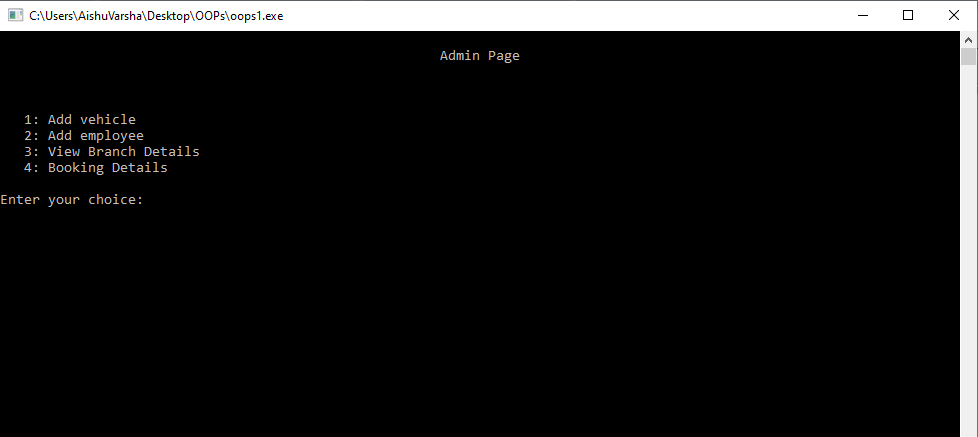
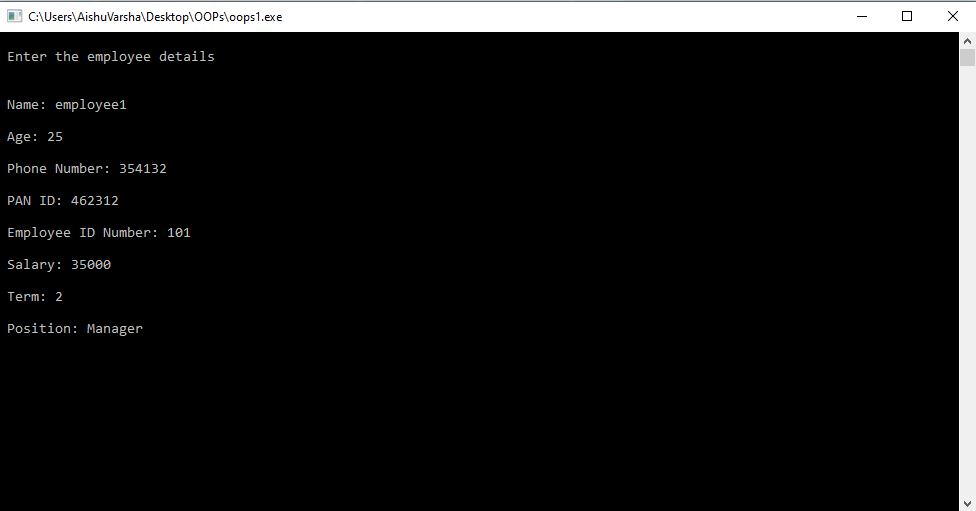
|  |  |  |
| --- | --- | --- |
| **1.** | **Introduction** | |
|  | **1.1** | **Overview of the problem statement**  The problem statement which we have chosen is regarding the Car Booking system in a showroom. The customers should be able to see the details of different types of vehicles available and the details of the vehicle. The customer thereafter should be able to book the desired vehicle. After the completion of booking the vehicle by a customer, the admin must be able to see the details of all the bookings.  The project should also be able to provide the admin to add a vehicle to the directory when a new vehicle comes and the admin must be able to add employee details to the directory. The admin should have the access to view all booking history, branch details like number of vehicles and employees present in the showroom and the number of bookings done. The project should also be able to handle any types of errors like accessing any empty directory i.e., when no record is present in the directory. |
|  | **1.2** | **Features of Application**   1. addEmployee 2. addVehicle 3. displayEmployee 4. displayVehicles 5. bookVehicle 6. displayBookings 7. NoOfEmployee 8. NoOfVehicles 9. NoOfBookings 10. DisplayBranchDetails |
| **2.** | **Design** | |
|  | 2.1 | Class Diagrams    **Class Diagram explanation:**   1. TataMotors:  |  | | --- | | TataMotors | | + companyid: int  +companyowner: string  +companyhelpline: int | | -company()  -company(int,string,int)  -displayCompanyDetails(): void |   Tata Motors is a company having many branches. This class has data members such as companyid, companyowner, companyhelpline to store the details of the company. This class also has constructors which are used to initialize and displayCompanyDetails function is used to display the company’s details like id, owner and its helpline no.   1. Branch:  |  | | --- | | Branch | | +branchid: int  +branchmanager: string  +no\_of\_vehicles: int  +no\_of\_employees: int | | -branch()  -branch(int,string,int,int)  -add\_vehicle(): void  -remove\_vehicle(): void  -add\_emp():void  -remove\_emp(): void  -displayBranchDetails(): void |   This class has aggregation relation with class TataMotors. Each branch has a manager, branch id/no, employees and vehicles. This class has constructors to initialize. When some vehicles are delivered to a branch, the function add\_vehicle() adds that vehicle into the system and store its details. When a vehicle is sold, the function remove\_vehicle() deletes the vehicle from the system. The function add\_emp() adds an employee to the directory when a new employee is hired. And the function remove\_emp() is used to remove an employee from the directory and deletes all his/her details. displayBranchDetails() is used to display the details of any particular branch.   |  | | --- | | Vehicle | | +vehiclename: string  +modelno: int  +vehicleid: int  +regno: string  +price: int | | -vehicle()  -vehicle(string,int,int,int,int)  -displayVehicleDetails(): void |  1. Vehicle:   Every vehicle has a name, model number, vehicle id, registration number and a certain price. This class has constructors which are used to initialize and the function displayVehicleDetails() is used to display all the details of the vehicle.   1. Employee:  |  | | --- | | Employee | | +employeeid: int  +employeesalary: int  +employeeterm: int  +emp\_position: string | | -employee()  -employee(int,int,int,string)  -update(): void  -displayEmpDetails(): void |   Each employee has their own id, a certain amount of salary, fixed amount of term and a particular position. Constructors are used to initialize and the update() function is used to update the information such as position and salary when anyone is promoted or got a salary hike.   1. Customer:  |  | | --- | | Customer | | +custid: int  +custlicense: string | | -customer()  -customer(int,string)  -displayCustDetails(): void |   All customers are persons. So this customer class is inherited by the class person. Each customer has an id and a driving license. Constructors are used to initialize and the displayCustDetails() function is used to display customer details.   1. Person:  |  | | --- | | Person | | +name: string  +age: int  +phoneno: int  +PANid: int | | -person()  -person(string,int,int,int)  -displayPersonDetails(): void |   A person has name, age, phone number and an address. The class person is a parent class which is inherited by the class employee and the class customer. This class has constructors to initialize and a displayPersonDetails() function to display the person’s details.   1. Booking:  |  | | --- | | Booking | | +orderid: int  +orderdate: int  +vehicle: Vehicle  +employee: Employee  +customer: Customer  +payment: Payment | | -booking(int,int,Vehicle, Employee,Customer,Payment)  -displayOrder(): void |   An employee will sell a particular vehicle to a customer. When a vehicle is booked, an order id, order date and payment is recorded. The class booking doesn’t have default constructor because there must be a customer, employee and a vehicle to book. Parameterized constructor is used to initialize and the displayOrder() function is used to display.   1. Exception:  |  | | --- | | Exception | | -errorno: int  -errormsg: string | | +excpetion(int,string)  +showerror(): void |   This class is used to handle all the exception cases and negative cases throughout the code. |
| **3.** | **Unit Test Plan**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Features | Constructors | Inheritance | Exception Handling | getdata()/ putdata() | Virtual function | | Admin/ Customer login |  |  |  | Yes |  | | Fetching vehicle details |  | Yes | Yes | Yes |  | | Fetching employee details |  | Yes |  | Yes | Yes | | Fetching branch details |  |  | Yes | Yes |  | | Booking vehicle | Yes |  |  |  |  | | About |  |  |  | Yes |  | | Number of vehicles |  |  |  | Yes |  | | Number of employee |  |  |  | Yes |  | | Number of bookings |  |  |  | Yes |  | | Fetching bookings history |  | Yes | Yes | Yes |  | | |
| **4** | Implementation | |
|  | 4.1 | Results: |

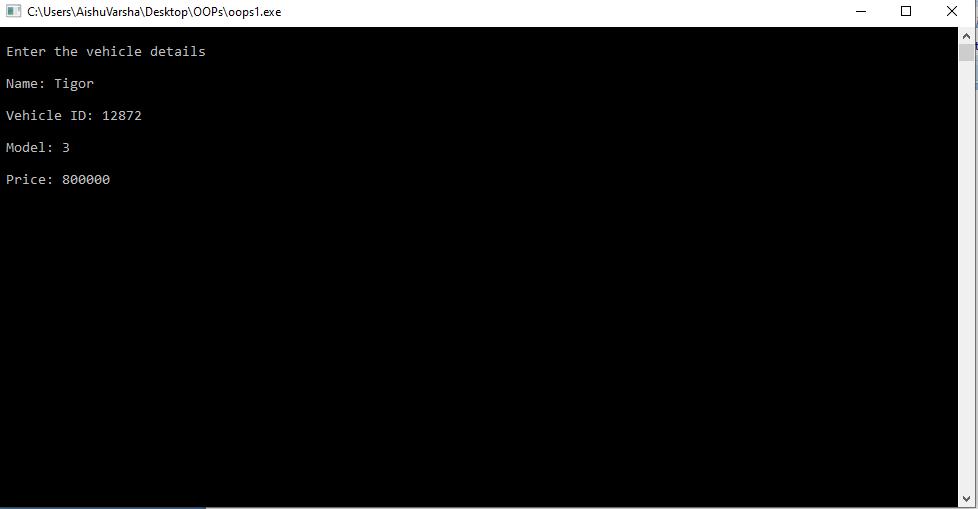


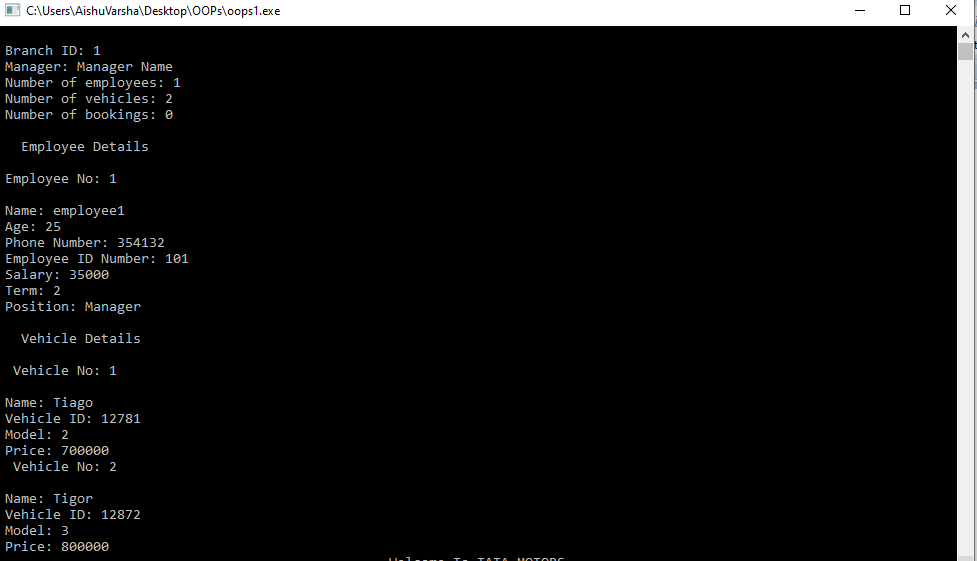
Admin Page:



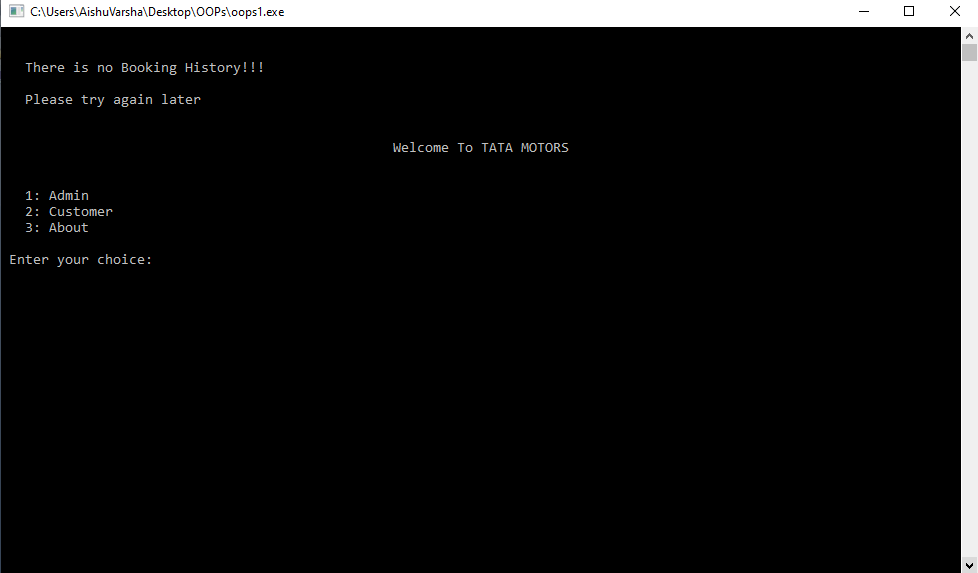
Add Employee :

Add Vehicle:

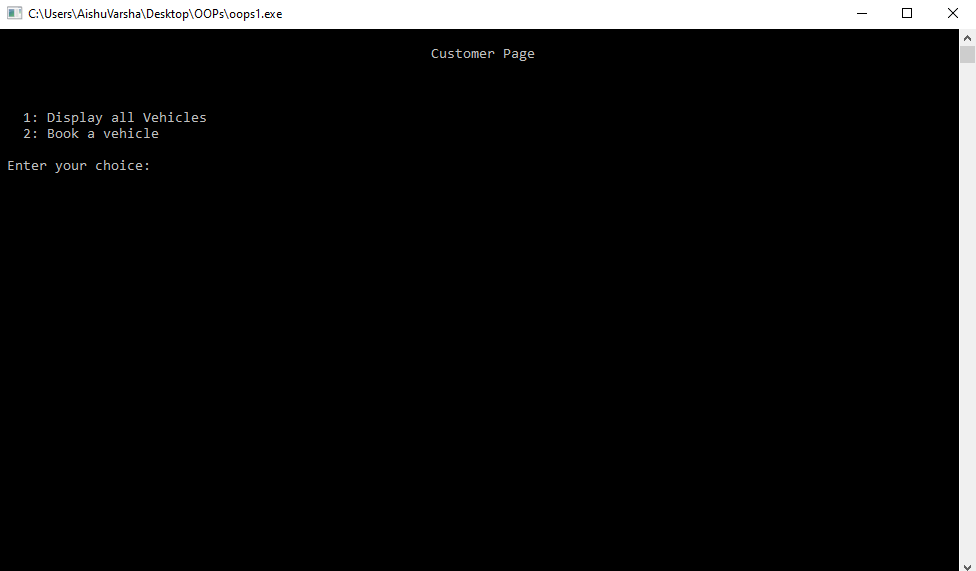


View Branch Details : 

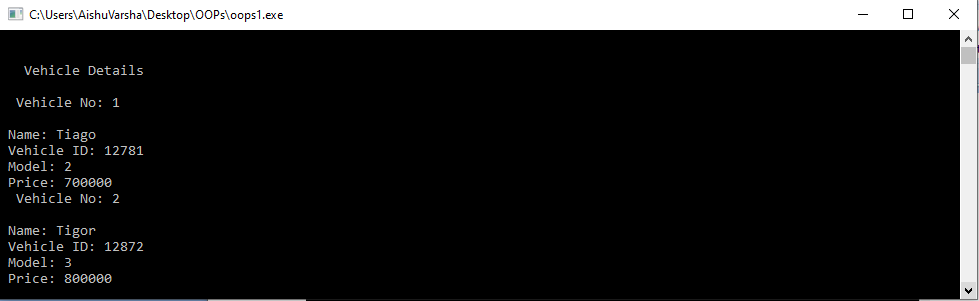
Error while accessing booking history when no booking has done previously :



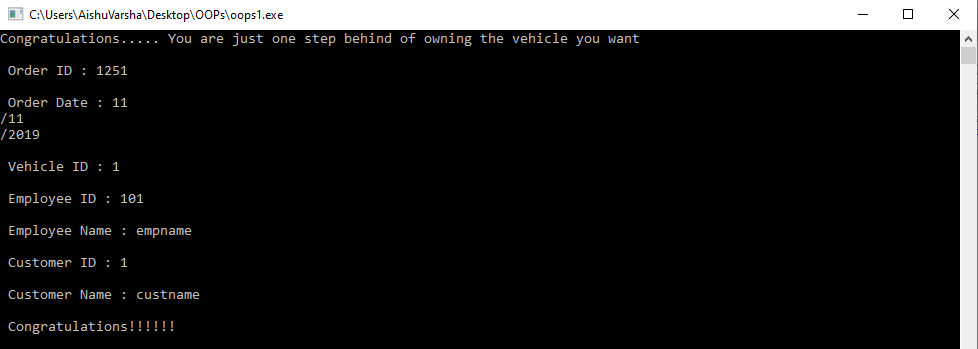
Customer Page :



Display All Vehicles :



Booking a Vehicle :



Booking Details:

