CS-112 Project : Car Rental and Buying Management System.

**\_\_\_**

By

M.Hamdoon: 2024387, Basil Joseph: 2024138 ,

Yahya Khalid Khan : 2024664 ,M.Zeeshan Naseem:2024495

# 1.INTRODUCTION

The projects purpose is for the automation of Car renting and purchasing processes for both customers and administrators alike. This C++ project provides:

1.Proper protected authentication and verification for both customers and admins of the system

2.comprehensive management of different vehicles and their subtypes depending on the customers demands

3. Transaction process for either rent or purchasing of the chosen cars

4. Proper in depth receipts for any transaction made

This system not only solves the hardwired problem of manually keeping records of any transactions but also provides a digital solution to track transactions made by users .

## 2.Summary Of Functional Requirements:

Functional requirements are a set of basic instructions or functionalities that a system follows to be able to smoothly perform the tasks given by the users

1.User Authentication:

⇢ Registration and Login for customers

⇢Admin Login based on a predetermined login info providing special privileges to manage the system

2.Car Managing:

⇢ Add and Remove any Cars and their subtypes from inventory (**admin only**)

⇢Search Vehicles by inputting their make and model and then also displaying the available vehicles (**customer specific**)

3.Transactions:

⇢ Process Renting of cars with their return dates

⇢Process purchasing of vehicles with ability to add warranty.

4.Administrative Functions:

⇢ Adding and removing of cars from inventory

⇢Managing car and their subtypes inventory

5.Other Features:

⇢ Menus are displayed and repeated for each action and part of the system

⇢upon exiting the program the user is thanked!

## 3.Tools and Technologies Used:

⇢ C++ is used as a programming language

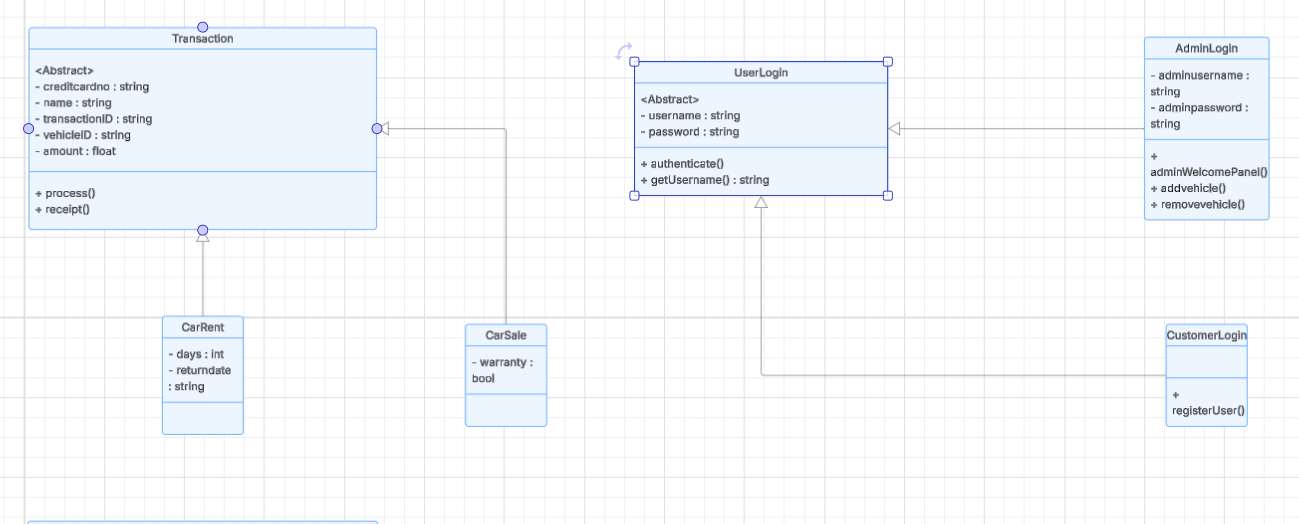
⇢Some of the libraries used are :**<iostream>,<ifstream>,<ctime>,<iwindows.h>,<string>**

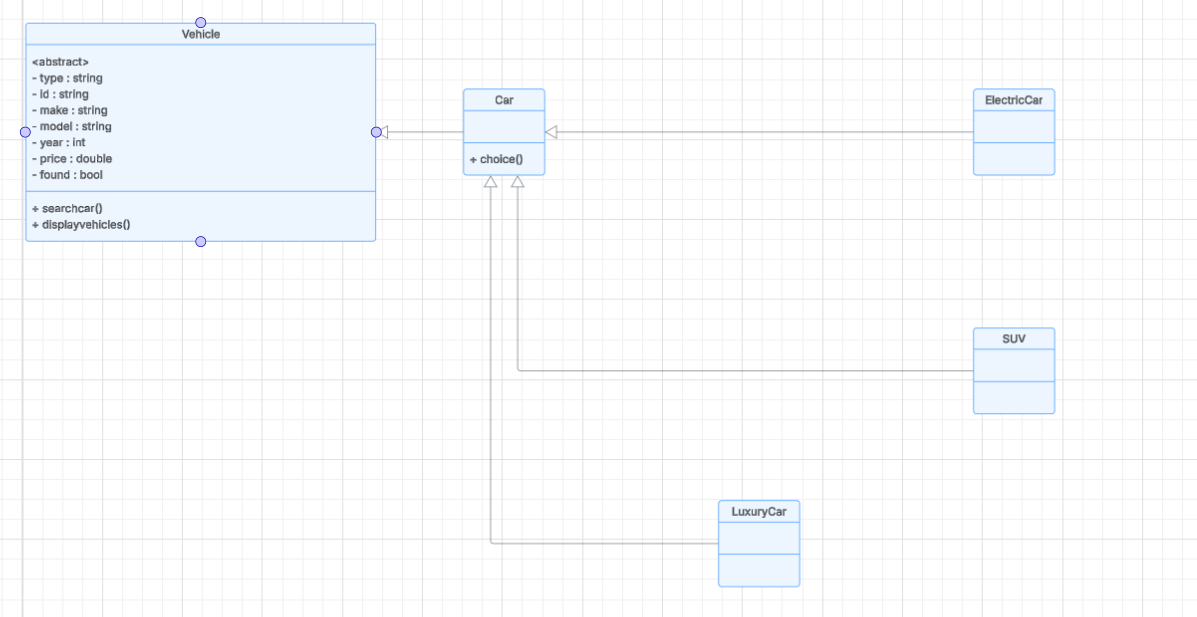
⇢ Interface used is Command line interface

⇢VsCode and Devc++ were used as the IDEs

⇢Windows was used for the OS.

4.System design:





## 5.Concepts Used And Their Implementation:

1.Encapsulation:

It is used throughout the program to hide details regarding the said processes

⇢private/protected members are used throughout all classes

⇢For e.g in the transaction class private data like credit card number is private whereas interface processes like process() and receipt() are made public

⇢user login class privates the username and password only being able to access them using getter methods

2.Inheritance:

It is used throughout the program to create hierarchy between classes

⇢**Vehicle abstract base class** is inherited by car which is further inherited by specialized types like **suv,electric car,luxury car.**

⇢Transaction base class is inherited by carrent and carsale classes aswell

⇢**User login** abstract baseclass is inherited by **adminlogin and customerlogin**

3.Polymorphism:

It is implemented in several ways:

⇢Transaction base class showcases the use of pure virtual functions like **process() and receipt()** which are after that overridden in derived classes like **carrent and carsale** with their individual implementations

⇢**vehicle and userlogin** are abstract base classes that contain pure virtual functions like **searchcar( ), display vehicles( ), authenticate( )** forcing derived classes to implement these methods so that they dont get abstract themselves

4.Abstraction:

It is implemented in several ways:

⇢Abstract base classes like v**ehicle, userlogin and transaction** define interfaces without any implementation

⇢hiding complex operations like process() in transaction class

**6.CHALLENGES:**

⇢ **File permission issues:**Soemtimes it would say permission denied when trying to remove files or access files in the program. So we tried to implement proper file handling with error checking

**⇢Irregular looping problems:** sometimes when printing the receipt the program would give improper loops that go on till infinite resolution fixed it by adding breaks in the loops where required

**⇢Files not displaying:** sometimes the cars would not display properly due to invalid string formatting in files so we fixed it with proper formatting

**7.TESTING:**

⇢ **Unit Testing:**

Verified correct login/logout functionality and tested all admin operations to add and remove cars. Also tested all transaction and receipt processes

**⇢Integration Testing:**

Verified all the work done from login to transaction part of the program

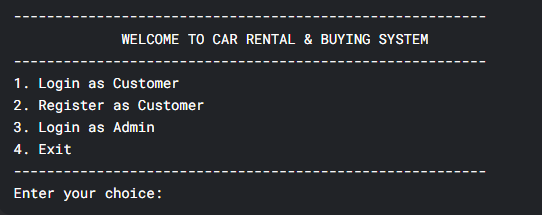
**⇢Edge cases:**

Using numbers as input instead of string when required causes infinite loops.

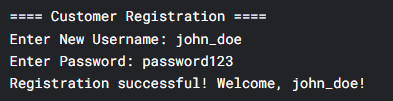
Empty files causes nothing to display aswell.

**8.OUTPUTS:**

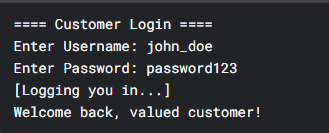
1.Main Menu:



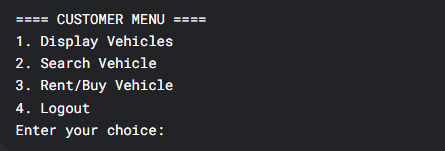
2.Customer Reg:



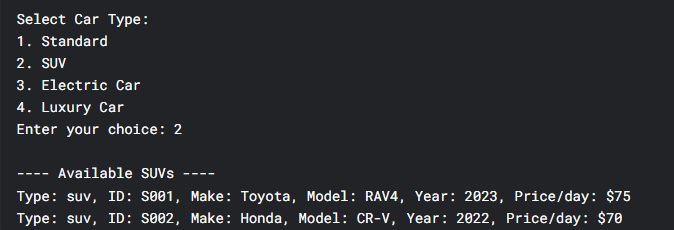
3.Customer Login:



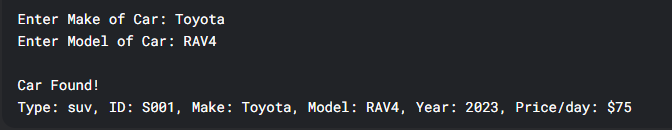
4.Customer Menu:



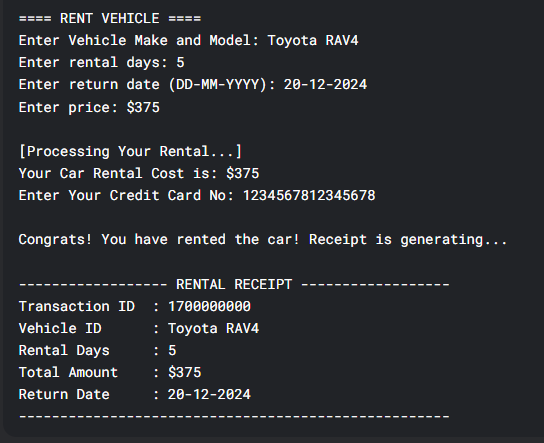
5.Display Vehicles:



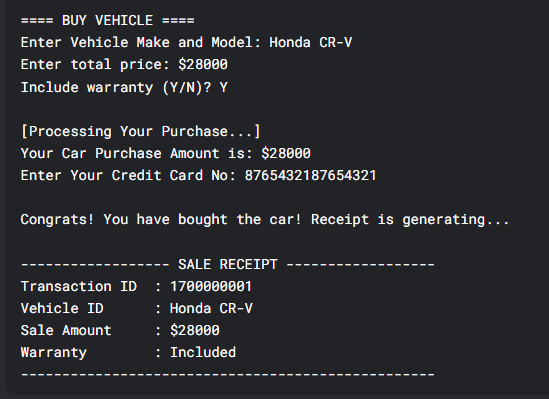
6.Search Vehicles:



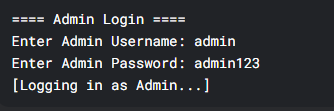
7.Rent Vehicles:



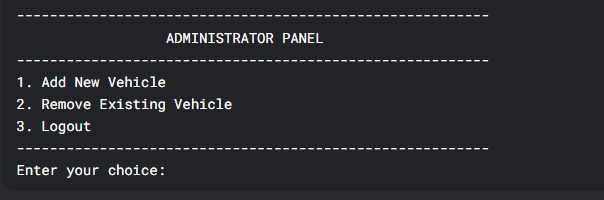
8.Buy Vehicle:



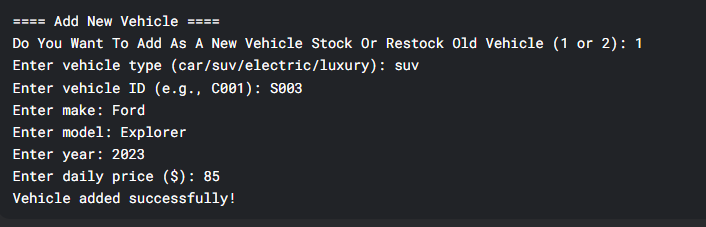
9.Admin LOGIN:



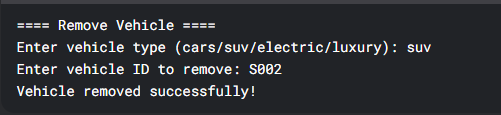
10.Admin Panel:



11.Add Vehicle:

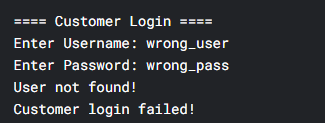


12.Remove Vehicle:

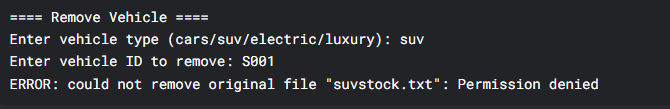


13.ERROR CASES:

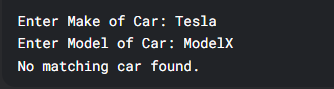
A.Invalid Login:



B.File Permission Error:



C.Vehicle Not Found:



14.Program Exit:



**9.Areas to Improve Upon:**

1.Better reports for the admin:  
⇢ Generate daily and weekly reports for the admin to analyze and export them to an excel sheet

2.Filter Options:

⇢ Filter vehicles based on their price ranges and search by more criterias aswell

3.Better Payment:

⇢ Support for multiple payment method and confirmation email for payment

4.Gui Implementation:

⇢ switch from CLI to GUI