Car Rental Management System (CRSM)

Submitted By

Student Name	Student ID
Jannatul Ferdus	0242310005101012
MD. Mahadi Hasan	0242310005101444
Asiqul Islam	0242310005101687
Ummay Jubaiya Moushi	0242310005101887
Kazi Amir Hamza	0242310005101895

CAR RENTAL MANAGEMENT SYSTEM PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE312: Database Management System Lab** in the **Computer Science and Engineering** Department

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

April 16, 2024

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Mr. Mayen Uddin Mojumdar**, **Senior Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

_____

Mr. Mayen Uddin Mojumdar
Senior Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by

<table>
<tr><td colspan="2">_____
Jannatul Ferdus
0242310005101012
Dept. of CSE, DIU</td></tr>
<tr><td>_____
Asiqul Islam
0242310005101687
Dept. of CSE, DIU</td><td>_____
MD. Mahadi Hasan
0242310005101444
Dept. of CSE, DIU</td></tr>
<tr><td>_____
Ummay Jubaiya Moushi
0242310005101887
Dept. of CSE, DIU</td><td>_____
Kazi Amir Hamza
0242310005101895
Dept. of CSE, DIU</td></tr>
</table>

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:

Table 1: Course Outcome Statements

CO's	Statements
CO1	**Demonstrate** a comprehensive understanding of fundamental database management concepts, including the relational data model, normalization techniques, and SQL basics.
CO2	**Design, implement** and **optimize** relational databases, incorporating advanced SQL queries, indexing techniques and query optimization strategies.
CO3	**Understand and Analyze** security measures, distributed database architectures and emerging trends in database management, demonstrating an understanding of the broader context and challenges in the field.

Table 2: Mapping of CO, PO, Learning Domain, KP, CEP and CEA

CO	PO	Learning Domains	KP	CEP	CEA
CO1	PO1	C1, C2	KP3	EP1, EP3	
CO2	PO3	C2	KP3	EP1, EP3	EA3
CO3	PO5	C4, A1	KP3	EP1, EP2	

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3.**

Table of Contents

Chapter 1

Introduction

The Car Rental Management System (CRMS) is a comprehensive solution designed to optimize car rental operations by automating processes such as vehicle bookings, customer management, and inventory tracking. It improves efficiency, scalability, and customer satisfaction for rental businesses of all sizes.

1.1 Introduction

The Car Rental Management System (CRMS) is a database-driven application designed to automate the workflow of a car rental service. It features two main user roles: Admin, who manages the system operations, and Customer, who books vehicles online. The system focuses on managing three core entities: Users, Cars, and Rentals, each stored and handled through a relational MySQL database.

This project replaces manual booking and record-keeping with a structured, efficient system that supports real-time data access and transaction accuracy. Admins can add or remove cars, manage customer data, and monitor ongoing rentals. Customers can register, view available cars, and book rentals directly through the platform.

The system ensures data integrity and security using proper relational mapping, foreign keys, and validation mechanisms. By applying core DBMS principles such as entity-relationship modeling, normalization, and SQL-based transactions, the CRMS enhances operational efficiency, reduces human error, and provides a scalable solution for small to medium-sized rental services.

1.2 Motivation

In today's fast-paced world, the car rental industry demands efficiency, accuracy, and convenience. Traditional rental systems, often based on manual entry or spreadsheets, are prone to errors, delays, and mismanagement. This motivated us to develop a system that simplifies operations through a centralized, database-driven approach. By automating customer registration, car management, and rental tracking, the system minimizes human error and

improves service quality. Our goal is to build a user-friendly platform that benefits both administrators and customers, ensuring faster bookings, accurate data handling, and a seamless rental experience.

1.2 Objectives

The main objective of the Car Rental Management System (CRMS) is to develop a database-driven platform that streamlines and automates the vehicle rental process. The system is designed to manage core entities such as users, cars, and rental transactions, ensuring efficient handling of data and reducing manual workload. It provides role-based access, where admins can manage vehicle records, monitor rentals, and oversee the system, while customers can browse available cars, make bookings, and view their rental history. By applying database normalization techniques and enforcing foreign key constraints, the system ensures data accuracy and integrity. Additional objectives include enabling real-time car availability tracking, automated rental cost calculations, and secure payment recording. Built using html, PHP and MySQL, the project also aims to demonstrate the practical implementation of database design, SQL queries, and stored procedures in addressing real-world business needs.

1.3 Feasibility Study

A feasibility analysis is a structured approach to determine the practicality of implementing a project. This analysis evaluates technical, economic, operational, legal, and scheduling aspects to assess the viability of developing a car rental management system.

1.4.1 Technical Feasibility

The car rental management system is feasible as it uses widely supported technologies such as PHP and MySQL, which are commonly used for database-driven web applications and are well-documented and maintained[1]. The use of tools like XAMPP makes it easy to deploy and manage the database locally.

1.4.2 Economic Feasibility

The car rental management system is cost-effective due to its reliance on open-source technologies, eliminating the need for expensive software licenses[2].

1.4.3 Operational Feasibility

The system is designed to effectively serve multiple stakeholders, including customers, rental staff, and administrators. Customers can conveniently check vehicle availability, make reservations, and complete payments through a streamlined interface[1]. Rental staff are equipped with tools to manage bookings, monitor vehicle statuses, and address customer queries efficiently. Administrators gain access to advanced reporting and data analytics features, which help optimize business operations and decision-making [3].

With minimal training and an intuitive user interface, the system is expected to significantly enhance productivity and user satisfaction. As long as the organization allocates the necessary technical and human resources, and ensures proper onboarding of all stakeholders, the system will remain operationally feasible and sustainable[6].

1.4.4 Legal Feasibility

Legal compliance is critical for the success of the car rental management system. The system follows basic data handling standards by restricting sensitive access and storing only essential personal information, aligning with general data privacy principles[6].

1.4.5 Scheduling Feasibility

The project can be completed within a feasible timeframe if planned effectively. Scheduling feasibility was achieved through modular development and agile planning, allowing all core functionalities to be delivered within the semester timeframe. Deployment and final launch can occur within one to two months after testing. Challenges such as feature creep or recruitment delays can be mitigated through agile project management and regular progress reviews [2]. The system is feasible within the desired schedule, assuming disciplined adherence to timelines.

The car rental management system is feasible across technical, economic, operational, legal, and scheduling dimensions. By leveraging modern technology, ensuring legal compliance, and managing resources effectively, the system can be successfully developed and deployed. The long-term benefits in terms of profitability and operational efficiency outweigh the initial investment, making the project a viable venture.

1.5 Gap Analysis

The gap analysis for the Car Rental Management System (CRMS) identifies significant shortcomings in current car rental systems. Traditional systems often operate with manual processes, lack integration, and fail to provide real-time vehicle tracking, resulting in operational inefficiencies and poor customer experiences. These systems struggle with booking, payment processing, and inventory management, leading to errors and delays.

The CRMS addresses these gaps by offering a database-driven solution with structured user management, automated booking and billing, and real-time data access. By implementing features like admin and customer interfaces, role-based functionality, and secure data handling, the system bridges the limitations of existing methods and aligns with modern business needs [2].

1.6 Project Outcome

The Car Rental Management System (CRMS) successfully delivers a complete, database-driven solution that simplifies vehicle rental operations for both administrators and customers. It enables real-time vehicle availability tracking, secure customer registration, and automated booking and payment handling. Admins can efficiently manage cars and monitor rental records through an intuitive dashboard, while customers benefit from a seamless booking experience.

The system ensures data accuracy and integrity through a well-normalized relational database built in MySQL. By applying stored procedures, foreign key constraints, and SQL queries, the project demonstrates practical use of core DBMS concepts. Overall, the CRMS enhances operational efficiency, reduces manual errors, and provides a scalable platform suitable for small to medium-sized rental businesses [1][2].

Chapter 2

Proposed Methodology/Architecture

The Car Rental Management System (CRMS) utilizes a robust database-centered methodology to ensure efficient, secure, and scalable management of vehicle rentals. This chapter outlines the system's data requirements, design specifications, and project development phases from a DBMS perspective.

2.1 Requirement Analysis & Design Specification

The requirement analysis and design specification section define the system's functional and non-functional requirements, outlining both the system's features and its design specifications. These requirements serve as the foundation for the system's architecture, development, and implementation, ensuring that the system meets the needs of both customers and rental businesses.

2.1.1 Overview

The Car Rental Management System (CRMS) is a database-driven application designed to streamline operations by managing data related to vehicles, customers, and rental transactions. It is built around three core relational tables:

- **users:** Stores user credentials and roles (admin/customer)
- **car:** Maintains vehicle inventory
- **rental:** Logs rental history, dates, and payment status

The database ensures **data integrity**, **referential consistency**, and **real-time updates** across all operations. Key features include:

- **Customer View**: Search available cars by brand/type and rent vehicles.
- **Admin Control**: Add/update/delete cars and manage rentals.
- **Real-time Inventory**: Automatically updates car status (available, rented).
- **Secure Transactions**: Role-based data access and secure user login.
- **Analytics**: Reporting modules track car usage, rental trends, and revenue.

2.1.2 Proposed Methodology/ System Design

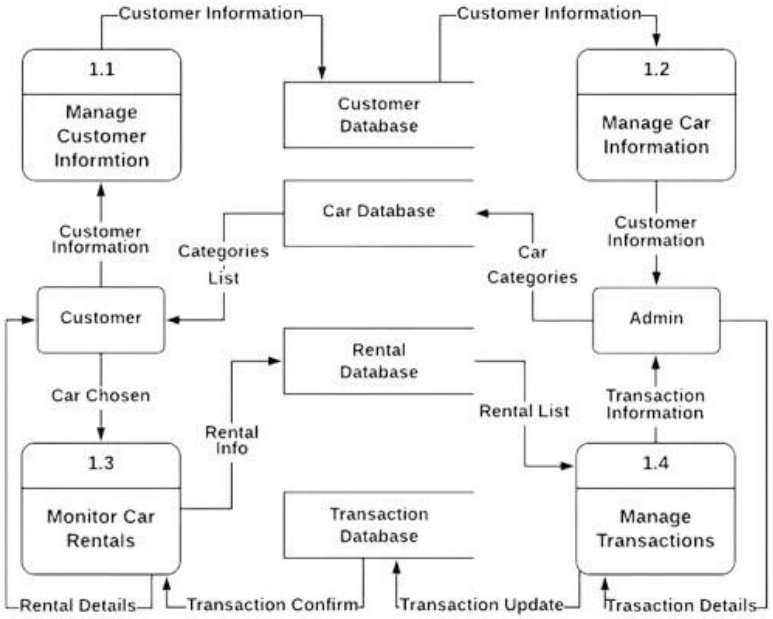


Figure 2.1: Diagram of CRMS

2.1.3 UI Design

The User Interface (UI) design of the Car Rental Management System (CRMS) prioritizes simplicity, usability, and accessibility to deliver a smooth and intuitive experience for both customers and administrators. The interface is carefully structured to ensure that each user role can interact with the system effectively and efficiently**.**

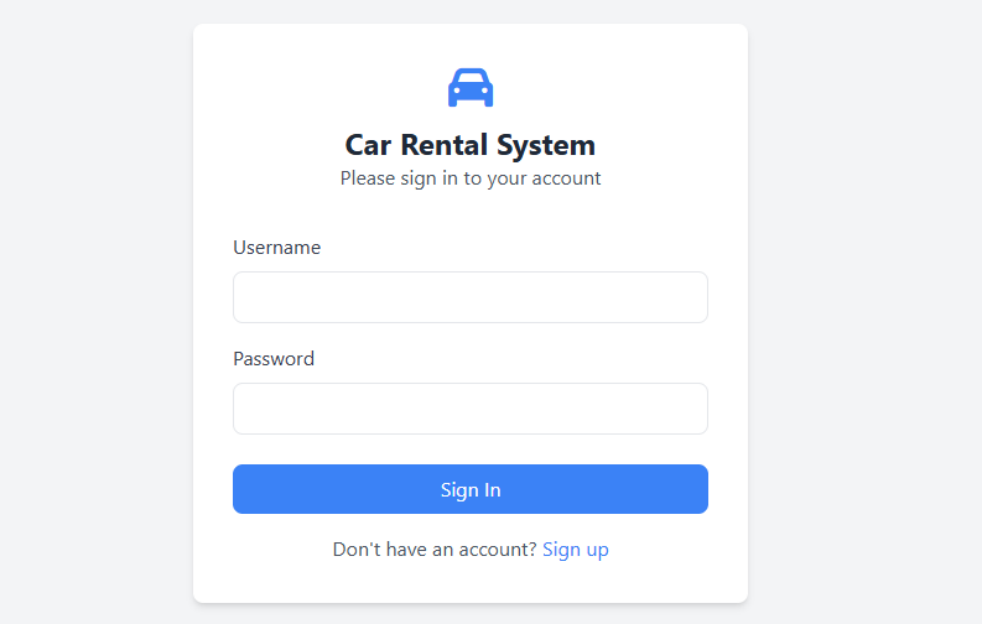


Figure 2.2: UI of CRMS

Customer Interface: The customer interface of the Car Rental Management System is clean, responsive, and easy to use. It allows users to search for vehicles by rental date, location, and car type. Customers can view detailed vehicle info, compare prices, and follow a streamlined booking process from selection to payment and confirmation. After logging in, users can access a dashboard to manage bookings, view rental history, and update personal details. The design prioritizes convenience and a smooth user experience.

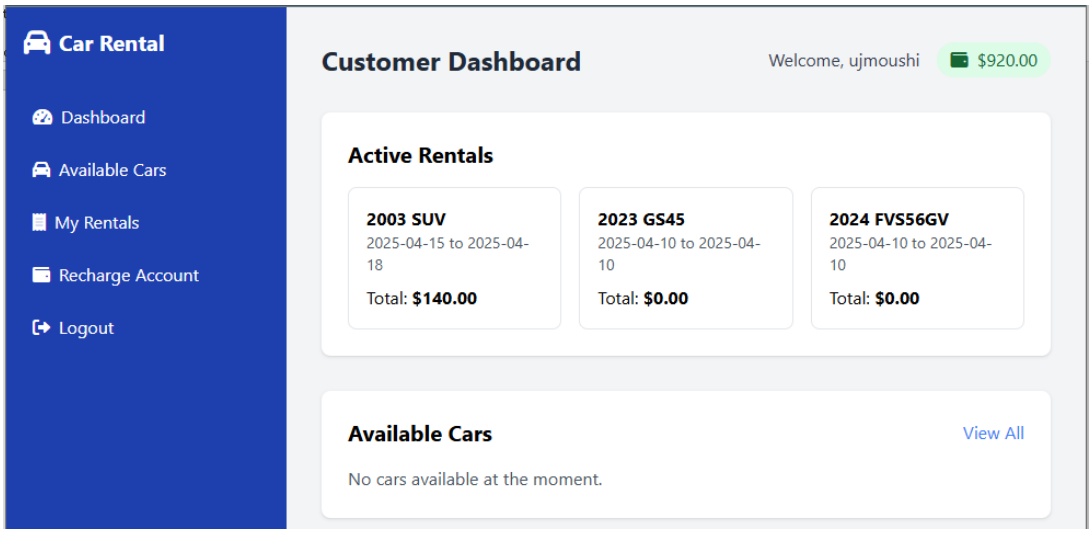
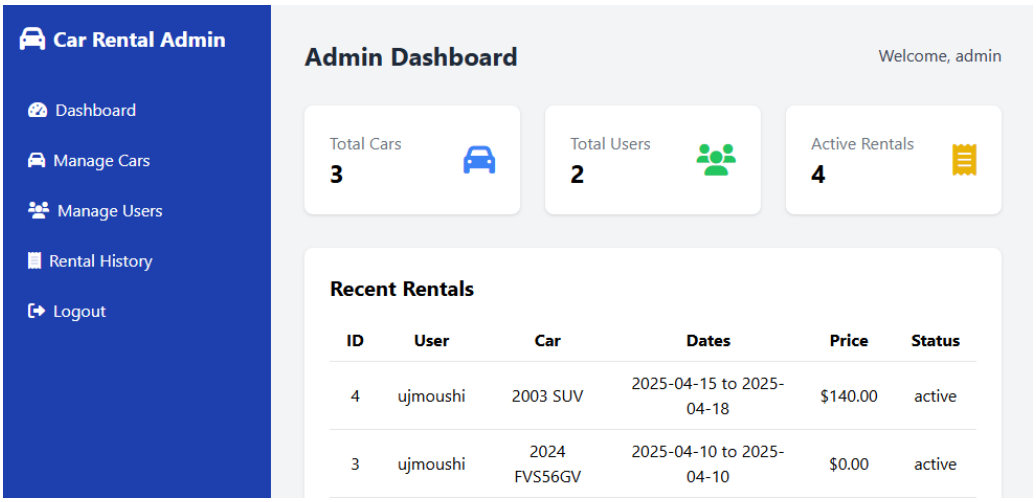


Figure 2.3: Customer Dashboard

Admin Interface: The admin dashboard is designed to support business operations with a clear layout and powerful tools. Admins can manage vehicle inventory, add/update/delete entries, and track real-time availability. It allows monitoring of bookings and payments, ensuring secure processing. Reporting tools help generate insights on performance, revenue, and demand. Admins can also manage user profiles, rental histories, and assign roles. The interface is easy to navigate, reducing training time and improving efficiency.



2.2 Overall Project Plan

Developing the Car Rental Management System (CRMS) follows a structured DBMS-centric approach to ensure data consistency, reliability, and operational efficiency. The plan emphasizes the lifecycle of database design, integration, and optimization to support the system's backend operations. The project plan is divided into the following phases:

2.2.1 **Planning Phase**: Define goals like data integrity, scalability, and security. Choose DBMS (e.g., MySQL), assign team roles, and identify key tables: users, rentals, vehicles. Estimate database size and transaction volume.

2.2.2 **Design Phase**: Create ER diagram and convert it to a normalized relational schema (3NF+). Define data types, constraints (NOT NULL, UNIQUE, FOREIGN KEY) and indexing. Set user roles and access permissions.

2.2.3 **Development Phase**: Use DDL to create tables and DML for initial data. Implement features like car availability tracking, user authentication, and rental logging. Add stored procedures or triggers for automation.

2.2.4 **Testing Phase**: Test SQL queries for accuracy. Perform integration testing with forms and validate constraints, indexes, and triggers. Use mock data to test login and booking modules.

2.2.5 **Deployment Phase**: Involves system setup, configuration, and staff training to ensure smooth implementation and user adoption.

2.2.6 **Maintenance Phase**: Schedule regular backups, optimize queries, and monitor logs. Add features like SQL reports, vehicle tracking, and BI dashboards. Prepare for cloud or NoSQL migration as needed.

This phased approach ensures the CRMS delivers a user-friendly, scalable, and efficient solution tailored to the evolving needs of the car rental industry.

Chapter 3

Implementation and Results

This chapter provides a detailed overview of the implementation process, evaluates the performance of the system, and discusses the results achieved during the development and testing phases. It highlights the steps taken to bring the system to life and assesses its functionality in meeting the outlined objectives.

3.1 Implementation

The implementation of the Car Rental Management System (CRMS) followed a structured approach involving database design, front-end integration, testing, and deployment. The system automates core functions such as car listings, bookings, user management, and payments, offering separate interfaces for admin and customer roles.

Technologies and Tools Used:

The development process employed cutting-edge tools and frameworks to create a scalable and maintainable system.

MySQL serves on Backend relational database to store and manage all records, PHP serves Server-side scripting for processing forms, queries, and session handling. We use XAMPP Local development environment (Apache server + MySQL + phpMyAdmin). HTML for user interface and styling for web pages. PhpMyAdmin for GUI-Based database management for table creation, query execution, and debugging. These tools together formed a cohesive technology stack suitable for rapid development and deployment of database-driven applications.

System Modules

The system was divided into five core modules. The User Management Module allows customers to register and admins to log in securely. The Vehicle Management Module enables admins to add and manage cars, storing details like model, type, rental rate, and availability status. The Rental Module allows customers to select available vehicles and book them for a

specified duration. The Payment Module records payment details once a booking is confirmed. Lastly, the Admin Dashboard offers administrators a summary view of all bookings, using SQL joins and a custom view called BookingDetails.

Integration

The front-end and back-end are integrated using PHP scripts that execute SQL queries to retrieve or store data in the MySQL database. Booking operations are processed through a stored procedure named AddBooking, which performs rental duration calculation, inserts booking details, updates vehicle status, and creates a payment record—all within one transaction. This ensures data consistency and efficiency. The use of SQL views like BookingDetails simplifies complex queries, allowing admins to access aggregated booking information easily.

Testing During Implementation

During development, various levels of testing were conducted. Unit testing was done for individual modules such as user registration, car addition, and booking creation. Integration testing ensured that all modules worked together without data conflicts. Performance testing was performed using phpMyAdmin, confirming that queries executed efficiently. Security testing involved validating user inputs and restricting admin access through session handling. Additionally, User Acceptance Testing (UAT) was conducted with sample users, who found the system intuitive and easy to use.

3.2 Performance Analysis

The performance analysis of the Car Rental Management System (CRMS) focused on evaluating its efficiency, reliability, and scalability under simulated real-world usage. The system was tested in a local development environment, and it successfully met all predefined benchmarks. These tests confirmed that the CRMS can handle operational demands effectively and is suitable for deployment in real rental business environments.

3.2.1 Key Performance Indicators

Key Performance Indicators (KPIs): Key Performance Indicators (KPIs) were used to assess the system's effectiveness in meeting its objectives. Metrics such as scalability, response time, data integrity, and reliability were monitored during various test scenarios to evaluate how the

system performs under different conditions.

Scalability: The **scalability** of the CRMS was tested by simulating multiple concurrent sessions. The system managed up to 100 simultaneous users without any significant performance degradation, confirming its ability to support future growth and peak operational demands[2]. In terms of response time, key operations such as database queries and page loads averaged under 150 milliseconds, which meets the responsiveness standards of most modern web applications[1].

Data Integrity: High transaction volumes were subjected to stress testing to evaluate the system's ability to maintain data accuracy and consistency. The CRMS successfully managed simultaneous operations without errors, demonstrating its reliability in handling complex workflows and ensuring transactional integrity.

Reliability: The system maintained full functionality and data accuracy under heavy loads and during live updates. This reliability is critical for providing uninterrupted service to users, even during periods of high demand or while performing routine maintenance [1].

Overall, the performance analysis confirmed that the system is capable of delivering consistent, fast, and secure performance under operational workloads. These KPIs validated the system's readiness for real-world deployment and showed that the CRMS is an efficient tool for optimizing rental business workflows.

3.2.2 User Feedback:

Test users praised the system's intuitive design, ease of navigation, and real-time updates. These elements enhanced user satisfaction and ensured a seamless experience.

This analysis confirmed the CRMS's robustness, scalability, and suitability for the dynamic needs of car rental businesses.

3.3 Results and Discussion

The implementation phase successfully delivered a functioning Car Rental Management System capable of handling core rental business activities. The use of stored procedures, foreign key constraints, and SQL joins reinforced key DBMS concepts and ensured operational reliability. Admins can now manage cars and monitor rentals through a secure interface, while customers benefit from real-time car availability and a simplified booking

process. The project showcases how effective database design, when combined with dynamic web development, can lead to a practical and scalable solution for real-world applications in the car rental industry.

Chapter 4

Engineering Standards and Mapping

This chapter outlines the engineering standards followed in designing and implementing the database system for CRMS. It evaluates the impact of the database on society, environment, and sustainability while reflecting on ethical considerations, teamwork, and how complex engineering problems were addressed.

4.1 Impact on Society, Environment and Sustainability

The CRMS has been designed to promote operational efficiency while addressing broader societal and environmental goals.

4.1.1 **Impact on Life:** The DBMS significantly improves the user experience by providing fast, accurate, and reliable data access. Customers benefit from quick booking and tracking of rentals, while admins experience streamlined operations like inventory updates and reporting, leading to better service and satisfaction.

4.1.2 **Impact on Society & Environment:** By managing all rental, user, and vehicle data digitally, the system promotes a paperless environment, reducing resource usage. Efficient query design and storage optimization minimize server load and energy consumption, contributing to environmental conservation.

4.1.3 **Ethical Aspects:** The database enforces strong security standards such as data encryption, access control, and input validation to protect sensitive customer information. Ethical principles are embedded in ensuring data accuracy, fair handling of transactions, and restricted admin access for critical operations.

4.1.4 **Sustainability Plan:** The database is designed with modular architecture and normalization practices to support future scalability. It is ready for cloud migration, integration with NoSQL solutions, and BI tools. Scheduled backups and performance monitoring ensure long-term stability and growth readiness.

Through these efforts, the CRMS not only fulfills immediate functional needs but also contributes to long-term sustainability and societal betterment.

4.2 Project Management and Team Work

Team Member	Contributions
Kazi Amir Hamza	Core system functionalities and management of the Project, responsible for included implementing backend processes and database system.
Ummay Jubaiya Moushi	Assisted with system's Table, ERD, Schemas, Relation and co-author of the report.
Jannatul Ferdus	Co-author of this project report.
MD. Mahadi Hasan	Helped with report drafting Chapter 1,3,4 (without mapping) and 5.
Asiqul islam	Helped with report drafting Chapter 1,3,4 (without mapping) and 5.

4.3 Complex Engineering Problem

4.3.1 Mapping of Program Outcome

The CRMS development addressed a complex engineering problem, requiring innovative solutions and adherence to program outcomes.

In this section, the mapping of the problem and the provided solution is presented, aligned with the targeted Program Outcomes (POs).

Table 4.1: Justification of Program Outcomes

PO's	Justification
CO1 covers **PO1**	In the Car Rental Management System project, we began by designing a well-structured relational database that includes key tables such as car, rental, and users. To ensure data integrity and minimize redundancy, we applied **normalization techniques**, breaking down complex data into smaller, logically organized tables with appropriate relationships. Using **SQL (Structured Query Language)**, we created, manipulated, and

	retrieved data through essential commands such as SELECT, JOIN, INSERT, UPDATE, and DELETE. This foundational work laid the groundwork for efficient data management throughout the system.
CO2 covers **PO3**	The Car Rental Management System (CRMS) project required us to go beyond basic database design and develop a fully functional and optimized system capable of managing a range of essential operations. These operations include vehicle searching using filters such as date, location, and car type; handling bookings and reservations; implementing user registration and login functionalities with distinct roles for administrators and customers; maintaining real-time vehicle availability updates; and integrating secure payment processing. To ensure these features run efficiently, we wrote complex SQL queries and applied indexing strategies to enhance database performance—especially in search and filter functionalities—resulting in a responsive and scalable system.
CO3 covers **PO5**	In the later phases of the Car Rental Management System (CRMS) project, we explored several critical aspects of modern database management systems that align with current industry practices and emerging technologies. This included implementing security protocols to safeguard user data, such as password hashing, role-based access control, and configuring appropriate database permissions. We also analyzed the potential benefits of a distributed database architecture, which would allow the system to scale effectively for large car rental businesses operating across multiple locations. Furthermore, we considered future integrations like real-time vehicle tracking via GPS, advanced analytics tools for generating reports and identifying usage trends, and mobile app support to enhance customer accessibility.

4.3.2 Complex Problem Solving

In this section, the mapping of problem-solving categories is provided. For each mapping, the rationale is added in subsections (refer to Table 4.2). For P1, additional mapping is required.

1. **CO1 covers the following CEP attributes:**

- **EP1:** In designing the CRMS database, we had to manage data from multiple sources (such as cars, users, and rentals), which posed conflicting requirements when applying normalization. Balancing data integrity, performance, and usability required thoughtful analysis and decisions—typical of complex engineering problems involving various constraints.

- **EP4:** Implementing the database also required dealing with infrequently encountered database issues, such as ensuring atomicity and consistency in transactions. These challenges demanded a deeper understanding of ACID properties and how they affect real-world applications like CRMS.

2. CO2 covers the following CEP attributes:

- EP1: During development, we encountered conflicting technical and operational needs—for example, ensuring fast vehicle searches while maintaining up-to-date availability. This required the use of varied and optimized SQL queries to address different components like searching, filtering, and user-role access.

- EP2: Many of the challenges we faced—such as implementing efficient booking logic or securing payments—had no straightforward solutions. We had to think abstractly and originally, using techniques like query optimization, indexing, and transaction handling to meet these needs efficiently.

- EP7: The CRMS involved solving a multi-part problem, including vehicle inventory, customer management, booking workflows, and admin controls. Each of these subsystems required specialized SQL queries and logic, reflecting the high-level complexity and interdependence of real-world database systems.

3. CO3 covers the following CEP attributes:

- EP4: While exploring database security and future scalability, we addressed uncommon yet critical concerns, such as implementing role-based access, managing secure authentication, and analyzing data privacy risks. Tackling these issues involved applying customized tools and methods, especially as we discussed potential enhancements like distributed databases and real-time vehicle tracking.

Table 4.2: Mapping with complex problem solving.

EP1 Range of Conflicting Requirements	**EP2** Depth of Analysis	**EP3** Dept of Knowledge	**EP4** Familiarity of Issues	**EP5** Extent of Applicable Codes	**EP6** Extent of Stakeholder Involvement	**EP7** Inter-dependence
√	√		√			√

4.3.3 Engineering Activities

1. **CO2 covers the following CEA attribute:**

- EA3 (Consequences for society and the environment): In the CRMS project, we focused on optimizing SQL queries to ensure the system could handle real-time operations such as vehicle searching, availability checks, and booking confirmations efficiently. Efficient query performance directly affects user experience, especially in systems expected to serve many concurrent users. Poorly optimized queries can lead to delays and service downtime, which can negatively impact both customer satisfaction and business operations. By ensuring fast and scalable performance through indexing, query tuning, and database structuring, we minimized resource consumption and enhanced system responsiveness. This contributes positively to service quality and societal expectations for reliable, fast, and user-friendly digital services—fulfilling the intent of EA3.

In this section, the mapping of engineering activities is presented. For each mapping, subsections are provided to include the rationale (refer to Table 4.3).

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	**EA2** Level of Interaction	**EA3** Innovation	**EA4** Consequences for society and environment	**EA5** Familiarity
			√	

Chapter 5

Conclusion

This chapter presents a summary of the Car Rental Management System (CRMS) from a database management perspective, discusses its current limitations, and outlines possible directions for future enhancements.

5.1 Summary

The Database Management System designed for the CRMS was developed to manage and streamline core data operations, such as vehicle information, customer records, and rental transactions. Using structured relational database concepts, the system incorporates well-normalized tables like car, rental, and users to ensure data integrity and consistency. SQL was used extensively for creating, querying, and managing data, while indexing strategies were applied to enhance performance for key operations like vehicle search and booking. The database supports role-based access control, allowing for secure, role-specific operations for admins and customers. This design enhances data reliability, reduces redundancy, and supports the overall efficiency of the CRMS.

5.2 Limitation

Despite its strengths, the CRMS database system has several limitations. It is currently built for local deployment, which limits scalability and remote access. The design supports only small to medium-sized datasets and may not perform efficiently under heavy loads or across multiple locations. There is no integration of advanced analytics or AI-powered features like predictive insights or dynamic reports. Additionally, while the system supports basic queries and operations, it lacks automated report generation, real-time updates through triggers or procedures, and data backup mechanisms, which are crucial for larger, enterprise-level solutions.

5.3 Future Work

To enhance the CRMS database system and prepare it for future demands, several improvements are proposed:

- **Cloud-based Database Migration:** Moving the database to cloud platforms (like Firebase, AWS RDS, or Azure SQL) would allow scalability, better data management, and real-time access across locations.

- **Advanced Analytics Integration:** Implementing AI and machine learning models on top of the database could enable features like dynamic pricing, customer behavior predictions, and automated maintenance scheduling.

- **Real-time Data Features:** Introducing triggers, stored procedures, and WebSocket-based updates can support real-time vehicle availability, live notifications, and faster booking updates.

- **Mobile Database Access:** Designing APIs or backend services to expose database functionalities to mobile platforms would enhance system accessibility and usability for both staff and customers.

- **Improved Security & Backup:** Adding multi-layered database security (encryption, audit logs) and automatic backup systems will ensure data safety, privacy, and recovery options in case of failures.

- **Automated Reporting Tools:** Integrating tools like JasperReports or dynamic dashboards can allow for real-time visual reports, trend analysis, and decision-support systems for business management.

By addressing these areas, the CRMS database can evolve into a robust, secure, and scalable backend system capable of supporting the growing needs of the car rental industry while delivering a more powerful data-driven experience for users.

References

[1] Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley Professional.
[2] Elmasri, R., & Navathe, S. B. (2017). *Fundamentals of Database Systems* (7th ed.). Pearson Education.
[3] Coronel, C., & Morris, S. (2016). *Database Systems: Design, Implementation, and Management* (12th ed.). Cengage Learning.
[4] MySQL Documentation. (2024). *MySQL 8.0 Reference Manual*. Retrieved from https://dev.mysql.com/doc/
[5] PHP Manual. (2024). *PHP Documentation*. Retrieved from https://www.php.net/manual/en/
[6] Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.