Car Rental Management system

- Project Title: Car Rental Management

- Your Name: Amal Roy

- Roll Number: 12

- Course Name: Programming in C

- Date:

Introduction

Brief overview of the project

The Car Rental Management System is a software application designed to manage car rentals efficiently. It allows the user to add new cars, book cars for customers, and return cars while calculating the rental charges.

Problem statement

Managing car rentals manually can be cumbersome and prone to errors. There is a need for a systematic way to keep track of available cars, customer bookings, and returns.

Objective

The objective of this project is to develop a simple car rental management system in C that can:

* Add new cars to the system.
* Allow customers to book cars.
* Handle car returns and calculate rental charges.

System Requirements

Hardware requirements

\*Processor: Intel i3 or higher

\* RAM: 4GB or more

\* Hard Disk: 100MB free space

Software requirements

\*Operating System: Windows, Linux, or macOS

\*Compiler: GCC or any C compiler

\*IDE: Visual Studio Code, Code::Blocks, or any text editor

Design and Development

Description of the program logic

The program maintains two main lists: one for cars and another for customers. Each car has an ID, model name, and availability status. Each customer has an ID, name, car ID they rented, and the number of days rented.

The system provides three main functionalities:

1. Allows adding new cars to the system.
2. Allows customers to book available cars.
3. Handles car returns and calculates the rental charges based on the number of days rented.

Psuedocode

Start

Initialize arrays for cars and customers

Initialize counters for the number of cars and customers

Display menu

If choice is 1:

* Add a new car

If choice is 2:

* Book a car

If choice is 3:

* Return a car

If choice is 4:

* Exit

End

Testing and Results

Test cases

Test Case ID: 1

Description: Test adding a new car to the system.

Preconditions: The system is running and ready to accept input.

Test Steps:

\*Run the Car Rental Management System.

\*Select the "Add Car" option by entering 1.

\*Enter car ID: 101.

\*Enter car model: Toyota.

\*Expected Result: The system should display "Car added." and the car should be listed in the system as available.

\*Actual Result: Car added

\*Status: Pass

A screenshot of a computer program

Description automatically generated

The successful execution of test case 1 demonstrated that the "Add Car" functionality in the Car Rental Management System works as expected. The system correctly handled user inputs, stored the new car details, and provided appropriate feedback to the user, indicating that the car was added successfully. This confirms that the system accurately manages the addition of new cars, ensuring they are available for booking in subsequent operations. The actual results matched the expected results, validating the correctness of this functionality and the robustness of the system's input handling and data storage mechanisms.

Conclusion

The Car Rental Management System is a straightforward C program designed to manage car rentals. It provides basic functionalities such as adding new cars, booking cars, and returning cars. This system aims to simplify the process of managing car rentals and reduce errors associated with manual handling.

Future enhancements

\*User Authentication: Add user authentication to secure access to the system.

\*Database Integration: Integrate a database to handle larger data sets and provide persistent storage.

\*GUI: Develop a graphical user interface to make the system more user-friendly.

\*Enhanced Reporting: Add features for generating detailed reports on car rentals.

References

\*C Programming Language, 2nd Edition, by Brian W. Kernighan and Dennis M. Ritchie

\*GeeksforGeeks

\*Stack Overflow