

# North South University

# Department of Electrical and Computer Engineering

Semester : Summer 2023

Project Title : **Vehicle Rental Management System**

Course Title : CSE115- Programming Language I Lab

Section : 15

Group No : 02

Group Member’s Name : 1. Pranjal Bhowmik (2031333043)

2. Tawhid Janan Avik (2121844643)

3. Samiul Hasnat Taseen (2233054642)

Name and Initial of Faculty : Rifat Ahmed Hassan (RIH)

Monamy Islam

Date of Submission : 9 November, 2023

**Features**

A full-featured car rental management system serves both clients and rental agency employees with a variety of functions. These functions improve client happiness, expedite the renting process, and improve business operations.

**Customer Features:**

Vehicle Search and Availability: Renters can quickly look for cars that meet their needs by entering a number of parameters, such as price range, location, car type, and length of rental. Real-time availability data ought to be provided by the system to guarantee a flawless booking process.

Online Reservation and Booking: Users should be able to reserve cars straight from the system, doing away with the need for calls or in-person meetings. Reservations, including payment processing, email confirmations, and reminders, ought to be managed by the system.

Account management: Renters should be able to view upcoming reservations, manage their past rentals, make changes to existing reservations, and update personal data via a safe online account.

Rental Agreement and Documentation: To ensure clarity and legal compliance, the system should produce rental agreements and other required documentation for each rental transaction.

**Rental Agency Staff Features:**

Vehicle Fleet Management: Employees should be able to oversee the whole fleet of vehicles, including scheduling maintenance, adding, deleting, and keeping thorough records of each vehicle.

Management of the Rental Process: Employees should be in charge of processing reservations from clients, creating rental contracts, handing out keys, and confirming client identity and driver's licenses.

Processing payments and billing: The system must manage all financial transactions, including those made online, as well as the creation of invoices and the management of client deposits and refunds.

User Management and Access Control: Staff members should be able to control their access privileges and prevent unwanted access by using the system's secure user authentication and authorization.

Inventory management: To maximize inventory levels and rental operations, staff members should be able to track vehicle availability, manage vehicle locations, and keep an eye on vehicle usage.

Performance analytics and reporting: To help with business decision-making, the system should produce reports on rental trends, vehicle utilization, customer demographics, and revenue generated.

**Description of each function**

registe() function:

This function handles the user registration or signup process. This takes users information from the user through user input and stores them for further validation while login.

login() function:

This function handles the user login process. This takes users username and password from the user through user input and validates them with the stored data. If both the username and password are correct only then then user shall be allowed to login and access the system.

imp() function:

This function handles the main car rental process. It prompts users to select car categories, car types, and rental durations. It calculates the total rental cost based on the chosen car and duration. It then prompts users to choose a payment method (via card or outlet) and processes the payment accordingly. Finally, it generates a receipt for the user.

displayInventory() function:

This function to display available car inventory. Provides user the list of cars along with prices to choose from.

premium(), normal(), low() functions:

These functions handle the selection of premium, normal, and low-cost cars within each category (Sedan, SUV, Pickup Truck). They prompt users to choose specific car models and display their corresponding rental prices.

Case 1, Case 2, Case 3:

These cases handle the selection of Sedan, SUV, and Pickup Truck categories, respectively. They call the premium(), normal(), low() functions based on the user's choice of car type.

Payment by VIA CARD and payment by OUTLET sections:

These sections handle the processing of card payments and outlet payments, respectively. For card payments, it prompts users to enter their card number, expiry date, and CVV, and simulates a delay for processing. For outlet payments, it informs users of the outlet locations and instructs them to make the payment at their earliest convenience.

RECEIPT.txt file generation:

The code generates a receipt file (RECEIPT.txt) for both card payments and outlet payments. The receipt includes the user's card information (for card payments), total rental cost, and payment amount.

main() function:

This function serves as the entry point for the program. It prompts users to register or login and calls the appropriate functions based on their choice. It also removes the Car.txt file after the program execution to prevent stacking of information.

Overall, the code demonstrates a comprehensive car rental management system with various functionalities, including car selection, cost calculation, payment processing, and receipt generation.

**Applications**

1. User Registration and Login: Allows users to create accounts and log in securely, ensuring access control.
2. Car Category Selection: Provides options for users to choose from different categories of cars, including Sedan, SUV, and Pickup trucks.
3. Type of Car Selection: Allows users to select the type of car they want to rent from options such as Premium, Normal, and Low Budget.
4. Billing Calculation: Calculates the total bill based on the user's choices and the duration of the rent, considering different car types and categories.
5. Payment Options: Supports two payment methods - payment via card and payment through an outlet, providing flexibility to users.
6. Receipt Generation: Generates a detailed receipt for the transaction, including chosen car details, payment amount, and payment method.
7. Record Keeping: Maintains a record of user transactions, enabling car rental businesses to track customer activity and payments.
8. Potential for Expansion: Can be expanded to include additional features such as reservation systems, real-time availability checks, and integration with online payment gateways for further convenience.
9. Business Management: Assists car rental businesses in managing their vehicle fleet, customer interactions, and financial transactions efficiently.
10. Customer Satisfaction: Enhances customer satisfaction by providing a seamless rental experience, accurate billing, and well-documented receipts for transparency and trust.

**Limitations:**

1. Security Concerns: The project lacks robust security features. Storing sensitive user information (such as passwords) in plain text files is highly insecure.
2. User Interface: The project does not have a user-friendly graphical interface. A graphical user interface (GUI) would enhance user experience significantly.
3. Error Handling: There is limited error handling in the code. It does not handle various edge cases, input validation, or unexpected user inputs, which can lead to unexpected behavior.
4. Data Validation: The project does not validate the input data thoroughly. For instance, it assumes users will always input the correct data format and type, leading to potential issues if users provide unexpected inputs.
5. File Management: The code deletes the "Car.txt" file at the end of the program, which means all the data is lost after each run. A better approach might be appending new data to the existing file or using a database for more structured data storage.

**Future Work:**

1. Enhanced Security: Implement robust security features, such as encryption for storing sensitive user data, to ensure the protection of user information.
2. User Interface Development: Design and implement a user-friendly graphical interface to enhance the user experience and make the application more accessible.
3. Input Validation: Implement comprehensive input validation to handle various scenarios, preventing unexpected inputs and enhancing the overall robustness of the system.
4. Database Integration: Integrate a database system (like MySQL or SQLite) to store user information and rental data securely. This would allow for better data management and persistence across different runs of the program.
5. Error Handling: Implement proper error handling mechanisms to provide informative error messages to users in case of incorrect inputs or system failures.
6. Payment Gateway Integration: Integrate a secure payment gateway for handling online payments securely, ensuring the safety of financial transactions.
7. Reservation System: Implement a reservation system to allow users to reserve cars for specific dates and times, providing a more comprehensive rental management solution.

**Conclusion**

By addressing these limitations and working on the future enhancements, the Rental Management System can become a reliable and secure tool for car rental businesses and a user-friendly platform for customers**.**

**Contribution**

1. Registration and login functionality were implemented by Pranjal Bhowmik (2031333043).
2. Payment method and receipt generation functionality were implemented by Tawhid Janan Avik (2121844643)
3. Inventory, premium, normal, low functionality and some parts of imp function were implemented by Samiul Hasnat Taseen (2233054642)