User Manual: Car Rental System

For Developer

These functions are part of the **LoginUser class**, and they handle various operations related to user login, booking cars, managing bookings, and displaying information to the user.

1. `string getPasswordInput()`:

- This function is used to securely input a password from the user.
- It masks the password input with asterisks (*) to keep it hidden.
- It works differently based on the platform:
  - On Windows (`_WIN32` defined):
    - It uses `_getch()` to read characters without displaying them.
    - It handles backspace to remove characters from the password.
  - On other platforms:
    - It uses `termios` to turn off echoing of characters while typing.
    - It reads the password using `getline()`.

2. `bool login()`:

- This function allows users to log in by entering their username and password.
- It prompts the user for a username and password.
- It checks if the provided username and password match any user in the `users` vector.
- If a match is found, it sets the `isLoggedIn` flag to `true` and updates the `currentUser` variable with the logged-in user's information.
- If no match is found, it displays an error message and returns `false`.

3. `void registerUser()`:

- This function allows users to register a new account.
- It prompts the user for a username, password, and role (admin or customer).
- It checks if the entered username is already taken.
- If the username is available, it adds the new user to the `users` vector and saves the user data to a file named "users.txt".
- If the username is taken, it displays an error message.

4. `void saveCarsToFile()`:

- This function saves the car records to a CSV file named "cars.csv".
- It creates the file if it doesn't exist or overwrites it if it does.
- It writes the car records including fields like ID, make, model, year, mileage, availability, and rent periods to the CSV file.

5. `void addCar()`:

- This function allows an admin user to add a new car to the system.
- It checks if the user is logged in and has the "admin" role.
- It generates a unique ID for the new car based on existing car IDs.
- It prompts the admin for car details such as make, model, year, mileage, availability, and rent periods.
- It validates the input, ensuring that the year, mileage, and rent periods are within specified ranges.

- It appends the new car data to the "cars.csv" file.

6. `void updateCar()`:

- This function allows an admin user to update an existing car's information.
- It checks if the user is logged in and has the "admin" role.
- It prompts the admin to enter the car ID they want to update.
- It loads existing car data from "cars.csv" and creates a temporary file ("temp_cars.csv") to hold updated data.
- If the specified car ID is found, it updates the car's details (make, model, year, mileage, availability, and rent periods).
- It validates the updated data, and if successful, replaces the original "cars.csv" file with the temporary file containing the updated data.

7. `void deleteCar()`:

- This function allows an admin user to delete an existing car from the system.
- It checks if the user is logged in and has the "admin" role.
- It prompts the admin to enter the car ID they want to delete.
- It loads existing car data from "cars.csv" and creates a temporary file ("temp_cars.csv") to hold updated data.
- If the specified car ID is found, it marks the car as deleted by skipping its data while copying other car data to the temporary file.
- It replaces the original "cars.csv" file with the temporary file containing the updated data if the car is found.

8. `saveBookingsToFile()`

This function saves the booking data to a file named "bookings.csv". It first opens the file for writing, writes a header line, and then iterates through the `bookings` vector, saving each booking's details to the file.

9. `loadCarsFromFile()`

This function loads car data from the "cars.csv" file. It opens the file for reading, clears the existing car data, and then parses the file line by line. It extracts car details (such as ID, make, model, year, etc.) from each line and populates the `cars` vector with this data.

10. `manageBookings()`

This function provides options to manage bookings. It displays a menu with the following options:

- List All Bookings: Displays all booking details, including booking ID, car ID, start date, end date, username, and booking status.
- Approve Booking: Allows an admin to approve a booking by entering the booking ID. The status of the selected booking is changed to "Approved," and the updated data is saved to the file.
- Reject Booking: Allows an admin to reject a booking by entering the booking ID. The status of the selected booking is changed to "Rejected," and the updated data is saved to the file.
- Go back: Returns to the main menu.

11. `showAvailableCars()`

This function displays a list of available cars to the user. It checks if the user is logged in and has the role of "customer" before displaying the list. It reads car data from the "cars.csv" file and prints the details of cars that are marked as available (car's `availableNow` field is set to 1).

12. `calculateRentalFee(int durationInHours)`

This function calculates the rental fee for a given rental duration in hours. It uses a simple formula based on the duration, where the rental fee is $20 per hour plus an additional $20 for each day (24 hours). The calculated fee is returned as a double.

13. `calculateDurationInHours(const string& start, const string& end)`

This function calculates the rental duration in hours based on the start and end dates provided as strings in the format "YYYY-MM-DD." It parses the dates, converts them to timestamps, calculates the time difference in seconds, and converts it to hours. The calculated duration in hours is returned as an integer.

14. `bookCar()`

This function allows customers to book a car. It checks if the user is logged in as a customer and then prompts the user to enter the ID of the car they want to book. It verifies if the car is available and, if so, creates a new booking with the car ID, the current user's username, start date, and end date. It calculates the rental fee based on the duration and updates the car's availability. The booking is added to the `bookings` vector, and car and booking data are saved to their respective files.

15. `loadBookingsFromFile()`

This function loads booking data from the "bookings.csv" file. It opens the file for reading, clears the existing booking data, and then parses the file line by line. It extracts booking details (such as booking ID, car ID, username, start date, etc.) from each line and populates the `bookings` vector with this data.

16. `showBookings()`

This function allows customers to view their own bookings. It checks if the user is logged in as a customer and then displays a list of their bookings. It prints the booking ID, car ID, start date, and end date for each booking.

These functions collectively enable users to book cars, manage bookings (for admins), and view their own bookings (for customers). The data is read from and saved to CSV files to persist the information between program runs.

For User(Admin & Customer)

Welcome to the User Manual for the Car Rental System. This manual provides comprehensive instructions on how to effectively navigate and use the system. The Car Rental System allows users to log in, register as new users, manage cars, and make car bookings.

Table of Contents

1. Getting Started

System Introduction

The Car Rental System allows users to interact with the system based on their roles. There are two user roles: admin and customer.

- Admins have the ability to manage cars and bookings.
- Customers can view available cars, book cars, and view their bookings.

Starting the Program

1. To start the program, execute the compiled program file (e.g., `car_rental.exe`).
2. The program will initialize and load data from existing files.

Logging In

1. If you have an existing account, select option “1” to log in.
2. Enter your username and password when prompted.
3. If the login is successful, you will be presented with the appropriate menu based on your role.

Registering as a New User

1. If you are a new user, select option “2” to register.

2. Enter a unique username, a password, and specify your role (admin or customer).

3. Your account will be registered, and you can log in with your credentials.

Exiting the Program

1. To exit the program, select option “3”.

2. The program will save any changes and data to the appropriate files.

3. The program will then terminate.

2. User Functions

Logged-In User Menu

Once logged in, you will see a menu based on your role (admin or customer). You can select options by entering the corresponding number.

Adding a Car (Admin)

- Admins can add new cars to the system.

- Select option “1” to add a car.

- Follow the prompts to provide car details such as make, model, year, mileage, availability, and rent periods.

- The car will be added to the system.

Updating Car Information (Admin)

- Admins can update existing car information.

- Select option “2” to update a car.

- Enter the car's ID and follow the prompts to update its details.

- The updated information will be saved.

Deleting a Car (Admin)

- Admins can delete existing cars from the system.

- Select option “3” to delete a car.

- Enter the car's ID to confirm deletion.

Viewing Available Cars (Customer)

- Customers can view available cars.

- Select option “4” to see the list of available cars.

Booking a Car (Customer)

- Customers can book available cars.

- Select option “5” to make a booking.

- Follow the prompts to select a car and specify booking details.

- A booking will be created.

Viewing My Bookings (Customer)

- Customers can view their bookings.

- Select option “6” to see a list of your bookings.

Logging Out

- Select option “7” to log out of your account.

- You will be returned to the login/registration menu.

Exiting the Program

- Select option “8” to exit the program.
- The program will save any changes and data to the appropriate files.

3. Admin Functions

Managing Bookings (Admin)

- Admins can manage bookings, including approving or rejecting them.
- Select option **9** to manage bookings.
- Follow the prompts to view and manage bookings.

4. System Data

Loading Data from Files

- The program loads user, car, and booking data from files when it starts.
- User data is loaded from "users.txt".
- Car data is loaded from "cars.csv".
- Booking data is loaded from "bookings.txt".

Saving Data to Files

- The program automatically saves car and booking data to files before exiting.
- Car data is saved to "cars.csv".
- Booking data is saved to "bookings.txt".