Parking Reservation System

Description:

The purpose of this project is to manage and maintain the multi level parking garage. It allows the customer to reserve an available spot, check availability and price, and check out of the garage. Parking Reservation System is done using Object Oriented Programming concepts Inheritance, Data Encapsulation, Abstraction and Polymorphism. The customer data (the vehicle number and vehicle type) is secured using Data Encapsulation. The Vehicle class is an abstract class from which the Automobile class inherits. When a car arrives the customer is given the following menu options.

1. Reserve a spot
2. Check availability
3. Check Parking charges
4. Check out of Parking lot
5. Exit the menu

When a customer reserves a spot he/she has to provide their Vehicle number and the type of Vehicle they drive from the following options.

car - 1
compact_car - 2
motorcycle - 3
bus - 4
electric_car - 5
truck - 6

The customer receives a confirmation display message after successfully reserving a spot. Alternatively the customer can choose to check the availability of open slots or check the charges of the parking lot before going for reservation. When the customer checks out, they need to enter their vehicle number and the system calculates the amount of time it remains in the parking lot and displays the bill.

Program Flow:

1. Gets the user input - Vehicle number and Vehicle type (car,bus,truck so on)
2. Creates an Automobile object and store it in a vector inside the Parking lot class
3. The Automobile object consists of Vehicle number, Vehicle type, the number of parking spots the particular vehicle can occupy, and the entry time of the vehicle.
4. The Parking lot object is updated. The number of slots available in each level of the parking lot and the details of vehicles currently occupied.
5. When the user checks out the bill is calculated by taking the current time as exit time. Billing object returns the total fee with tax. The Automobile object is deleted from the Parking lot object and the number of cars in the parking lot is updated.

Input:

- Main choice
- Vehicle number
- Vehicle type

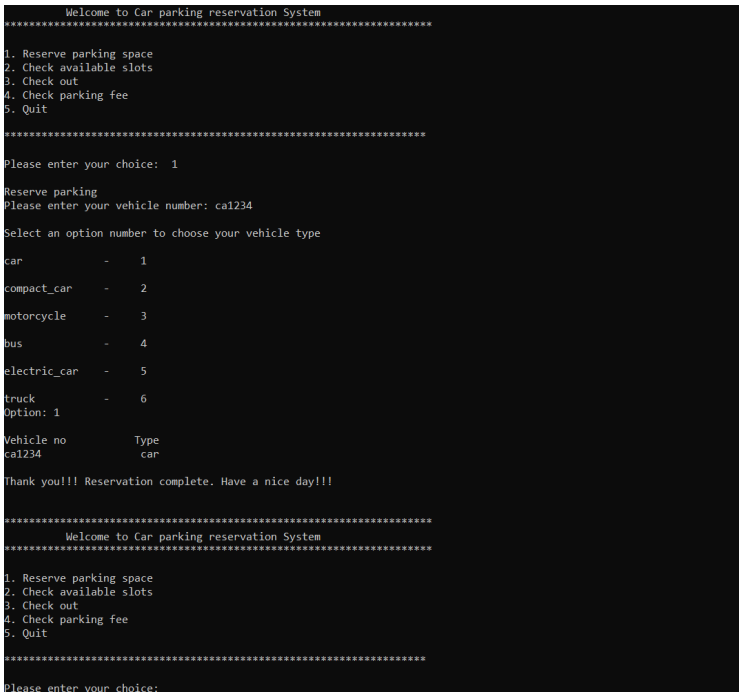
Constraints:

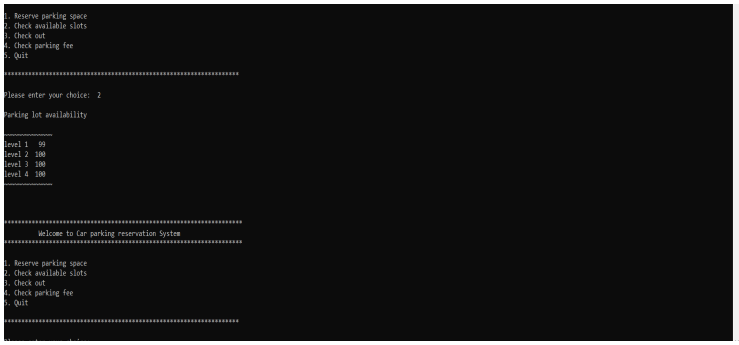
Main choice should be an integer. The program throws an exception and exits if it's not an integer. If the input is not within a valid range the user is given another option to reenter the choice.

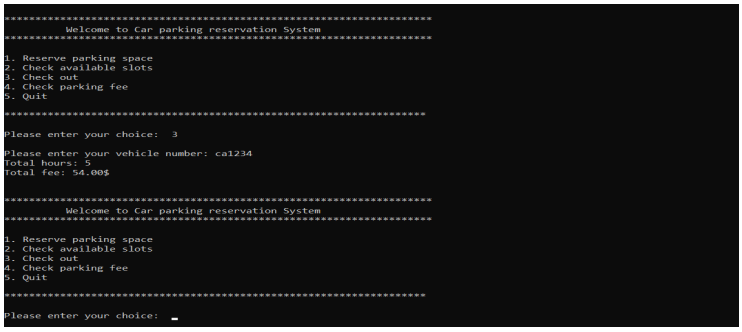
Vehicle number is a string.

Vehicle type is an integer. The program throws an exception and exits if it's not an integer. It also throws an exception if the integer is out of range.

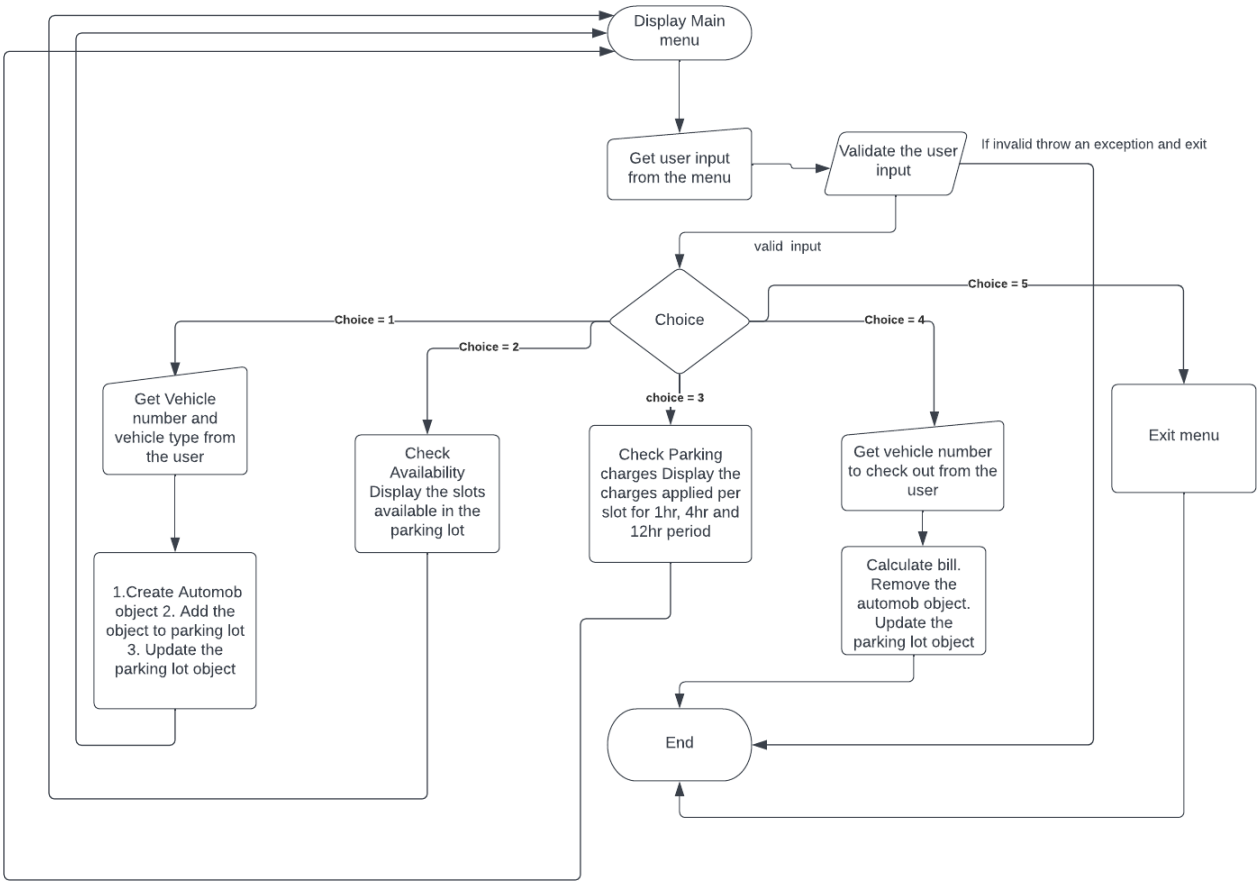
Sample Output:



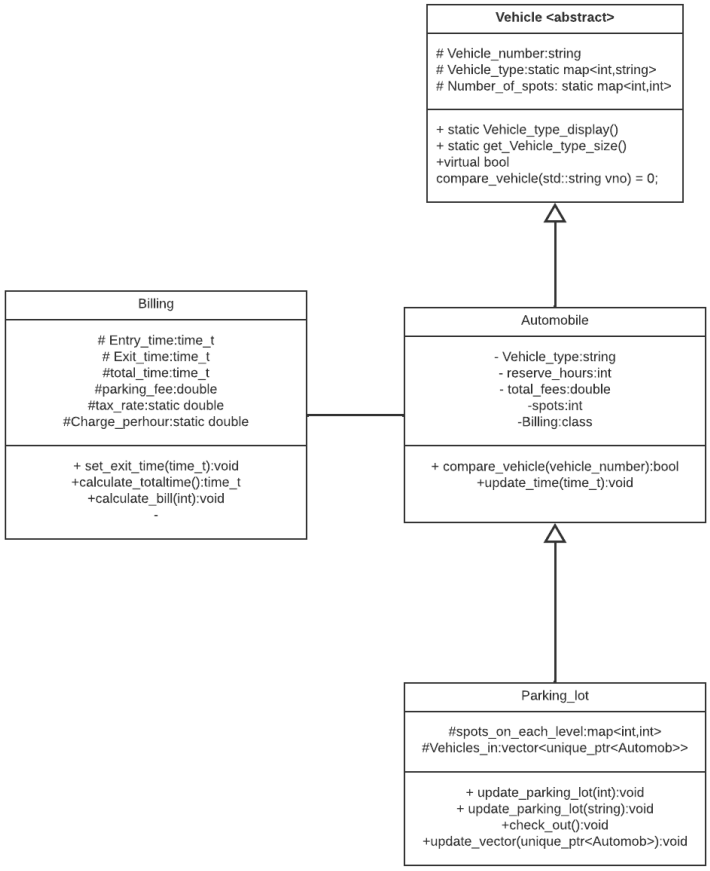




Flow chart:


Display Main menu
Get user input from the menu
Validate the user input
If invalid throw an exception and exit
valid input
Choice
Choice = 1
Choice = 2
choice = 3
Choice = 4
Choice = 5
Get Vehicle number and vehicle type from the user
1.Create Automob object 2. Add the object to parking lot 3. Update the parking lot object
Check Availability Display the slots available in the parking lot
Check Parking charges Display the charges applied per slot for 1hr, 4hr and 12hr period
Get vehicle number to check out from the user
Calculate bill. Remove the automob object. Update the parking lot object
Exit menu
End

Class Diagram:



Extensions:

1. Get input from sensors instead of the user entering the input.
2. Let the customer choose an available spot before parking.
3. Assign spots for specialised vehicles like disabled spot, electric car charging spot, compact car spot and motorcycle parking.

Issues :

The following issues were faced and resolved during the development.

- Error1: First defined here

  Caused by the static variable. The static variable was first declared and defined inside the class header file which caused this error. Defined the static variables in the .cpp file of the class instead of the header file.
- Error2: Out of range error

  Caused when the index of the map variable went out of the size of the map. Fixed it by checking whether the index is less than the size of the map. If the index goes out of range the program throws an exception and exits the program.
- Error3: mismatch of data type

  Caused when the user enters a character instead of integer in the choice given at the main menu. Fixed by reading the data into a stringstream and validating the data before

passing it on to the switch statement. The program throws an exception when the input is not an integer and exits the program.