CAR RENTAL SYSTEM DOCUMENTATION



Name: IRTIZA ALI

Roll No. 2023-SE-02

1st Semester Final Project Documentation

Submitted to: Sir Shehzad Asif

**Introduction:**

The Car Rental Management System is a comprehensive C++ program designed and implemented as a semester project for the Programming Fundamentals. The primary objective of this system is to facilitate the management of car rentals with distinct functionalities for both customers and administrators. Through the utilization of fundamental programming concepts, the project addresses key aspects such as user authentication, efficient data management, and a user-friendly interface.

**User Roles:**

* **Customers:** Can rent, return cars, view available cars, and generate rental reports for their transactions.
* **Admins:** Have the authority to manage the car inventory by adding or removing cars, view rental reports for all customers and cars, and perform administrative tasks.

**File Handling:**

File handling is a pivotal feature in the Car Rental Management System, streamlining data storage and retrieval. The system employs distinct text files to manage various aspects efficiently. "admin.txt" securely stores admin credentials, ensuring authenticated access. "cosid.txt" dynamically track the count of customers , adapting to changes in the user base. Individual customer details are maintained in "costumer.txt," while the "cars.txt" file tracks the car inventory. Furthermore, the system creates unique files for each registered user, enhancing personalized data management. This file handling approach enhances data security, accessibility, and organization, contributing to the overall effectiveness of the Car Rental Management System.

**Functions Implementation:**

1. \*\*`adminSide`:\*\*

void adminSide(cardata\* carptr, int\* carsnumber, costumerdata\* costumerptr, int\* costumernumber)

this function login the admin by providing its name and password

- Calls `adminoptions` on successful login.

1. \*\*`adminoptions`:\*\*

*adminoptions(cardata\* carptr, int\* carsnumber, costumerdata\* costumerptr, int\* costumernumber)*

This function displays the menu from which admin can select the option of his own choice.

- Displays admin menu.

1. \*\*`changeadminpass`:\*\*

void changeadminpass()

this function simply enables admin to change his password and stores the new password in file of admin.

4. \*\*`addcar` Function:\*\*

*cardata\* addcar(cardata\* car, int\* carnumber)*

In this function I have made a new dynamic array by increasing its size to car number +1.I copy the whole data into new array and enters the new data at the last of array anad return it.

5.\*\*`viewcars` Function:\*\*

*void viewcars(cardata\* carptr, int\* carsnumber)*

this function simply uses a for loop to display all the cars by calling

print car function

6.\*\*`printCar` Function:\*\*

void printCar(cardata s)

- Purpose: Prints detailed information about a single car.

7.\*\*`PrintcarReport` Function:\*\*

*void PrintcarReport(cardata\* car, int\* carsnumber)*

this function takes the id of car and displays complete information related to it.

8.\*\*`searchCarData` Function:\*\*

*int searchCarData(cardata\* car, int\* carsnumber)*

this function simply searches a car by putting its id and returns the index of that very car.

9.\*\*`updateCarData` Function:\*\*

*void updateCarData(cardata\* car, int\* carsnumber)*

this function takes input the id of car you want to update data of; by calling searchcardata function and then you can update the data of that car

10.\*\*`deleteCarData` Function:\*\*

*cardata\* deleteCarData(cardata\* car, int\* carsnumber)*

This function takes the id of car you want to delete and then make dynamic array of one less size than car array and copies the whole data into it except the one you want to delete the data of and returns that array.

11.\*\*`addCostumer` Function:\*\*

*costumerdata\* addCostumer(costumerdata\* costumer, int\* costumernumber)*

this function makes a new array of 1+ size than costumer and copies the whole data of costumer into new array and enter the new costumer data at the last index of that new array named newcostumer.

12.\*\*`viewcostumer` Function:\*\*

*void viewcostumer(costumerdata\* costumer, int\* costumernumber)*

this function simply uses a for loop upto the number of costumers and prints the data of all costumers by calling the printcostumer function.

13. \*\*`printCostumer` Function:\*\*

- Purpose: Prints detailed information about a single customer.

14.\*\*`searchCostumer` Function:\*\*

int searchCostumer(costumerdata\* costumer, int\* costumernumber)

this functions searches for the id of the costumer and returns the index of that costumer in array.

- Purpose: Searches for a customer based on their ID and prints their details.

15.\*\*`updateCostumerData` Function:\*\*

void updateCostumerData(costumerdata\* costumer, int\* costumernumber)

this function takes the index of that costumer of which you want to update the data and then you can change the data of that id.

16. \*\*`deleteCostumerData` Function:\*\*

*costumerdata\* deleteCostumerData(costumerdata\* costumer, int\* costumernumber)*

this function takes the inedex of costumer you want to delete by calling the searchCostumer function and then make dynamic array of size 1 less than costumernumber and then copies the whole data into that new array except for that index of which you want to delete the data and returns that new array.

17.\*\*`costumerSide` Function:\*\*

*void costumerSide(cardata\* car, int\* carnumbers, costumerdata\* costumer, int\* costumernumber)*

this function logins the costumer by entering the id and password given by admin.

18.\*\*`costumerOptions` Function:\*\*

*void costumerOptions(cardata\* car, int\* carnumbers, costumerdata\* costumer, int\* costumernumber, int id)*

this function displays the menu to the user if the user logins correctly and then costumer can choose the option whichever he likes.

19.\*\*`rentCar` Function:\*\*

cardata\* rentCar(cardata\* car, int\* carnumbers, costumerdata\* costumer, int\* costumernumber, int& id)

this function let the user take a car on rent if the car is available and user has not rented any other car and also writes the history of costumer into the file it also asks the number of days for which the user want to take the car on rent.

20.\*\*`returnCar` Function:\*\*

*cardata\* returnCar(cardata\* car, int\* carnumbers, costumerdata\* costumer, int\* costumernumber, int& id)*

this function let the costumer return a car if the user has rented any car and update the status of car and costumer in the program as available

21. \*\*`printRenalReport` Function:\*\*

*void printRenalReport(cardata\* car, int\* carnumbers, costumerdata\* costumer, int\* costumernumber, int& id)*

this function prints the rental report of the costumer which includes all the history and current rented car details and payment details.

22. \*\*`timefunction` Function:\*\*

*string timefunction()*

This function returns the current date and time as string by using different functions and structures of time library

- Uses `ctime\_s` and `put\_time` functions for time formatting.

23. \*\*`carpointerdata` and `costumerpointer` Functions:\*\*

*cardata\* carpointerdata()*

*costumerdata\* costumerpointer()*

these to function reads the data of costumers and cars from file into structure dynamic array and returns those pointers.

24. \*\*`noOfCars` and `noOfcostumers` Functions:\*\*

*int\* noOfCars()*

*int\* noOfcostumers()*

these two function reads the number of cars and costumers registered into dynamic pointer and return those pointers.

25. \*\*`exitfunction` Function:\*\*

*void exitfunction(const cardata car[], const int carsnumber, const costumerdata costumer[], const int costumernumber)*

this function writes the whole data from car and costumer structure back into the files

26. \*\*Validation Functions:\*\*

*bool validateCarNumber(const string& carNumber)*

*bool validatePhoneNumber(const string& phoneNumber)*

*bool validateCNICNumber(const string& CNICNumber)*

*void validateNum(int& num)*

these functions validate the inputs that are entered

27. \*\*Print Messages Functions:\*\*

*void printWelcomeMessage()*

*void printCongratulationsAdmin()*

*void printCongratulationsCustomer()*

printWelcomeMessage, printCongratulationsAdmin, and printCongratulationsCustomer functions display welcome messages and messages for logging in as an admin or customer.