Car Rental System

Documentation

Omar Samir 211000372

Abdullah El Khateeb 211001730

Mahmoud Reda 211000489

Nile University

Dr. Soha

Car Rental System

The Car Rental System is a program implemented in C++ that simulates a basic car rental service. It allows users to perform operations such as renting a car, returning a car, reserving a car, canceling a reservation, repairing a car, and viewing the available, rented, and reserved cars.

Algorithm

1. Start the program and prompt the user for a username and password to access the car rental system.

2. Validate the provided credentials. If the credentials are correct, grant access to the main menu; otherwise, display an error message and terminate the program.

3. Display the main menu with available options.

4. Based on the user's choice, execute the corresponding function.

- If the user chooses to rent a car, prompt for the car index, customer name, pickup date, and return date. Call the rent_car function.
- If the user chooses to return a car, prompt for the car index. Call the return_car function.
- If the user chooses to reserve a car, prompt for the car index and customer name. Call the reserve_car function.
- If the user chooses to cancel a reservation, prompt for the car index. Call the cancel_reservation function.
- If the user chooses to repair a car, prompt for the car index and mechanic name. Call the repair_car function.
- If the user chooses to show available cars, call the show_available_cars function.
- If the user chooses to show rented cars, call the show_rented_cars function.
- If the user chooses to show reserved cars, call the show_reserved_cars function.
- If the user chooses to exit, terminate the program.
- If the user chooses an invalid option, display an error message and return to the main menu.

5. After executing the chosen function, return to the main menu and repeat the process until the user chooses to exit.

Car Struct

The Car struct represents a car object and contains the following properties:

- **model (string):** The model of the car.
- **is_rented (bool):** Indicates whether the car is currently rented or not.
- **is_repaired (bool):** Indicates whether the car has been repaired or not.
- **is_reserved (bool):** Indicates whether the car is currently reserved or not.
- **rented_by (string):** Stores the name of the customer who rented the car.
- **reserved_by (string):** Stores the name of the customer who reserved the car.
- **repaired_by (string):** Stores the name of the mechanic who repaired the car.
- **pickup_date (string):** Stores the date when the car was picked up for rental.
- **return_date (string):** Stores the date when the car is scheduled to be returned.
- **next (Car*):** Points to the next car in the linked list.

CarRentalSystem Class

The CarRentalSystem class manages the car rental system and provides the following functions:

add_car(string model): O(1)

- Adding a new car involves creating a Car object, setting its properties, and adding it to the linked list. Since the car is added at the end of the list, this operation has a constant time complexity.

get_car(int car_index): O(N)

- Retrieving a car from the linked list based on its index requires traversing the list until the desired index is reached or the end of the list is reached. Therefore, the time complexity is linear and depends on the size of the list.

rent_car(int car_index, string customer_name, string pickup_date, string return_date): O(N)

- Renting a car involves validating the pickup and return dates and marking the car as rented. Additionally, the function calls the calculateRentalDays function, which iterates day by day between the pickup and return dates. Therefore, the time complexity is linear, as it depends on the number of days between the pickup and return dates.

calculateRentalDays(const string& pickup_date, const string& return_date): O(D)

- The calculateRentalDays function calculates the number of days between the pickup and return dates. It converts the dates from strings to integers and iterates day by day until the pickup date reaches the return date. The time complexity is proportional to the number of days (D) between the pickup and return dates.

return_car(int car_index): O(1)

- Returning a rented car involves updating the relevant properties of the car object to mark it as available. This operation has a constant time complexity as it doesn't depend on the size of the list.

reserve_car(int car_index, string customer_name): O(1)

- Reserving a car involves updating the relevant properties of the car object to mark it as reserved. This operation has a constant time complexity.

cancel_reservation(int car_index): O(1)

Canceling a car reservation involves updating the relevant properties of the car object to mark it as available. This operation has a constant time complexity.

repair_car(int car_index, string mechanic_name): O(1)

- Repairing a car involves updating the relevant properties of the car object to mark it as repaired. This operation has a constant time complexity.

show_available_cars(): O(N)

- Displaying the list of available cars requires traversing the linked list and checking the availability of each car. Therefore, the time complexity is linear and depends on the size of the list.

show_rented_cars(): O(N)

- Displaying the list of rented cars requires traversing the linked list and retrieving the details of each rented car. The time complexity is linear and depends on the size of the list.

show_reserved_cars(): O(N)

- Displaying the list of reserved cars requires traversing the linked list and retrieving the details of each reserved car. The time complexity is linear and depends on the size of the list.

CarRentalSystem(): O(1)

- The constructor initializes the CarRentalSystem object by setting the head pointer to nullptr. It has a constant time complexity.

Main Function

The main function is the entry point of the program. It starts by prompting the user for a username and password to access the car rental system. If the provided credentials match the predefined ones ("admin" and "123"), the user is granted access to the main menu.

The main menu provides the following options:

1. Rent a Car: Allows the user to rent a car by providing the car index, customer name, pickup date, and return date.

2. Return a Car: Allows the user to return a rented car by providing the car index.

3. Reserve a Car: Allows the user to reserve a car by providing the car index and customer name.

4. Cancel Reservation: Allows the user to cancel a car reservation by providing the car index.

5. Repair a Car: Allows the user to mark a car as repaired by providing the car index and mechanic name.

6. Show Available Cars: Displays the list of available cars in the car rental system.

7. Show Rented Cars: Displays the list of rented cars in the car rental system.

8. Show Reserved Cars: Displays the list of reserved cars in the car rental system.

9. Exit: Terminates the program.

The program uses a switch statement to execute the corresponding function based on the user's choice. If the provided choice is invalid, an error message is displayed.