



**Dharmsinh Desai University, Nadiad**

**Faculty of Technology**

**Department of Computer Engineering**

**B. Tech. CE Semester – IV**

**Subject: Software Engineering Practice**

**Project title: Online Car Rental System,**

**By:**

**1) Harsh Thumar,**

**Roll no: CE134,**

**Id: 19CEUOG046**

**2) Shail Shah,**

**Roll no: CE123,**

**Id: 19CEUON088**

# **Contents:**

1. Introduction.....	2
2. Software Requirement Specifications.....	4
3. Design	
I) Use Case diagram .....	10
II) Class diagram.....	13
III) Sequence diagram .....	14
IV) Activity diagram .....	16
V) Data Flow diagram .....	19
VI) Structure Chart .....	23
4. Implementation Detail	
I) Modules.....	24
II) Major Functionality .....	25
5. Work-Flow/Layout .....	28
6. Conclusion .....	34
7. Limitation and Future extension.....	35
8. Bibliography .....	36

## **Introduction**

- It is web based system which allows the customer to register and reserve a car online through the website.
- Online Car Rental is a system that can be used for renting cars for a temporary amount of time with some fees. The person who want to Rent a car need to be Register online through the website. The rental cars are categorized by economy ,premium and luxury. The customer has to firstly register himself/herself if he comes for the first time.
- The main objective of the application car Rental System require a temporary vehicle, for example those who do not own their own car, or have a damaged vehicles who are awaiting repair or travelers who are out of town.
- The customer can search for his/her requirements for a particular car and its availability also the availability of the driver and its details. The customers can also send us the feedback of their experience. The website also shows that the customers can contact our branches and locations.
- Meanwhile the admin user can login and add a new vehicle with its full details in the database, the admin can also manage the booking details ,can manage all registered details, and also can manage the driver details ,and finally the branch details, and etc.

### **Technologies/tools used :**

- Platform used: Visual Studio 2019, Pycharm  
Technology: Django Framework
- Platform used: MySQL (Database)

# **Software Requirement Specifications:**

## **Online Car Rental System**

### **THE USERS OF THE ONLINE CAR RENTAL SYSTEM ARE:**

#### ➤ **ADMIN:**

The admin is the superuser of the website who manages and maintains the system. The admin has the power to update the car database, manage bookings of the rental and check and reply to the feedback of customer. Also he has the power to update its employees details.

#### ➤ **Employee:**

All employee of the company can manage the customer's order i.e they can keep track of customer's booking details, trip, payments etc.

#### ➤ **CUSTOMER:**

The customer can look up to details of the car rental. Look for the car availability, register the car, look for the driver availability, raise the queries.

# **1. MANAGE USER/CUSTOMER**

## **R.1.1 Register customer:**

Description: the details of the new customer are registered such as name, age, phone number, address, email-id etc. and a unique customer id is generated.

**Input:** customer details.

**Output:** unique customer id.

### **R.1.1.1 CONFIRM REGISTRATION:**

Description: if the customer confirms that he/her is authentic then the account is added in the database.

**Input:** customer confirms for himself/herself.

**Output:** home page of the website.

## **R.1.2 Account update:**

Description: the details of the registered user is updated.

**Input:** the register user details (e.g address change name, city change name, etc) to be changed.

**Output:** a message on the screen after the updation.

## **R.1.3 Delete account:**

Description: it deletes the account of the customer as per their request.

**Input:** customer name/customer id and Password.

**Output:** conformation message.

## **2.MANAGE EMPLOYEE :**

### **R.2.1 Add Employee:**

Description: a employee must be add his/her details such as name,address,phone number, email address, etc.

**Input:** employee details

**Output:** conformation message.

### **R.2.2 Delete Employee:**

Description: it deletes the employee details,

**Input:** Employee name

**Output:**conformation message.

### **R.2.3 Update Employee Details:**

Description: it updates the the employee information/data.

**Input:** employee name

**Output:**conformation message.

#### **R.2.3.1 Display Employee Details:**

Description: it displays the all the details of the employee.

**Input:** employee name

**Output:** employee details.

## **3 Car Details Module:**

### **R.3.1 Add Cars:**

Description: a car must be registered in the database of the system including its details for the rental of it.

**Input:** car details.

**Output:** conformation message.

### **R.3.2 Delete Cars:**

Description: it deletes the car details from the database

**Input:** car name

**Output:** conformation message.

### **R.3.3 Car Details:**

Description: it shows all the details of the cars with their insurance.

**Input:**employee input

**Output:**car details.

## **4 BOOKING DETAILS:**

### **R.4.1 Selection of car:**

Description: The car specified is selected.

**Input:** the requirements of the user.

**Output:** display car information.

### **R.4.1.1Availability of car:**

Description: the specified cars is checked were it is available or not.

**Input:** user requirement.

**Output:**output screen.

### **R.4.2 Driver booking:**

Description: the driver availability is .

**Input:** user requirement.

**Output:** details of the driver.

### **R.4.3 Booking:**

Description:The user selected car is selected and confirmed by the customer.

**Input:** user selection.

**Output:** conformation message.

#### **R.4.3.1 Review Booking:**

Description: The selected booking is reviewed by the customer.

**Input:** user selection.

**Output:** Booking details.

#### **R.4.3.2 Reset Booking:**

Description: the booking done can be reseted by the customer.

**Input:** reset selection.

**Output:** confirmation message.

#### **R.4.4 Location details:**

Description: the travelling route can be selected by the user.

**Input:** location of pick up and drop.

**Output:** route from the pick up and drop location.

#### **R.4.5 Booking Cancellation**

Description: this is help for cancel the booking of customer.

**Input:** user selection on that booked car.

**Output:** conformation message.

#### **R.4.6 Booking history:**

Description: the previous rentals can be viewed in the history section.

**Input:** user selection.

**Output:** booking history.

#### **R.4.7 Booking Payment:**

Description: customer can pay their rent of car

**Input:** Payment Details.

**Output:** conformation message



## **5. Manage statistics**

### **R.5.1 Display car count**

Description: count the total no of cars which are in garriage and which are in currently on rent

**Input :** employee selection

**Output:** count of cars which are on rent and which are in garriage.

### **R.5.2 Display number of transactions**

Description: the number of user in last 1 year per car

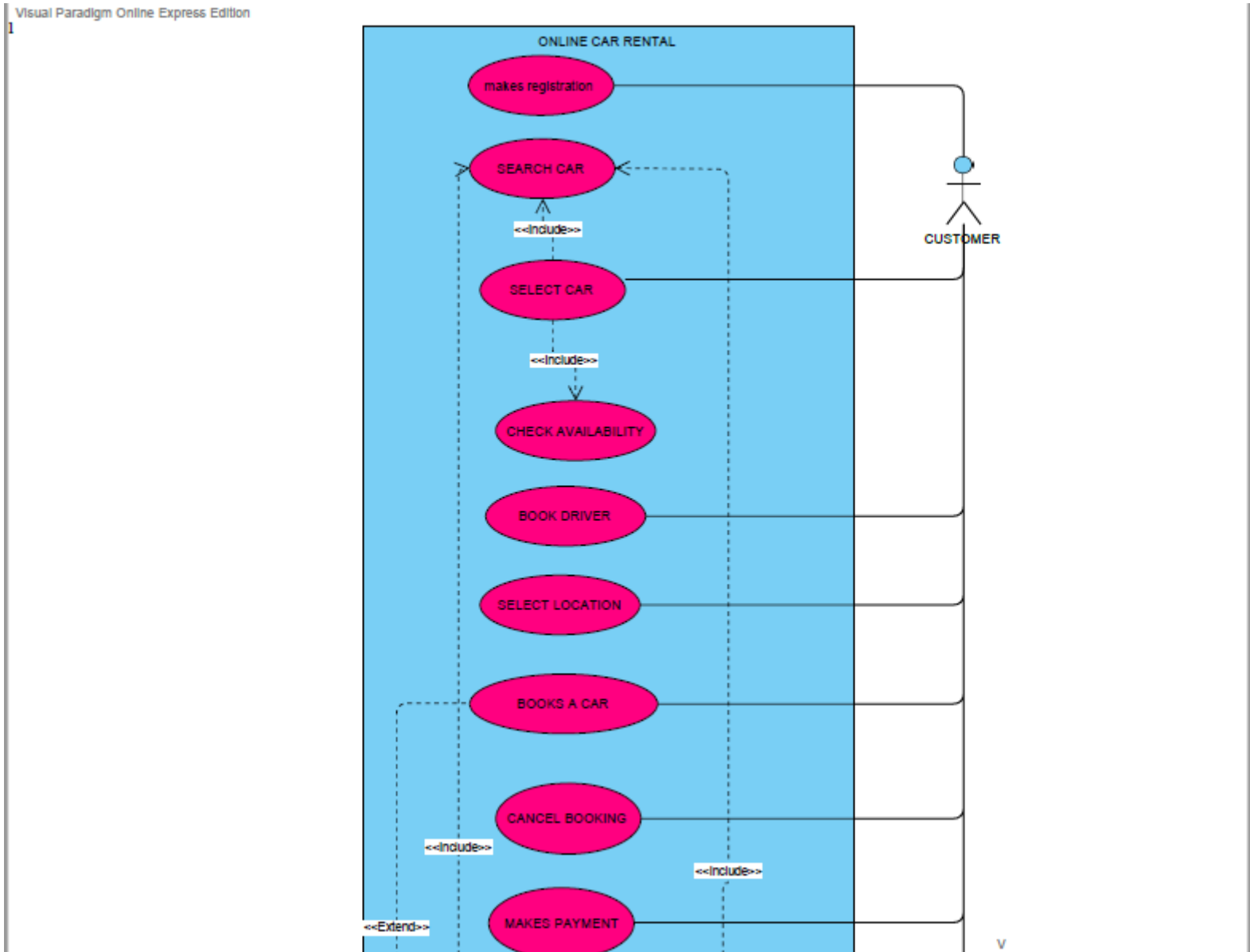
Will be display

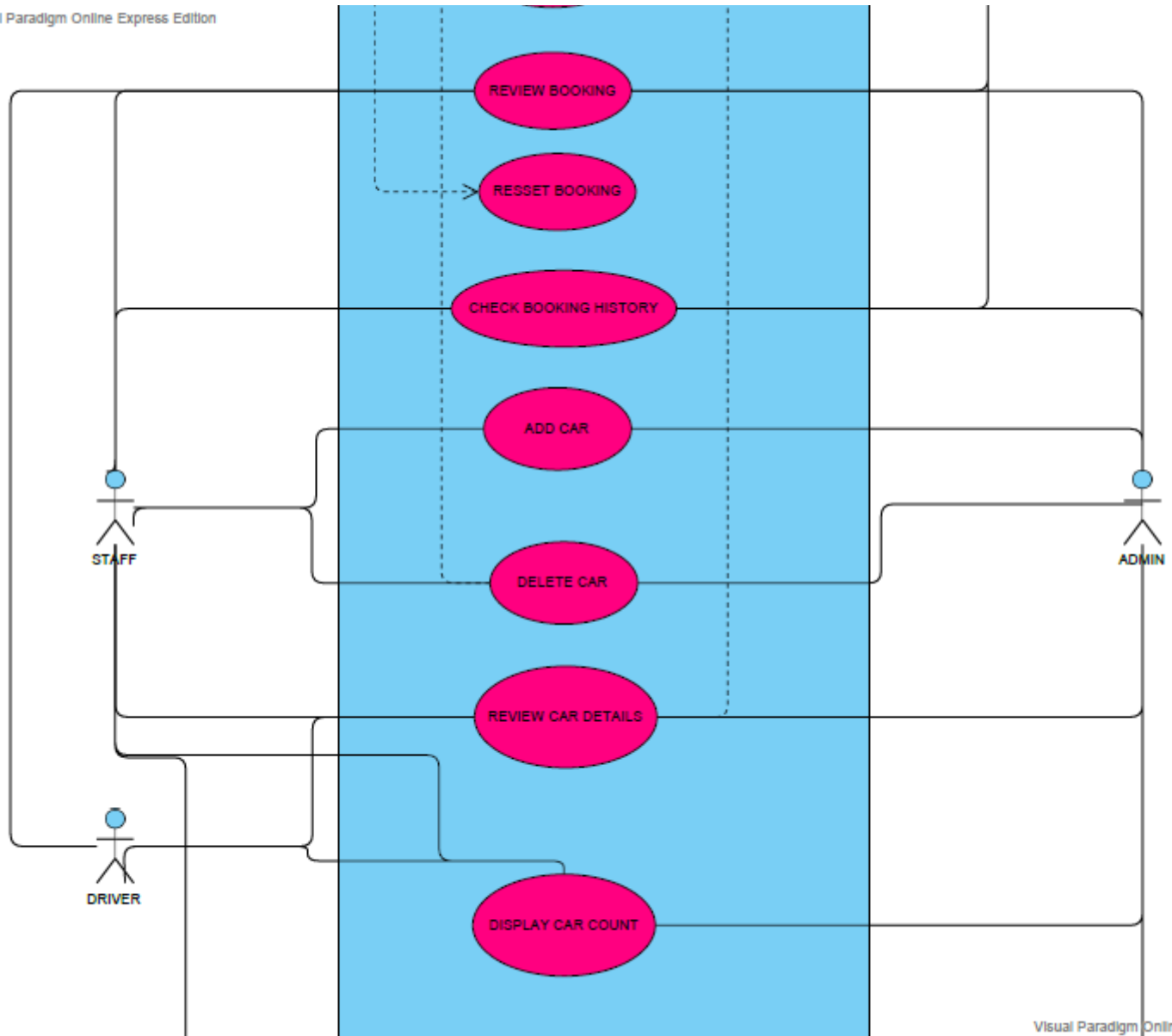
**Input :** employee selection

**Output :** display the car name with number of user

# Design:

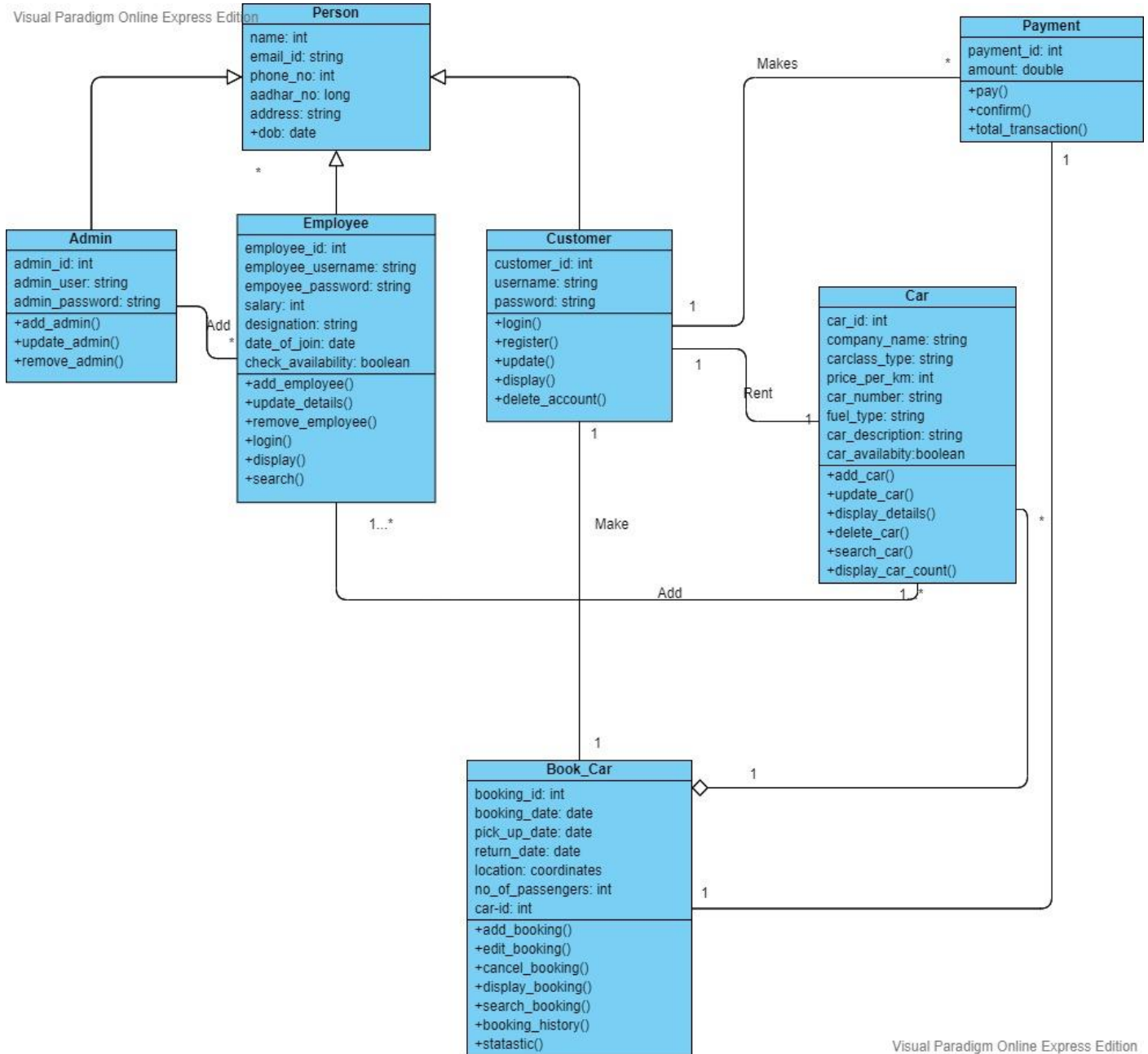
- Use Case diagram





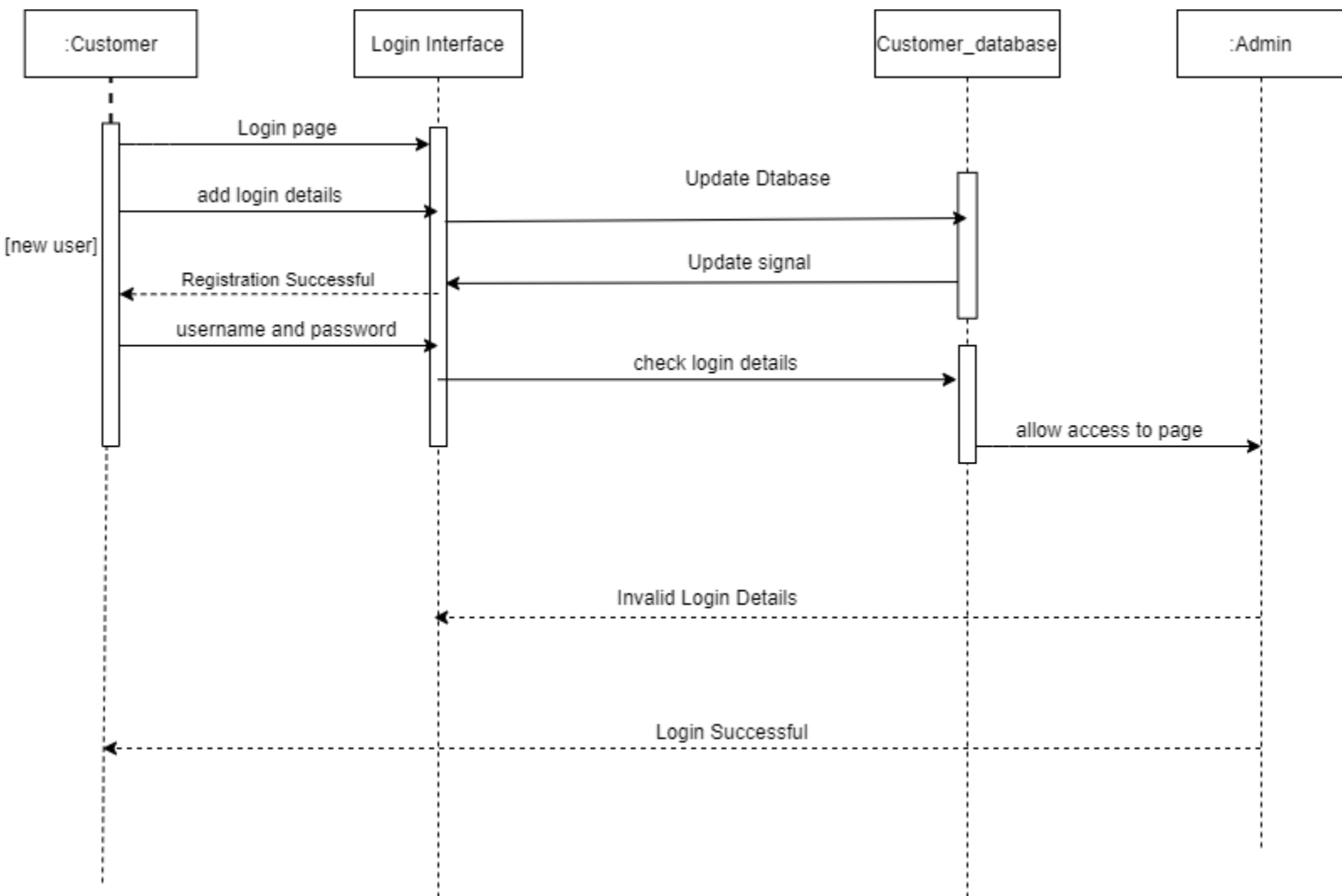


- Class Diagram



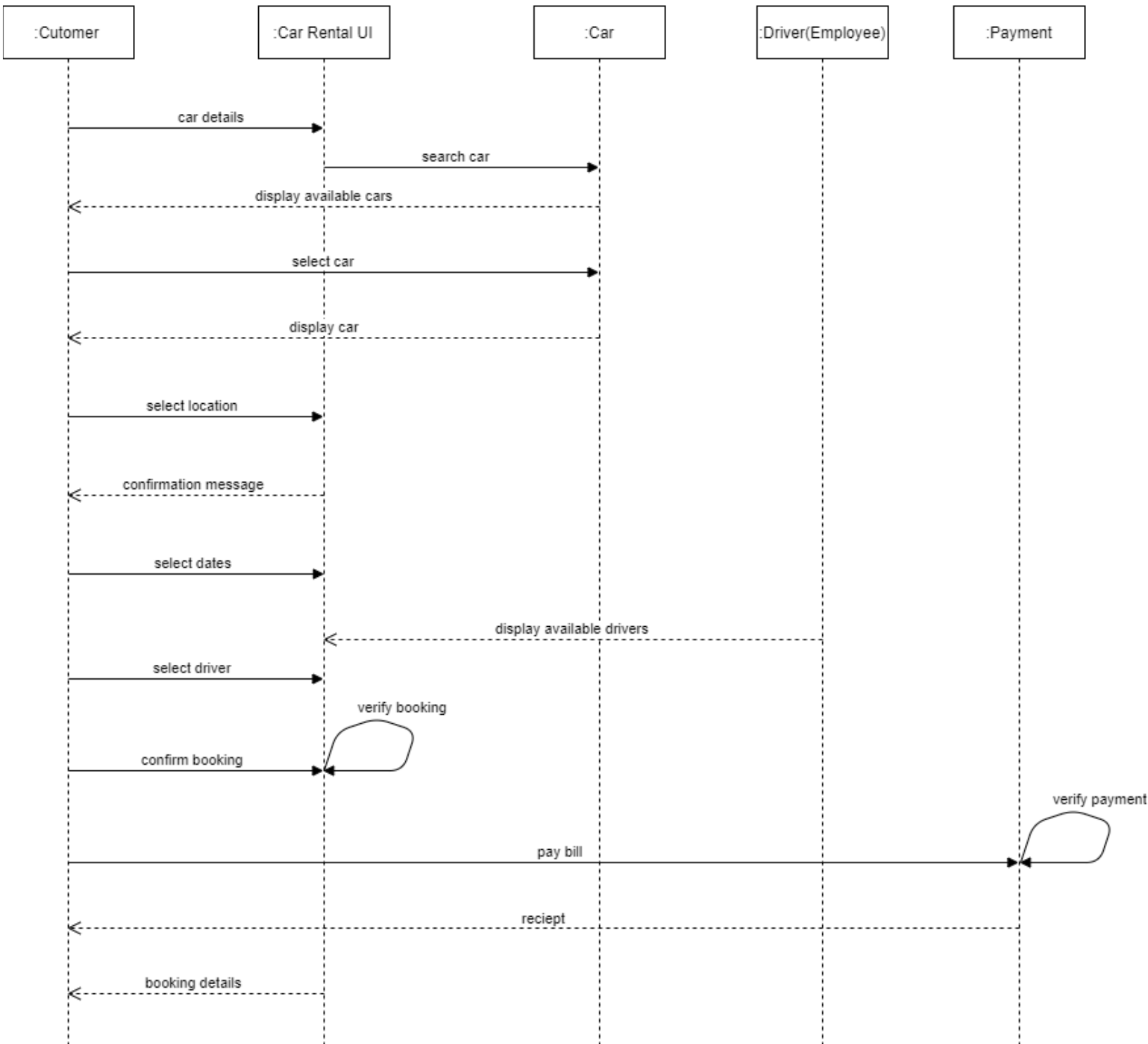
- Sequence Diagram

- sequence diagram of customer registration/login.



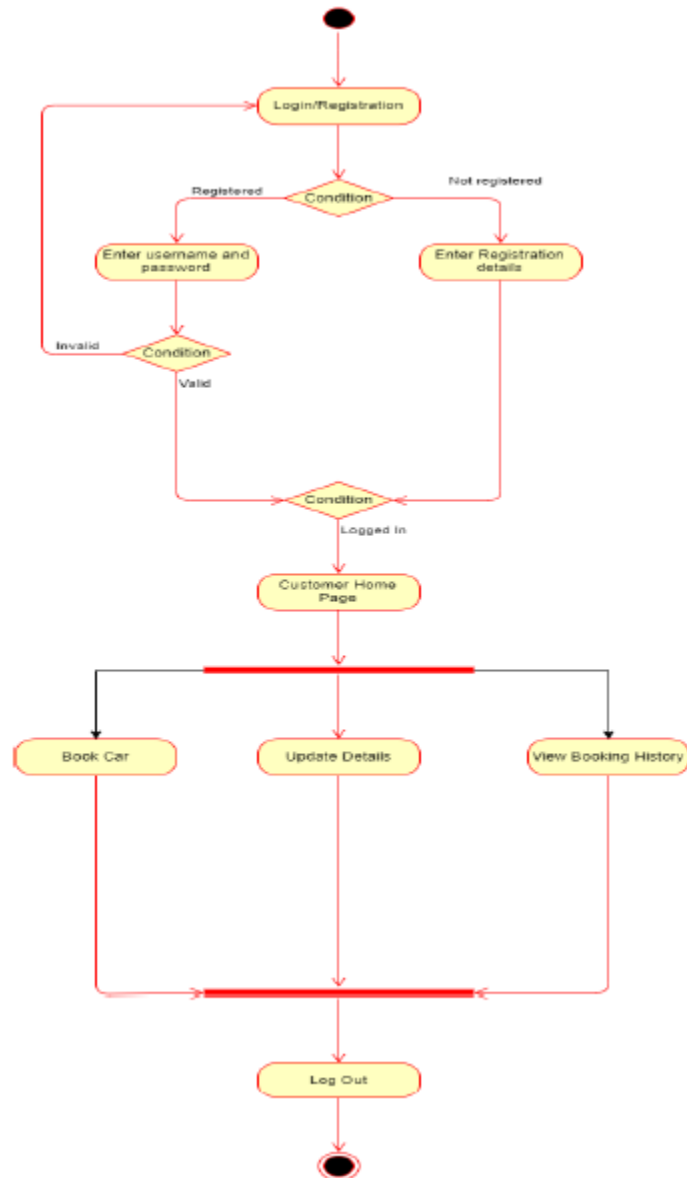
- sequence diagram of Book a car.

Book a car



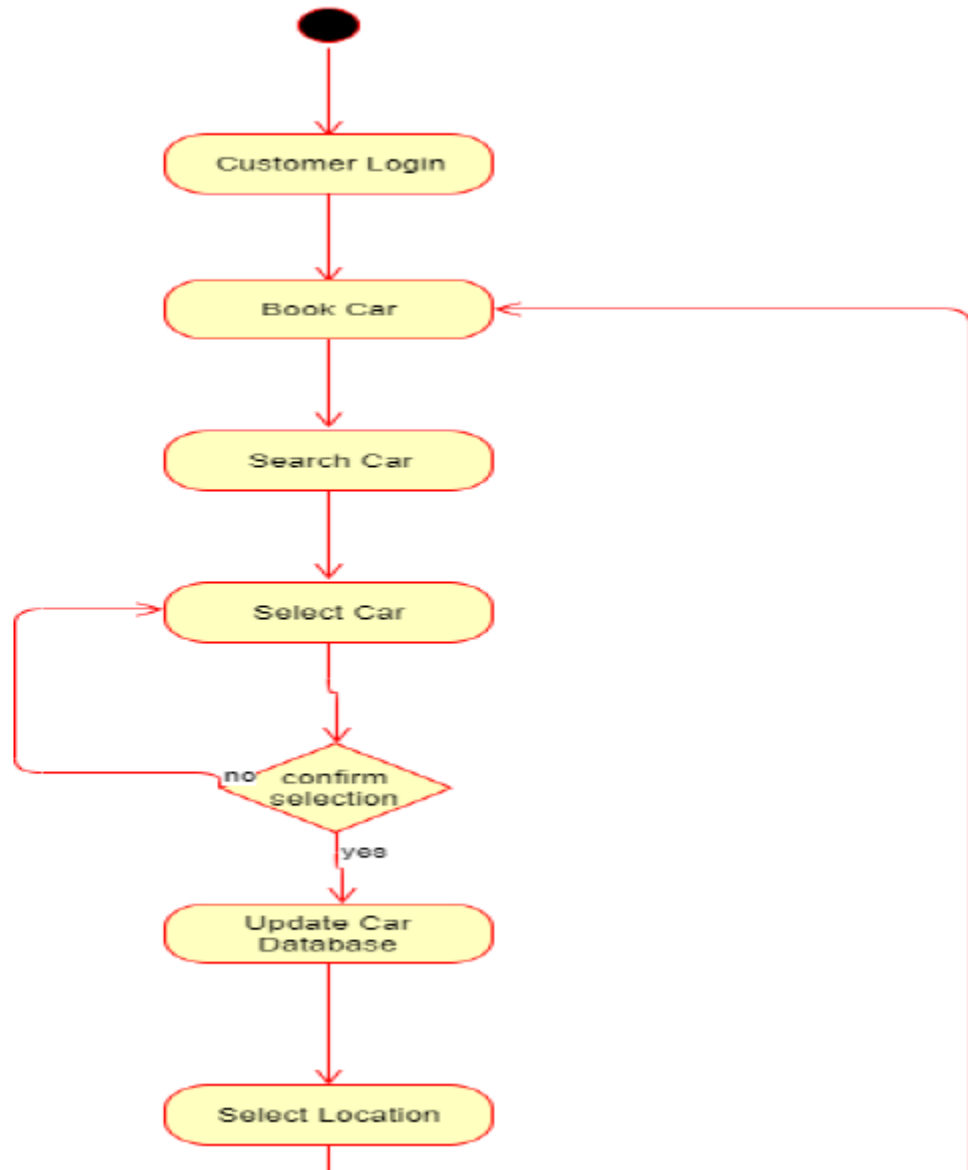
- Activity Diagram

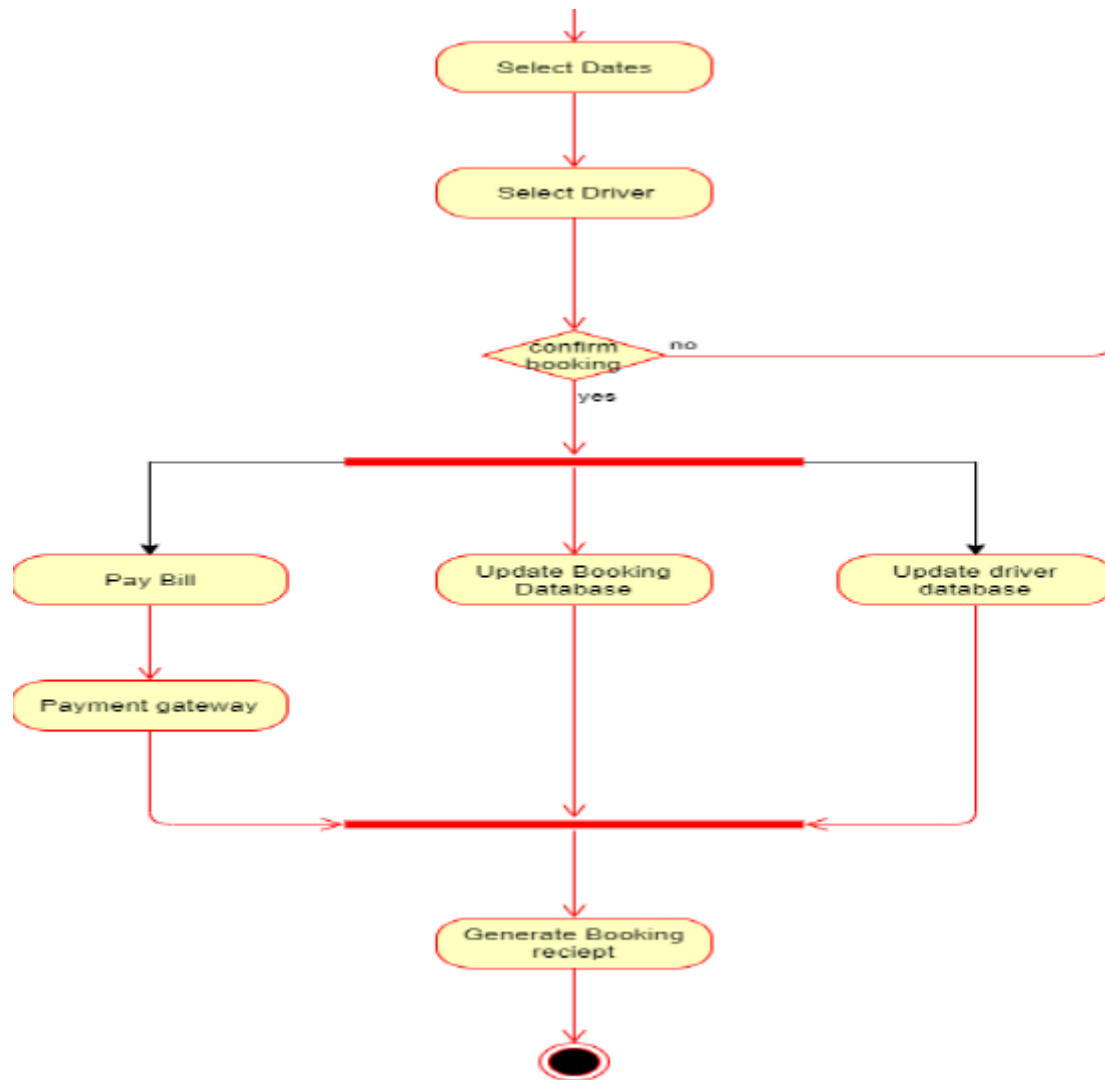
Activity diagram for customer login/registration:





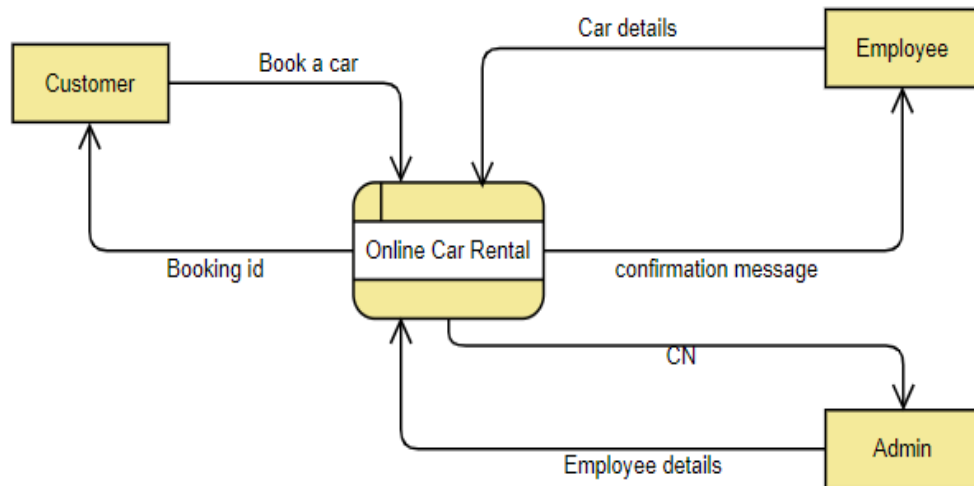
- Activity diagram for book a car:





- Data Flow Diagram

### Contex Diagram

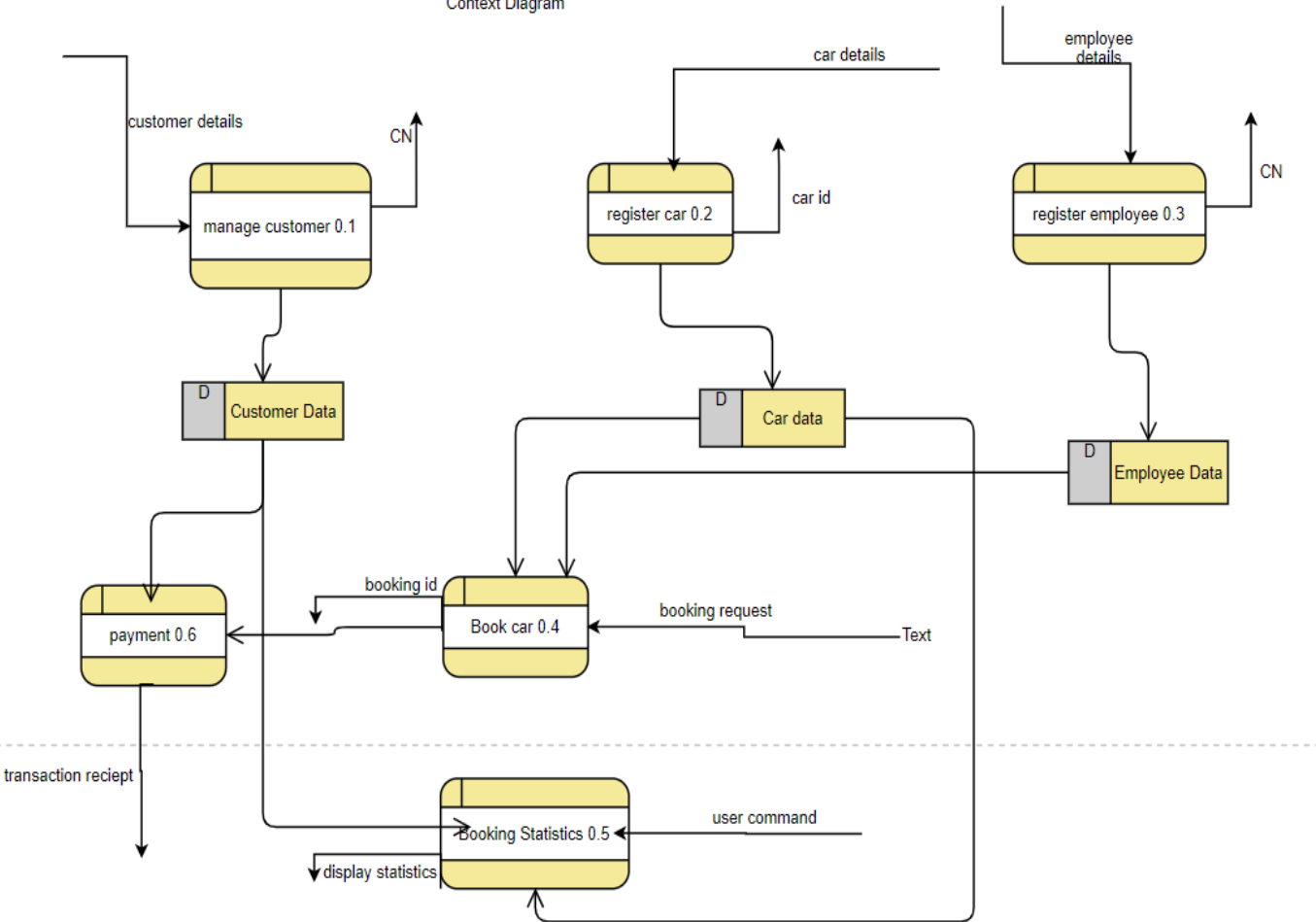


Context Diagram

I

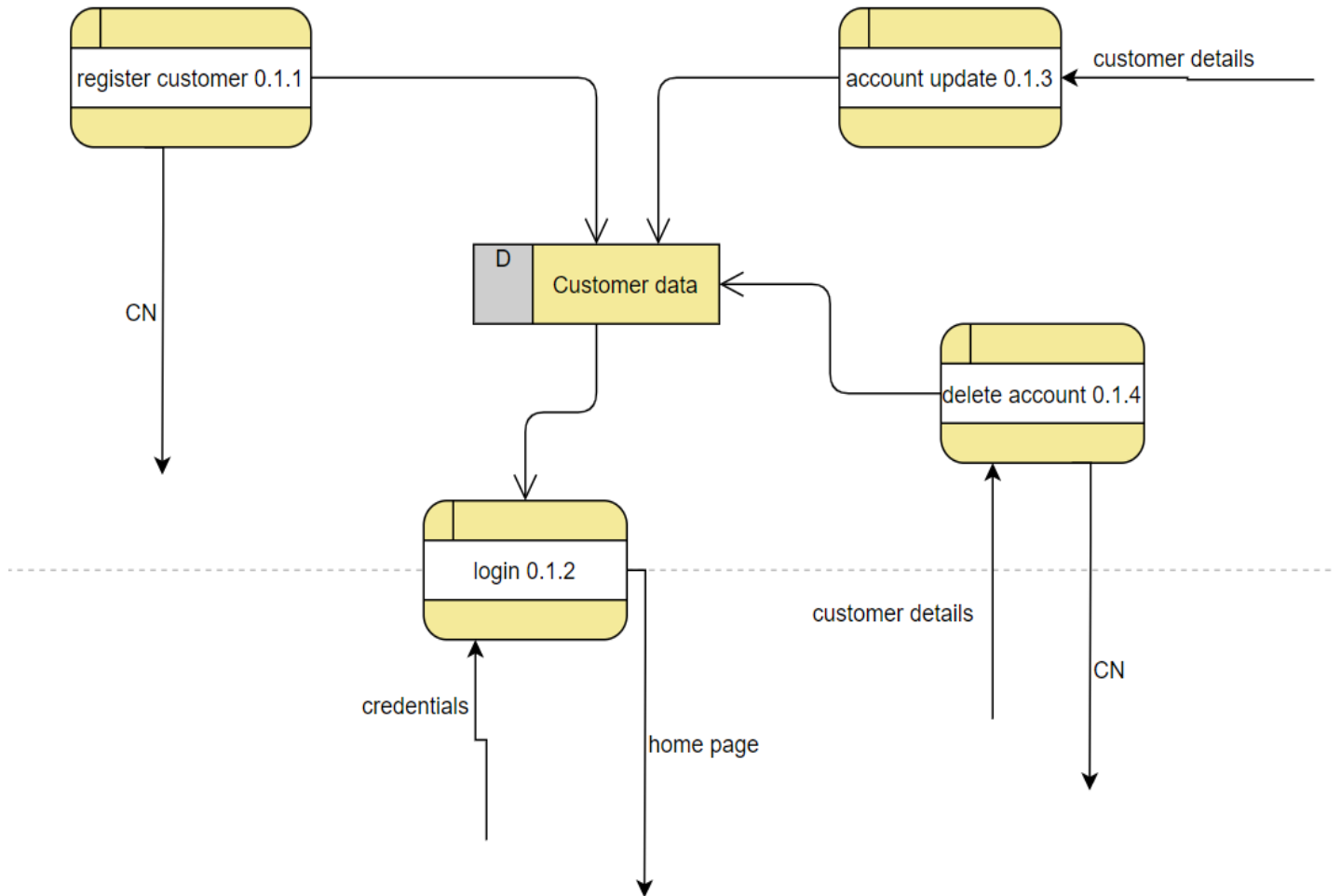
# Level 1

Context Diagram

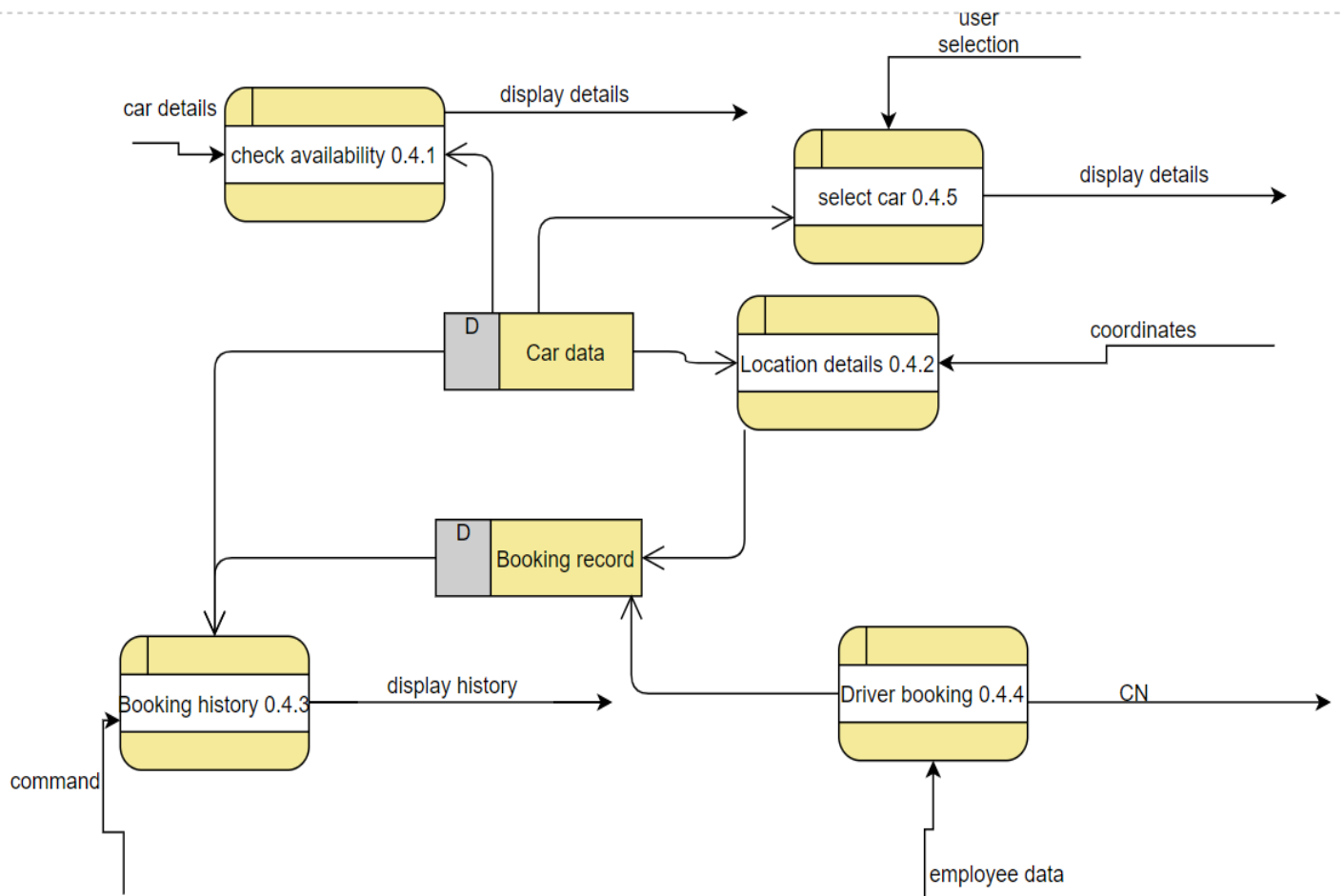


Level 1

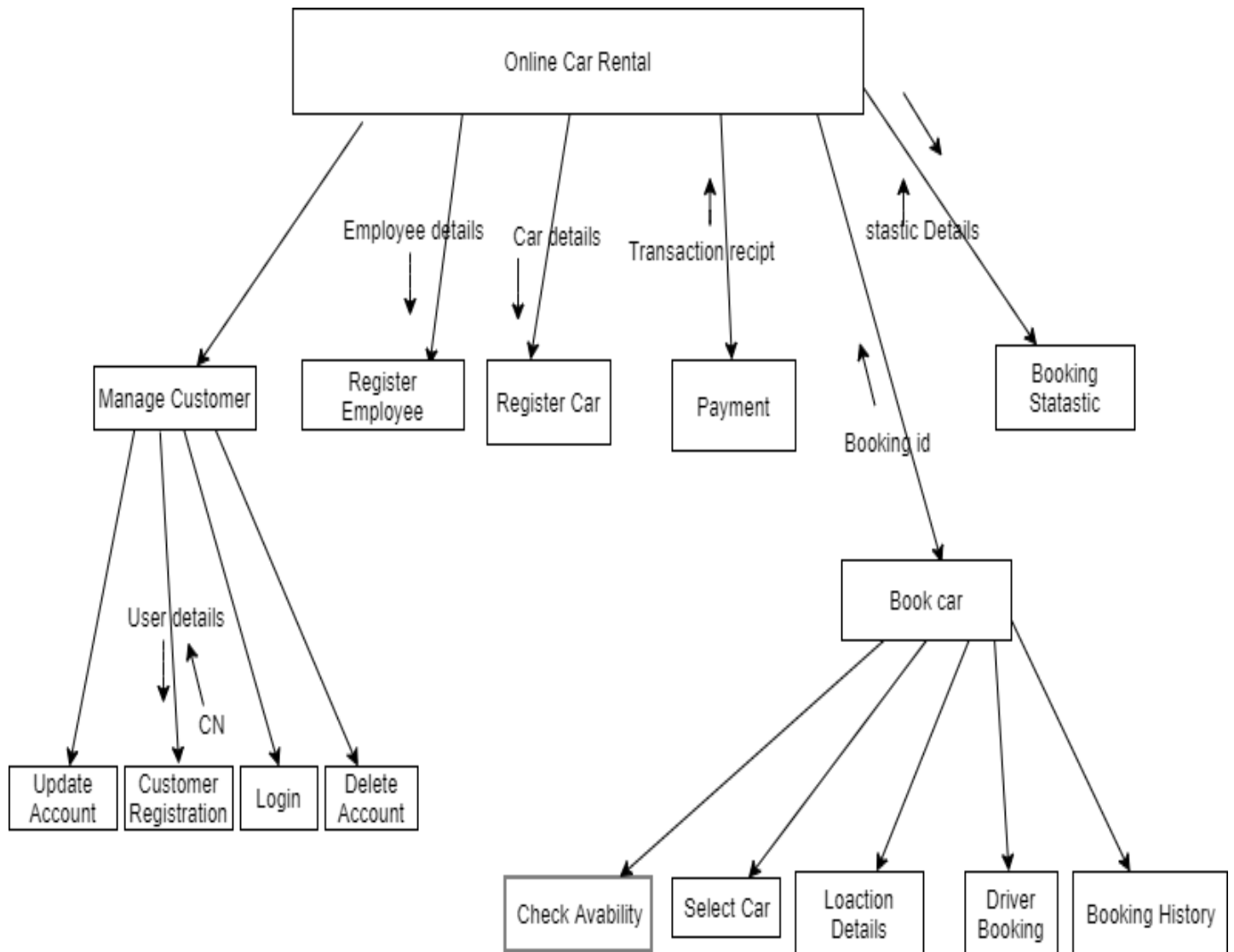
## Level 2(Register customer)



## Level 2(Car Booking)



- **Structure Chart**



# **Implementation Detail:**

## **1. Modules:**

In the following section a brief description of each module is given. Related screenshots are attached in separate sections.

### **Manage User/customer Module**

In Manage User modules employee and admin have access to add user to their System at the first time. user can also register by him/herself. they can delete and update their existing account.

### **Manage Employee-Module:**

Manage Employee modules appear in the admin side. Admin have all access to add Employee with their associated designation. And can perform crud operations on Employee objects..

### **Manage Car -Module:**

In this Module all car related data can be store, update and delete. admin and employee has a access to perform a crude operation in this module.

### **Manage Booking -Module**

This module manage all the booking releted stuff. customer can book a Car ,they can make a payment and also view the booking history.



## 2. Major Functions prototypes

### ➤ REGISTRATION/LOGIN:

The customer can register himself/herself for the first time if he wants to rent a car. He/she has to provide his/her details .

```
def customerinfo(request):
    customerusername = request.POST.get('username', '')
    customerpassword = request.POST.get('password', '')
    customername = request.POST.get('name', '')
    customeremail = request.POST.get('email', '')
    customerphone = request.POST.get('phoneno', '')
    customeraadhar = request.POST.get('aadhar', '')
    customeraddress = request.POST.get('address', '')
    customerdob = request.POST.get('dateofbirth', '')
    s = Customer(customer_id=None, customer_username=customerusername, customer_password=customerpassword,
                 customer_name=customername, customer_email=customeremail, customer_phone_no=customerphone,
                 customer_aadhar_no=customeraadhar, customer_address=customeraddress,
                 customer_dob=customerdob)
    user = User.objects.create_user(username=customerusername, email=customeremail, password=customerpassword)
    user.save()
    s.save()
    return HttpResponseRedirect('/customer_login/registersuccess/')

def registersuccess(request):
    return render(request, 'customerhomepage.html')
```

```
def customerlogin(request):
    username = request.POST.get('username', '')
    password = request.POST.get('password', '')
    user = auth.authenticate(username=username, password=password)
    if user is not None:
        auth.login(request, user)
        return render(request, 'customerhomepage.html')
    return HttpResponseRedirect("Invalid Credentials")
```

### ➤ BOOKING OF CAR:

The booking can be made by the customer which includes the listing of the user requirements of the (sometime 3D visualization is also possible).

```

def b(request):
    carid = request.POST.get('carid')
    total_days = request.POST.get('total_days')
    total_amount = request.POST.get('total_amount')
    cardate = request.POST.get('cardate')
    returndate = request.POST.get('returndate')
    username = request.POST.get('username')
    password = request.POST.get('password')
    car = Car.objects.get(car_id=carid)
    if Customer.objects.filter(customer_username=username, customer_password=password).exists():
        customer = Customer.objects.get(customer_username=username, customer_password=password)
        s = Booking(customerid=customer.customer_id, customername=customer.customer_name, carid=carid,
                    carname=car.car_company, carnumber=car.car_number, carbookdate=cardate, carreturndate=returndate,
                    totalpayment=total_amount, total_no_of_days=total_days)
        s.save()
        car.car_availability = False
        car.save()
        return render(request, 'payment.html')
    else:
        return render(request, 'invalid_details.html')

```

### ➤ VEHICLE UPDATE:

This is done by the admin which maintains the new entry of the new vehicle registering or the car which is no longer in need. Also there can be changes in the price i.e the admin can change the price.

```

def carupdated(request):
    carid = request.POST.get('carid', '')
    carname = request.POST.get('carcompany', '')
    carclasstype = request.POST.get('carclass', '')
    price = request.POST.get('price', '0')
    carnumber = request.POST.get('carnumber', '')
    fueltype = request.POST.get('fuel', '')
    cardescription = request.POST.get('description', '')
    caravailability = request.POST.get('availability', 'True')
    c = Car.objects.get(car_id=carid)
    c.car_company = carname
    c.car_class_type = carclasstype
    c.price_per_day = price
    c.car_number = carnumber
    c.fuel_type = fueltype
    c.car_description = cardescription
    c.car_availability = caravailability
    c.save()
    return render(request, 'employeehomepage.html')

```

## ➤ EMPLOY MANAGEMENT :

All the records of the company's employs with their details and salary, designation etc.

```
def addemployeeinfo(request):
    try:
        username = request.POST.get('employeeusername')
        user = User.objects.get(username=username)
        return HttpResponse("This Username already exists")

    except:
        username = request.POST.get('employeeusername', '')
        password = request.POST.get('employeepassword', '')
        employeeename = request.POST.get('name', '')
        employeeemail = request.POST.get('email', '')
        employeeephone = request.POST.get('phoneno', '')
        employeeaadhar = request.POST.get('aadhar_no', '')
        employeeaddress = request.POST.get('address', '')
        employeeedob = request.POST.get('dob', '')
        employeesalary = request.POST.get('salary', '0')
        employeeedesignation = request.POST.get('designation', '')
        employeejoin = request.POST.get('join', '')
        employeeavailability = request.POST.get('availability', 'True')
        user = User.objects.create_user(username=username, email=employeeemail, password=password)
        user.save()
        s = Employee(employee_id=None, employee_username=username, employee_password=password,
                      employee_name=employeeename,
                      employee_email=employeeemail, employee_phone_no=employeeephone, employee_aadhar_no=employeeaadhar,
                      employee_address=employeeaddress, employee_dob=employeeedob, employee_salary=employeesalary,
                      employee_designation=employeeedesignation, date_of_join=employeejoin,
                      employee_join=employeejoin,
                      employeeavailability = request.POST.get('availability', 'True')
        user = User.objects.create_user(username=username, email=employeeemail, password=password)
        user.save()
        s = Employee(employee_id=None, employee_username=username, employee_password=password,
                      employee_name=employeeename,
                      employee_email=employeeemail, employee_phone_no=employeeephone, employee_aadhar_no=employeeaadhar,
                      employee_address=employeeaddress, employee_dob=employeeedob, employee_salary=employeesalary,
                      employee_designation=employeeedesignation, date_of_join=employeejoin,
                      check_availability=employeeavailability)
        s.save()
        return HttpResponseRedirect('/employee_login/success/')

def success(request):
    return render(request, 'addemployee.html')
```

## Work Flow/Layouts

Online Car Rental Home Dashboard Login About Us Contact Us

Username  
hansh

Password  
\*\*\*\*\*

Name  
hansh thummar

Email address  
hansh@gmail.com

Mobile Number  
6785698463

Aadhar-card Number  
45785478385

Date of Birth  
02-03-2007

Address  
anand, Gujarat

Register

Already have account? Login

Register

127.0.0.1:8000/

Online Car Rental Home Dashboard Login About Us Contact Us

Customer-Login  
Employee-Login  
Admin-Login

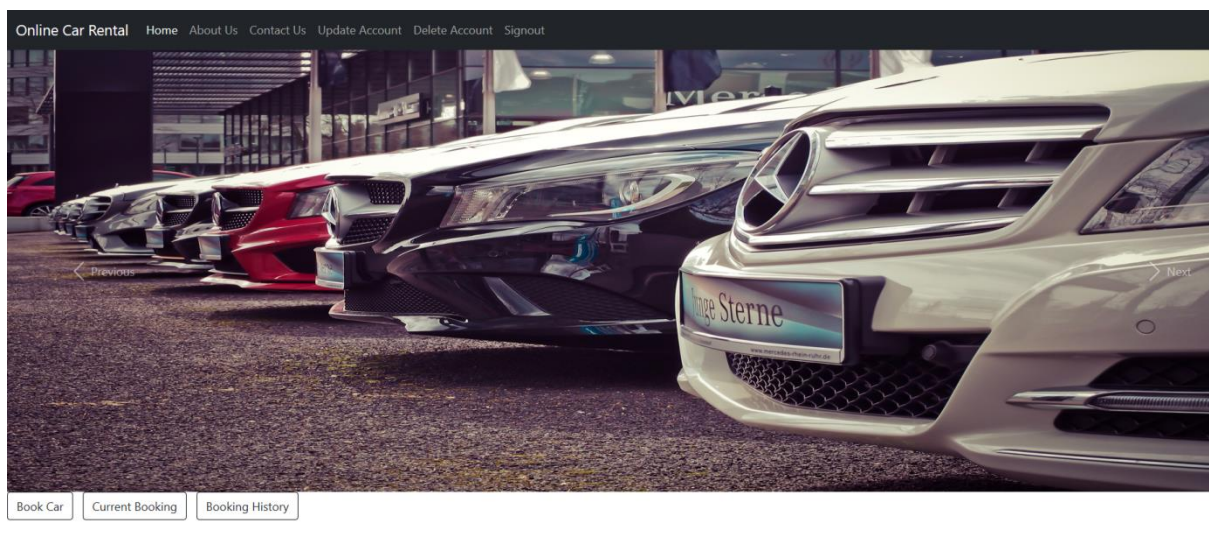
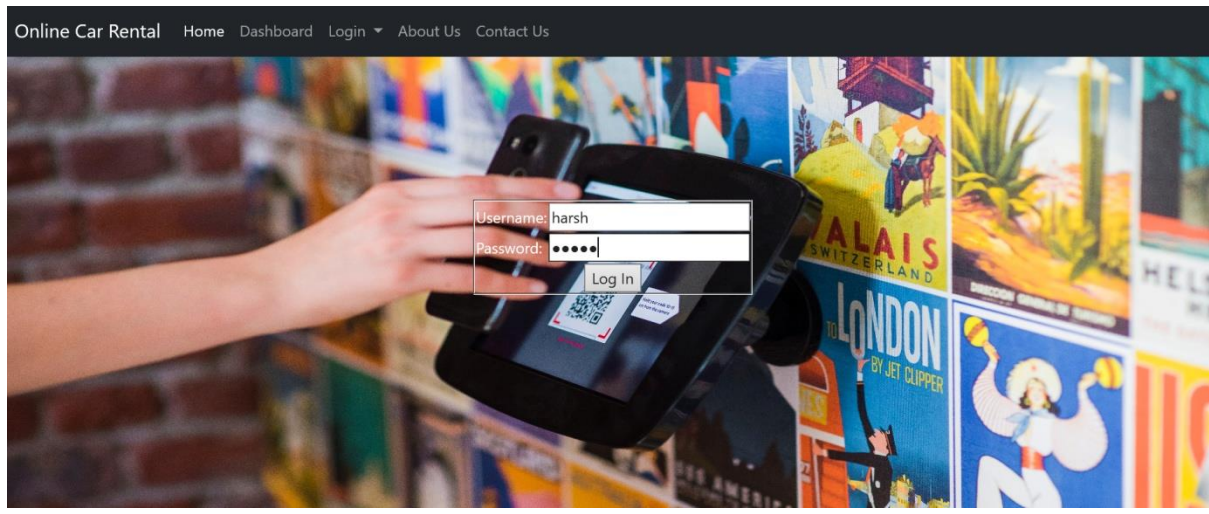
Username  
Enter Your Username

Password  
Password

Name  
Enter Your Fullname

Email address  
email@example.com

Mobile Number  
0123456789



The cars available for booking are:

**CAR\_ID: 5**  
Car Company: SWIFT  
Car Type: SEDAN  
Price(Per Day):200  
Car Number: GJ-27-QW-2390

**CAR\_ID: 6**  
Car Company: i20  
Car Type: SEDAN  
Price(Per Day):1000  
Car Number: GJ-27-QW-2390

## Enter Your Booking Details

**Your SELECTED CAR**  
Car: i20  
Car Class: SEDAN  
Price: 1000  
Car Number:GJ-27-QW-2390

Enter the date of car to take: 01-03-2021  
Enter the car Return date: 04-03-2021

BOOKING HISTORY

127.0.0.1:8000/Car\_Booking/history/

Online Car RentalHomeAbout UsContact UsDelete AccountSignout

# You Have No Bookings History

---

BOOKING HISTORY

127.0.0.1:8000/Car\_Booking/current/

Online Car RentalHomeAbout UsContact UsDelete AccountSignout

Booking Id	3
Car Id	4
Car name	TOYOTA
Car number	GJ-01-OR-1010
Booking Date	March 31, 2021, 1:25 p.m.
Car Taken Date	March 31, 2021
Car Return Date	April 4, 2021
Total Days	4
Total Payment	1000

---



← → ↻ 🏠 ⓘ 127.0.0.1:8000/Car\_Booking/book/ ⓘ ☆ ☆= 🔍

Online Car Rental Home About Us Contact Us Delete Account Signout

Go Back

<b>Your SELECTED CAR</b>	
Car:	i20
Car Class:	SEDAN
Price:	1000
Car Number:	GJ-27-QW-2390
Car taking date:	3000
Car return date:	2021-03-04
Total Days:	3
Your total amount:	3000

**Enter your username and password to confirm your Booking**

Enter Your username:

Enter Your password:

BOOK

---

← → ↻ 🏠 ⓘ 127.0.0.1:8000/employee\_dashboard/authorize/ ⓘ

Online Car Rental Home About Us Contact Us Signout

ti raju

Click Here to add a car -> [ADD CAR](#)

Click Here to remove a car -> [DELETE CAR](#)

Click Here to view the Cars -> [View Cars](#)

Click Here to update Car details -> [Update Cars](#)

---



127.0.0.1

127.0.0.1:8000/employee\_dashboard/getcars/

Online Car Rental

HomeAbout UsContact UsSignout

CAR COMPANY TOYOTA

Car ClassSUV

Price250

Car NumberGJ-01-OR-1010

Car FuelPETROL

Car description It is a car for long distance travelling.

CAR COMPANY SWIFT

Car ClassSEDAN

Price200

Car NumberGJ-27-QW-2390

Car FuelDIESEL

Car description it is car with good ratings

CAR COMPANY i20

Car ClassSEDAN

Price1000

Car NumberGJ-27-QW-2390

Car FuelDIESEL

Car description it is car with good ratings!!!

## **Conclusion: -**

- Car rental business has emerged with a new goodies compared to the past experience where every activity concerning car rental business is limited to a physical location only. Even though the physical location has not been totally eradicated; the nature of functions and how these functions are achieved has been reshaped by the power of internet. Nowadays, customers can reserve cars online, rent car online, and have the car brought to their door step once the customer is a registered member or go to the office to pick the car.
- The web based car rental system has offered an advantage to both customers as well as Car Rental Company to efficiently and effectively manage the business and satisfies customers' need at the click of a button.

## **Limitations:**

- 1) Here customer can't show the present working condition of the car on this software.
- 2) If any car are booked for Rent then other customer can't see that how long that car will be booked.
- 3) Customer can't hire a driver through this software

## **Future Extension:**

To take over the limitations we are planning this future extension in our system.

- 1) We can add the feature of 3D view of car.
- 2) We can add the feature of driver booking.
- 3) We can track the current location of the Rented car.
- 4) We can add defaulter list.

# **Bibliography**

## **References/resources used for developing project:**

### **Books Used:**

- Fundamentals-of-software-engineering- Rajib-mall
- Software Engineering - Roger S. Pressman

### **References Used:**

- <https://docs.djangoproject.com/en/3.1/>
- <https://www.geeksforgeeks.org/django-tutorial/>