Final Project Report

Rent-a-car - Car Rental Management System

Bacha Wondimu- 114935

A report submitted in part fulfilment of the degree of

BSc in Computer Science

Supervisor: Obinna Kalu

Department of Computer Science

Maharishi International University

February 10, 2024

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:

Student Name: Bacha Wondimu

Date of Submission: February 10, 2024

Signature: Bacha Wondimu

Table of Contents

Abstract

The Rent-a-Car Car Rental Management System offers a comprehensive solution tailored to the specific needs of car rental businesses. By efficiently managing vehicle fleets, reservations, rentals, and customers tasks, this system aims to streamline operations and enhance overall customer satisfaction.

Project Specification

The Rent-a-Car Car Rental Management System must fulfil the following functional requirements:

1. Vehicle Management
   - The system should allow the addition, modification, and deletion of vehicles in the rental fleet.
   - Each vehicle entry should include details such as make, model, year, registration number.
   - The system keeps track of how many cars of a certain brand and type are available in the car fleet.

2. Reservation System
   - Customers should be able to check the availability of vehicles based on specified criteria like date, location, and vehicle type.
   - Users should be able to make online reservations, specifying the pickup and drop-off locations, dates, and vehicle preferences.
   - The system should prevent double bookings by updating the availability status in real-time.

3. Borrowing functionality.
   - A customer can pick up the car at the car renting location.
   - A customer can return the car to the car renting location.
   - The customer pays for the car when the car is returned.

4. Customer functionality:
   - We can add, remove, update and find customers in the car rental application. You can search customers on customer number, name and email address. Every customer has a unique customer number.

5. Rental Pricing and Invoicing
   - The system should calculate rental charges.

Chapter 1: **Introduction**

Car rental businesses face various challenges in efficiently managing their operations, including vehicle fleet management, reservation handling, customer management, and rental pricing. To address these challenges, the Rent-a-Car Car Rental Management System provides a comprehensive solution tailored to the specific needs of car rental businesses.

1.1 Background

The primary objective of the Rent-a-Car Car Rental Management System is to streamline the operations of car rental businesses and enhance customer satisfaction by providing efficient management of vehicle fleets, reservations, rentals, customers, and administrative tasks.

Chapter 2: **Project Design and Conceptualization**

2.1 User Story Mapping and Acceptance Criteria

User Stories

1. As a car rental staff, I want to add a new vehicle to the rental fleet so that it becomes available for customers.
2. As a customer, I want to check the availability of vehicles based on my specified criteria like date, location, and vehicle type so that I can make a reservation.
3. As a customer, I want to pick up a reserved car at the rental location so that I can start using it.
4. As a customer, I want to return the rented car to the rental location so that I can complete my rental.
5. As an admin, I want to add, remove, update, and find customers in the car rental application so that customer records are accurate and up to date.
6. As a car rental staff, I want to calculate rental charges accurately based on the duration of the rental period and any additional charges.

Acceptance Criteria

User Story 1: Add a New Vehicle

- Admin should be able to access the "Add Vehicle" feature.
- Required vehicle details (make, model, year, registration number) must be provided.
- Upon submission, the vehicle should be added to the rental fleet.
- The system should display a confirmation message upon successful addition.

User Story 2: Check Vehicle Availability

- Customers should be able to access the "Check Availability" feature.
- Criteria such as date, location, and vehicle type should be provided for the search.
- The system should display a list of available vehicles meeting the specified criteria.
- Availability status should be updated in real-time to prevent double bookings.

User Story 3: Pick Up Reserved Car

- Customers should be able to pick up their reserved car at the rental location.
- The system should verify the customer's reservation and allocate the corresponding vehicle.
- Staff should process any necessary paperwork (e.g., rental agreement) efficiently.
- Payment details should be confirmed before handing over the car to the customer.

Main Scenario

User Story 1: Add a New Vehicle

1. Admin navigates to the Add Vehicle section.
2. Admin fills in the required details for the new vehicle (make, model, year, registration number).
3. Admin submits the form.
4. System validates the input and adds the vehicle to the rental fleet.
5. Confirmation message is displayed.

User Story 2: Check Vehicle Availability

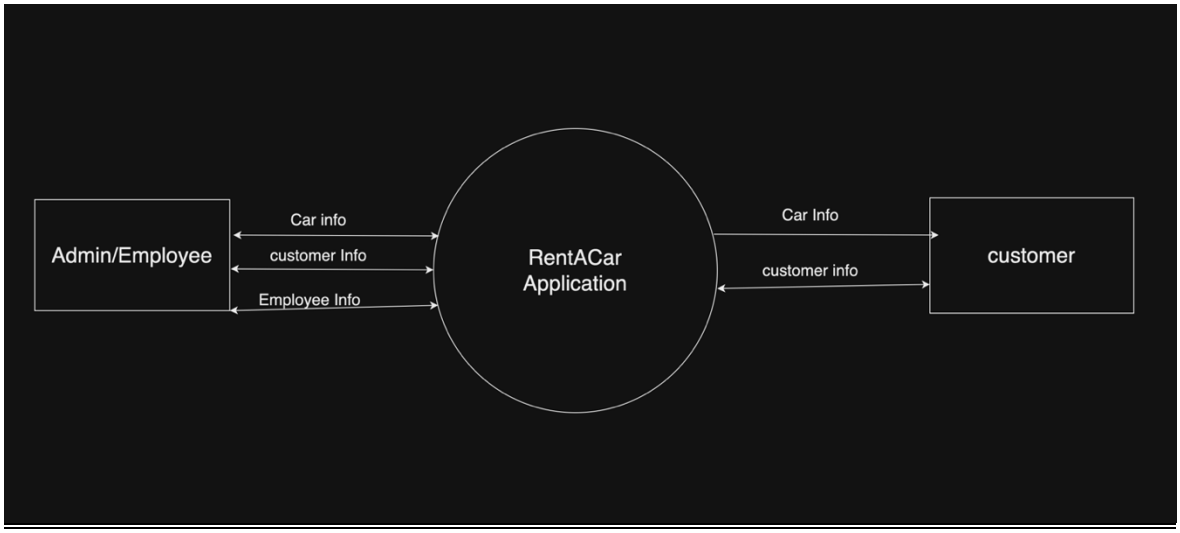
1. Customer accesses the Check Availability feature.
2. Customer specifies the criteria (date, location, vehicle type) for the search.
3. System retrieves and displays a list of available vehicles meeting the criteria.
4. Customer selects a preferred vehicle.
5. System updates availability status in real-time.

User Story 3: Pick Up Reserved Car

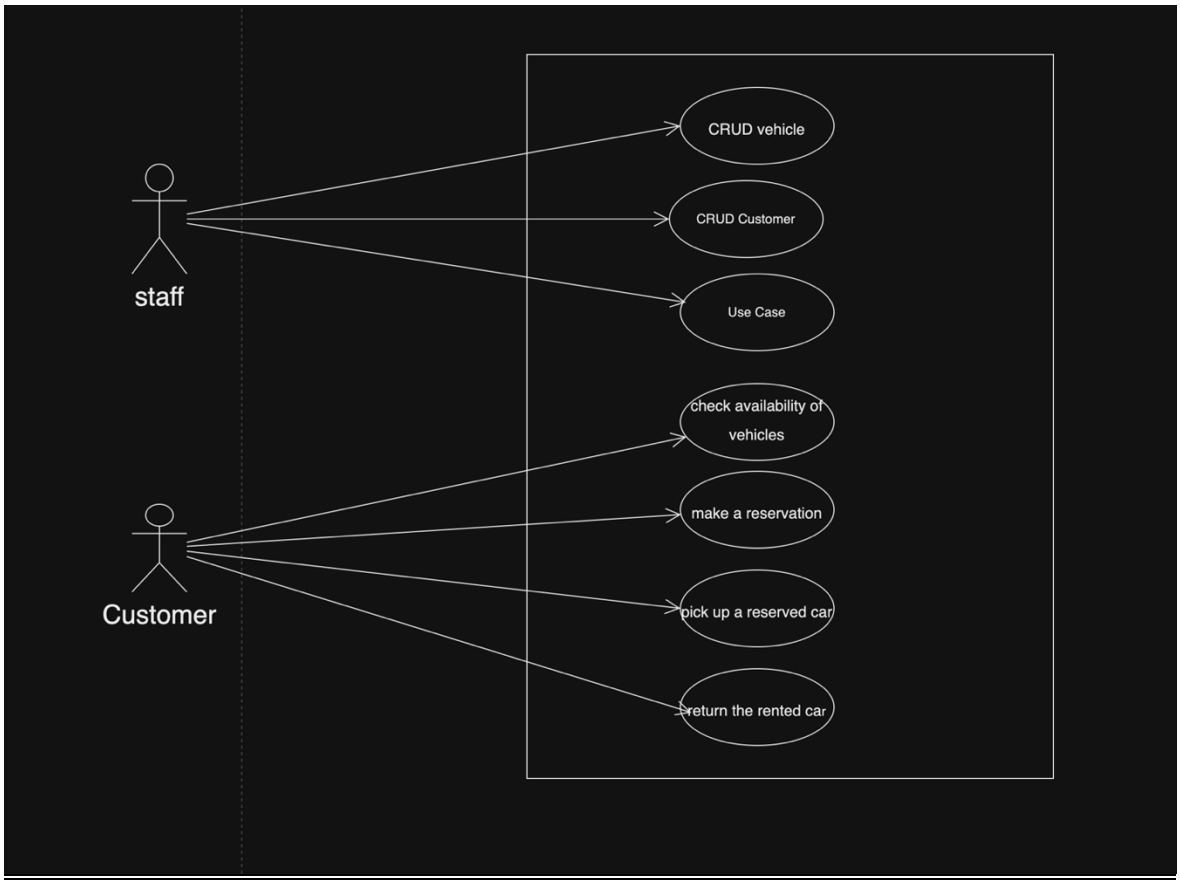
1. Customer arrives at the rental location to pick up the reserved car.
2. Staff member verifies the customer's reservation.
3. Staff allocates the corresponding vehicle to the customer.
4. Necessary paperwork (e.g., rental agreement) is processed.
5. Customer confirms payment details.
6. Staff hands over the car keys to the customer.

2.2 Context Diagram and Use-Case Diagram

Context Diagram

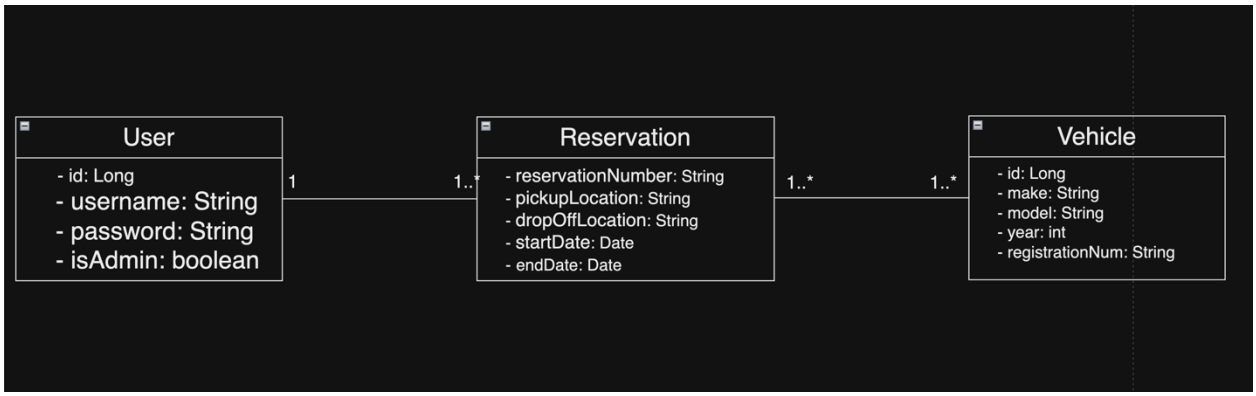


Use-Case Diagram

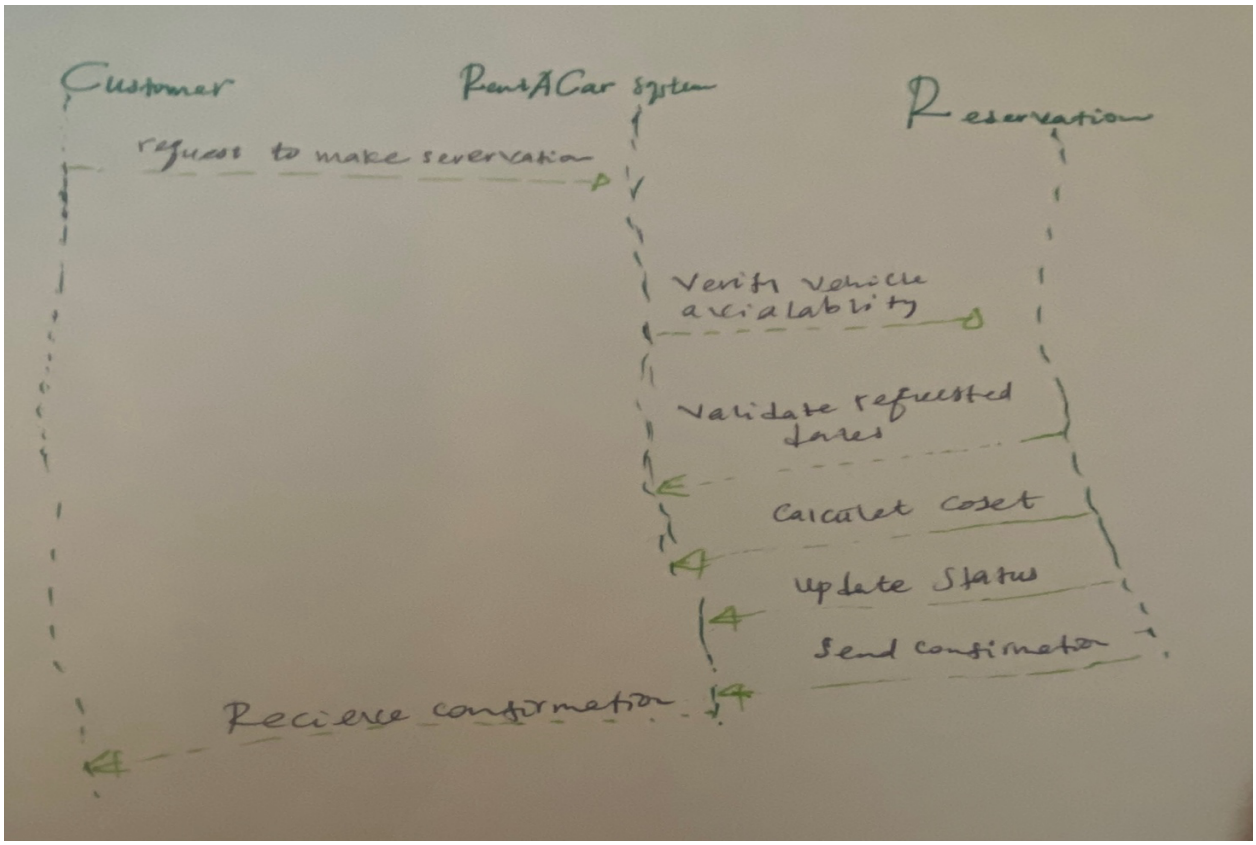


2.3 Domain Model

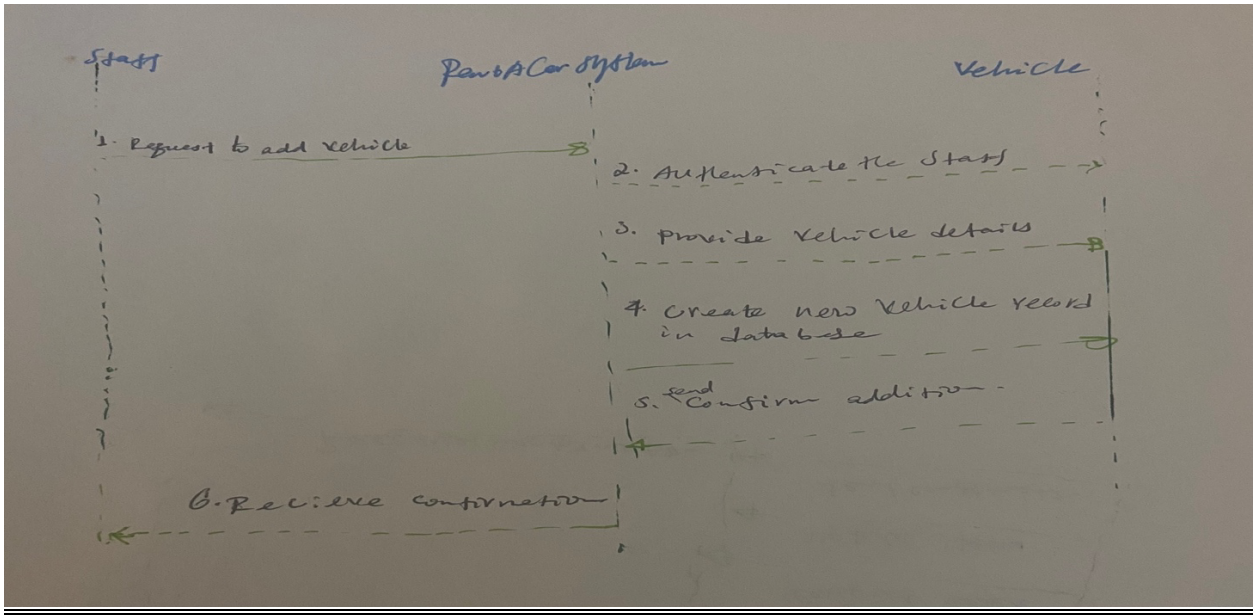
Domain Model (Class Diagram)



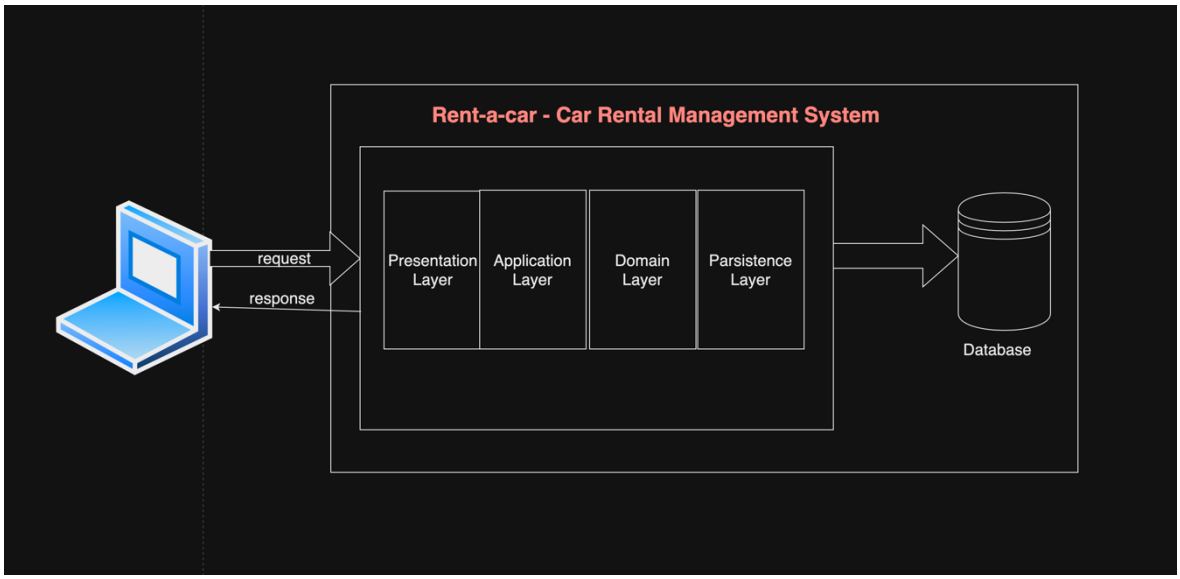
Sequence Diagram-Customer Making a Reservation


Customer
RentACar system
Reservation
request to make severvation
Verify vehicle avcialability
Validate refuested dates
Calcalet coset
Update Status
Send confirmation
Recieve confirmation

Sequence Diagram-Staff Adding a Vehicle



2.4 System Architecture



Chapter 3: **Implementation**

3.1 Presentation Layer

The presentation layer of the Rent-a-Car Car Rental Management System encompasses functionalities related to user interaction and interface design. It includes features such as user authentication, navigation menus, forms for data input, and displays for presenting information to users.

3.2 Domain Layer

The domain layer contains the core business logic and entities of the Rent-a-Car Car Rental Management System. It encapsulates the rules and processes related to vehicle management, reservation handling, customer management, and rental pricing. This layer ensures that business operations are carried out according to the defined rules and constraints.

3.3 Service Layer

The service layer acts as an intermediary between the presentation layer and the domain layer. It encapsulates application logic that spans multiple domains or requires coordination between different parts of the system. In the Rent-a-Car Car Rental Management System, the service layer may include functionalities such as transaction management, logging, and security enforcement.

3.4 Persistence Layer

The persistence layer is responsible for storing and retrieving data from the database. It abstracts the details of data storage and retrieval, providing a standardized interface for accessing the underlying database. In the Rent-a-Car Car Rental Management System, the persistence layer interacts with the database to store information about vehicles, reservations, customers, invoices, and other relevant entities.

Chapter 4: **Demos**

Chapter 5: **Project Analysis and Outlook**

5.1 Methodology

The methodology employed in developing the Rent-a-Car Car Rental Management System integrates Agile principles, fostering adaptability and collaboration throughout the project lifecycle. Agile methodology promotes iterative development, allowing for frequent feedback and adjustments to deliver high-quality software efficiently. Additionally, leveraging Transcendental Meditation principles, which advocate simplicity and stress reduction, complemented the Agile approach. Just as Agile streamlines development processes, Transcendental Meditation simplifies life, promoting clarity and flow.

Despite working on the project individually, I adopted an Agile methodology to guide the development process. This approach proved invaluable in managing time effectively and prioritizing tasks to ensure progress within the allotted timeframe. By breaking down the project into manageable iterations, I was able to focus on delivering incremental functionality while continuously adapting to evolving requirements. The iterative nature of Agile allowed me to remain flexible and responsive to changes, maximizing productivity and mitigating risks associated with time constraints.

5.2 Results and Discussion

Working individually presented unique challenges, particularly with regards to time constraints and the complexity of system design. Initially planning for a full-stack implementation, I encountered limitations that necessitated a shift in focus to developing a web API using Spring Boot technology. Despite this adjustment, I leveraged various tools and technologies such as IntelliJ, GitHub, Model Mapper, and Maven to streamline the development process and enhance productivity.

One significant struggle was in designing the system architecture and class diagrams. Agile methodology provided a structured approach, guiding me through the design process step by step. I began by defining requirements and creating a basic outline, gradually refining my designs as I progressed. Each completed requirement and modelling task was pushed to GitHub, allowing for continuous integration and collaboration with myself.

Despite the challenges, my adherence to Agile principles enabled me to overcome obstacles systematically. By breaking down tasks into manageable chunks and maintaining a steady pace of development, I successfully implemented a fully functional web API within the constraints of time. This iterative approach not only facilitated efficient development but also ensured the delivery of high-quality software that met my project objectives.

5.3 Future Enhancements

Future enhancements for the Rent-a-Car Car Rental Management System extend beyond technical refinement to encompass the holistic evolution of the system. In addition to optimizing user interfaces and system performance, future endeavors may include deeper integration of Agile methodologies, fostering continuous improvement and innovation. Furthermore, the exploration of advanced features such as predictive analytics promises to elevate the system's capabilities, enabling data-driven decision-making and enhanced customer experiences. Moreover, the ongoing exploration of emerging technologies ensures that the system remains at the forefront of industry trends, poised to adapt to evolving business needs.

Chapter 6: **Acknowledgement**

I would like to extend my sincere gratitude to Professor Obinna Kalu for his invaluable support and guidance throughout the Software Engineering CS499 course and the completion of my graduation project. His expertise and mentorship have played a crucial role in my success in this class.

Additionally, I wish to express my appreciation to my academic advisor, Dr. Anne Dow, for her unwavering assistance with any academic challenges I encountered during my time at MIU. Her advice and encouragement have been instrumental in my academic journey.