

ABSTRACT

The **Car Rental Management System (CRMS)** is a software solution designed to streamline car rental operations. It caters to the needs of car rental businesses by providing efficient management of vehicle inventory, reservations, customer information, and financial transactions. The system's user-friendly interface allows rental staff to navigate seamlessly, while its inventory management module keeps track of available cars, including details such as make, model, year, and condition. Customers can make reservations by rental staff in person, and DCRMS handles reservation requests, availability checks, and booking confirmations. The billing and payments module calculates rental charges based on factors like duration and mileage, generating invoices and maintaining financial records. Additionally, the system generates insightful reports on revenue, utilization rates, and popular car models. Security measures ensure data privacy and prevent unauthorized access. Future enhancements may include mobile app integration and additional payment gateways.

DECLARATION

We, Rajbir Ahmed (ID: 21225103170), Md. Abu Sayem (ID: 21225103140), Saurav Baral Joy (ID: 21225103147), Fahim Al Farid (ID: 21225103155), Anik Hasan (ID: 21225103255), Md. Shahariar Nabi (ID: 21225103164) are declaring that this project called “Car Rental Management System” and the work presented in it, are our own. We are confirming that:

We have done this whole work mainly while in student for a B.Sc. Engineering in CSE at this University. This has been clearly stated that any part of this software development project has previously been submitted for a degree or any other qualification at this university or any other institution. This is always clearly assigned that we have discussed the published work of others. We always stated that we have quoted from the works of others this. Without this quotation this thesis is entirely our own work. All main sources of help have been acknowledged by us. Where the project is based on work done by us jointly with others. Exactly what was done by others and what we have contributed by us, we have made that clear.

Rajbir Ahmed
(ID: 21225103170)

Md. Abu Sayem
(ID: 21225103140)

Saurav Baral Joy
(ID: 21225103147)

Fahim Al Farid
(ID: 21225103155)

Anik Hasan
(ID: 21225103255)

Md. Shahariar Nabi
(ID: 21225103164)

CERTIFICATION

This project “Car Rental Management System” report submitted by Rajbir Ahmed (ID: 21225103170), Md. Abu Sayem (ID: 21225103140), Saurav Baral Joy (ID: 21225103147), Fahim Al Farid (ID: 21225103155), Anik Hasan (ID: 21225103255), Md. Shahariar Nabi (ID: 21225103164) students of Department of Computer Science and Engineering, Bangladesh University of Business and Technology(BUBT), under the supervision of **M.M. Fazle Rabbi**, Assistant Professor, Department of Computer Science and Engineering has been accepted as satisfactory for the partial requirements for the degree of Bachelor of Science Engineering in Computer Science and Engineering.

(M.M. Fazle Rabbi)
Assistant Professor,
Department of CSE

Md. Saifur Rahman
Assistant Professor & Chairman(Acting)
Department of CSE

DEDICATION

Dedicated to our parents and teachers for all their love and inspiration.

ACKNOWLEDGEMENT

In the name of Allah, most Beneficent and most Merciful. Praise be to Allah who gave us strength, determination, staying power and ability to complete this work. We are thankful to Almighty Allah for helping us in each step. This project required a lot of work, patience and dedication. Still, implementation would not have been possible if we didn't have the support of many individuals and team members. Therefore, we would like to extend our sincere gratitude to all of them. First, we are thankful to our supervisor, **M.M. Fazle Rabbi**, Assistant Professor, Department of Computer Science and Technology, for providing necessary guidance, continuous support and motivation pertaining to the supervision and completion of this project. Without her superior knowledge and experience, the project would lack quality outcomes, and thus her support has been essential. We would like to express our sincere thanks to all our team members who devoted their time and knowledge and put all their efforts into the implementation of this project. Nevertheless, we express our gratitude towards our families and especially to our parents for their kind co-operation and encouragement which helped us in the completion of this project. May Allah reward all of them for their continuous support, for helping us sincerely and putting all their efforts into this whole time of our project.

Approval

This project “Car Rental Management System” Submitted by Rajbir Ahmed (ID: 21225103170), Md. Abu Sayem (ID: 21225103140), Saurav Baral Joy (ID: 21225103147), Fahim Al Farid (ID: 21225103155), Anik Hasan (ID: 21225103255), Md. Shahariar Nabi (ID: 21225103164) Department of Computer Science and Engineering (CSE), Bangladesh University of Business and Technology (BUBT) under the supervision of **M.M. Fazle Rabbi**, Assistant Professor, Department of Computer Science and Engineering has been accepted as satisfactory for the partial fulfillment of the requirement for the degree of Bachelor of Science (B.Sc. Eng.) in Computer Science and Engineering and approved as to its style and contents.

Supervisor:

M.M. Fazle Rabbi,
Assistant Professor,
Department of CSE
Bangladesh University of Business
and Technology (BUBT)

Chairman:

Md. Saifur Rahman
Assistant Professor and Chairman (Acting)
Department of CSE
Bangladesh University of Business
and Technology (BUBT)

TABLE OF CONTENTS

Abstract	i.
Declaration	ii.
Certification	iii.
Dedication	iv.
Acknowledgement	v.
Approval	vi.
Chapter 1: Introduction.....	1-2
1.1 Introduction	1
1.2 Existing model	1
1.3 Problem Statement.....	1
1.4 Motivation	1
1.5 Conclusion.....	2
Chapter 2: Existing System.....	3-5
2.1 Introduction	3
2.2 Existing System.....	3
2.3 Supporting Literature.....	3
2.3.1 Used Diagram.....	3-4
2.3.2 Technology Used.....	4
2.3 Analysis of Existing System.....	4-5
2.4 Conclusions	5
Chapter 3: Proposed Model.....	6-18
3.1 Introduction	6
3.2 Feasibility Study.....	6
3.2.1 Technical Feasibility.....	6
3.2.2 Operational Feasibility.....	7
3.2.3 Economical feasibility.....	7
3.3 Requirement Analysis.....	7-8
3.3.1 Non-Functional requirement.....	8-9
3.3.2 Functional requirement	9
3.4 System Design.....	10
3.4.1 SDLC Method	10-11
3.4.2 ER Diagram.....	12
3.4.3 Use Case Diagram	13
3.4.4 Level 0 Diagram.....	14
3.4.5 Context level Diagram.....	15
3.4.6 Activity Diagram.....	16
3.4.7 Sequence Diagram.....	17
3.6 Conclusions.....	18
Chapter 4: Experimental Result.....	19-20
4.1 Introduction.....	19
4.2 Result Analysis.....	19
4.3 Applications.....	19-20
4.4 Conclusions	20

Chapter 5: User Manual	21-25
5.1 Introduction.....	21
5.2 System Requirements	21
5.2.1 Hardware Requirements.....	21
5.2.2 Software Requirements.....	21
5.3 User Interface.....	21
5.5.1 Splash.....	22
5.5.2 login.....	22
5.5.3 Manage Car.....	23
5.5.4 Manage Driver.....	23
5.5.5 Manage Customer.....	24
5.5.6 Reservation of Car.....	24
5.5.7 Available Car.....	25
5.4 Conclusions	25
Chapter 6: Conclusion And Further Work.....	26
6.1 Conclusion.....	26
6.2 Further Work.....	26
References.....	27

Chapter 1

Introduction

1.1 Introduction

Car rental management system is a desktop-based system for a company that rents out car. This is a company that rents cars for a short period of time for a few days or week. To develop a desktop-based system that will help manage the business transactions of car renting. This desktop-based application is designed to simplify and enhance experience when it comes to renting and managing cars. This comprehensive solution is tailored to meet the specific needs of car rental company, providing a user-friendly interface and robust features to efficiently manage fleet, reservations and overall operations.

1.2 Existing model

The car rental agency for which we are developing this desktop-based application, that agency also works like the other existing agency work. But we notice some extra activities do for ensure their service. Those are

1. Note their financial statements in notebook.
2. Booked and available car details note down on a notebook.
3. They also communicate customer view to social media.
4. They don't give opportunity to drivers to handle the customer.
5. Fixed a rated fare as the customers requirement.
6. Keep all the information on notebook for use.

1.3 Problem Statement

Most of the car renting business of Bangladesh doesn't have an online reservation system for their business although the systems are not user friendly.

1. Unable to directly search for available and booked vehicles.
2. Trouble for financial manual transactions.
3. Hazard for listing available cars and drivers.
4. Can't have appropriate cars and drivers details properly.

1.4 Motivation

In today's fast-paced world, efficiency and convenience in service delivery are paramount. The motivation behind developing a Desktop-Based Car Rental Management System stems from the need to modernize the traditional car rental process. With the increasing demand for rental services, businesses face challenges in managing reservations, vehicle maintenance, and customer service effectively. DCRMS addresses these challenges by providing a centralized platform that automates operations, reduces manual workload, and enhances customer experience. It empowers car rental companies to manage their fleet more effectively and provides customers with a quick and easy way to book vehicles, fostering a more dynamic and responsive service environment.

1.5 Conclusions

The Car Rental Management System project aimed to enhance the efficiency and convenience of car rental services. In conclusion, the Car Rental Management System significantly improves the car rental process, benefiting both customers and the rental company. With continuous enhancements, it will remain a valuable asset in the car rental industry.

Chapter 2

Existing System

2.1 Introduction

Before we develop the system, we have to know about the Existing system are Belongs. We don't find any existing system. But there is model of man-made system. That system works like a paper work. That system has many drawbacks that is also not automated. Our system is able to recover all the drawbacks.

2.2 Existing System

The system which exists is too much convenient. So that system is not useful. That systems work procedure is like Note their financial statements in notebook. Booked and available car details note down on a notebook. They also communicate customer view to social media. They don't give opportunity to drivers to handle the customer. Fixed a rated fare as the customers requirement. Keep all the information on notebook for use.

2.3 Supporting Literature

supporting literature for a project report refers to relevant scholarly articles, books, and other sources that provide context, theoretical foundations, and evidence related to your project topic.

2.3.1 Used Diagram

1) Use case Diagram:

A use case diagram is a visual representation that illustrates the interactions between actors (users or external systems) and the system under consideration. It shows the various use cases (functional requirements) and their relationships. Here's a concise summary:

- (i) Actors: Represent users or external systems interacting with the software.
- (ii) Use Cases: Depict specific functionalities or actions the system performs.
- (iii) Associations: Show how actors relate to use cases (e.g., "actor A uses case B").
- (iv) System Boundary: Encloses all use cases within the system.

2) Level 0 Diagram:

A Level 0 Data Flow Diagram (DFD), also known as a context diagram, provides a high-level overview of the entire system or process being analyzed. It shows a single process node and its connections to external entities, offering a broad view with minimal detail.

3) Context level Diagram:

A context diagram, also known as a Level 0 data flow diagram, provides a high-level view of a system. It illustrates the interactions between the system being developed and its external entities (such as users, other systems, or processes). The focus is on how the system fits into its environment without delving into internal detail.

4) Activity Diagram:

An activity diagram visually represents the flow of actions or control within a system. It focuses on execution rather than implementation, showing how activities (operations) relate to one another.

5) Sequence Diagram:

A sequence diagram is an interaction diagram that details how operations occur—specifically, what messages are sent and when. It represents the chronological order of interactions between objects in a system, showing how different parts work together in a sequence to achieve a specific function.

2.3.2 Technology Used

Java

Java is a high-level, object-oriented programming language released in 1995 by Sun Microsystems (now owned by Oracle). Key features include:

1. Simple: Designed for easy learning and use, especially if you know C++.
2. Object-Oriented: Everything is an object, following OOP principles.
3. Distributed: Supports creating networked applications.
4. Compiled and Interpreted: Compiles to Java bytecode, then interprets it.
5. Robust and Secure: Emphasizes error handling and security.
6. Portable: Runs on any device with a Java Virtual Machine (JVM).
7. Multithreaded: Enables concurrent execution.
8. Dynamic: Supports dynamic class loading and reflection.

2.3 Analysis of Existing System

Mainly maximum Existing car rental agencies in Bangladesh complete their reservation system work by paper. That procedure is fully manual and record all information in website. The working process is described below.

Name	Working Procedure
Sheba.xyz	Customers can book car rental services through the Sheba.xyz website or mobile app. The platform provides a user-friendly interface where customers can choose their desired vehicle type, rental duration and any additional services. After selecting the options, customers provide personal details and confirm the booking. They cater to different needs such as daily commutes, intercity travel and special events. The platform also provides options for both self-drive and chauffeur-driven rentals. Customers can pay online through various payment methods including credit/debit cards, mobile banking and other digital payment options. Upon successful payment, customers receive a booking confirmation via email and SMS including details of the rental and contact information for the driver. (https://www.sheba.xyz/car-rental)

Name	Working Procedure
Haque Rent A car	Customers can book vehicles through their website, phone or by visiting their office. Management offers a range of rental durations: hourly, daily and monthly. The booking process involves selecting the type of vehicle, rental duration and providing necessary personal details. Also offers a diverse fleet including cars, minibuses, pickups, covered vans and trucks. They provide vehicles for various purposes such as personal use, business travel and tourism. Customers can choose from different vehicle models and brands depending on their requirements and budget. (https://www.haquerentacar.com/)
Zoom Rent A Car	Customers can make bookings through the Zoom Rent a Car website or by contacting their customer service via phone. The booking process involves selecting the type of vehicle, rental period and providing necessary details such as pickup and drop-off locations. Customers can choose between self-drive and chauffeur-driven options. Payment can be made online or at the time of vehicle pickup, with options including cash, credit/debit cards and online payment gateways. After booking, customers receive a confirmation with rental details and contact information for any further communication. (https://www.zoomrentacar.com.bd/)

2.4 Conclusions

In the conclusion of chapter 2 Existing system we can say the system which is exist that has so many lacking and using that is full of risk. Our system Car Rental Management system is more convenient and useful then the existing system. Our can be used with more easily and help the rental agency to make their more efficient and present a shameless rental experience.

Chapter 3

Proposed Model

3.1 Introduction

Car rental management system is a desktop-based system for a company that rents out car. This is a company that rents cars for a short period of time for a few days or week. To develop a desktop-based system that will help manage the business transactions of car renting. This desktop-based application is designed to simplify and enhance experience when it comes to renting and managing cars. This comprehensive solution is tailored to meet the specific needs of car rental company, providing a user-friendly interface and some features to efficiently manage fleet, reservations, and overall operations.

3.2 Feasibility Study

To develop a system we have to know first is the system is feasible for both developer and client. If the system is feasible for both then we have to approach farther else we can't go ahead to develop the system. Mainly a developer study three types of feasibility to complete a system. A feasibility study is crucial when evaluating the viability of a proposed project. The feasibility study serves as the bedrock upon which your project plan is built. It determines whether the project is practical and achievable. Decision-makers gain a comprehensive understanding of potential benefits, drawbacks, barriers, and constraints. This holistic view guides project direction and minimizes risks. The primary purpose of a feasibility study is to assess whether the project is not only viable but also beneficial. It evaluates technical, financial, legal, and market aspects.

3.2.1 Technical Feasibility

Technical feasibility in software development refers to assessing whether a software project can be successfully implemented based on available resources and technology.

Here the technology we need:

Hardware Technology: (For Developer and Client)

1. Processor: Intel Core i3
2. RAM: 2GB or more
3. Hard-disk: 256GB or more

Software Technology:

1. Programming Language: Java
2. Platform: Netbeans
3. Database: MySQL

All the hardware and software technologies are available for both developer and client. So this project is technology feasible.

3.2.2 Operational Feasibility

Manual Training: After completing manual training, There are enough manpower and skill for use and operate this application. Though this is operationally feasible for admin.

3.2.3 Economical feasibility

Economic feasibility in the context of software development refers to assessing whether the expected benefits of a proposed software system outweigh the costs associated with its development and operation. It involves analyzing factors such as cost savings, increased revenue, and profits.

Here the Development Cost of our proposed System

1. Design – 5000tk
2. Programming – 7000tk
3. Testing – 2000tk
4. Debugging – 1000tk
5. Database – 5000tk

Total cost $(5000+6000+2000+1000+5000) = 20000\text{tk}$

The upper costing is 20000tk which, the client agrees to pay. So this project is financially feasible.

3.3 Requirement Analysis

Purpose of Requirements Analysis:

1. Gathering Requirements: The initial step is to communicate with users to gather the requirements for the system.
2. Analyzing Requirements: In this phase, evaluate system feasibility and confirm with the quality assurance team that the requirements are testable.
3. Improving Requirement Quality: Enhance the clarity, completeness, and correctness of the requirements.
4. Modeling Requirements: Create visual representations (such as use cases or data flow diagrams) to understand system behavior.
5. Documenting and Reviewing Requirements: Properly document the requirements and review them with stakeholders.

Software Requirements:

1. Essential Needs: Software requirements represent the essential needs that the software must fulfill to deliver a quality product.
2. User Expectations: These requirements reflect what end users expect from the software to achieve their goals.
3. Fulfillment: The goal is to ensure that genuine and necessary requirements are met to address the problem at hand.

Activities in Requirements Analysis:

1. Problem Recognition: Identify the problem or need that the software aims to solve.
2. Evaluation and Synthesis: Thoroughly examine existing processes and potential solutions.
3. Modeling: Create visual representations (models) to understand system behavior.
4. Specification: Clearly define the requirements in a structured format.

5. Review: Validate the requirements with stakeholders to ensure accuracy and alignment with business goals.

An effective requirements analysis sets the foundation for successful software development. It ensures that the system meets user needs, aligns with business objectives, and operates within specified constraints.

3.3.1 Non-Functional requirement

Non-Functional requirements of our proposed Car Rental Management System are given bellow:

- 1) Handling 500 Concurrent Users:
 - (a) To handle concurrent users, consider the following:
 - i. Scalability: Design the system architecture to scale horizontally (adding more servers) or vertically (increasing server resources) as needed.
 - ii. Load Balancing: Distribute incoming requests across multiple servers to prevent overload.
 - iii. Caching: Use caching mechanisms (e.g., Redis, Memcached) to reduce database load.
 - iv. Optimized Queries: Optimize database queries to minimize response time.
 - v. Asynchronous Processing: Offload time-consuming tasks (e.g., email notifications) to background workers.
- 2) Helpful Error Messages and Tooltips:
 - (a) Provide clear error messages:
 - i. User-Friendly: Avoid technical jargon; explain errors in plain language.
 - ii. Contextual: Include details about what went wrong and how to resolve it.
 - (b) Tooltips:
 - i. Add tooltips to UI elements (buttons, forms) to guide users.
 - ii. Keep tooltips concise and relevant.
 - iii.
 - iv.
- 3) Consistent and Accurate Data:
 - (a) Data Validation: Validate input data to ensure consistency and accuracy.
 - (b) Database Constraints: Set constraints (e.g., unique keys, foreign keys) to maintain data integrity.
 - (c) Regular Data Audits: Periodically review and clean up data.
- 4) Access Controls to Prevent Unauthorized Data Access:
 - (a) Authentication and Authorization:
 - i. Implement user authentication (login) and authorization (role-based access control).

- ii. Restrict access to sensitive data based on user roles (admin, customer, driver).
- (b) Secure APIs and Endpoints:
 - i. Use tokens (JWT, OAuth) for API authentication.
 - ii. Validate user permissions before allowing access to specific endpoints.

3.3.2 Functional requirement

Functional requirements of our proposed Car Rental Management System are given bellow:

- 1) Car Management:
 - (a) Add, Delete, Edit Details of Available Cars:
 - i. Implement functionality to add new cars to the system.
 - ii. Allow administrators to delete cars that are no longer available.
 - i. Provide an interface to edit car details (e.g., model, make, year, rental rates).
 - ii.
- 2) Driver Management:
 - (a) Add, Delete, Edit Details of Drivers:
 - i. Create a section to add new drivers to the system.
 - ii. Enable administrators to remove drivers or update their information (e.g., license details, contact information).
- 3) Customer Management:
 - (a) Add, Delete, Edit Details of Customers:
 - i. Implement customer profiles where their personal information (name, contact, address) can be added.
 - ii. Allow administrators to delete customer records or modify their details.
- 4) Financial Statements:
 - (a) Record Keeping:
 - i. Develop a system to store financial data related to transactions (e.g., rental fees, payments, expenses).
 - ii. Generate reports (monthly, yearly) summarizing revenue, expenses, and profits.
- 5) Car Availability:
 - (a) Show Available and Occupied Cars:
 - i. Display a real-time view of cars that are currently available for rent.
 - ii. Indicate which cars are currently rented out or reserved.

3.4 System Design

System design is a crucial step in software engineering where we create a detailed plan or blueprint for a software solution. It encompasses defining the architecture, components, modules, interfaces, and data to meet specific requirements.

- 1) Definition of System Design:
 - (a) System design involves creating a coherent and efficient plan for a software system that addresses the needs and objectives of stakeholders.
 - (b) It transforms high-level concepts and requirements into tangible structures.
- 2) Components of System Design:
 - (a) Architecture: Define the overall structure of the system, including its layers, components, and interactions.
 - (b) Modules: Break down the system into smaller functional units (modules) that can be developed and tested independently.
 - (c) Interfaces: Specify how different modules communicate with each other.
 - (d) Data Design: Design the data storage and retrieval mechanisms (databases, file systems).
 - (e) Security Design: Plan for authentication, authorization, and data protection.
 - (f) Scalability and Performance: Consider how the system will handle increased load and traffic.
 - (g) Error Handling and Recovery: Define strategies for handling errors and ensuring system reliability.
- 3) Importance of System Design:
 - (a) Efficiency: Well-designed systems perform optimally and use resources effectively.
 - (b) Maintainability: A good design makes it easier to maintain and enhance the system.
 - (c) Scalability: Scalable designs accommodate growth without major rework.
 - (d) Reliability: Properly designed systems minimize failures and downtime.
 - (e) Security: Design considerations impact system security.
- 4) Case Studies and Real-World Examples:
 - (a) Study successful systems (e.g., Facebook, Google, Amazon) to understand their design principles.
 - (b) Learn from their architecture choices, scalability solutions, and data management strategies.

3.4.1 SDLC Method

The System Development Life Cycle (SDLC) is a structured approach used in software development to design, develop, and test high-quality software or hardware modules. For Car Rental Management System our process of SDLC is given below:

- 1) Planning Phase:
 - a) Objective: Define the project scope, goals, and requirements.
 - b) Activities:
 - i) Gather requirements for the car rental system (e.g., car details, customer profiles, financial statements).

- ii) Identify stakeholders (users, administrators, developers).
 - iii) Create a project plan, including timelines and resource allocation.
- 2) Analysis:
 - a) Objective: Understand the existing system (if any) and identify areas for improvement.
 - b) Activities:
 - i) Evaluate the current manual or legacy car rental processes.
 - ii) Conduct feasibility studies to assess the practicality of the proposed system.
 - iii) Define functional and non-functional requirements.
- 3) Design Phase:
 - a) Objective: Create a blueprint for the system.
 - b) Activities:
 - i) Architectural design: Define system components (e.g., car management, customer profiles).
 - ii) User interface design: Plan the layout, forms, and screens.
 - iii) Database design: Design the data schema for car details, rentals, and financial records.
- 4) Development and Testing Stages:
 - a) Objective: Build and validate the system.
 - b) Activities:
 - i) Coding: Develop the car rental system using appropriate programming languages (e.g., Java, C++, Python).
 - ii) Unit testing: Verify individual components (car management, customer profiles) for correctness.
 - iii) Integration testing: Ensure seamless interaction between system modules.
- 5) Implementation/Deployment:
 - a) Objective: Roll out the system for use.
 - b) Activities:
 - i) Deploy the desktop application on users' machines.
 - ii) Migrate data from the existing system (if applicable).
 - iii) Train users (administrators, staff) on using the new system.
- 6) Maintenance:
 - a) Objective: Ensure the system remains functional and up-to-date.
 - b) Activities:
 - i) Regularly update the system to fix bugs and enhance features.
 - ii) Monitor performance and address any issues.
 - iii) Provide user support and handle maintenance requests.

3.4.2 ER Diagram: The ER Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.

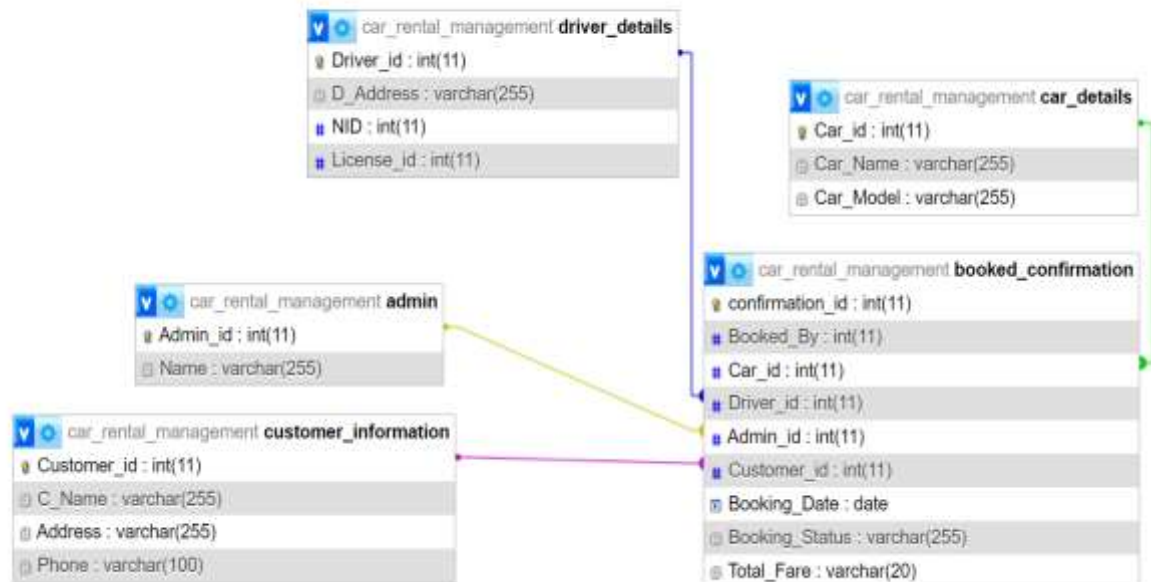


Figure:01

In figure 01, there is a sample ER diagram for desktop-based car rental management system. Admin, Cars details, Drivers details, Customer information and Booked confirmations details can be some table for stored data.

3.4.3 Use Case Diagram: A Use Case Diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

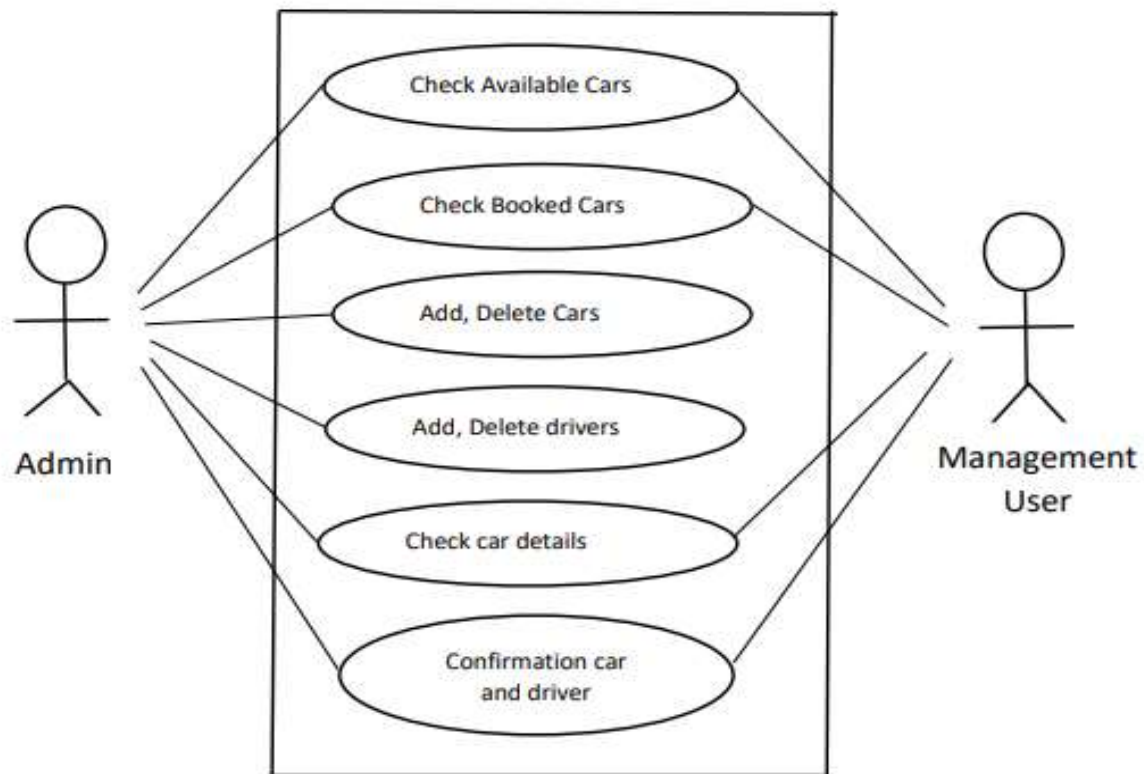


Figure:02

In figure 02, admin and management user are two actors. Admin can use all kind of functions and management user can use few of functions. Check available cars, check booked cars, add and delete cars and drivers, check car details, confirmation car and driver are main functions.

3.4.4 Level 0 Diagram: This is the highest-level DFD, which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes.

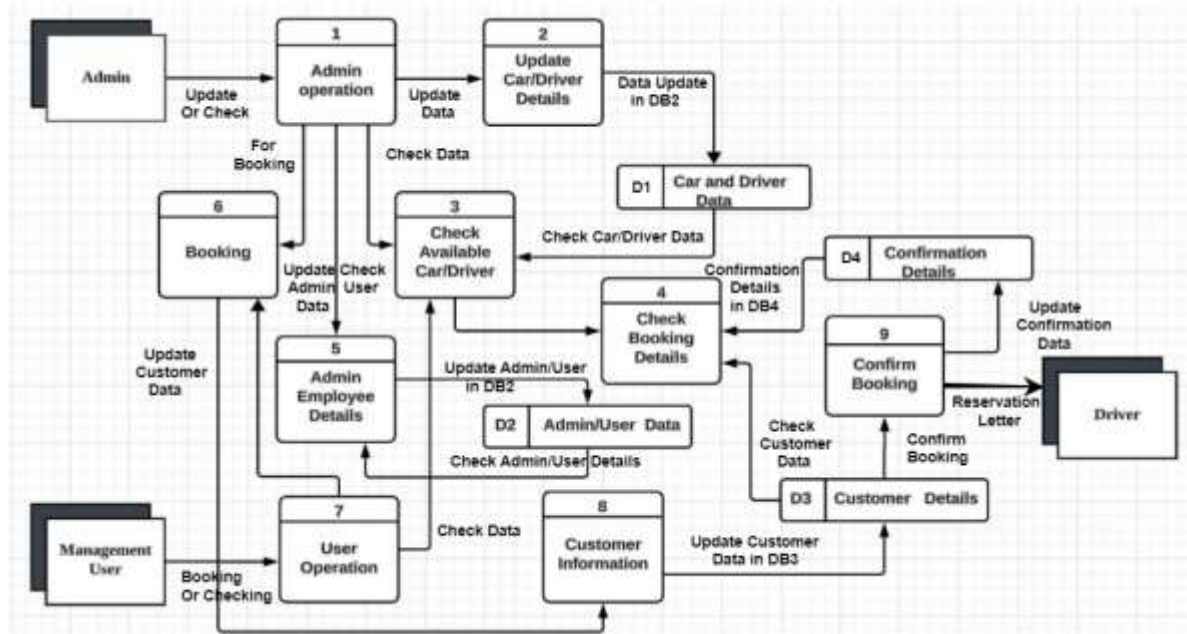


Figure:03

In figure 03, There are three entities and total nine processes. Admin, driver and management user are three entities. Admin operation, Update car/driver details, check available car/driver, Check booking details, Admin employee details, booking, user operation, customer information, confirm booking are processes. Car and driver data, admin and user data, customer details, confirmation details are database.

3.4.5 Context level Diagram: A context level diagram outlines how external entities interact with an internal software system. Context diagrams are high-level diagrams, meaning they don't go into the detailed ins and outs of the system. Instead, they map out an entire system in a way that's simple, clear, and easy to understand.

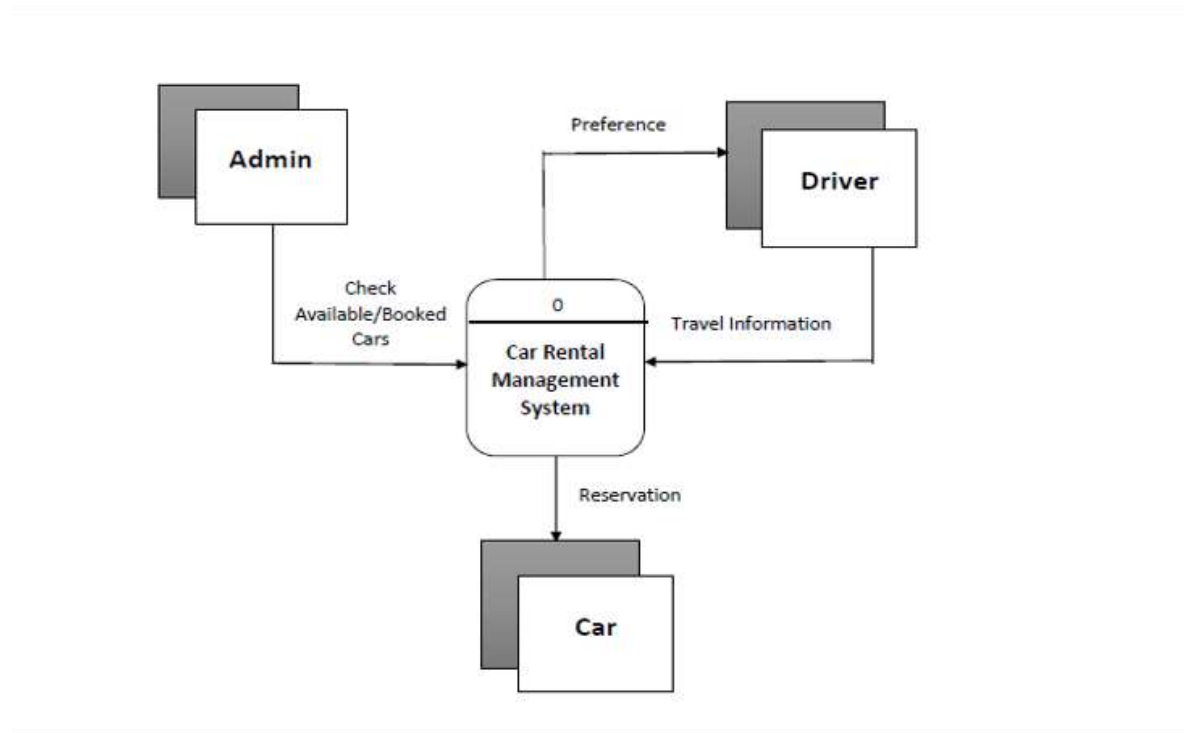


Figure:04

In figure 04, there are three entities and one process. All the drivers and vehicles information will be checked and car rental management system will be processed by admin section. Admin, driver and car are three entities and car rental management system is only process.

3.4.6 Activity Diagram: Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

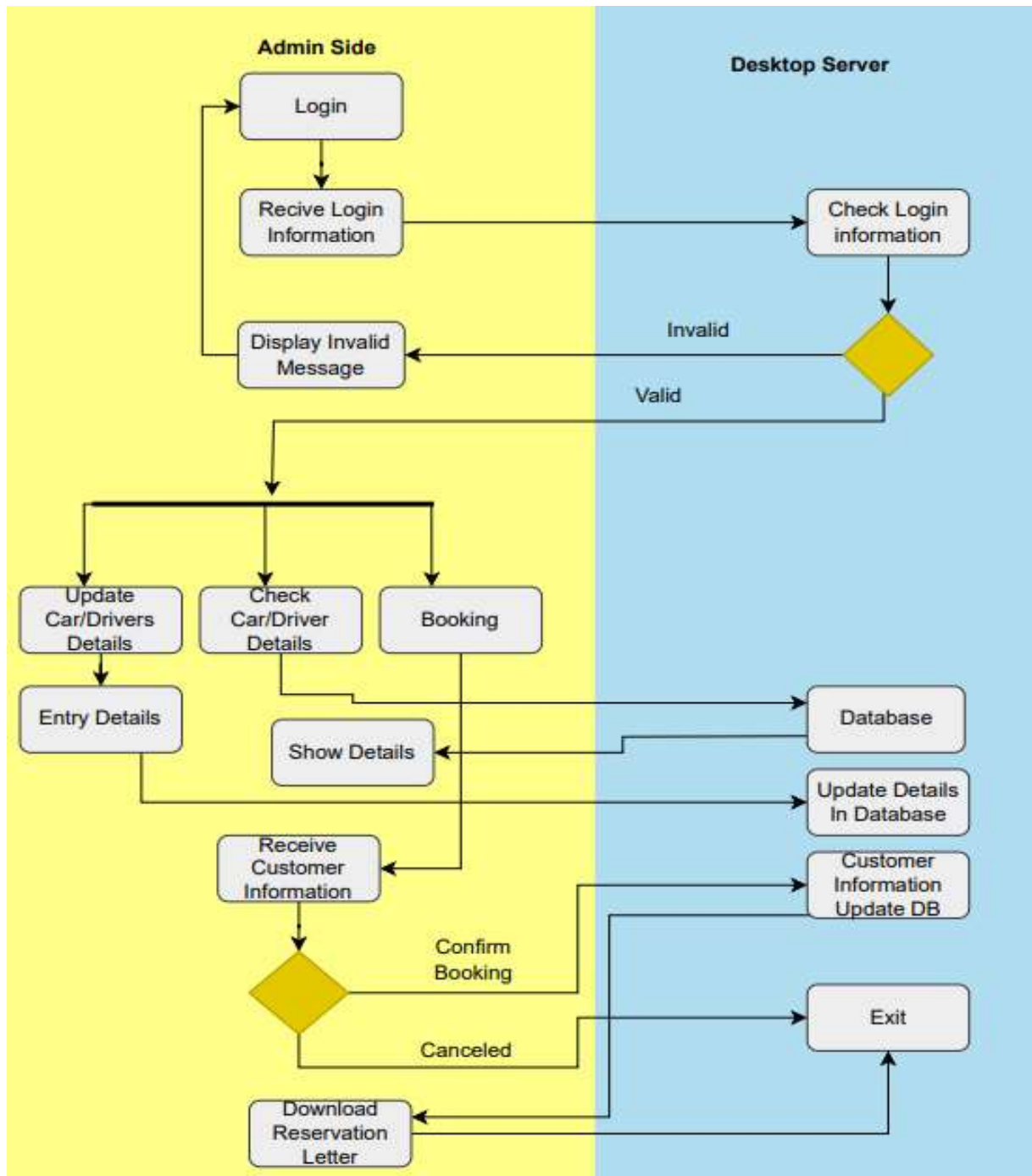


Figure:05

Admin and Desktop server are swim lanes. Here is a full process of reservation a car.

3.4.7 Sequence Diagram : Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

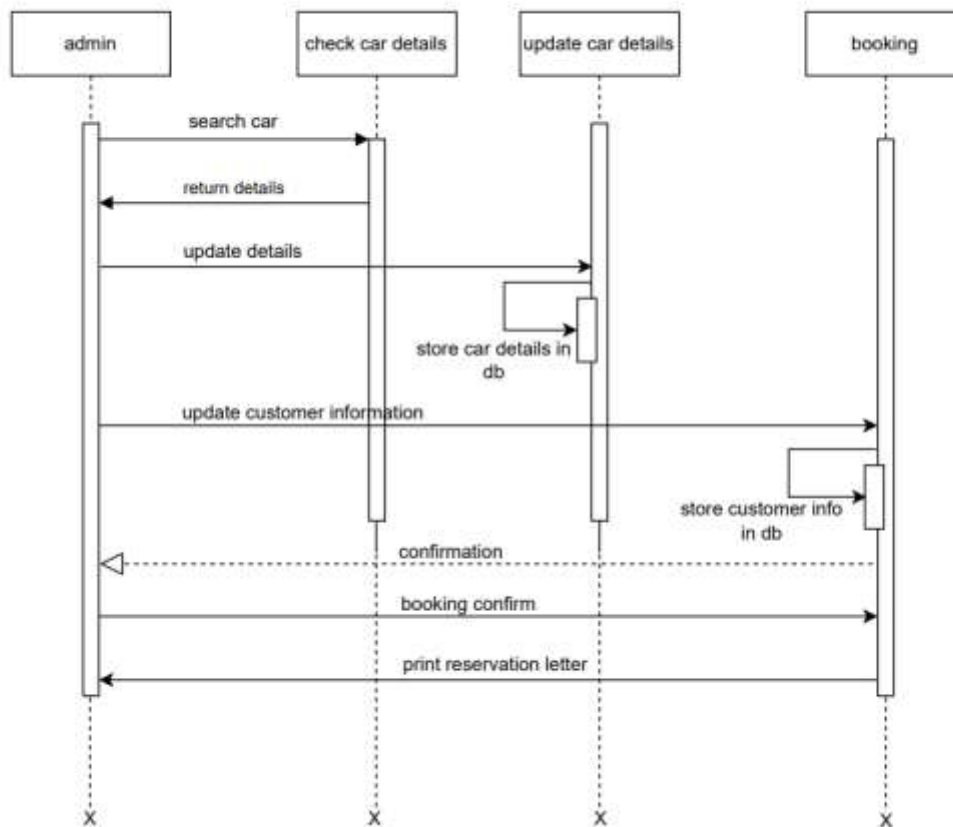


Figure:06

Here Admin is actor. Check car/driver details , update car/driver details, booking(reservation) are objects. Admin can access all processes.

3.6 Conclusion

Our proposed car rental management system follows the principles of the SDLC. Throughout the development process, we meticulously adhered to the various phases, ensuring a robust and efficient system. Our system aims to streamline car rental operations by providing an efficient desktop application. It facilitates tasks such as vehicle booking, inventory management, and customer interactions.

1) SDLC Alignment:

a) We followed the SDLC phases:

- i) Requirements Analysis: Gathered detailed requirements from stakeholders.
- ii) Design and Planning: Designed the system architecture and user interfaces.
- iii) Implementation: Developed the desktop application using appropriate technologies.
- iv) Testing: Rigorously tested the system to ensure functionality and reliability.
- v) Deployment: Rolled out the system for use by car rental staff.

2) Key Features:

- a) Vehicle Reservation: Customers can easily reserve cars through the desktop application.
- b) Inventory Management: Staff can track available vehicles, maintenance schedules, and availability.
- c) User Profiles: The system maintains customer profiles, transaction history, and preferences.

3) Business Impact:

- a) Our proposed model enhances efficiency, reduces manual paperwork, and improves customer service.
- b) By automating processes, the car rental company can serve customers more effectively.

In conclusion, our car rental management system, aligned with the SDLC, promises to revolutionize the company's operations and enhance customer satisfaction.

Chapter 4

Experimental Result

4.1 Introduction

The experimental results for our desktop-based car rental management system represent a critical phase in assessing the system's functionality, performance, and usability. Through rigorous testing and evaluation, we gained insights into various aspects of the system. In this section, we present key findings, highlight achievements, and discuss any challenges encountered during the experimentation process.

4.2 Result Analysis

The following areas were explored during the experiments:

- 1) **Functionality Testing:**
 - a) We verified that all core features—such as vehicle reservation, inventory management, and user profiles—functioned as intended.
 - b) Test scenarios covered booking, cancellation, and modification of reservations.
 - c) Any discrepancies or unexpected behavior were thoroughly documented.
- 2) **Performance Metrics:**
 - a) We measured system response times, resource utilization, and scalability.
 - b) Performance benchmarks were established to assess the system's efficiency under varying loads.
 - c) The goal was to ensure optimal responsiveness even during peak usage.
- 3) **Usability and User Experience:**
 - a) User feedback played a crucial role in evaluating the system's usability.
 - b) We conducted user acceptance testing (UAT) to gauge how well the interface met user expectations.
 - c) Improvements were made based on user suggestions.
- 4) **Security and Data Integrity:**
 - a) The system underwent security testing to identify vulnerabilities.
 - b) Measures were taken to safeguard sensitive data, prevent unauthorized access, and ensure data integrity.

In the subsequent sections, we delve into specific results, analyze trends, and draw conclusions based on empirical evidence. Our experimental findings guide further refinements and enhancements to deliver a robust and reliable car rental management system.

4.3 Applications

Desktop-based car rental management systems find applications in various scenarios within the car rental industry. Let's explore some key use cases:

- 1) **Fleet Management:**
 - a) Car rental companies use desktop systems to efficiently manage their vehicle fleets. This includes tracking vehicle availability, maintenance schedules, and ensuring optimal utilization.
 - b) Fleet managers can monitor vehicle status, allocate resources, and plan maintenance activities.

- 2) Reservation and Booking Management:
 - a) Desktop systems handle reservations and bookings seamlessly. Staff can reserve vehicles online, check availability, and receive confirmation.
 - b) Rental staff can manage bookings, allocate vehicles, and handle modifications or cancellations.
- 3) Billing and Invoicing:
 - a) Car rental software generates accurate invoices based on rental duration, vehicle type, and additional services.
 - b) It ensures transparent billing for both customers and the rental company.
- 4) Customer Database and Profiles:
 - a) Desktop-based systems maintain customer records, including personal details, rental history, and preferences.
 - b) This information helps personalize services, offer loyalty discounts, and improve customer satisfaction.
- 5) Reporting and Analytics:
 - a) Car rental managers rely on desktop software to generate reports on revenue, utilization rates, customer trends, and operational efficiency.
 - b) Analytics help identify growth opportunities and areas for improvement.
- 6) Rate Management:
 - a) Car rental rates can vary based on factors like demand, seasonality, and location. Desktop systems allow dynamic rate adjustments.
 - b) Rate tables ensure consistent pricing across different channels.
- 7) Maintenance Tracker:
 - a) Rental companies use desktop software to schedule and track vehicle maintenance, inspections, and repairs.
 - b) Preventive maintenance reduces downtime and enhances safety.

4.4 Conclusions

In conclusion we can say that our Car Rental management system is pass successfully and ready to use. Now we have to go for final deployment and train people to operate the system.

Chapter 5

User Manual

5.1 Introduction

The Car Rental Management System is a comprehensive software solution designed to streamline car rental operations. Whether you're a car rental company, an administrator, or a customer, this user manual provides essential information to navigate the system efficiently.

The purpose of this user manual is to:

1. Familiarize Users: Introduce users to the system's features, functionalities, and user interface.
2. Guide Operations: Provide step-by-step instructions for common tasks such as booking, vehicle management, and reporting.
3. Ensure Efficiency: Help users maximize the system's capabilities and minimize errors.
4. Rental Staff: Learn how to manage reservations, handle customer inquiries, and maintain the vehicle fleet.
5. Administrators: Understand system configuration, security settings, and reporting.

User manual is our go-to resource for using the Car Rental Management System effectively

5.2 System Requirements

System requirements refer to the specific hardware, software, and other conditions necessary for a particular application or software program to run effectively on a computer or device. These requirements ensure that the system can handle the demands of the software and provide optimal performance

5.2.1 Hardware Requirements

Hardware Technology: (For Developer and Client)

- a. Processor: Intel Core i3
- b. RAM: 2GB or more
- c. Hard-disk: 256GB or more

5.2.2 Software Requirements

Software Technology: (For Developer and Client)

- a. Programming Language: Java
- b. IDE: Netbeans
- c. Database: MySQL

5.3 User Interface

The user interface (UI) is the visual and interactive part of a software application or system that allows users to interact with it. It serves as the bridge between users and the underlying functionality of the software.

5.5.1 Splash

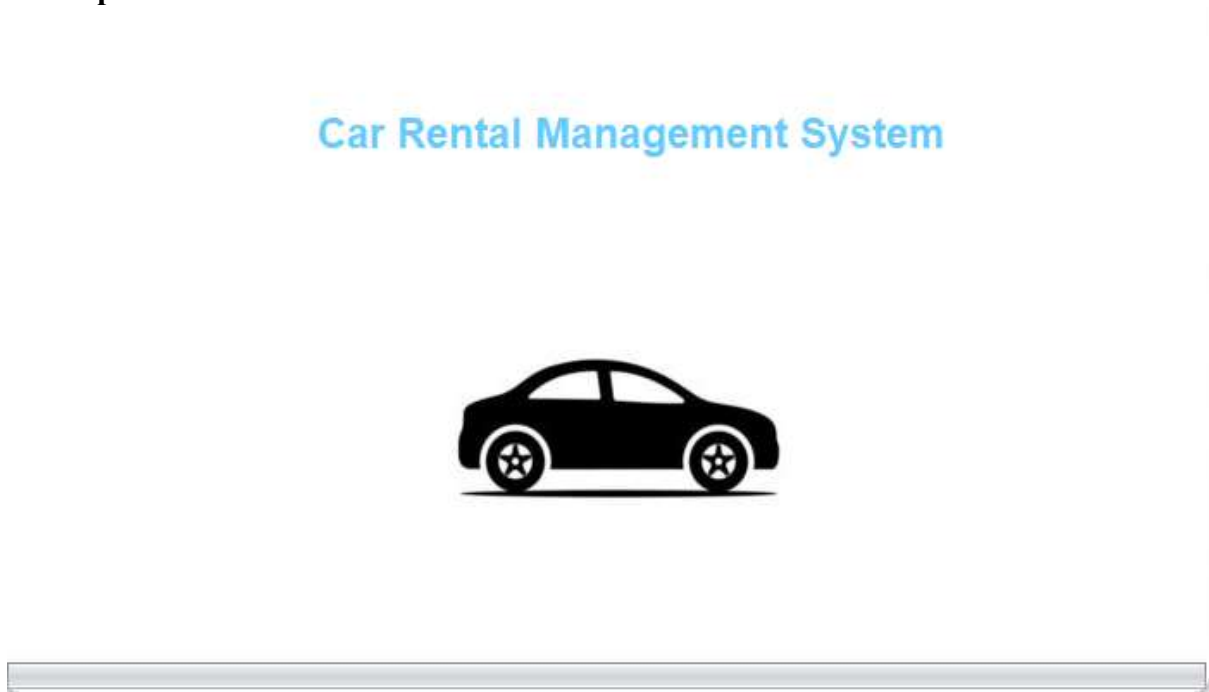


Figure: 07

This is first Interface this software.

5.5.2 Login

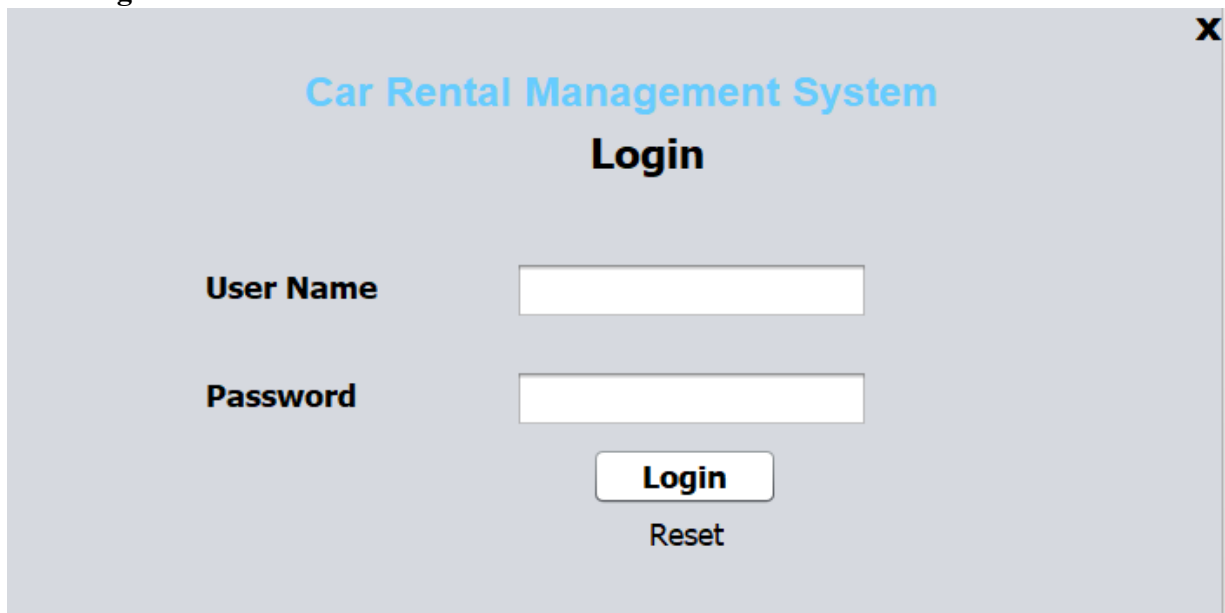
The login form is displayed on a light grey background. At the top, it has the title "Car Rental Management System" in blue, followed by the word "Login" in bold black. There are two input fields: "User Name" and "Password", both with white text labels and empty white input boxes. Below the password field is a "Login" button with black text, and a "Reset" link with black text. A small "X" icon is located in the top right corner of the form area.

Figure: 08

In login page, user gives user name and password.

5.5.3 Manage Cars

Figure: 09

Here manage all cars details. User can add new cars, edit cars details, delete cars and also reset information.

5.5.4 Manage Drivers

Figure: 10

Here manage all driver details. User can add new drivers, edit driver details, delete and also reset drivers information.

5.5.5 Manage Customers

Figure: 11

Manage all customer details. User can add, edit details, delete and also reset information of customers.

5.5.6 Reservation of Car

Figure: 12

Here confirm all reservations. Add all information of reservations.

5.5.7 Return Cars

RESERVAT	CAR_REGI...	CUSTOMER	DESTINATL	DESTINATL	PRICE	RESERVAT	RETURN	DRIVER_LI
2	B00123...	Rajbir	Dhaka	London	40000	2024-05...	2024-05...	AB00001...
6	B00123...	Sakib	Dhaka	London	100	2024-05...	2024-05...	AB00001...
8	B00070...	Rajbir	achu	poam	9000	2024-05...	2024-05...	AB00001...
9	B00182...	Rajbir	port	nara	1000	2024-05...	2024-05...	AB00001...
10	B09100...	Rajbir	ctq	dhaka	4300	2024-05...	2024-05...	AB00001...
11	B09100...	Rajbir	Dhaka	Sylhet	10000	2024-05...	2024-05...	AB00059...
12	B09182...	Sakib	mirpur	Rhob	1000	2024-05...	2024-05...	AB00001...

RETURN_ID	CAR_REGISTR...	CUSTOMER_N...	RETURN_DATE	DRIVER_LICE...	DELAY	FINE
1	B00123456...	Rajbir	2024-05-20	AB000011134	0	0
2	B00123456...	Rajbir	2024-05-19	AB000011134	0	0

Figure: 13

Add all details of return cars.

5.4 Conclusions

In this project, we focused on creating an intuitive and user-friendly interface for our application. The UI design plays a crucial role in enhancing user experience and ensuring efficient interaction. Let's recap the key points:

- 1) Final Product Showcase:
 - a) We successfully implemented the UI components, including buttons, forms, navigation menus, and feedback elements.
 - b) Users can now interact seamlessly with the application, thanks to the thoughtfully designed interface.
- 2) Impacts and Benefits:
 - a) The well-crafted UI positively influences user engagement, productivity, and satisfaction.
 - b) By prioritizing usability and consistency, we've created an environment where users can achieve their goals effortlessly.
- 3) Reflections and Learnings:
 - a) Throughout the design process, we encountered challenges and made informed decisions.
 - b) Reflecting on our choices helps us refine our skills and adapt to future projects.

Chapter 6

Conclusion And Further Work

6.1 Conclusion

The Desktop-based Car Rental Management System offers a reliable and efficient solution for managing car rental businesses. By leveraging desktop technologies, this system ensures optimal performance, offline functionality, and a user-friendly experience for users. Designed to be a comprehensive solution for car rental businesses, providing administrators with the tools they need to efficiently manage their fleets, reservations, and overall operations. Embrace the future of car rental management with this intuitive and robust desktop application.

6.2 Further Work

The future of car rental management systems holds exciting possibilities. Here are some areas where advancements and improvements are likely to occur:

- 1) Predictive Analytics and Demand Forecasting:
 - a) Car rental systems will leverage data analytics to predict demand patterns.
 - b) By analyzing historical data, weather conditions, events, and seasonal trends, companies can optimize fleet allocation and pricing strategies.
 - c) This will lead to better resource utilization and improved revenue.
- 2) Contactless Solutions:
 - a) The ongoing shift toward contactless experiences will impact car rental systems.
 - b) Expect touchless check-ins, digital contracts, and mobile app-based interactions.
 - c) QR codes for unlocking vehicles and seamless payment processes will enhance convenience and safety.
- 3) Enhanced Customer Experience:
 - a) Personalization will play a significant role. Systems will tailor recommendations based on user preferences, past rentals, and travel history.
 - b) Virtual assistants and chatbots will handle customer inquiries, bookings, and support.
 - c) Loyalty programs and rewards will further engage customers.
- 4) Collaboration with Mobility Services:
 - a) Car rental systems will integrate with ride-sharing, bike-sharing, and public transportation services.
 - b) Seamless multimodal travel options will emerge, allowing users to combine different modes of transport within a single app.

References

1. Accident Rate. Newsroom: Road Traffic Injuries. Geneva: World Health Organization; 2022. [20 June 2022]. <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>
2. Rental management. Guidelines on assessing donor suitability for blood donation. Annex. National Library of vehicle-<https://www.who.int/publications/i/item/9789241548519>
3. Aide-mémoire. Rent safety. Geneva: World Health Organization; 2002. [17 August 2012]. http://www.who.int/rentsafety/publications/who_bct_02_03/en/index.html.
4. The Melbourne Declaration on 100% voluntary car rent. Geneva: World Health Organization; 2009. [17 August 2012]. http://www.who.int/worldcarrent/Melbourne_Declaration_VNRBD_2009.pdf.
5. WHO/CDC/IFRC. Car rent counselling: Implementation guidelines. Geneva: World Health Organization; 2012. [17 August 2012]. http://www.who.int/carsafety/selection_counselling/en/
6. Geeksforgeeks- <https://www.geeksforgeeks.org/>