PES University

100 Feet Ring Road, BSK III Stage, Bangalore-560085

Object-oriented analysis and design (UE21CS352B)

Project report on

"CAR RENTAL MANAGEMENT SERVICE"

Project carried out by

Manish Bhat	**PES1UG21CS326**
Nayonika S Rao	**PES1UG21CS370**
Navya Govil	**PES1UG21CS368**
Ayman Hasib	**PES2UG21CS297**

Table of contents

CHAPTER 1
INTRODUCTION TO ONLINE CAR RENTAL SYSTEM

1.1 Introduction

This project is intended to be utilized by a car rental company that specializes in client automobile rentals. Customers can view available cars, register, view profiles, and book cars using this online system.

1.2 Problem Statement

An automobile rental is a vehicle that may be rented out for a set amount of time. Even for those who lack access to or do not possess a vehicle at all, having a personal car makes it easier for them to travel around. The person in need of a car must make contact with and sign a rental automobile agreement with a corporation. This technology makes managing people and vehicles easier while also increasing customer retention.

1.3 Aims and Objectives

The creation of a web-based system that will enable customers to register and make reservations for cars online and will enable the business to run its vehicle rental business efficiently.

To make it easier for customers to rent a car anytime they need to.

1.4 Scope

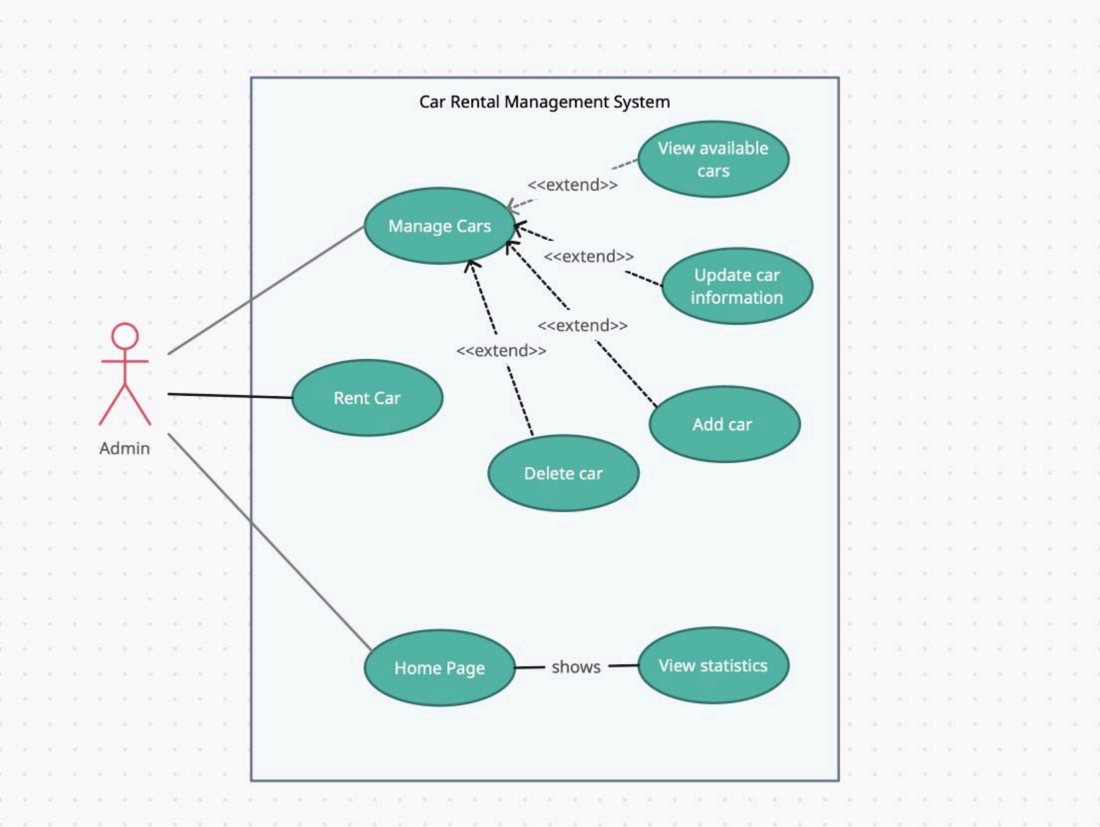
This project traverses a lot of areas ranging from business concept to computing field, and required to perform several researches to be able to achieve the project objectives.
The area covers include:

- Car rental industry: This covers research on the methods, procedures, and areas for development in the car rental industry.
- Both the company's employees and general customers will be able to use the system efficiently.
- Because it is web-platform based, the system will always be accessible, barring any brief server outages, which should be rare.

CHAPTER 2
MODELS USED

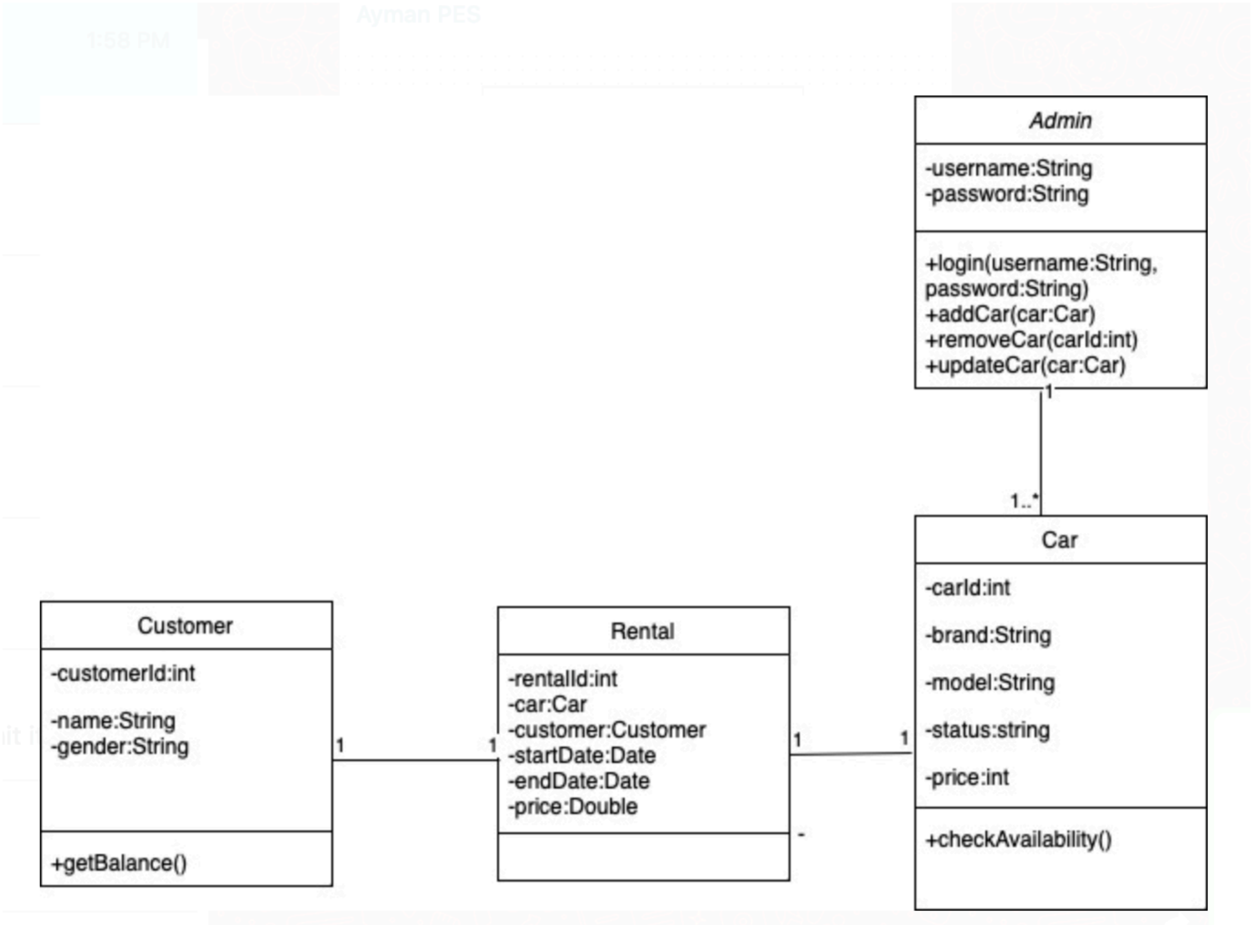
2.1 Use-Case Model

A use-case model is a model of how different types of users interact with the system to solve a problem. As such, it describes the goals of the users, the interactions between the users and the system, and the required behavior of the system in satisfying these goals.



2.2 Class Model

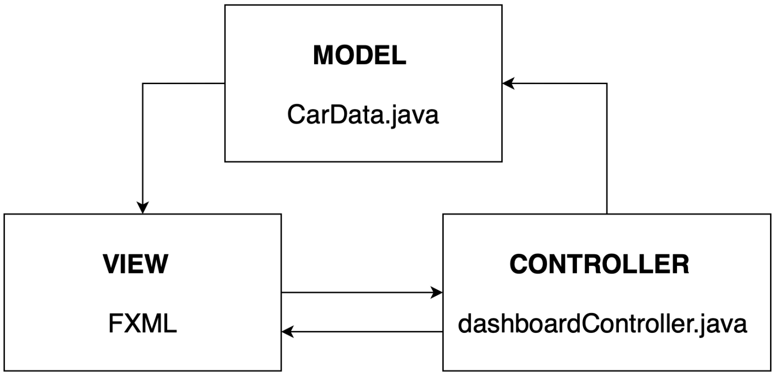
The Class Model describes the various domains in the system and the Classes and instances within them. It sets the underlying foundation upon which objects will be put to work, providing a flexible foundation upon which systems can be assembled in component-like fashion. Development of the object architecture starts with an identification of domains and an initial set of object classes using a Class Diagram.


Admin
-username:String
-password:String
+login(username:String, password:String)
+addCar(car:Car)
+removeCar(carId:int)
+updateCar(car:Car)
1
1..*
Car
-carId:int
-brand:String
-model:String
-status:string
-price:int
+checkAvailability()
Customer
-customerId:int
-name:String
-gender:String
+getBalance()
1
1
Rental
-rentalId:int
-car:Car
-customer:Customer
-startDate:Date
-endDate:Date
-price:Double
1
1
-

Chapter - 3
Applying Design Concepts

3.1 Architecture Patterns

This project utilizes the Model-View-Controller architecture pattern. The Model-View-Controller (MVC) is a well-known design pattern in the web development field. It is a way to organize our code. It specifies that a program or application shall consist of data model, presentation information and control information. The MVC pattern needs all these components to be separated as different objects.



3.2 Design Principles (SOLID)

The set of guidelines used as guidelines for design creation in Java is known as the design principles. The idea of design patterns and Java are comparable in terms of design principles. The design principles are more abstract and broad than the design patterns; that is the only distinction between the two. There is a lot more real and useful guidance in the design pattern. Not simply generalized coding approaches, but the class problems as a whole are tied to the design patterns.

3.2.1 Single Responsibility Principle (SRP)

The database class in the provided code follows the Single Responsibility Principle (SRP) as it has a single responsibility of establishing a connection to the database.In the database class, the sole responsibility is to create and return a connection to the MySQL database. The connectDb() method encapsulates the logic for loading the MySQL JDBC driver, creating the connection object, and returning it.

3.3 Design Patterns

Design patterns are essentially well-established solutions to common problems that arise during software design and development. They provide reusable templates that developers can apply to solve recurring design problems in their projects. These patterns encapsulate best practices, design principles, and years of collective experience within the software development community.

3.3.1 Observer Pattern

The code uses JavaFX's built-in event handling and property binding mechanisms, which can be considered an implementation of the Observer pattern. For example, the availableCarSearch() method uses textProperty().addListener() to observe changes in the search text field and update the filtered list accordingly.

3.3.2 Singleton Pattern

The code utilizes the database class to connect to a MySQL backend. The singleton pattern is employed to ensure that only one database connection is established, which can be accessed globally throughout the code.

Github Link

Link - **https://github.com/210manish/CarRentalManagement.git**

Individual contributions of team members

Manish Bhat - I worked on connecting the frontend to the MySQL backend using the MySQL jar provided by Java. I developed parts of the project's dashboard GUI using JavaFX library and Scene Builder, along with the corresponding controller class to manage button functionalities. I defined various handlers in Java to handle button actions and also worked on the CSS styling for the GUI.

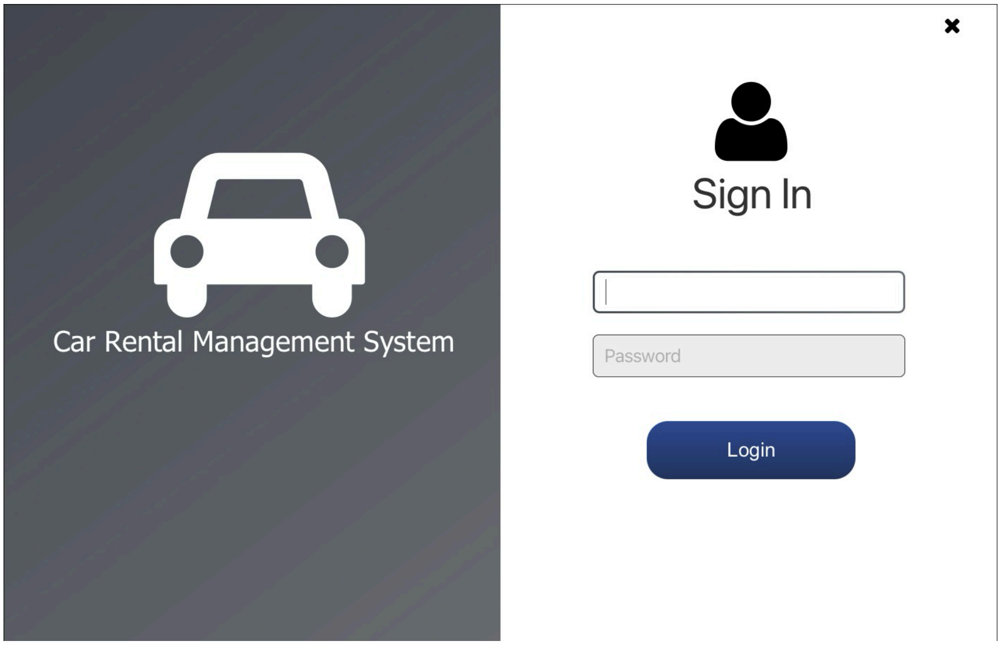
Navya Govil - I handled the model class to communicate with the database and give the necessary information to present to the user. I also implemented the Car and the Admin class with the necessary attributes and methods.

Ayman Hasib - I worked on creating the UI and webpage using JavaFX with Scene Builder and CSS styling , also helped with the use of design pattern such as singleton.

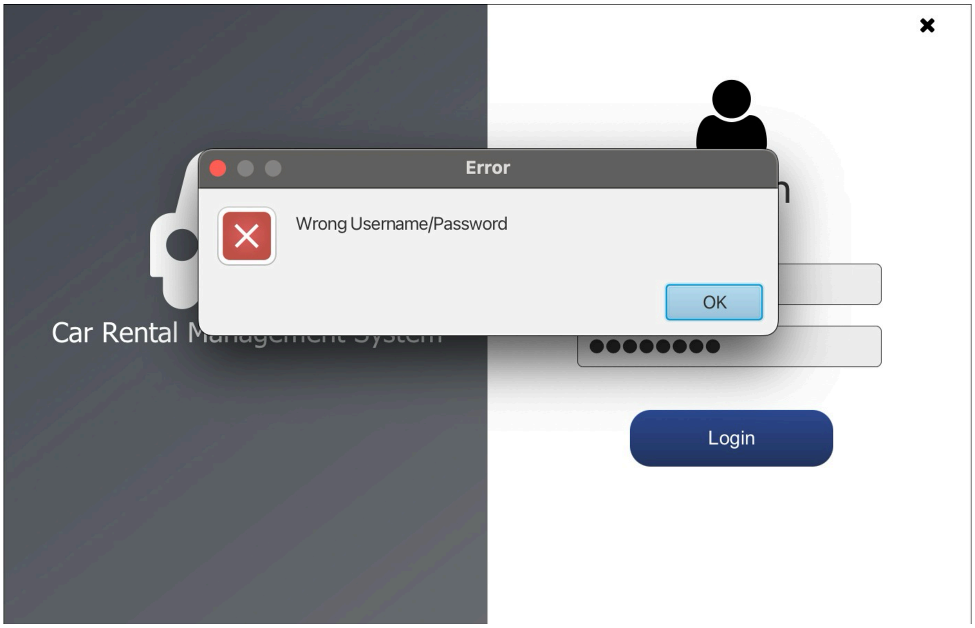
*Nayonika S Rao -*I focused on enhancing the project's UI, ensuring it was user-friendly and visually appealing. Additionally, I helped with implementing error handling mechanisms, particularly when users input invalid details during the car rental process. Moreover, I contributed in documenting various design principles and patterns incorporated into the project, ensuring clarity and maintainability.

Output Screenshots

Main log in screen -



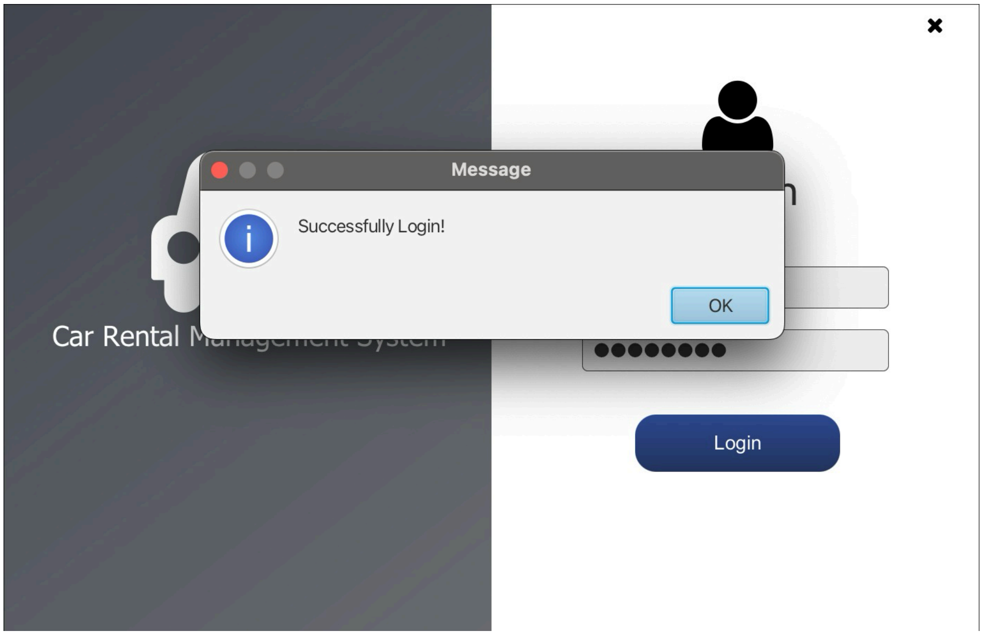
Trying to log in with the wrong username and password -



Error

Wrong Username/Password

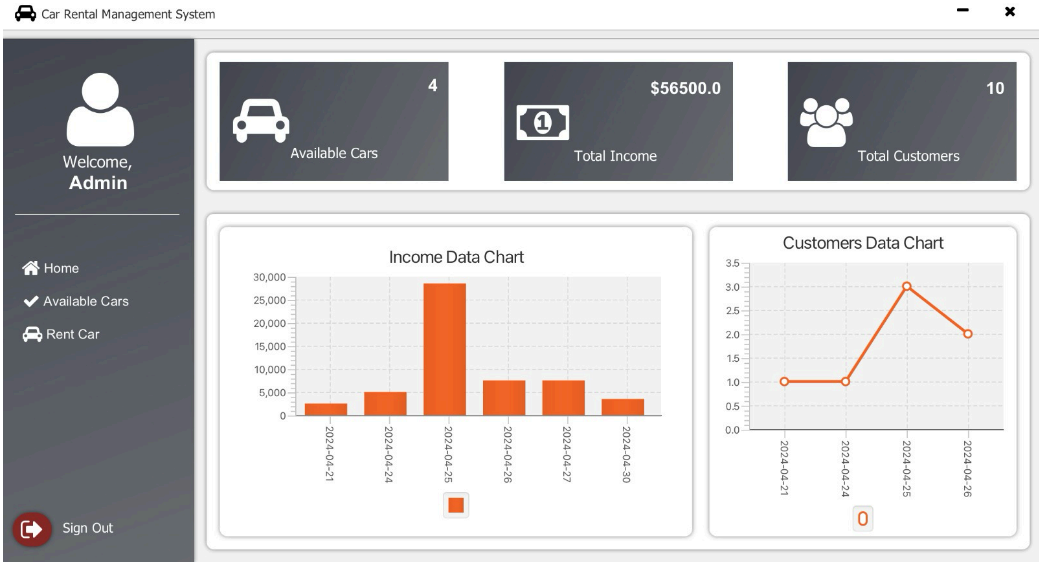
Successfully logging in -



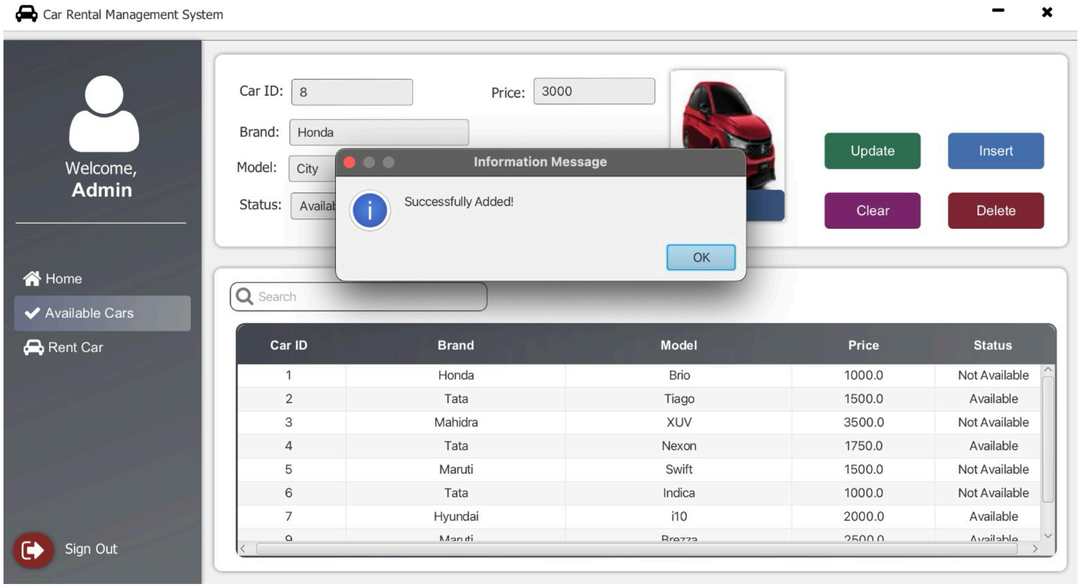
Message

Successfully Login!

Dashboard -



Adding a new car to the list of cars for rental -



Car ID: 8

Price: 3000

Brand: Honda

Model: City

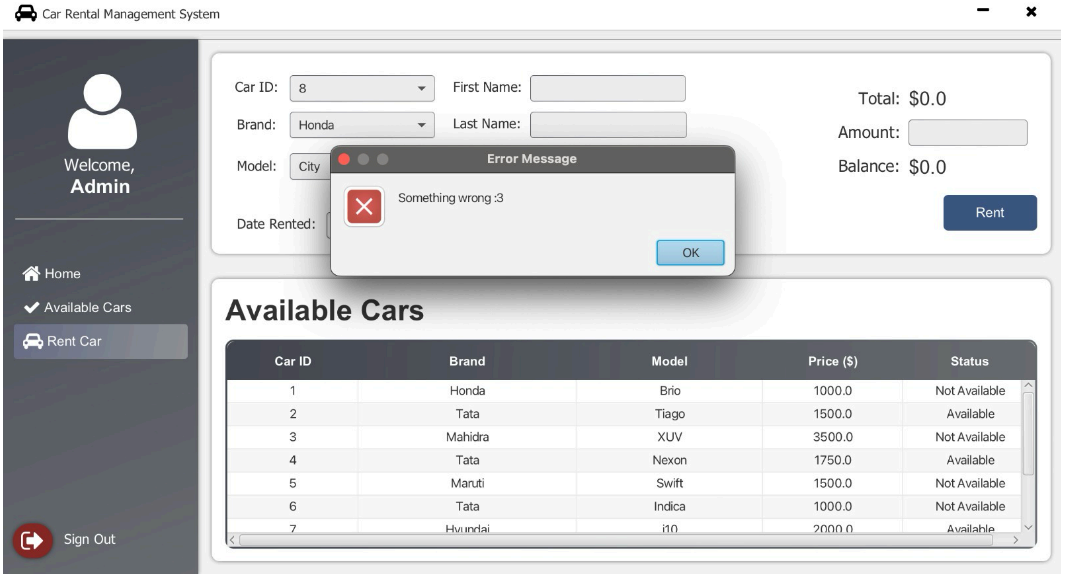
Status: Availal

Information Message

Successfully Added!

Car ID	Brand	Model	Price	Status
1	Honda	Brio	1000.0	Not Available
2	Tata	Tiago	1500.0	Available
3	Mahidra	XUV	3500.0	Not Available
4	Tata	Nexon	1750.0	Available
5	Maruti	Swift	1500.0	Not Available
6	Tata	Indica	1000.0	Not Available
7	Hyundai	i10	2000.0	Available
9	Maruti	Brezza	2500.0	Available

Unsuccessful if all values are not added -



Car ID: 8

Brand: Honda

Model: City

Date Rented:

First Name:

Last Name:

Total: $0.0

Amount:

Balance: $0.0

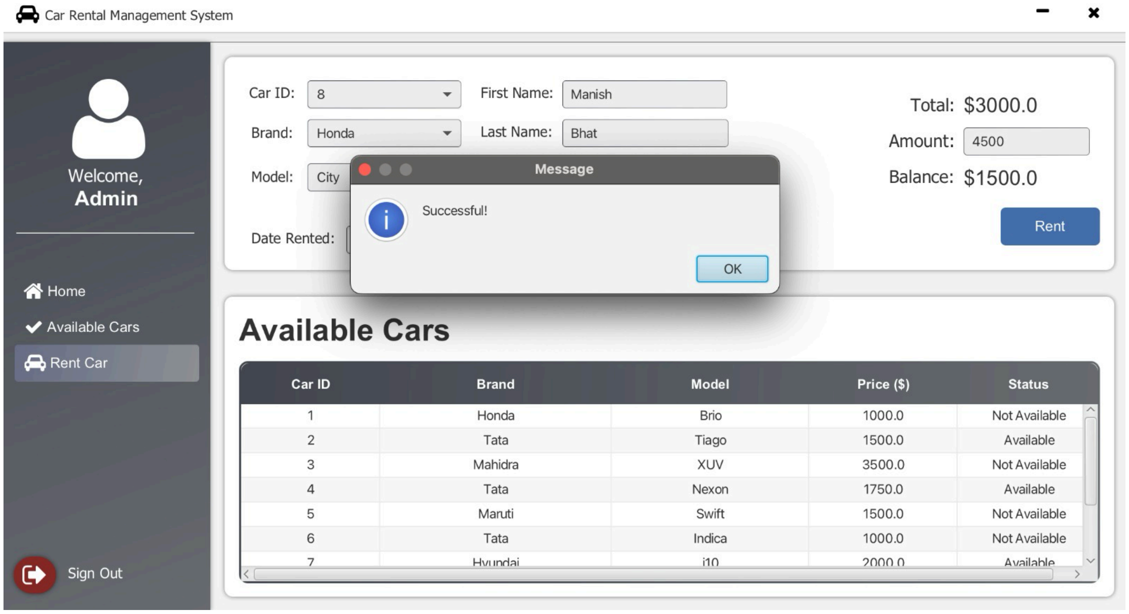
Error Message

Something wrong :3

Available Cars

Car ID	Brand	Model	Price ($)	Status
1	Honda	Brio	1000.0	Not Available
2	Tata	Tiago	1500.0	Available
3	Mahidra	XUV	3500.0	Not Available
4	Tata	Nexon	1750.0	Available
5	Maruti	Swift	1500.0	Not Available
6	Tata	Indica	1000.0	Not Available
7	Hyundai	i10	2000.0	Available

If the value inserted to pay is higher than the total amount then payment and rental is successful and the balance amount to be returned to the customer is returned -



Car ID: 8

Brand: Honda

Model: City

Date Rented:

First Name: Manish

Last Name: Bhat

Total: $3000.0

Amount: 4500

Balance: $1500.0

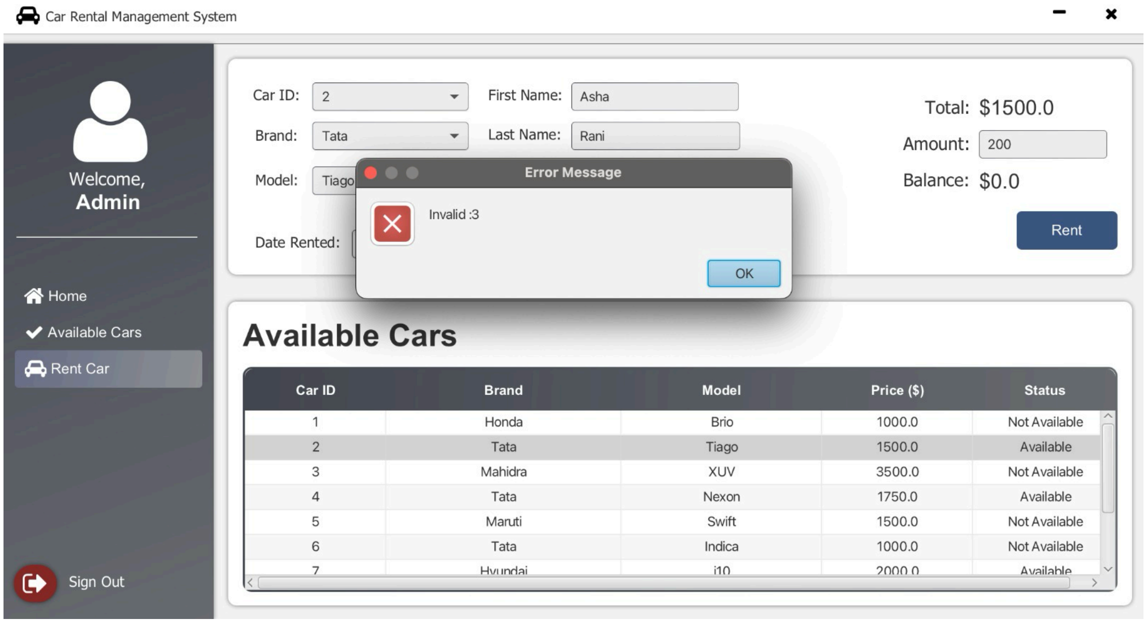
Message

Successful!

Available Cars

Car ID	Brand	Model	Price ($)	Status
1	Honda	Brio	1000.0	Not Available
2	Tata	Tiago	1500.0	Available
3	Mahidra	XUV	3500.0	Not Available
4	Tata	Nexon	1750.0	Available
5	Maruti	Swift	1500.0	Not Available
6	Tata	Indica	1000.0	Not Available
7	Hyundai	i10	2000.0	Available

If the Amount entered is lower than the total amount to pay then the payment fails -



Car ID: 2

Brand: Tata

Model: Tiago

Date Rented:

First Name: Asha

Last Name: Rani

Total: $1500.0

Amount: 200

Balance: $0.0

Error Message

Invalid :3

Available Cars

Car ID	Brand	Model	Price ($)	Status
1	Honda	Brio	1000.0	Not Available
2	Tata	Tiago	1500.0	Available
3	Mahidra	XUV	3500.0	Not Available
4	Tata	Nexon	1750.0	Available
5	Maruti	Swift	1500.0	Not Available
6	Tata	Indica	1000.0	Not Available
7	Hyundai	i10	2000.0	Available