KLE Society's

KLE Technological University

Open Ended Activity Report

On

Parking Management

Object Oriented Programming with C++ (20ECSC204)

Object Oriented Programming with C++ Lab (20ECSP203)

Submitted By

NAME	ROLLNO	USN
Sourabh Kumar	228	01FE19BCS102
Sahana Kubsad	218	01FE19BCS091
Sneha Sangeet	219	01FE19BCS092
Samarth H	221	01FE19BCS094

TEAM NUMBER: 4B06

Faculty In-charge:

Manjula Pawar

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBLI – 580 031 (India)

Academic year 2020-21

1. Introduction

1.1 Overview of Problem Statement

As more and more individuals and companies expand their ownership of vehicles, the complexities and conflicts of parking swells. Nowadays there is a crucial problem of vehicle parking in malls. . Every parking area needs a system that records the details of vehicles to give the facility.

Vehicle parking management system is a semi-automatic system which delivers data processing at a very high speed in a systematic manner. Our application will track the entry and exit of cars, maintain a listing of cars within the parking lot, and determine if the parking lot is full or not. It will determine the cost of per vehicle according to their time consumption.

1.2 Features of Application

1.2.1 Arrival of a vehicle

When a new vehicle enters the parking lot, the details of the vehicle like the vehicle number, driver name, arrival time and date are noted under the respective type of vehicle (two-wheeler, three-wheeler, four-wheeler).

1.2.2 Parking space

The application tells about the total number of vehicles parked in the parking lot. If it is full and a new vehicle enters then, it throws an exception and displays a message to wait till a vehicle is being departed.

1.2.3 Parking Fees Collection

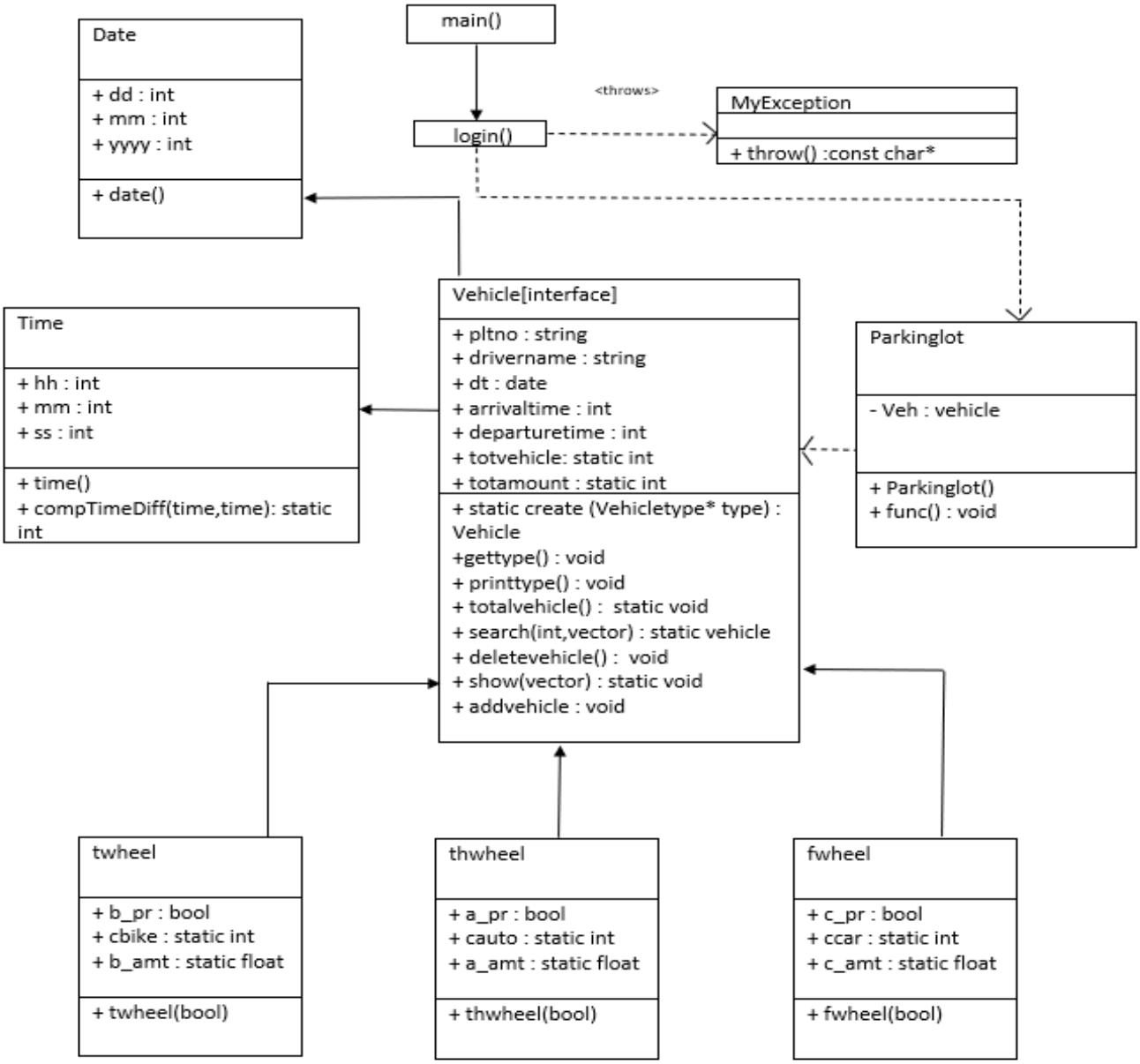
The total amount to be paid by the vehicle owner is calculated by noting the arrival and departure times and by charging a fixed rate on the basis of the time for which it was parked and the vehicle type. The total amount collected by the management is the total sum of all the payments done by the vehicle owners.

1.2.4 Search a Vehicle

The application can also search the vehicle if it’s present in the parking lot on the basis of vehicle number. This saves up a lot of time and manual labor.

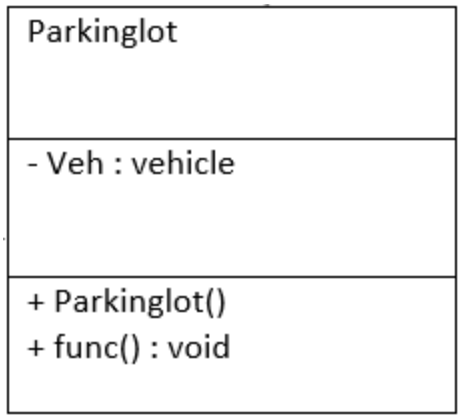
2. Design

2.1 Class Diagram



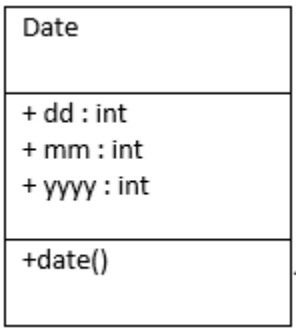
2.2 Description of Each Class

1. Parking lot



- This is a client (factory) class where the "*func ()"* function is called and using a factory design pattern we will create an object for each subclass.

2. Date



- This is a class used to enter the date of arrival and date of departure of the vehicle in the parking lot.
- This is used to store date in the format dd-mm-yyyy, where the data members are date (dd), month (mm) and year (yyyy).

3. Time

Time
+ hh : int + mm : int + ss : int
+time() + compTimeDiff(time,time): static int

- This is the class used to enter time in the format hh:mm: ss.
- There is a function, *"compTimeDiff (time , time)"* in this class which is used to calculate the difference between the departure time and arrival time.

4. twheel, thwheel, fwheel

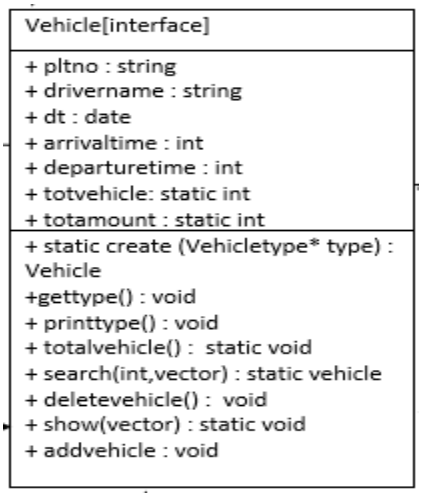
twheel
+ b_pr : bool + cbike : static int + b_amt : static float
+ twheel(bool)

thwheel
+ a_pr : bool + cauto : static int + a_amt : static float
+ thwheel(bool)

fwheel
+ c_pr : bool + ccar : static int + c_amt : static float
+ fwheel(bool)

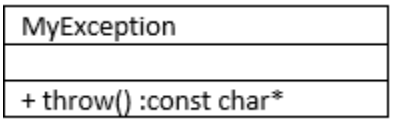
- There are three subclasses: *twheel, thwheel*, and *fwheel.*
- The above subclasses are the classes whose object is to be created.
- Based on which type of vehicle enters the parking lot the object of that subclass will be created.

5. Vehicle



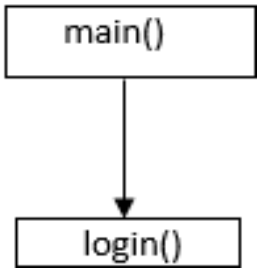
- This class acts as an interface to create an object of type subclass.
- It has 3 sub-classes: *twheel, thwheel & fwheel.*
- An *Exception* is thrown when the parking spots are full or the number of vehicles entering parking lots exceeds parking spots in the parking lot.
- Another *Exception* is thrown when there is an invalid input.

6. MyException



- This is the class which throws an *Exception* when the user enters the wrong password, parking lot is full, search not found.

2.3 Main Function



In the main function, we call the login system.

In the login system, the user is asked to enter the password. If the password matches, then access to further functionalities is given or else throws an exception.

Then the object of parking lot type accesses the *"func ()"* where there is a menu for different options like adding, deleting, searching, total number of vehicles, display the vehicles in the parking lot are provided and then the choice of whether it is two-wheeler, four-wheeler and three-wheeler is entered.

According to the user input we will create a type of Vehicle using factory design pattern. The exceptions are handled using *MyException* class.

2.4 Use of Standard design Patterns

Factory design pattern is used for the above application.

❖ **Factory method pattern (Creational pattern):**

★ Definition: - The Factory Method pattern is a design pattern used to define a runtime interface for creating an object. It's called a factory because it creates various types of objects without necessarily knowing what kind of object it creates or how to create it.

★ Usage: - Factory method is suitable for this scenario because vehicle objects are to be created as per user demand, so to create the objects of the required vehicle type during the run time interface it becomes easier.

➢ Steps:

1. Create a common interface for factory methods. (Vehicle)
2. Create sub classes of different objects to be created. (twheel, thwheel, fwheel)
3. Create client class to use factory methods to create objects. (Parking lot)

3. Unit Test Plan

3.1 Add a Vehicle: void addVehicle ()

The adding vehicle process involves getting the type of vehicle (two-wheeler, three-wheeler, four-wheeler) and then getting details like vehicle number, arrival date, arrival time, driver name and all the details are stored further.

Given below is how a vehicle get added:

First, you need to login.

```
Parking Reservation System Login

Enter Password: ****

Access Granted! Welcome To Our System

Press any key to continue . . .
```

You need to choose the type of vehicle.

```
Add :
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
```

Then the vehicle details are asked.

Test case 1: Adding two-wheeler

Input:

Add :

Choose the type of vehicle.

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

1

Enter vehicle number: JH11A1234

Enter arrival time in hours minutes and seconds : 15

14

10

Enter date in day month and year : 04

05

2021

Enter driver name: Sourabh

Expected Output:

Vehicle added successfully

Do you want to continue, press y/n:

Y

Actual Output:

```
Add :
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : JH11A1234
Enter arrival time in hours minutes and seconds : 12
12
12
Enter date in day month and year: 04
05
2021
Enter Driver Name : Sourabh

Vehicle added successfully

Do you want to continue, press y/n :
```

Test case 2: Adding three-wheeler

Input:

Add:

Choose the type of vehicle.

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

2

Enter vehicle number: JH12A0001

Enter arrival time in hours minutes and seconds: 14

10

11

Enter date in day month and year: 04

05

2021

Enter driver name: Bodhi

Expected Output:

Vehicle added successfully

Do you want to continue, press y/n:

Y

Actual Output:

```
Add :
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
2
Enter vehicle number : JH12A0001
Enter arrival time in hours minutes and seconds : 14
10
11
Enter date in day month and year: 04
05
2021
Enter Driver Name : Bodhi

Vehicle added successfully

Do you want to continue, press y/n :
```

Test case 3: Adding four wheeler

Input:

Add :

Choose the type of vehicle.

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

3

Enter vehicle number: KA11B0008

Enter arrival time in hours minutes and seconds: 15

14

10

Enter date in day month and year: 04

05

2021

Enter driver name: Kshitij

Expected Output:

Vehicle added successfully

Do you want to continue, press y/n:

Y

Actual Output:

```
Add :
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
3
Enter vehicle number : KA11B0008
Enter arrival time in hours minutes and seconds : 15
14
10
Enter date in day month and year: 04
05
2021
Enter Driver Name : Kshitij

Vehicle added successfully

Do you want to continue, press y/n :
```

Test case 4: Parking space full exception

Input:

****
VEHICLE PARKING RESERVATION SYSTEM
1. Arrival of a vehicle
2. Total number of vehicles parked
3. Departure of vehicle
4. Total Amount collected
5. Display
6. Search a Vehicle
7. Exit
****
Enter your Choice : 1

Expected Output:

Add :
Parking Space is Full Kindly wait
Do you want to continue, press y/n :
y

Actual Output:

```
Add :
Parking Space is Full Kindly wait
Do you want to continue, press y/n :
y
```

3.2 Depart a vehicle:void deleteVehicle(vector<vehicle> veh)

Departing a vehicle process involves taking type of vehicle as input and then taking vehicle number, if found then takes departure time and calculates the parking fee.

Given below is how a vehicle get added :

First, you need to login.

```
                    Parking Reservation System Login

                          Enter Password: ****

                    Access Granted! Welcome To Our System
Press any key to continue . . .
```

You need to choose the type of vehicle.

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
```

Then the vehicle details are asked.

Test case 5 : Departing a two wheeler

Input:

Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : JH11A1234
Departure :
Enter departure time in hours minutes and seconds : 18
12
14

Expected Output:

Vehicle having vehicle number: JH11A1234 has left the parking after paying Rs. 20
Do you want to continue, press y/n :
y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : JH11A1234
Departure :
Enter departure time in hours minutes and seconds : 18
12
14

Vehicle having vehicle number  : JH11A1234 has left the parking after paying Rs. 20

Do you want to continue, press y/n :
y
```

Test case 6: Departing a three wheeler

Input:

Choose the type of vehicle

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

2

Enter vehicle number: JH12A0001

Departure:

Enter departure time in hours minutes and seconds: 20

11

15

Expected Output:

Vehicle having vehicle number: JH12A0001 has left the parking after paying Rs. 30

Do you want to continue, press y/n:

y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
2
Enter vehicle number : JH12A0001
Departure :
Enter departure time in hours minutes and seconds : 20
11
15

Vehicle having vehicle number  : JH12A0001 has left the parking after paying Rs. 30

Do you want to continue, press y/n :
y
```

Test case 7: Departing a four wheeler

Input:

Choose the type of vehicle

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

3

Enter vehicle number: KA11B0008

Departure:

Enter departure time in hours minutes and seconds: 21

19

53

Expected Output:

Vehicle having vehicle number: KA11B0008 has left the parking after paying Rs. 50

Do you want to continue, press y/n:

y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
3
Enter vehicle number : KA11B0008
Departure :
Enter departure time in hours minutes and seconds : 21
19
53

Vehicle having vehicle number  : KA11B0008 has left the parking after paying Rs. 50

Do you want to continue, press y/n :
y
```

Test case 8: Vehicle not in parking lot exception

Input:

Choose the type of vehicle

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

3

Enter vehicle number: KA11B0008

Expected Output:

Vehicle Not Found

Do you want to continue, press y/n :

y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : KA11B008
Vehicle Not Found
Do you want to continue, press y/n :
y
```

Test case 9: Parking space empty exception

Input:

VEHICLE PARKING RESERVATION SYSTEM
1. Arrival of a vehicle
2. Total number of vehicles parked
3. Departure of vehicle
4. Total Amount collected
5. Display
6. Search a Vehicle
7. Exit
****
Enter your Choice: 3

Expected Output:

No Vehicle in Parking...

Do you want to continue, press y/n:

y

Actual Output:

```
No Vehicle in Parking...

Do you want to continue, press y/n :
y
```

3.3 Search a vehicle: static vehicle searc(int a,vector<vehicle> v)

Searching a vehicle process involves choosing the type of vehicle and then giving the vehicle number as input. If found, details will be shown. If not the not found exception will be thrown.

Given below is how a vehicle is searched:

First, you need to login.

```
Parking Reservation System Login

Enter Password: ****

Access Granted! Welcome To Our System
Press any key to continue . . .
```

You need to choose the type of vehicle.

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
```

Then the vehicle number is being given as input.

Test case 10: Searching a two wheeler

Input:

Choose the type of vehicle

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

1

Enter vehicle number : JH11A1234

Expected Output:

Vehicle Found

Vehicle Type	Vehicle Number	Driver Name	Date	Arrival Time
Bike	JH11A1234	Sourabh	4/5/2021	14:14:10

Do you want to continue, press y/n :

y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : JH11A1234

Vehicle Found
Vehicle Type            Vehicle Number                  Driver Name                     Date                    Arrival Time
Bike                    JH11A1234                               Sourabh                 4/5/2021                        14:14:10

Do you want to continue, press y/n :
y
```

Test case 11: Searching a three wheeler

Input:

Choose the type of vehicle

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

2

Enter vehicle number : JH12A0001

Expected Output:

Vehicle Found

Vehicle Type	Vehicle Number	Driver Name	Date	Arrival Time
Auto	JH12A0001	Bodhi	4/5/2021	14:10:11

Do you want to continue, press y/n:

y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
2
Enter vehicle number : JH12A0001

Vehicle Found
Vehicle Type            Vehicle Number                  Driver Name                     Date                    Arrival Time
Auto                    JH12A0001                               Bodhi                   4/5/2021                        14:10:11

Do you want to continue, press y/n :
y
```

Test case 12: Searching a four wheeler

Input:

Choose the type of vehicle

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

3

Enter vehicle number: KA11B0008

Expected Output:

Vehicle Found

Vehicle Type	Vehicle Number	Driver Name	Date	Arrival Time
Car	KA11B0008	Kshitij	4/5/2021	15:14:10

Do you want to continue, press y/n :

y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
3
Enter vehicle number : KA11B0008

Vehicle Found
Vehicle Type        Vehicle Number        Driver Name        Date        Arrival Time
Car                 KA11B0008                  Kshitij       4/5/2021            15:14:10

Do you want to continue, press y/n :
y
```

Test case 13: Vehicle not found exception

Input:

Choose the type of vehicle

1. Two wheeler.

2. Three wheeler.

3. Four wheeler.

1

Enter vehicle number: KA11B008

Expected Output:

Vehicle Not Found

Do you want to continue, press y/n :

y

Actual Output:

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : KA11B008
Vehicle Not Found
Do you want to continue, press y/n :
y
```

3.4 Parking fees collection: void totalamount()

This method displays parking fees collected till now with collection of all types of vehicle too.

Given below is how parking fees collected is displayed:

First, a vehicle departs, on the basis of difference between arrival time and departure time parking fees is calculated, stored and displayed.

Test case 14: Parking fees collection is displayed

Input:

VEHICLE PARKING RESERVATION SYSTEM
1. Arrival of a vehicle
2. Total number of vehicles parked
3. Departure of vehicle
4. Total Amount collected
5. Display
6. Search a Vehicle
7. Exit

****
Enter your Choice: 4

Expected Output:

Total Parking Charges Collection till now: Rs. 100

Total Parking Charges Collection of Bike parked: Rs.20

Total Parking Charges Collection of Three wheeler parked: Rs.30

Total Parking Charges Collection of Car parked: Rs.50

Do you want to continue, press y/n:

Y

Actual Output:

```
Total Parking Charges Collection till now : Rs. 100
Total Parking Charges Collection of Bike parked : Rs.20
Total Parking Charges Collection of Three wheeler parked : Rs.30
Total Parking Charges Collection of Car parked : Rs.50

Do you want to continue, press y/n :
y
```

3.5 Parking space detail: static void totalveh()

This method displays how many parking lots are filled with which type of vehicle and how many lots are available.

Given below is parking space display:

First, a vehicle is added it fills the parking space, if a vehicle gets departed available parking space is updated.

Test case 15: Parking space display

Input:

VEHICLE PARKING RESERVATION SYSTEM
1. Arrival of a vehicle
2. Total number of vehicles parked
3. Departure of vehicle
4. Total Amount collected
5. Display
6. Search a Vehicle
7. Exit
****
Enter your Choice: 2

Expected Output:

Total number of vehicle parked : 3

Total number of Bike parked : 1

Total number of Three wheeler parked :1

Total number of Car parked : 1

Total number of Available Space : 0

Do you want to continue, press y/n :

y

Actual Output:

```
Total number of vehicle parked : 3
Total number of Bike parked : 1
Total number of Three wheeler parked :1
Total number of Car parked : 1
Total number of Available Space : 0

Do you want to continue, press y/n :
y
```

4. Implementation

4.1 Results

1. Login Screen

```
Parking Reservation System Login

Enter Password: ****

Access Granted! Welcome To Our System
Press any key to continue . . .
```

2. Login Screen exception

```
Parking Reservation System Login

Enter Password: *****

Wrong Password
Press any key to continue . . .
```

3. Main Menu

```
*****************************************************************
                 VEHICLE PARKING RESERVATION SYSTEM
1. Arrival of a vehicle
2. Total number of vehicles parked
3. Departure of vehicle
4. Total Amount collected
5. Display
6. Search a Vehicle
7. Exit
*****************************************************************
Enter your Choice :
```

4. Adding a vehicle

```
Add :
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : JH11A1234
Enter arrival time in hours minutes and seconds : 12
12
12
Enter date in day month and year: 04
05
2021
Enter Driver Name : Sourabh

Vehicle added successfully

Do you want to continue, press y/n :
```

5. Parking space full exception

```
Add :
Parking Space is Full Kindly wait
Do you want to continue, press y/n :
y
```

6. Departing a vehicle

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : JH11A1234
Departure :
Enter departure time in hours minutes and seconds : 18
12
14

Vehicle having vehicle number  : JH11A1234 has left the parking after paying Rs. 20

Do you want to continue, press y/n :
y
```

7. Search a vehicle

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : JH11A1234

Vehicle Found
Vehicle Type            Vehicle Number                  Driver Name                     Date                    Arrival Time
Bike                    JH11A1234                               Sourabh                 4/5/2021                        14:14:10

Do you want to continue, press y/n :
y
```

8. Search vehicle exception

```
Choose the type of vehicle
1. Two wheeler.
2. Three wheeler.
3. Four wheeler.
1
Enter vehicle number : KA11B008
Vehicle Not Found
Do you want to continue, press y/n :
y
```

9. Display details of all vehicles parked

```
Vehicle Type        Vehicle Number          Driver Name             Date                Arrival Time
Bike                KA11B0001                   Sourabh                 4/5/2021
10:10:10
Auto                KA12B0002                   Bodhi                   4/5/2021
11:11:11
Car                 KA12B0003                   Kshitij                 4/5/2021
12:11:10

Do you want to continue, press y/n :
y
```

10. Display available parking space

```
Total number of vehicle parked : 3
Total number of Bike parked : 1
Total number of Three wheeler parked :1
Total number of Car parked : 1
Total number of Available Space : 0

Do you want to continue, press y/n :
y
```

11. Display parking fees collection

```
Total Parking Charges Collection till now : Rs. 100
Total Parking Charges Collection of Bike parked : Rs.20
Total Parking Charges Collection of Three wheeler parked : Rs.30
Total Parking Charges Collection of Car parked : Rs.50

Do you want to continue, press y/n :
y
```