, and making choices on the system's growth and operation are all critical tasks performed by the management team. Employees directly contribute to the system's daily operation, including customer service agents and members of the operations team. Customers have a stake in the usefulness, practicality, and overall experience of the automobile rental system since they utilise it directly. Their approval and input are essential to the project's success. These direct stakeholders' viewpoints and needs are taken into consideration by the vehicle rental

2.2.2 Indirect Stakeholders

Organisations or individuals that aren't directly participating in the project but who might yet be impacted by its completion. Local communities around the operations of the vehicle rental company may be among these stakeholders as variables like traffic congestion, noise, or environmental concerns connected to the rental activities may have an influence on their day-to-day lives and surroundings. Indirect stakeholders include government organisations and regulatory authorities in charge of monitoring the vehicle rental market and guaranteeing adherence to rules and laws. In addition, companies offering insurance, manufacturers of vehicles, and technology partners that indirectly help the car rental industry through their offerings are also regarded as indirect stakeholders. Getting in touch with these parties, attending to their issues, and taking their opinions into consideration will help with the auto rental system.

2.2.3 Other stakeholders

Stakeholders in a car rental system project may include industry associations, trade organizations, financial institutions, and technology vendors or software providers. These stakeholders can provide expertise, financial support, or technological advancements that can significantly impact the project's outcomes. By recognizing and engaging with these stakeholders, the car rental system thesis can benefit from their expertise, financial support, and technological advancements, ultimately enhancing the overall success and ef-

fectiveness of the project.

2.3 Project Characteristics

This project's benefit is that it requires less time to book online more comfortable vehicle low budget because it's easily android apps of smartphone and web on laptop/pc. The vehicle can book manually which is more time-consuming. But in vehicle rental service app client register with Email and some personal information, then after the user can book a vehicle, save the history of booked vehicles and also check vehicles availability.

2.4 Project Activities

By integrating a mobile app with a web-based platform, the vehicle rental project provides a novel approach to car rental services. Travellers and people who want rapid access to services may now use their smartphones and tablets to access automobile rental services. The web-based platform guarantees smooth device compatibility and enables users to access the system using their default web browsers without having to download any special software. To ensure quick response and recovery in the case of unanticipated threats, the project team will place the highest premium on emergency preparation. They will put in place strict security measures, create a quick response team, and set aside a contingency fund to cope with financial risks and budget overruns. The project will be able to cover unanticipated expenses and resource shortages with the help of this reserve without jeopardising deadlines or deliverables. To ensure a smooth transition to the production environment and lower deployment-related risks, extensive deployment testing will be done, including compatibility checks on hardware and software settings. A rollback plan will be developed in the event that deployment issues occur.

2.4.1 Project Breakdown Structure/Work Breakdown Structure

The Work Breakdown Structure (WBS), which outlines the tasks necessary to create the different project deliverables, serves as the process' starting point. From the standpoint of the work packages, actions appear. For the task packages that make up the project's initial stage, there are precise blueprints at the start of the project. The project plan is updated when more in-depth, low-level work packages are added as the time to start new work approaches. The planning process is periodically reviewed to ensure that the detailed plans include the most recent knowledge about the project. Activities that were dependent on time constraints or input from other activities. For project including

renting cars that you specified, I can provide you a textual representation of the Work Breakdown Structure (WBS):

Project: Rent-a-Car System

Module 1: Manage Customer

- Develop customer registration feature
- Implement customer database and profile management
- Enable customer login and authentication
- Include car details and specifications in the system
- Integrate car availability and reservation status

Module 2: Manage Booking

- Develop online booking system
- Enable customers to search for available cars
- Implement booking request submission and confirmation
- Provide booking cancellation options

Module 3: Manage Database

- Design a database for storing rental history
- Capture rental details, including dates, customers, and cars.

This hierarchical structure presents the breakdown of the rent-a-car project into its major components and subsequent tasks under each component. Each task represents a specific activity or deliverable within the project.

2.5 Activities Estimation

We must estimate every resource needed to complete the project in order for the programme to be successful. An essential component of every software development project is project estimate. The inability to establish precise predictions of software size is the primary cause of the software project's failure. We frequently underfund or don't give enough time for development because we estimate size too low. Most often, poor size prediction occurs when estimating the project's cost, timeframe, or both. These types of issues may be avoided by accurately estimating the activities to lower the likelihood of project failure.

2.6 Activities Risk

Risk is an inevitable and inescapable component of the software development process. It continuously changes throughout the life of a project and can have an impact on the project, the product, or both. As a result, it becomes necessary to deal with and manage these risks in an effective and efficient way.
Risk management is a methodology or a mechanism used throughout the development process in the field of software engineering to detect, manage, and mitigate risks that arise both before and during the process.
The risk management method primarily covers three categories of activity.

2.6.1 Identify and Quantify Risks for Activities

Risk 1: Urgent Work

Probability: Low **Impact:** High **Risk Level:** Medium

Risk 2: Difficulties in Programming

Probability: Low **Impact:** High **Risk Level:** Medium

Risk 3: Can't find better Design

Probability: Low **Impact:** High **Risk Level:** Medium

2.6.2 Risk Reduction: Activity to Stop Risk Occurring

Risk 1: Urgent Work

To prevent Urgent Work, the following activities will be undertaken:

- if the project manager is preoccupied with pressing tasks, such as a family member's wedding.
- Focus on one achievement at a time.
- I'll put in more effort every day in the week leading up to the task.

Risk 2: Difficulties in Programming

To prevent Difficulties in Programming, the following activities will be undertaken:

- implementation-phase difficulties while creating a specific module.
- Utilise online tutorials or ask questions on social networks that are relevant to the area.
- Make my design as easy to understand as you can, while minimising coupling.

Risk 3: No better Design can be found

To prevent Difficulties in Can't find better Design, the following activities will be undertaken:

- if the design of the layout is not understandable.
- Ask for assistance and look up different HCI bases design.
- Create a user-friendly design using my own ideas, then share it with friends to get additional feedback.

2.6.3 Contingency: Action if Risk does Occur

The project team will give emergency preparedness top priority to guarantee prompt reaction and recovery in the event of unforeseen dangers. To deal with monetary risks and budget overruns, they will implement thorough security standards, form a fast reaction team, and set aside a contingency fund. With the use of this reserve, the project will be able to absorb unforeseen costs and resource shortages without jeopardising deadlines or deliverables. Thorough deployment testing will be carried out, including compatibility checks on hardware and software configurations, to guarantee a seamless transfer to the production environment and reduce deployment-related risks. In the event that deployment problems arise, a rollback strategy will be created.

Risk 1: Urgent Work

If an Urgent Work occurs, the following actions will be taken:

- A week before the work I will do more work on a daily basis.

Risk 2: Difficulties in Programming

If an Difficulties in Programming occurs, the following actions will be taken:

- use the internet to access online tutorials, or ask questions on domain-related social networks.

Risk 3: Can't find better Design

If an Can't find better Design occurs, the following actions will be taken:

- Take help and search for various HCI base design.

2.7 Methods of Communication

2.7.1 Internal Communication

To facilitate effective internal communication within the project, the following methods will be established:

- **Regular Team Meetings:** To review project progress, handle issues, and provide updates, have frequent team meetings. Depending on the requirements of the project, these sessions may be held weekly or biweekly.
- **Project Management Tools:** Utilize project management software or tools to facilitate collaboration, task tracking, and document sharing among team members. Examples include Flutter, Firebase.
- **Watsapp Communication:** Use Watsapp as a primary mode of communication for formal announcements, sharing important documents or reports, and seeking approvals from relevant stakeholders.
- **Shared Document Repositories:** Establish shared repositories to store and manage project-related documents, guaranteeing simple access and version control. Examples include Google Drive and SharePoint.
- **Task Assignments and Reminders:** Implement a task management system where assignments are clearly defined, deadlines are set, and reminders are sent to team members to ensure timely completion of tasks.

These methods of communication will enable efficient and transparent communication with both internal team members and external stakeholders, promoting collaboration, timely information exchange, and satisfactory customer experiences.

2.8 Resources Allocation

A brief explanation has been included to the resource allocation table to highlight how this allocation would guarantee efficient project management and seamless execution. The timely completion of projects and achievement of milestones depend on the effective use of resources show in table2.1

Table 2.1: Resource Allocation

Name Of Task	**Resource**	**Allocation (Weekly)**
Planning	Project Manager	3
Design	UI	4
Development	Developer	19
Testing	Tester	4
Documentation	Planner	8

2.8.1 Resource Personal

The table 2.1 presents the resource allocation for the rent-a-car project:

The resource allocation table provides information about the allocated hours for each resource involved in the rent-a-car project. The roles of the resources include Project Manager, Marketing Specialist, Operations Manager, IT Specialist, and Customer Service Representative. The allocation column represents the initially planned hours for each resource.

The resource allocation ensures that the necessary expertise and human resources are assigned to specific roles in the project, allowing for the efficient execution of tasks and responsibilities. It provides a clear overview of how the workload is distributed among team members and facilitates effective resource management.

2.8.2 Gantt Chart

The project work plan lists all project tasks required to meet the project goals, together with precise estimates of their effort and cost, their timetable, and the time and resources they will need to complete. The project will be managed and its progress will be tracked using the project work plan as a guide.we use agile methodology for implementing this project. So in first stage we start initial planing that what can we do and how much time will required for every phase.After completion of initial planning start initial requirement.After Initial design comes to initial design. That how our project design look.Then start development and divide modules into sprints.after completing development and test whole system.And then start documentation.And this is thw whole work plan.Work plan is presented in Figure 2.1.

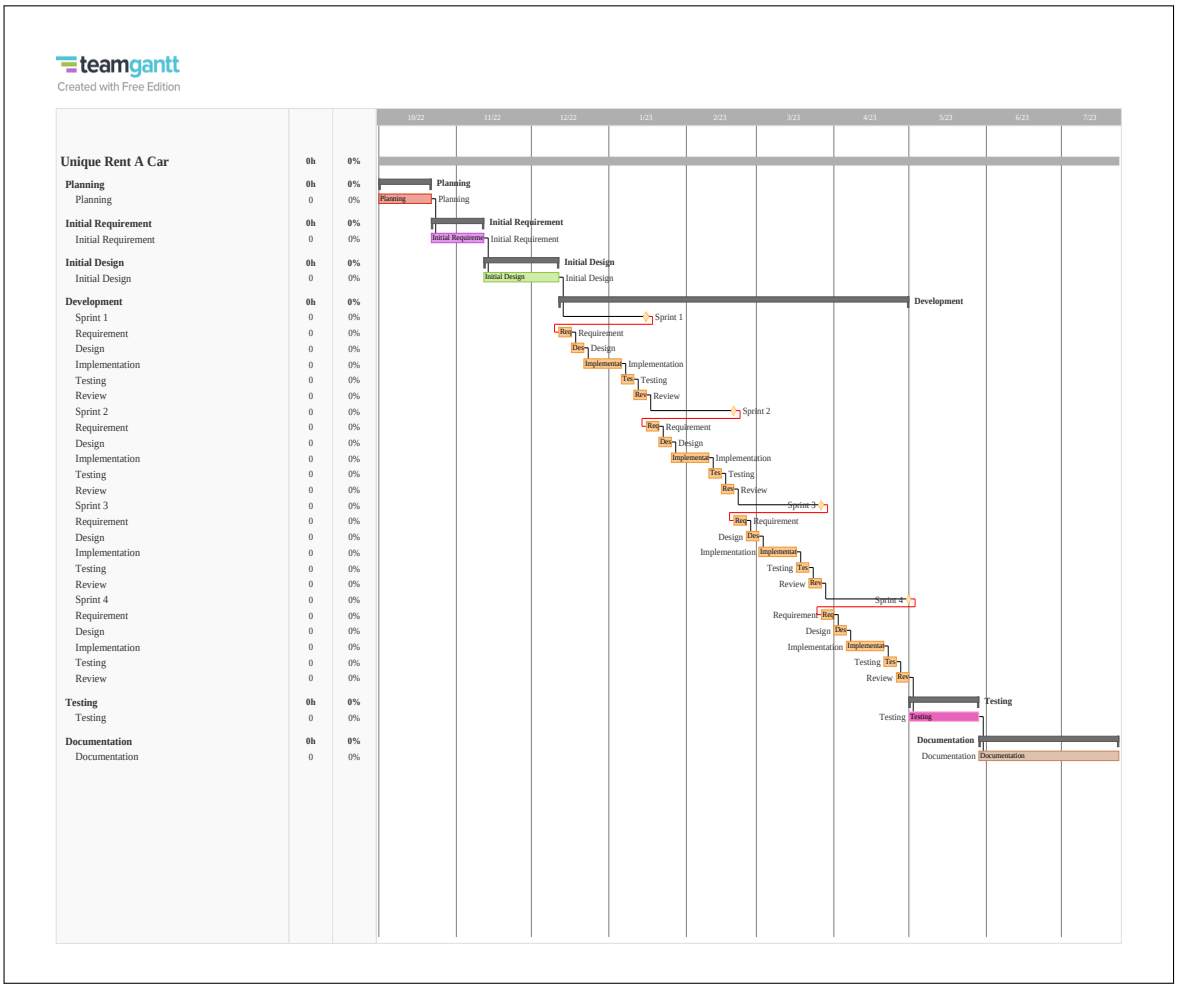

Figure 2.1: Gantt Chart

Chapter 3

Software Requirements Specification

This Software Requirements Specification (SRS) document's main goal is to give a thorough explanation of the specifications needed to create the rental vehicle software system. It describes the limitations, system user interfaces, and functional and non-functional needs that must be taken into account while developing software.

3.1 Requirement Elicitation

At this point, we want to identify the issue, provide remedies, and discuss the various options with one another. Meetings are planned with the stakeholders and the software engineering team to gain a more in-depth knowledge. The goal is to have a solid understanding of the system's goals, what has to be done, and how the whole thing fits into the car-rental industry. In general, lists will be made to identify the participants' stakeholder groups, technical environment descriptions, usage scenarios, and a list of needs were made at this point.
Functional Requirements and Non-Functional Requirements are the two main views of System Requirements.

3.2 Functional Requirements

Throughout the course of the project, any modifications that could arise should be handled with clarity and a vehicle. whether the stakeholders and the software engineering team agree, any prospective modifications would be reviewed, debated, and evaluated to see whether they could be made within the project's given time schedule. Since adjustments or alterations might happen in any situation, requirements management will take place throughout the project's process flow. Some of functional Requirement are:

- **Car Categories**
  A list of the many car classifications that are available for rental should be provided by the system, such as self-driven, corporate, outstation, and local driven cars. There could be unique features and price alternatives for each category.

- **Admin Panel**
  An administrative interface that enables authorised staff to control and keep an eye on the car rental system's user administration, vehicle inventory, and bookings.

- **Booking Request**
  Users can submit reservations for particular car classes and preferred rental dates. These requests need to be handled by the system.

- **Booking History**
  A feature that allows customers to browse prior rental reservations and related information, such as pick-up dates and vehicle categories.

- **Booking Car**
  Following confirmation of a booking request, the system should assign a specific car from the specified category to the user, guaranteeing the availability of the chosen vehicle.

- **Booking Cancellation**
  Users should have the ability to cancel their reserved car rentals, and the system should manage this task.

- **Forgot Password through Email**
  The system should provide a way to reset passwords in the event that users forget them by sending a link to their registered email addresses.

- **Authentication**
  To ensure security and privacy, the system should require users to log in using their credentials in order to access the booking and account-related capabilities.

- **Fare Estimator**
  A price estimator feature that enables users to get an approximation of the rental cost for various vehicle classifications and rental lengths should be included in the system.

- **Damage Policy**
  The system must to include thorough damage policies that are easy to understand and lay out what to do and how much it will cost if any hired vehicles are damaged during the rental time.

3.3 Non-Functional Requirements

Restrictions or requirements that the system must meet. The quality features of the system are determined by non-functional needs. Non-functional requirements, while not all of them do, often conclude with "ity".

- **Scalability**
  A system's capacity to manage rising workloads and user demands without experiencing performance loss is known as scalability. In order to accommodate concurrent users and vehicle bookings, a scalable auto rental web-based system should manage resources effectively, assuring a positive user experience during peak times.

- **Availability**
  Availability assesses a system's usability and accessibility throughout a predetermined time period. To guarantee that users can access, make reservations, and manage bookings without interruptions or downtime, high availability is essential for vehicle rental systems.

- **Reliability**
  For vehicle rental systems to guarantee users can access, make reservations, and manage bookings without downtime or service interruptions, availability assesses system accessibility and operationalness.

- **Maintainability**
  In order avoid downtime and guarantee system currentness, maintainability in car rental systems enables effective modification, upgrading, and repair of additions, bug fixes, and upgrades.

- **Serviceability**
  The goal of serviceability in car rental systems is to quickly detect and fix possible problems in order to reduce downtime, diagnose and repair malfunctions, and improve performance.

- **Regulatory**
  In order to secure user information and ensure legal compliance, car rental systems have to abide by data privacy, and industry-specific rules.

- **Manageability**
  In vehicle rental systems, manageability facilitates administration, monitoring, and control, allowing for effective supervision of stock, reservations, and user management.

- **Usability**
  Usability, or the system's user-friendliness, is essential for automobile rental systems to draw in new consumers and keep existing ones by providing user-friendly booking procedures and interfaces.

- **Performance**
  User experience is improved through performance optimisation in car rental systems by assuring speed, effectiveness, and seamless transaction processing.

3.4 Use-Cases/ Hierarchical Input Process Output

Use-cases, which include both consumers and administrators, are crucial to comprehending and describing interactions between users and the vehicle rental web-based system. They depict particular situations or sequences of activities that users take to accomplish certain objectives. Use-cases aid developers in identifying objectives, functions, and complexity in the context of the car rental system. Developers may create efficient solutions that satisfy user demands and adhere to project objectives by analysing and capturing use-cases. While use-cases for administrators define their administration and maintenance activities, such as maintaining car inventory, revising rental prices, and responding to customer support requests, use-cases for consumers give a clear picture of user journeys. This method aids in comprehending the administrative complexity of the system and guarantees effective administration of duties. Figure 3.1.

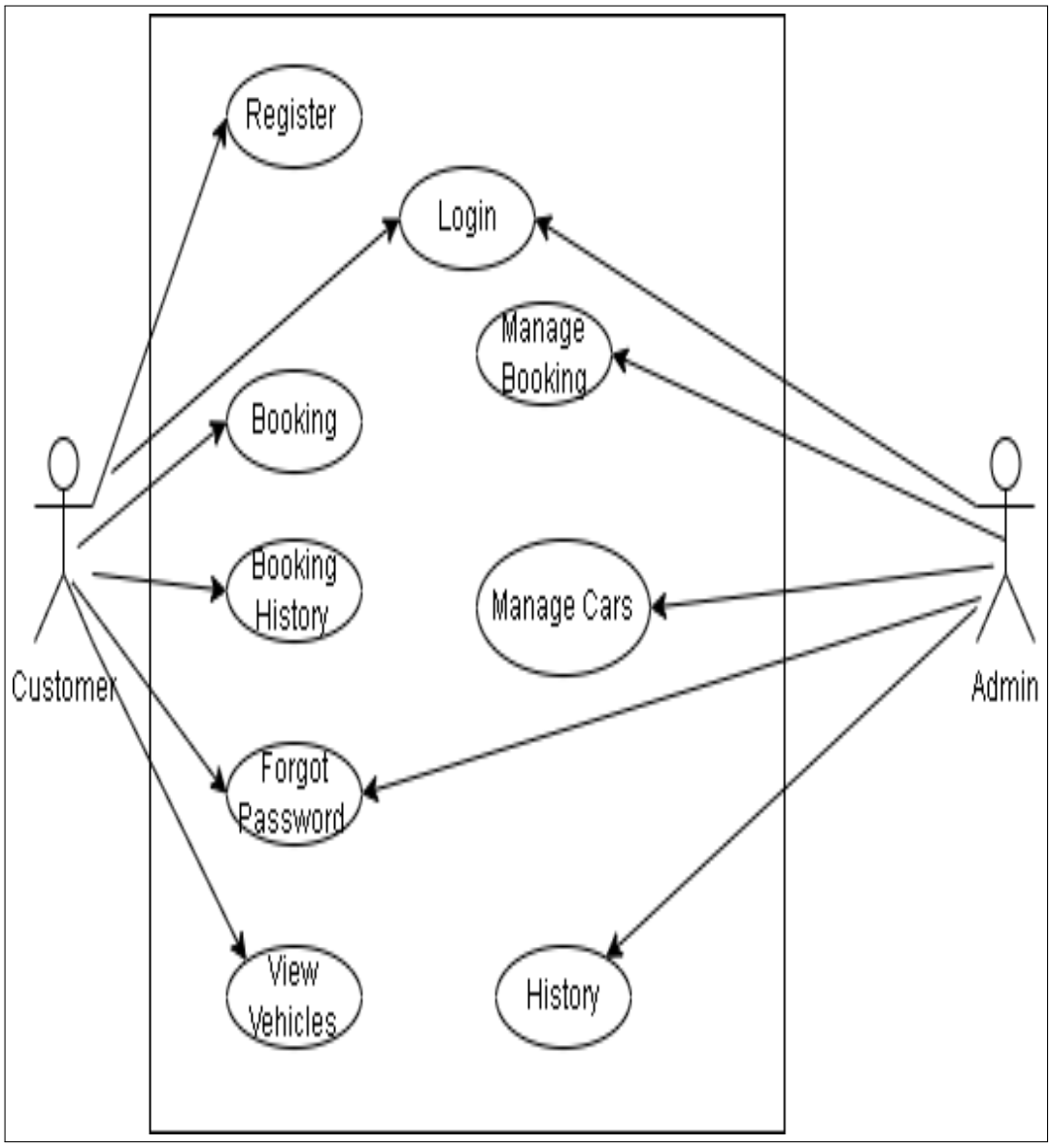

Figure 3.1: Use Case

Chapter 4

Software Design

Software design is a crucial phase in the creation of the car rental system since it helps to translate abstract requirements into a tangible and dependable system architecture. Software design lays the groundwork for the whole development process by serving as a link between basic concepts and realistic execution. In order to suit the unique requirements of the automobile rental system project, the envisioned features, interactions, and behaviour of the system are cautiously developed and structured during this phase. Software design produces a cogent plan that directs the development team in creating an effective, scalable, and user-friendly vehicle rental platform by integrating the thorough understanding of user needs, business objectives, and technological restrictions. This chapter explores the many facets of software design, emphasising its importance in transforming the hypothetical automobile rental system into a tangible, orderly, and useful reality.

4.1 System Diagram / Context Level Diagram

The System Diagram, sometimes called the Context Level Diagram, is a high-level illustration of the structure and external interactions of a vehicle rental system. The system, users (customers and administrators), and external entities like mapping services and car inventory databases are all represented by rectangular blocks in the diagram. Arrows are used to depict the interactions between various parts, showing how information and actions move between the system and outside entities. This diagram clarifies the general architecture of the system and how it interacts with other entities for improved functionality and user experience show in figure 4.1.

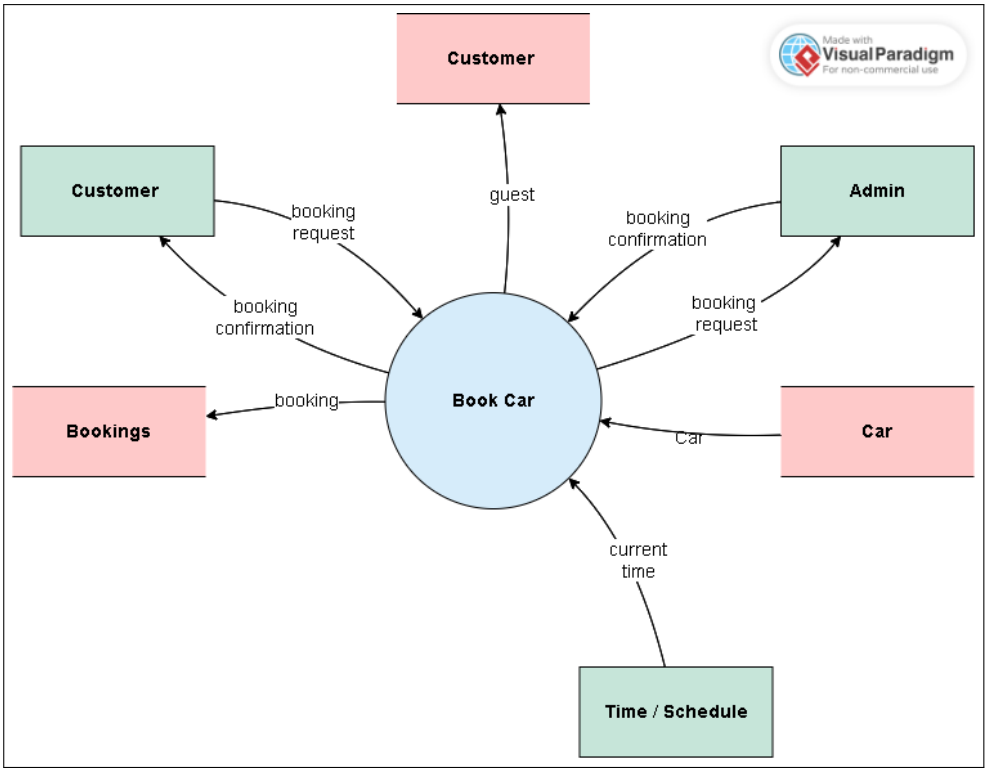

Figure 4.1: Context Diagram

4.2 Low Level Design

A low-level design is a detailed representation of the system's architecture that focuses on specific modules, components, and their interactions. It delves into the implementation details, such as algorithms, data structures, and interfaces. It addresses the technical aspects of the system, translating the high-level design into a more concrete and precise blueprint for developers to follow. The low-level design captures the granular details required for actual coding and provides a roadmap for efficient and effective implementation.

4.2.1 UML

Use Case

A visual representation of a system's functional needs is a UML use case diagram. It shows actors, use cases, and their connections to show how users interact with the system's features. It aids in comprehending the behaviour of the system and locating its essential components show in figure 4.2.

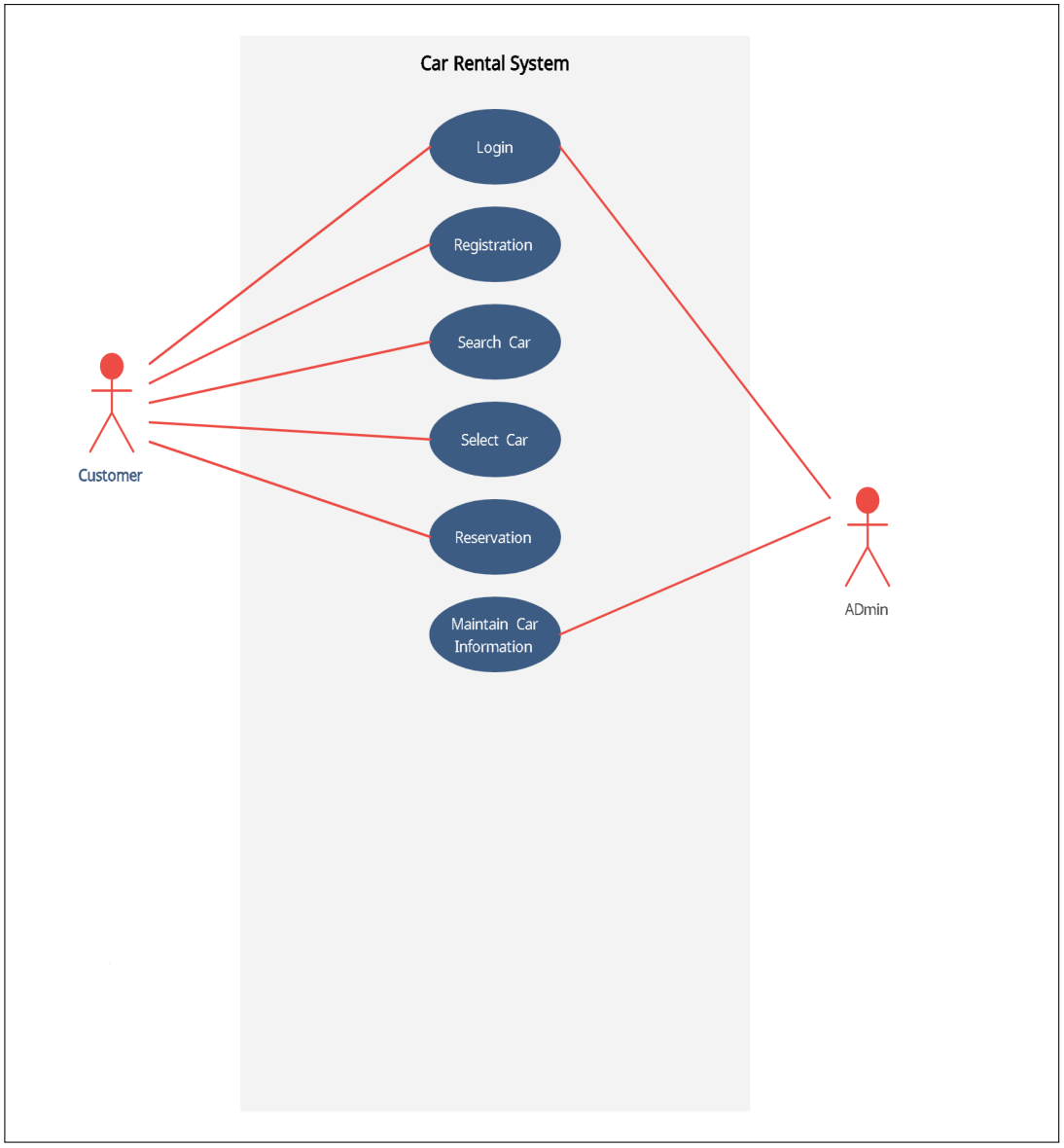

Figure 4.2: UML Use Case Diagram

Sequence Diagram

A UML sequence diagram is a graphic representation of how objects interact with one another in a system, displaying the sequence in which messages are passed back and forth. It illustrates the flow of control and the sequence of events in a scenario or use case, aiding in understanding system behavior and communication patterns show in figure 4.3.

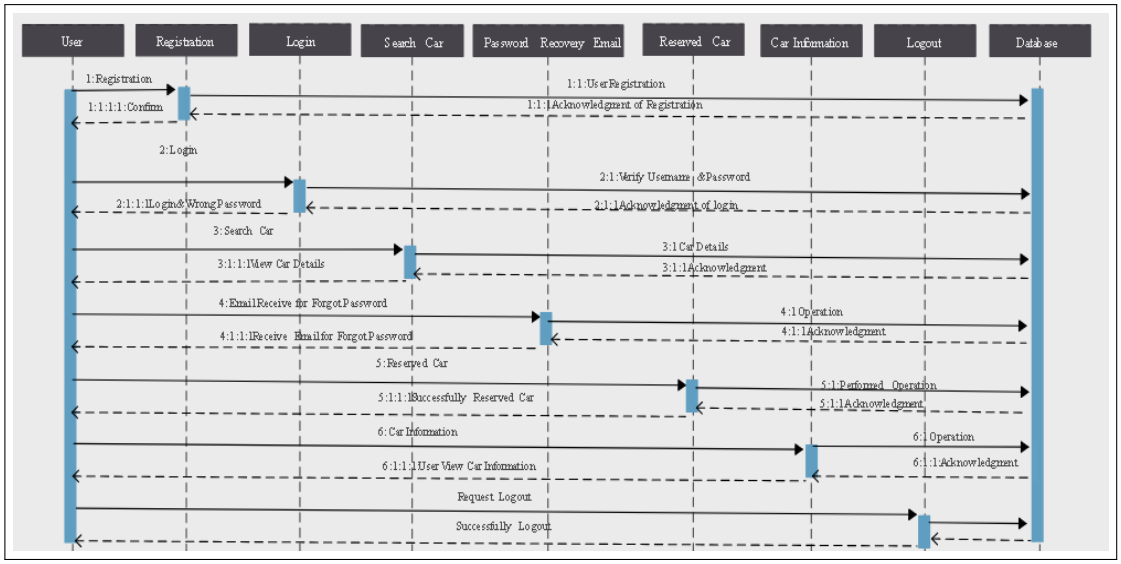

Figure 4.3: UML Sequence Diagram

4.2.2 DFD

Level 0 DFD

An overview of the system is given by a DFD (Data Flow Diagram) 0-level diagram, which displays the key operations and how they interact. It depicts the movement of information among external organisations, procedures, and data repositories. This high-level diagram helps in understanding the system's data flow and the relationships between its components show in figure 4.4.

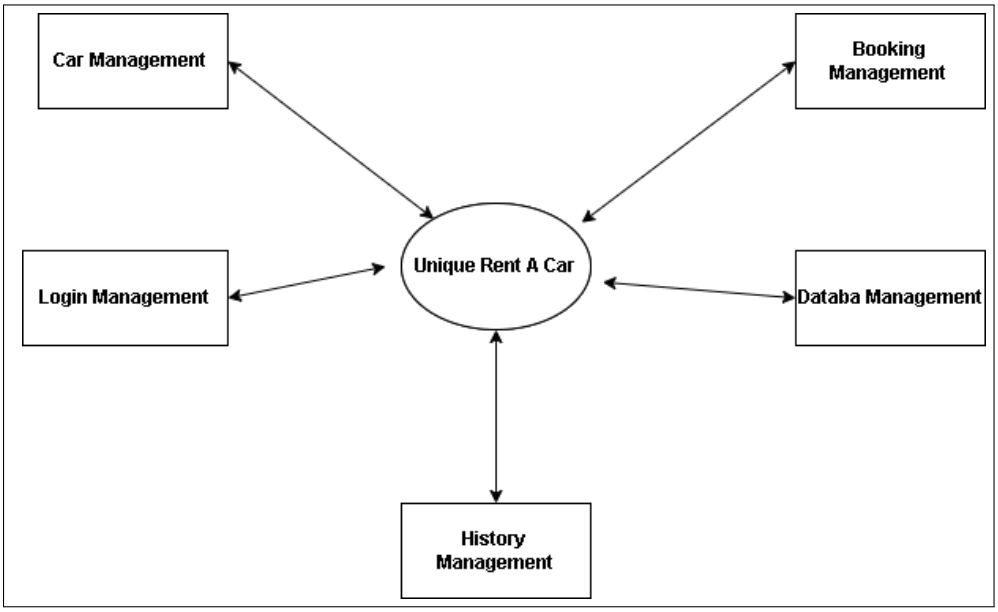

Figure 4.4: Level 0 DFD

Level 1 DFD

A DFD (Data Flow Diagram) 1 level diagram is a graphical representation of how data flows through a system or process. It depicts the inputs, outputs, and processes involved, showing the transformation of data from one form to another. It provides a high-level overview of the system's data flow and helps in identifying key data sources and sinks show in figure 4.5.

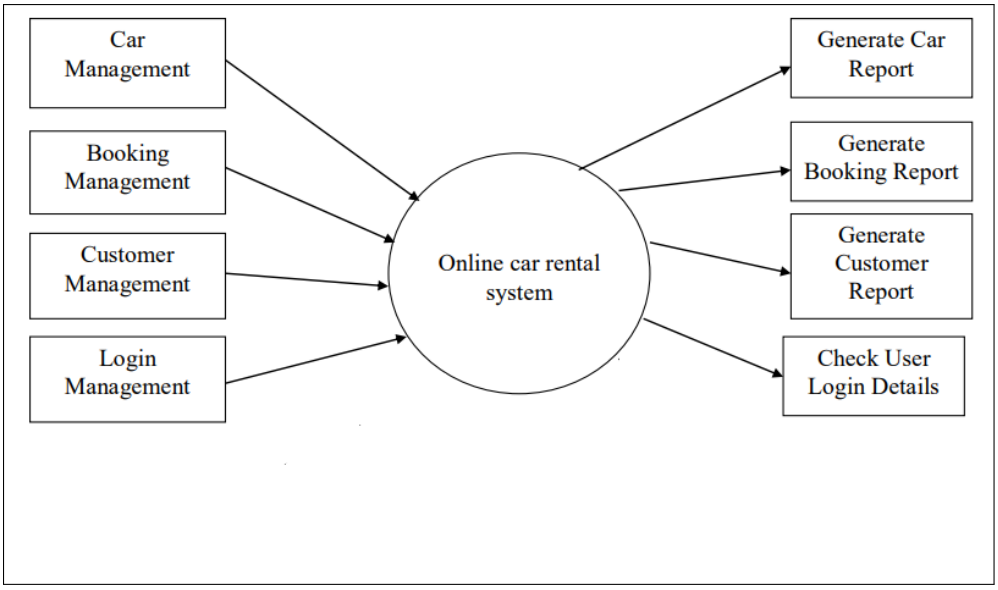

Figure 4.5: Level 1 DFD

Level 2 DFD

A DFD (Data Flow Diagram) 2-level diagram is a graphical representation of a system's data flow and processes. It consists of two levels, with the first level showing the major processes and data sources/sinks, and the second level providing a more detailed view of the processes and data flows within each major process. It helps visualize the flow of information and understand the system's data transformations and interactions show in figure 4.6.

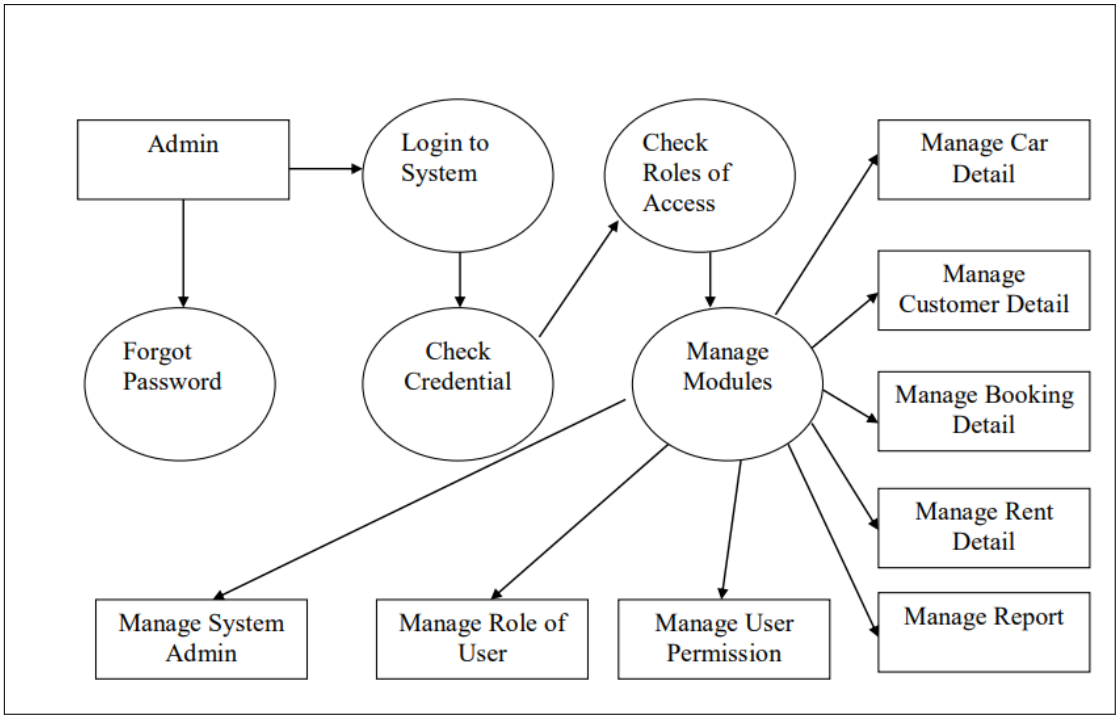

Figure 4.6: Level 2 DFD

4.2.3 ERD

An ERD (Entity-Relationship Diagram) is a visual representation of the relationships between entities in a database. It illustrates how different entities are related to each other through their attributes and associations. ERDs help design and understand database structures, ensuring data integrity and efficient data management in figure 4.7.

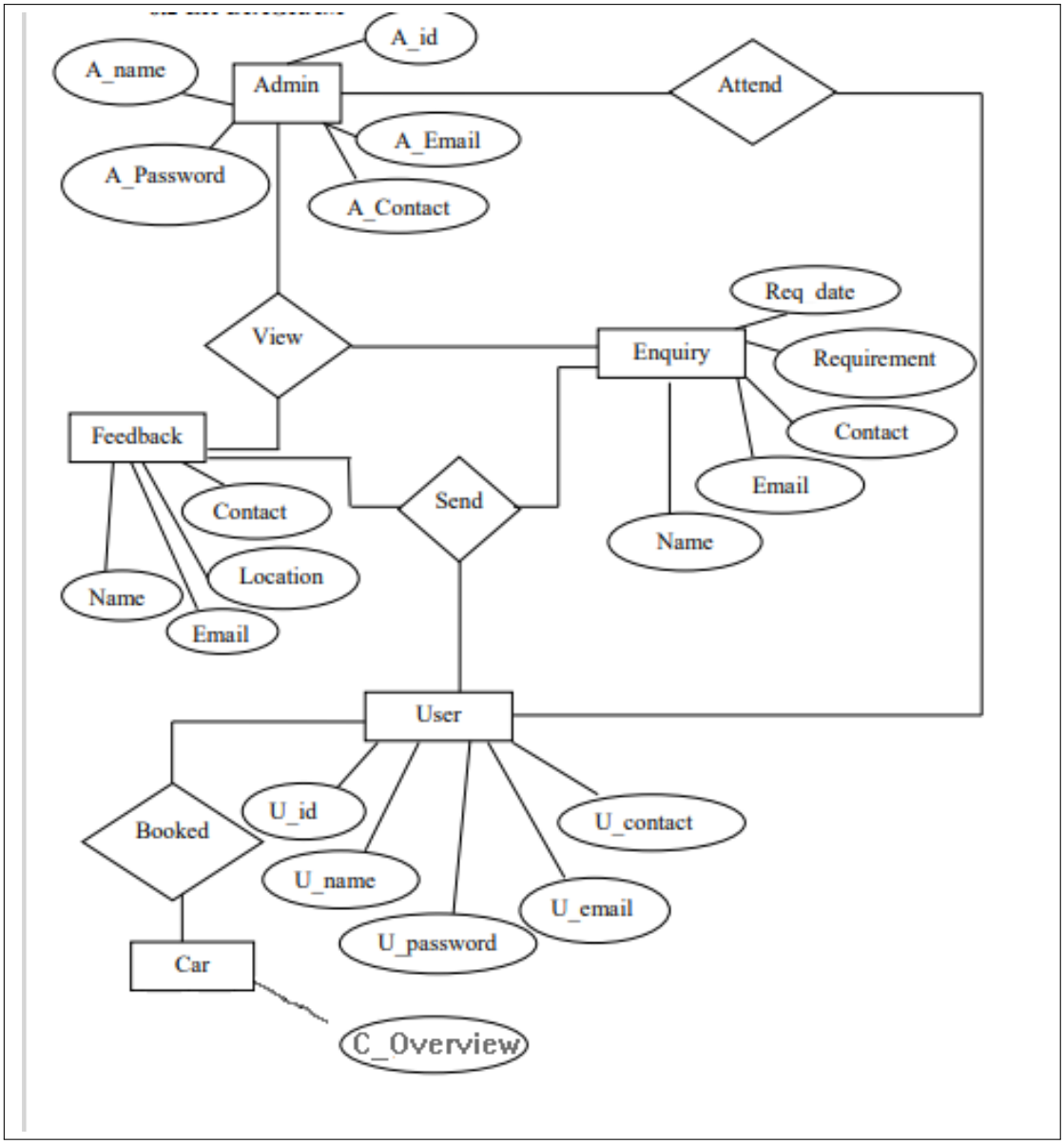

Figure 4.7: Entity Relationship Diagram

4.2.4 User Interfaces

Website Homepage

Users may access the car rental platform primarily through the homepage of the website. It features a clear and visually appealing design with menus that are simple to browse and displays featured autos and special deals. Customers may start their trip for a car rental from the homepage, look over their selections, and go forward with booking or login procedures in figure 4.8.

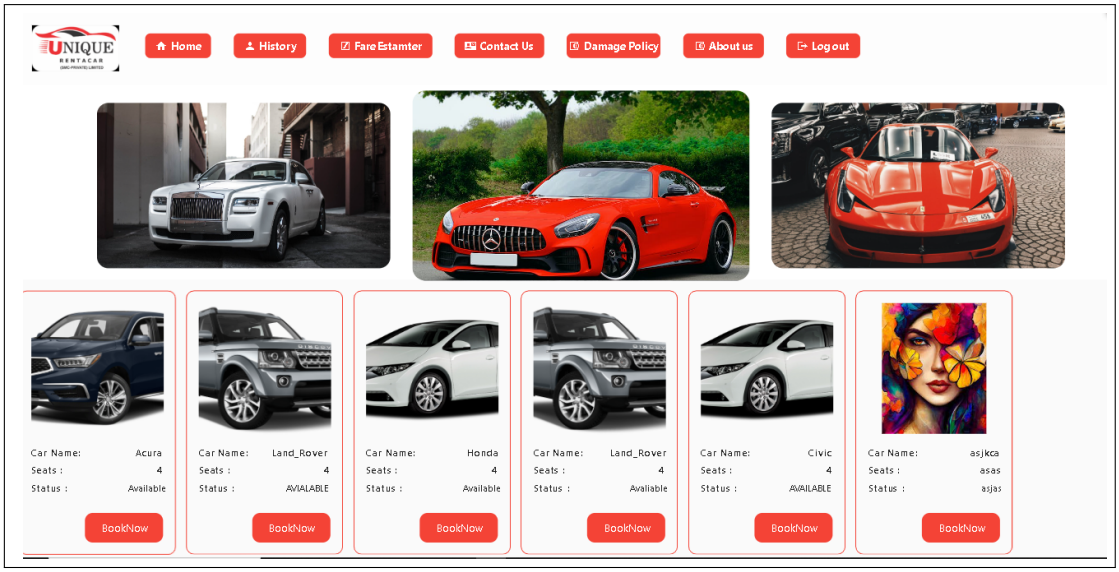

Figure 4.8: Home Page

AdminPanal

Only authorised administrators are allowed access to the Admin Panel, which is a secure and limited interface. Administrators may effectively control the whole automobile rental system with this interface. They can manage user accounts, answer customer support inquiries, add or delete cars from the inventory, and keep an eye on bookings. The Admin Panel simplifies backend activities so that administrators may maintain the integrity of the system and guarantee efficient company operations in figure 4.9.

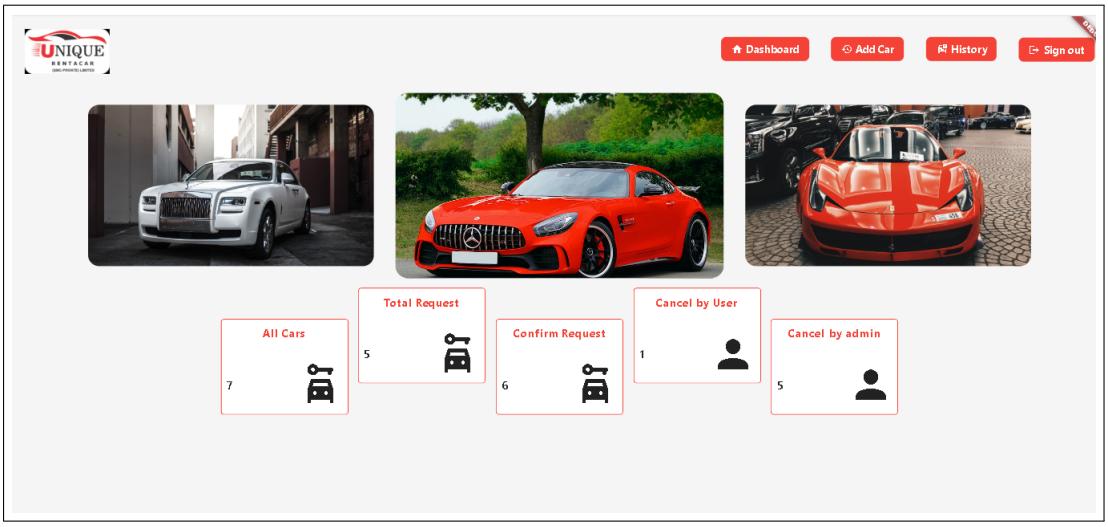

Figure 4.9: Admin Panal

Login

By facilitating user authentication, the Login interface makes sure that only registered users have access to personalised features and may make bookings. To access their accounts, modify current reservations, or examine their rental history, users can input their credentials, such as their email and password in figure 4.10.

Figure 4.10: Login

SignUp

New users can register for accounts in the automobile rental system via the SignUp portal. Users may register and join the platform by giving the required details, including name, email, and contact information. The system may gather crucial user information throughout the SignUp process, which will speed up future reservations and personalised services in figure 4.11.

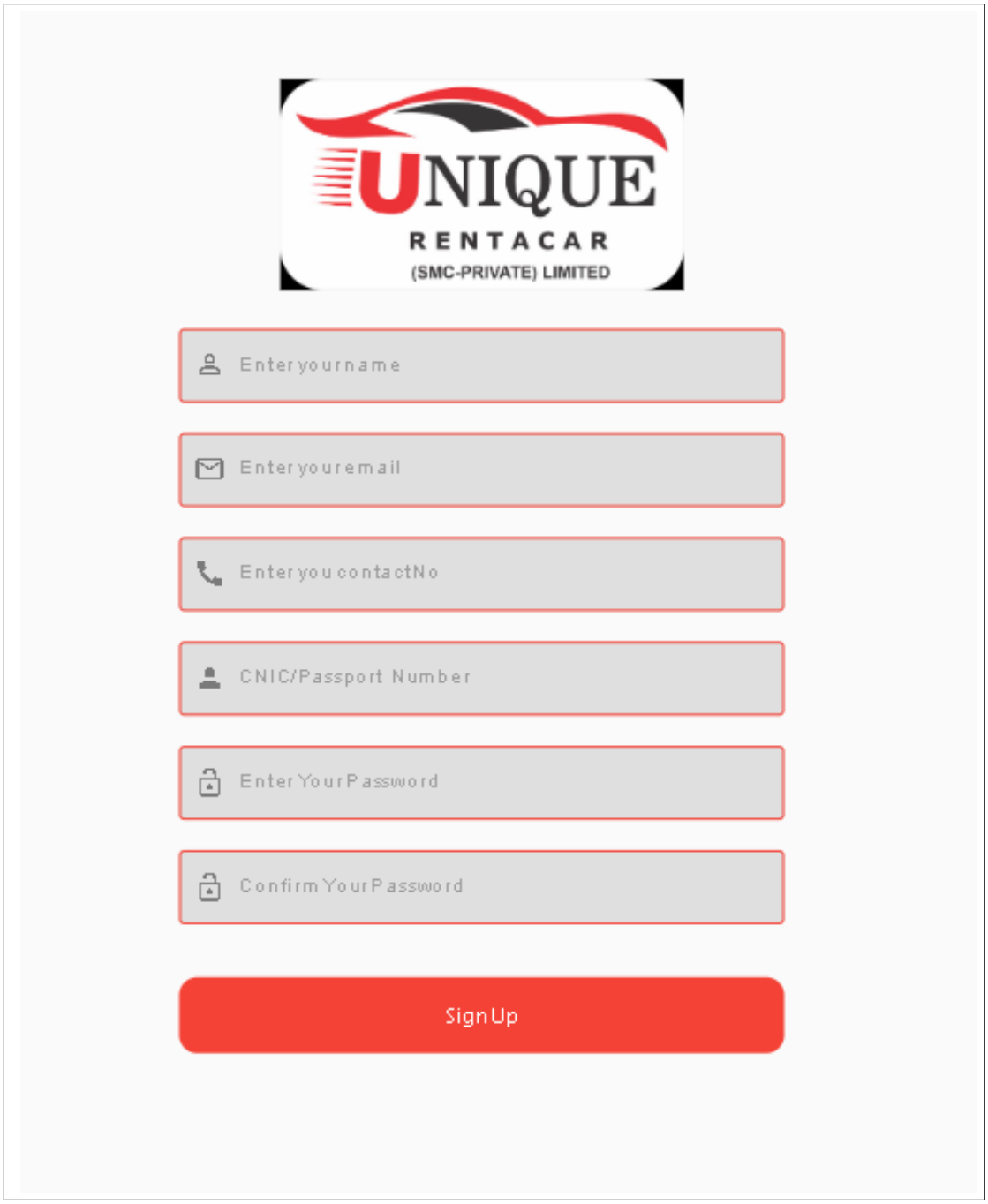

Figure 4.11: SignUp

Booking

Customers may use the Booking interface to look for available automobiles, choose preferred rental dates, and make bookings. Users may see vehicle data, explore a variety of automobile selections, and check availability in real time. The Booking interface walks users through the reservation procedure in a simple and welcoming manner in figure 4.12.

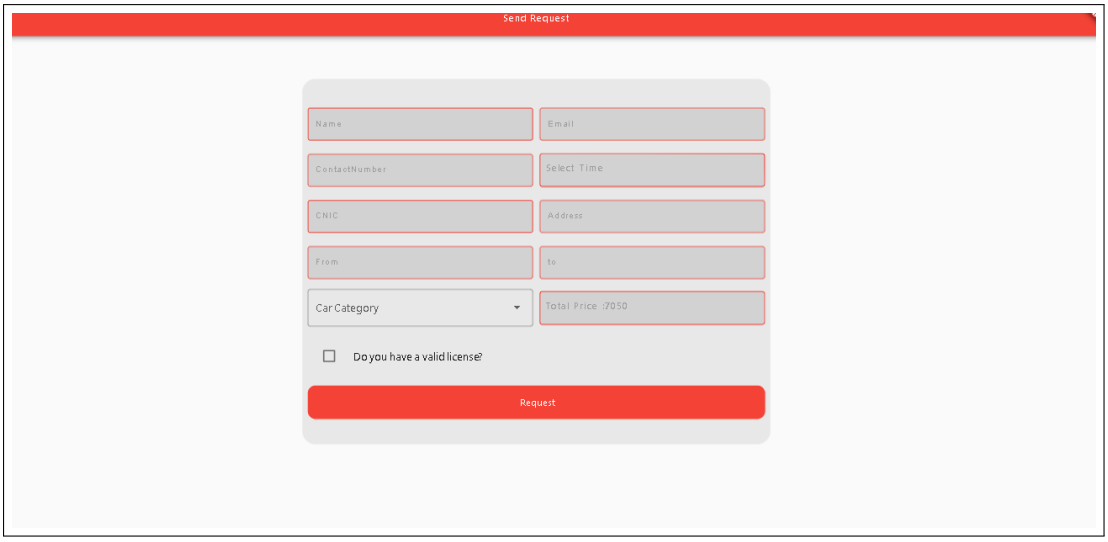

Figure 4.12: Booking

AddCar

Administrators are the only people with access to the AddCar interface, which gives them the ability to add new cars to the system's inventory. Administrators may provide vehicle characteristics, pictures, and rental information to make sure that buyers can pick from a variety of new cars show in figure 4.13.

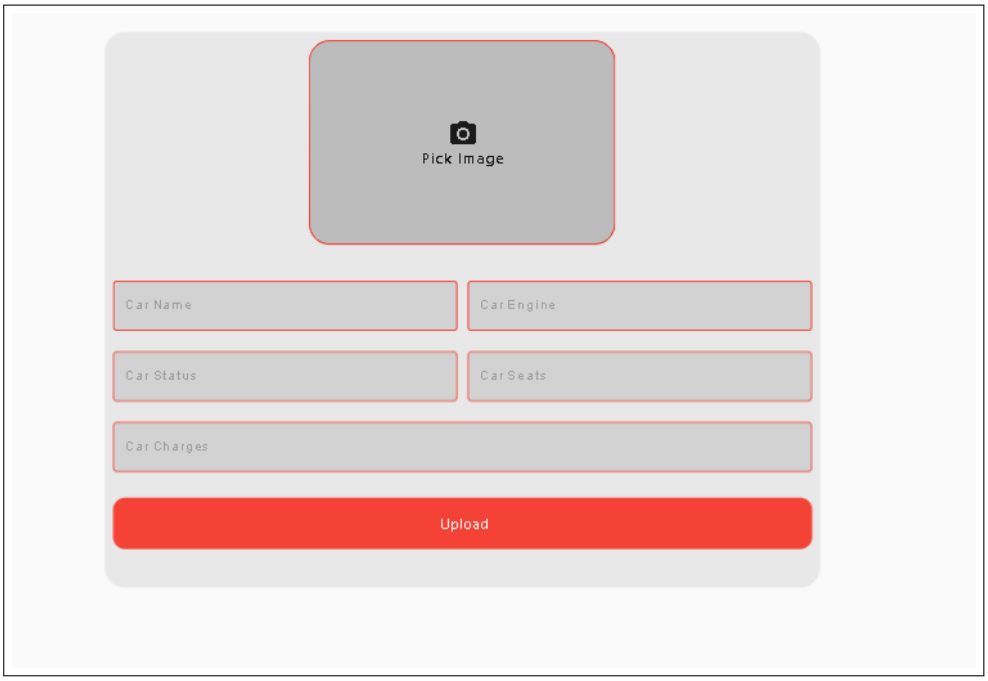

Figure 4.13: Add Car

AndroidAPP

For consumers who desire a more practical on-the-go experience, the Android App interface is a mobile application. Customers may use their Android handsets to access the automobile rental system thanks to the app's replication of key website functionality. Enabling mobile users expands the user base and improves accessibility show screen in figure 4.14.

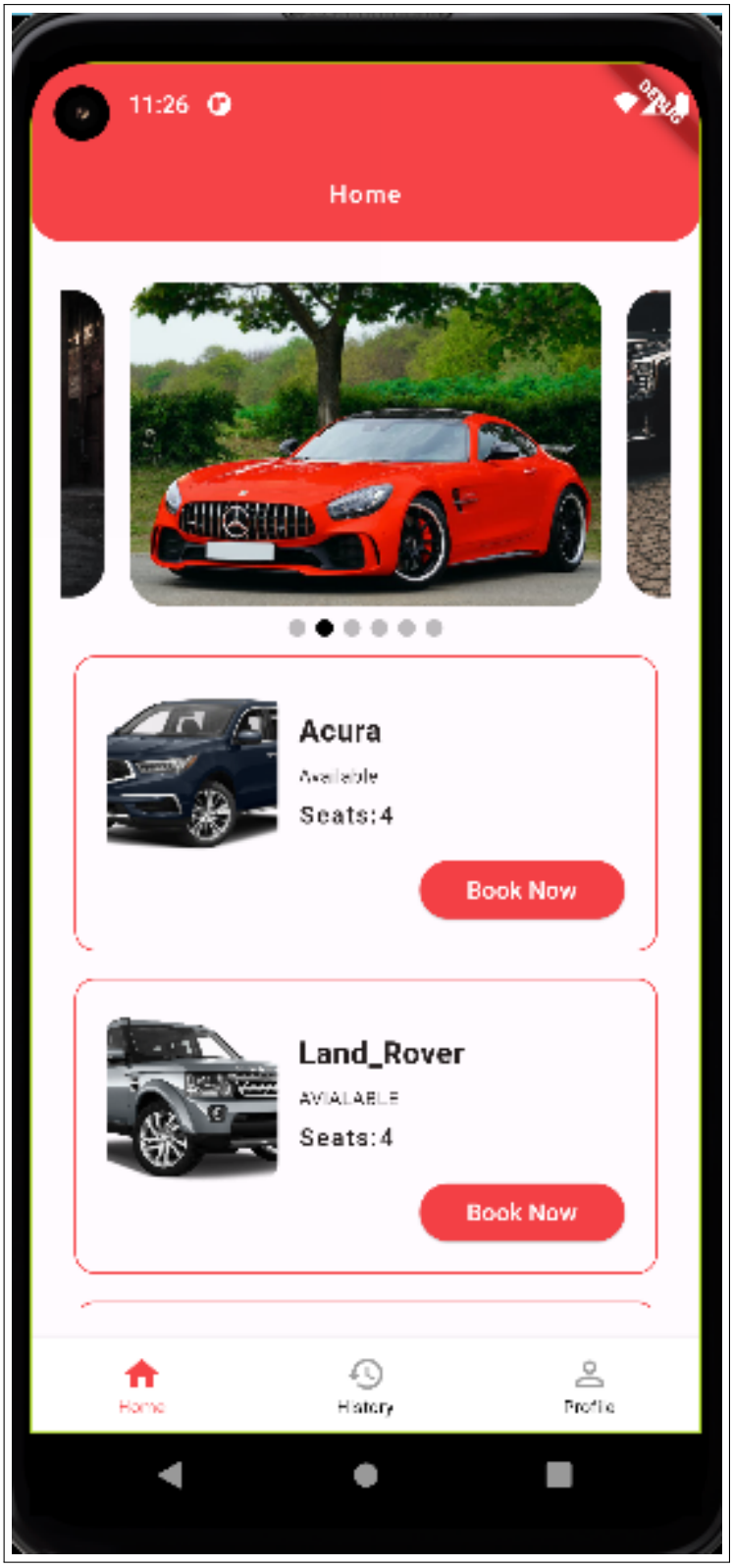

Figure 4.14: Application Home Page

Application Booking Page

The Android App's Booking-Page-Android interface is a particular screen that focuses on making smooth vehicle reservations. The automobile rental process is made more flexible and user-centric by allowing users to view vehicle details, select rental dates, and finish the booking procedure straight from their Android devices screen show in figure 4.15.

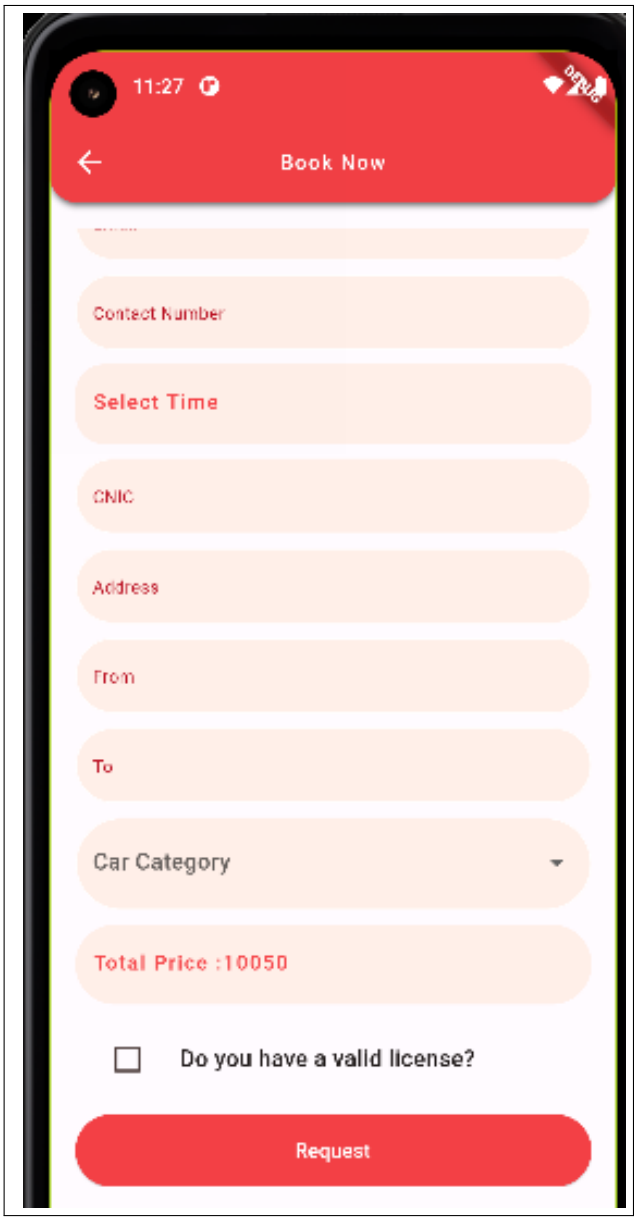

Figure 4.15: Application Booking Page

4.3 High Level Design

The High-Level Design, which describes the system's architecture and main components, is an essential blueprint for the Rent-A-Car initiative. Without going into detailed implementation details, it offers an abstract depiction of the system's architecture and features. User interfaces are managed by the presentation layer, while business logic and essential functions are handled by the application layer. Important data, such as user and vehicle information, is stored in the data layer. car Inventory oversees the car inventory, Booking administration handles reservations, and Admin Operations effectively runs the system. User Management handles user identification, registration, and account administration. The Android App will be integrated with the system, allowing mobile users to access key capabilities, according to the High-Level Design.The design lays the groundwork for further development and implementation by outlining the main components of the system and how they interact.With a shared knowledge of the system's architecture and a clear direction for the development process, this encourages good communication between the development team and stakeholders. It also enables the creation of a complete and efficient Rent-A-Car platform.

4.3.1 Deployment Diagram

The Rent-A-Car system thesis includes a deployment diagram that illustrates the physical deployment of the system and the distribution of software modules and components across hardware nodes. It shows how various software artefacts interact with the hardware infrastructure to make sure that it functions properly. Major components for the rent-a-car system, including the website, Android app, user management, vehicle inventory, and booking management, are divided across several hardware nodes, including web servers or cloud hosting services. The central data store that the modules share is also highlighted in the diagram. System administrators, developers, and other stakeholders can use the Deployment Diagram to plan and manage the deployment, guaranteeing scalability, redundancy, and efficient resource allocation. It offers useful insights into the physical structure of the system. The actual deployment of the system can be seen, which helps to provide a thorough knowledge of how the Rent-A-Car system is implemented and paves the way for reliable and effective system execution show in figure 4.16.

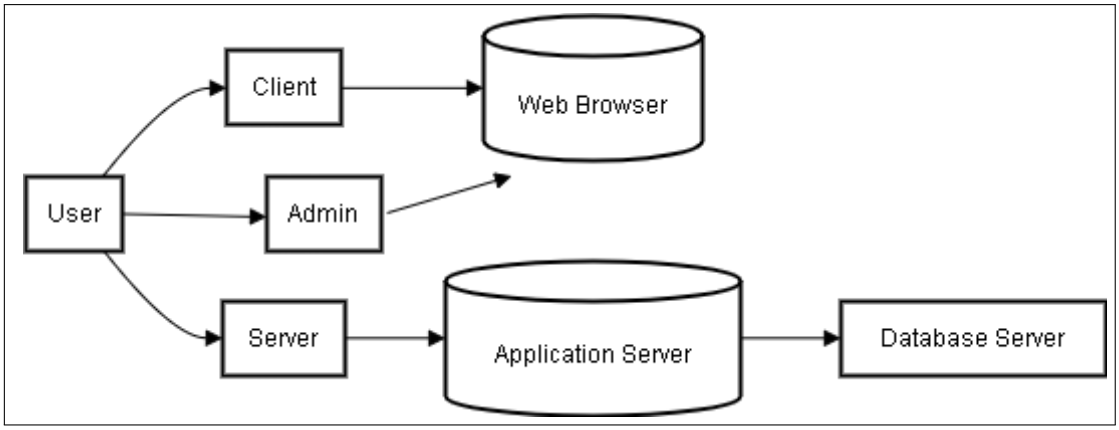

Figure 4.16: Deployment Diagram

4.3.2 Architecture Design

The project for the vehicle rental system focuses on developing a solid architecture that takes scalability, performance, and modularity into account. To manage user demands and expanding data quantities without sacrificing performance, scalability is essential. To scale horizontally, the design uses load balancing, distributed databases, and caching techniques. Another important factor is performance optimisation, which reduces latency and reaction times. To increase query efficiency and lessen database bottlenecks, strategies including data indexing, query optimisation, and caching are used. Faster content distribution is made possible by content delivery networks (CDNs), which also ensure optimal performance in various locales. By breaking the system down into separate modules for different functionalities, modularity is stressed in order to improve maintainability and adaptability. Collaboration between development teams is made simpler, which promotes faster iterations and more seamless system improvements. The automobile rental system's architecture guarantees a strong and adaptable foundation, well-equipped to fulfil the needs of a dynamic car rental market, by resolving scalability problems, optimising performance, and fostering flexibility show in figure 4.17.

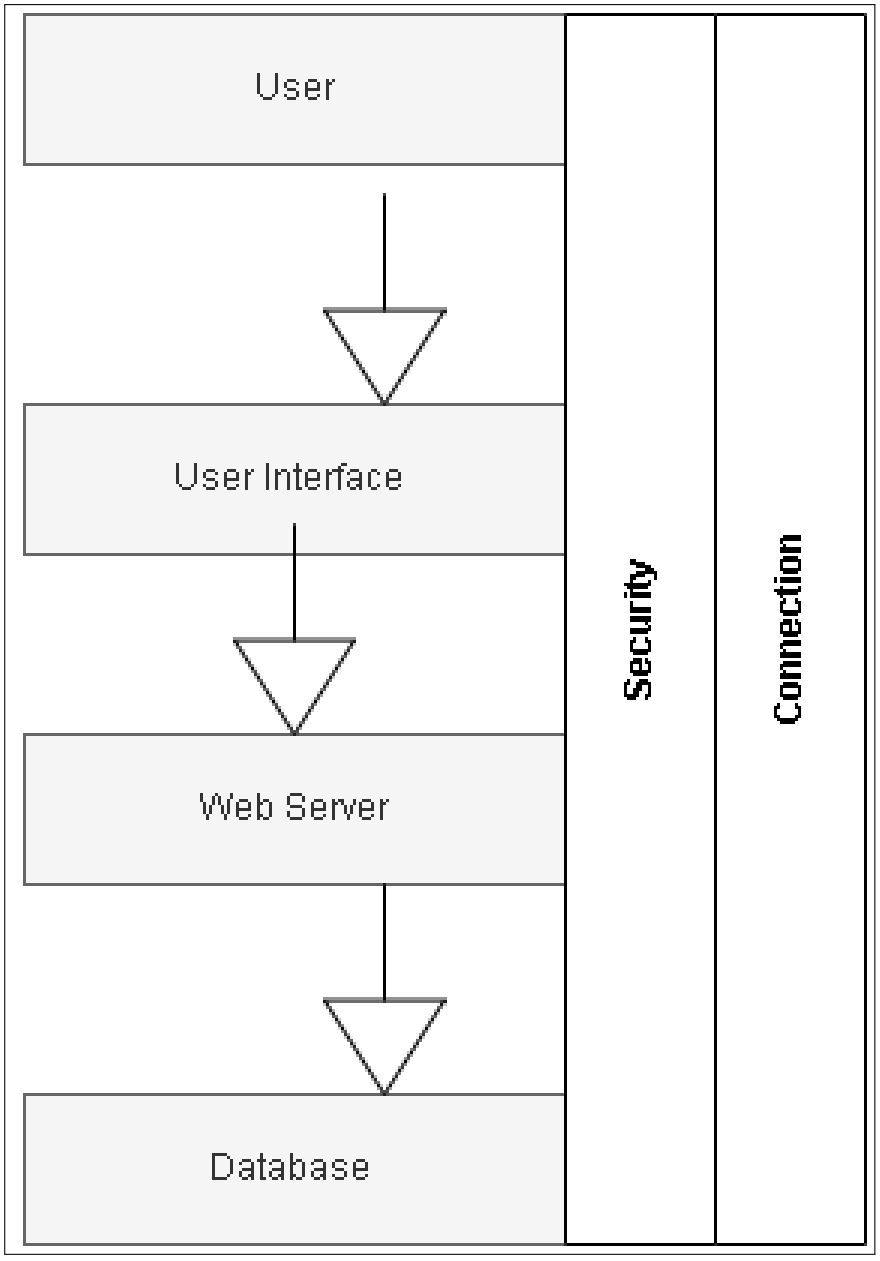

Figure 4.17: Architecture Design

Chapter 5

Software Testing

The car rental project goes into the crucial role of software testing in assuring the ultimate quality and dependability of the produced system in the Introduction to Software Testing chapter. To find and fix errors, faults, and discrepancies in the vehicle rental system, software testing is a crucial stage in the software development lifecycle. The chapter aims to confirm the system's conformance to given requirements and functional specifications by submitting it to rigorous testing scenarios and use-cases. Effective software testing reduces the possibility of errors and malfunctions, fostering trust in the functioning and performance of the system.The need of effective software testing cannot be overstated because the automobile rental platform serves a wide range of user demographics and manages sensitive data. This chapter examines the various testing methods used to evaluate the system's accuracy, usability, performance, and security. The goal is to make sure that the car rental system is strong, dependable, and capable of providing a first-rate user experience to both customers and administrators.

5.1 Unit Testing

Individual units or components of a codebase are tested in isolation as part of the core practise of unit testing in software development to assure their proper operation. Each unit is assessed independently to verify its behavior against expected outcomes, enabling early detection of bugs and facilitating code maintenance and refactoring. By automating these tests, developers can gain confidence in the reliability and correctness of their code, leading to more robust and maintainable software products show implementation in table 5.1.

Table 5.1: Unit Testing

Test Scenario ID		UT001		**Test Case ID**		TC001	
Test Case Description		Verify user rent car		**Test Priority**		High	
Pre-Requisite		User is logged in		**Post-Requisite**		Car successfully rented	
S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1	login	Valid user details (name, email, password)	Successfully Logged in	Login	Chrome	Pass	Login to User dashboard
2	Booking	User data	Save In History	Saved	Chrome	Pass	Booking Successfully

5.2 Integration Testing

Integration testing, as defined in the car rental project , includes assessing how various software components interact and are integrated to guarantee that the system as a whole functions as intended. The many modules and functions of the car rental system may be included in the integration testing components. For instance, interaction between the User Management and car Inventory modules will be investigated during integration testing to ensure that user information is accurately linked to the related car bookings. Additionally, the Booking Management module's communication will be tested as part of the integration process.In order to ensure consistent data interchange and user experience across both platforms, the integration between the Website and Android App will also be evaluated. Integration testing verifies the proper data flow, fluid communication, and seamless performance of the whole automobile rental system by closely examining the relationships between these interrelated components implementation show in table 5.2.

Table 5.2: Integration Testing

Test Scenario ID		2		**Test Case ID**		102	
Test Case Description		View and Book car		**Test Priority**		High	
Pre-Requisite		User is logged in		**Post-Requisite**		view and Book Cars	
S.NO	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1	Booking	User Data	Booking Request to admin	Booking Request	Chrome	Pass	Request Successful
2	Booking History	...	Save Booking History	Saved Booking History	Chrome	Pass	History Saved
3	Cancel Booking	...	Canceled Booking	Canceled	Chrome	Pass	Booking Canceled by User
4	Admin Add Cars Accept Request Saved History	...	Add Cars Accept Request View History	Car Added Request Accepted History Viewed	Chrome	Pass	Admin Add Car Accept Request View History

5.3 System Testing

System testing is a thorough software testing procedure that assesses the integrated system as a whole to make sure it complies with the requirements and performs as intended in a practical setting. System testing investigates the overall operation and behaviour of the entire system as opposed to integration testing, which focuses on interactions between specific components.To make that the automobile rental system fulfils its intended functional criteria and provides the anticipated user experience, functional testing is crucial.To guarantee appropriate functioning, it requires testing many use-cases and situations, including car booking, user authentication, and reservation administration. Non-functional testing examines aspects of the system other than its core functions, such as performance, usability, security, and compatibility. Performance testing, which includes load, stress, and endurance testing, evaluates the system's response, speed, and stability un-

der various situations. These tests are essential to ensuring the system's dependability and responsiveness during times of high usage. System testing is crucial in identifying possible problems and boosting overall dependability, performance, and user satisfaction by validating both functional and non-functional elements implementation show in table 5.3 .

Table 5.3: System Testing

Action	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
User Registration	Valid user details (name, email, password)	User is successfully registered	User registration successful	Chrome	Pass	New user can create an account without errors.
Car Search	Location, Pick-up dates	List of available cars	Correct car list displayed	Chrome	Pass	Car search functionality is working as expected.
Booking Confirmation	Car, Pick-up,	Booking confirmed, booking details displayed	Booking successful	Chrome	Pass	User can successfully book a car and see booking details.
User Cancel Booking	cancel	Cancel Booking	Canceled	Chrome	Pass	Booking Cancel Successfully

5.4 Deployment Testing

A crucial step in the software development lifecycle, deployment testing—also referred to as production or release testing—ensures a seamless and error-free transfer of a software

programme from the development environment to the production environment. Deployment testing's main goal is to confirm that the programme works as intended in the real world, with all of its dependencies correctly setup and running.In a project involving automobile rentals, the deployment testing stage seeks to guarantee a seamless transfer of the established system from the development environment to the production environment.Each test case in this phase of development has a particular goal to confirm the system's readiness for deployment. The installation verification test confirms the successful deployment of all required files, databases, and dependencies for the vehicle rental system on the target production servers. The compatibility test verifies the system's suitability for a range of hardware setups, operating systems, web browsers, and mobile devices, assuring the best possible user experience on all relevant platforms.The network connection test gauges how responsive the system is and how well it can communicate with other parties without experiencing network-related problems. Through testing under different user loads, load analysis determines the system's scalability and possible bottlenecks. In order to lower the possibility of customer discontent, the deployment testing step is essential for avoiding unanticipated downtime, performance issues, and compatibility issues. When the auto rental system is online, a successful deployment testing phase guarantees that it is stable, dependable, and fully functioning, providing a seamless and positive experience for both consumers and administrators implementation show in table 5.4.

Table 5.4: Deployment Testing

Test Case ID	**Objective**	**Description**	**Expected Output**
DT-001	Installation Verification	Validate the successful installation of the application	Application is installed without errors
DT-002	Compatibility Test	Check compatibility with different operating systems and browsers	Application functions correctly on all platforms
DT-003	Network Connectivity Test	Ensure the application can communicate with the server	Application connects to the server successfully

DT-004	Load Testing	Assess system performance under expected user loads	Application handles load without performance issues

5.5 Acceptance Testing

An essential phase in the software testing process is acceptance testing, which tries to ascertain if a software programme satisfies the needs and expectations of its target users or stakeholders. In a vehicle rental project, the acceptance testing phase is a vital time for collaboration between end users, clients, and stakeholders.End users actively interact with the system, offering insightful comments and observations that help to hone the programme for ready for production. Feedback from customers aids in the discovery of usability problems, user interface difficulties, and possible bottlenecks in the car rental procedure. In order to ensure that the system satisfies predetermined criteria and upholds contractual commitments, clients and stakeholders assess the system's alignment with the project's vision, business objectives, and industry standards. Open communication between the development team and end users or stakeholders is encouraged by the collaborative nature of acceptance testing. The system goes through several rounds to correct flaws or improvements, making sure it complies with end-user demands, corporate requirements, and industry standards before it is ready for production. The acceptance testing process makes that the automobile rental system is properly tested and modified before deployment to the production environment by incorporating comments and observations from end-users, clients, and stakeholders. This teamwork produces a strong, user-centric, and well-rounded product that offers a smooth and positive vehicle rental experience for everyone involved.

Chapter 6

Conclusion And Future Work

6.1 Conclusion

Lack of real-time vehicle rental for consumers was the issue that faced Vehicles Rental. They want a website that would make it simple and appealing for customers to acquire their service. Customers need to have had the option of seeing and renting cars with or without login into the system. To enable a good response and evaluation to start, requests should be entered immediately into the database and shown on the software inside the corporate premises. The answer was to provide Vehicles Rental a user-friendly online application that users could access and use on a variety of gadgets, including PCs, laptops, mobile phones, and tablets. By allowing admins to add and delete any automobiles from the site, the website is made to always be up to date. Any information a user enters on the website while making a request for a rental or purchase will be validated and saved. This will show up on the employees' software in real time. This programme has a user-friendly interface that makes it simple to retrieve rental requests and client data.

6.2 Future Work

To further enhance the service and meet the unique demands of the market, my future work for a car rental system may concentrate on a number of areas. Future projects might include the following:

6.2.1 Adaptation to regional payment systems

It would be more convenient for clients to rent a car if the car rental system in Pakistan was integrated with well-liked local payment methods and mobile wallets.

6.2.2 Programmes for promotions and loyalty

Promotional offers, discounts, and loyalty programmes might be implemented to draw in new clients and promote repeat business. The provision of special discounts to business clients or regular flyers may increase customer retention.

6.2.3 Enhanced Features for Mobile Apps

The entire user experience and convenience would be improved by spending money on cutting-edge mobile app features like real-time GPS monitoring of rental cars, mobile check-in and check-out, and seamless connection with customer care.

References

[1] "Zeeshan anwar (2012/2013) Research paper [online] Available:." https://www.studocu.com/row/document/arid-agriculture-university-rawalpindi/computernetworks/car-rental-management-systemproject-finalr/20854833. Accessed: 2022-09-29.

[2] "Amey Thakur (2021) Car rental system–[Online] Available." https://www.vixra.org/pdf/2108.0140v1.pdf. Accessed: 2022-09-25.

[3] "Interlink rent a car (Jun 9, 2020) Islamabad Car Rentals [online] Available:." https://rent-a-car-service.business.site/. Accessed: 2022-09-20.

[4] "Aswathy (Jan 02, 2020) Significance of mobile app for car rental business Available:." https://www.rentrabbit.io/blog/significance-of-mobile-app-for-car-rental-business. Accessed: 2022-10-05.

[5] "Kishore, K. (2020). Car Rental Mobile App Development – Cost and Key Features [Online] Available: ." https://www.octalsoftware.com/blog/car-rental-mobile-app-development-cost-and-keyfeatures/. Accessed: 2022-10-10.

[6] "SPROJECT NG (September 2019) Design and implementation of car rental system [online] Available: ." https://sprojectng.com/download/design-and-implementation-of-car-rental-system/. Accessed: 2022-10-25.

[7] "Amey Thakur (July 2021) Car Rental System– Research Gate [online] Available: ." https://www.researchgate.net/publication/353174644_Car_Rental_System. Accessed: 2022-10-15.