Online Car Rental

A Major Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

BY

Tanay Pandit (EN16CS301274)

Under the Guidance of

Prof. Sachin Solanki

Mr.Milind Nekadi

Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

May 2020

Report Approval

The project work **"Online Car Rental"** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the "Project Report" only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation:

Affiliation:

External Examiner

Name:

Designation:

Affiliation:

Declaration

I hereby declare that the project entitled **"Online Car Rental"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications in **'Computer Science Engineering'** completed under the supervision of **Milind Nekadi, .NET Developer,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Tanay Pandit

.....

Certificate

I, **Prof.Sachin Solanki** certify that the project entitled **"Online Car Rental"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications by **Tanay Pandit** is the record carried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Prof. Sachin Solanki	Mr. Milind Nekadi
Computer Science Enigneering	.NET Department
Medi-Caps University, Indore	Systematix Infotech Pvt .Ltd

Dr. Suresh Jain

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

Acknowledgement

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Sunil K Somani,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda,** Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Suresh Jain** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **Prof.Sachin Solanki, Mr. Manoj Kumar**, Project Lead, Systematix Infotech Pvt. Ltd as well as to my Internal Guide, Dr**. Debendra Kumar Panda,** Professor, Department of Computer Science, MU, without whose continuous help and support, this project would ever have reached to the completion.

I would also like to thank to my team at Systematix Infotech Pvt. Ltd Mr .Milind Nekadi, Mr.Dilip Namdeo,etc who extended their kind support and help towards the completion of this project.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Tanay Pandit

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

Abstract

Nowadays, there are online car reservations which give much benefit to user. A rental service is a service in which customers arrive to request the hire of a rental unit. It is more convenient than carrying the cost of owning and maintaining the unit.

A car rental or car hire agency is a company that rents automobiles for short period of time for a fee whether in a few hours or a few days or week. It is an extended form of a rental shop, often organized with numerous local branches (which allow a user to return a vehicle to a different location), and primarily located busy city areas and often complemented by a website allowing online reservations.

Car rental agencies primarily serve people who have a car that is temporarily out Of reach or out of service, for example travelers who are out of town or owners of damaged or destroyed vehicles who are awaiting repair or insurance compensation. Because of the variety of sizes of their vehicles, car rental agencies may also serve the self-moving industry needs, by renting vans or trucks, and in certain markets other types of vehicles such as motorcycles or scooters may also be offered.

In short, it is a system design specially for large, premium and small car rental business. The car rental system provides complete functionality of listing and booking car.

Table of Contents

List of Figures

List of Tables

Abbreviations

ER Diagram	Entity Relationship
SQL	Structured Query Language
DFD	Data Flow Diagram

Chapter 1

Introduction

1.1 Introduction

The Major project provides an opportunity to do something our own and under the supervision of our guide. We are working on the project, which involve fabrication, design or investigation of a software project that may take design, logical or analytical character or combine element of these areas. The title of project is **“Car Rental"**. It help people for Booking Car Online. It provides different types of cars for Customer and income source for vendors. This Project is used for Customer can Book Car Easley and travel between two city and state which is submitted to Medicaps University. This project will be a milestone in the facilities provided by college to the faculty. The Rent Car System is the student project developed by engineering students. This system will facilitate Rent a a Car System which is very popular now a days. It has lists of car with different rental tariff per day or per kilo meter. This project is developed using ASP.NET as front end and SQL Server as back-end.

The main idea of Rent Car System is using the internet technology with our daily life. Our project helps the rent car office and our main target who are the customers. The employee needs an automated system that helps him and speed up his work with professional way.

1.2 Objectives

Objective of online car rental system project is to maintain records of cars. Car rental system project report and documentation is attached with this page in C# and asp.net. Basically this system help car rental shopper to make daily record and easy billing of customers. Also help to maintain monthly revenues and help to analysis data to grow business. Car rental management system project is best suited for those car rental providers.

1.3 Significance

The main purposes of the Mobile Car Rental System include:

i)Able to recommend car to be rented by the users based on the three requirements which arebudget, number of passenger(s) and distance.

ii)Provide car catalog for users as an alternative for them to select car if they want to choose car bytheir own.

iii)Functioned in adding, deleting, updating and searching the data or information depends on thesecurity level.

iv)Allows the organization to search user information from the database based on the user'sidentification card number.

v)Provide an avenue for launching of complains if any to the administration of the companyand improve the customer experience.Administrator - allowed accessing all functions in the system.

1.4 Problem

The Manual car rental system provides services only during office hours. So; customers have limited time to make any transactions or reservation of the cars. However; there is still a few numbers of these online car rental systems in India and most of the systems offered reservation service for tourists or traveler. Besides that, there are some customers who faced a problem in choosing car to be rented which suitable with some of the important requirements.

1. To rent a car a prospective renter must first go to the nearest office to register as a client.

2. Cars that provide difficulties to rent out are normally advertised in local or national newspaper.it involves a lot of paper work and consumes time.

Chapter 2

System Requirement Analysis

2.1 Information Gathering

Vehicle rental project is a online web portal implemented in asp.net platform using c# programming language. This system is useful for cab organizing companies for efficiently managing and organizing cab service for large number of users through internet. Users can book cab service from any were in the world and take service when they visit that city. There are different services provided based on different level of cab standards.There are Three Module

Admin, Customer& Vander.Use email for messaging.Requirement of client is use SQL database ,object oriented programming language which is C#,for web designing ASP.NET .

2.2 System Feasibility

2.2.1 Economical

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available.

2.2.2 Technical

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.2.3 Behavioral

The project is operationally feasible as the user having basic knowledge about computer and Internet can use. Furthermore the project can be easily used if the computers have no internet access.

2.3 Platform Specification (Development & Deployment)

2.3.1 HARDWARE

- 8GB RAM
- 320GB HDD
- Intel Core i5

2.3.2 Software Implementation Language/ Technology

Operating System – Windows 7/8/10

Back End – C#,JavaScript

Front End – Asp.Net ,HTML,CSS

Server Software – Microsoft SQLServer Management Studio

Framework- .NET(Visual Studio 2019)

- **.NET Framework :-** .NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library called Framework Class Library and provides language interoperability across several programming languages.
- **C#:-** C# is a general-purpose, multi-paradigm programming language encompassing strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines.
- **Visual Studio 2019:-** Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.
- **ASP.NET:-** ASP.NET is an open-source, server-side web-application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, applications and services.
- **MSSMS:- SQL Server Management Studio** (SSMS) is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server.

Chapter 3

System Analysis

3.1 Information Flow Representation

3.1.1 ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

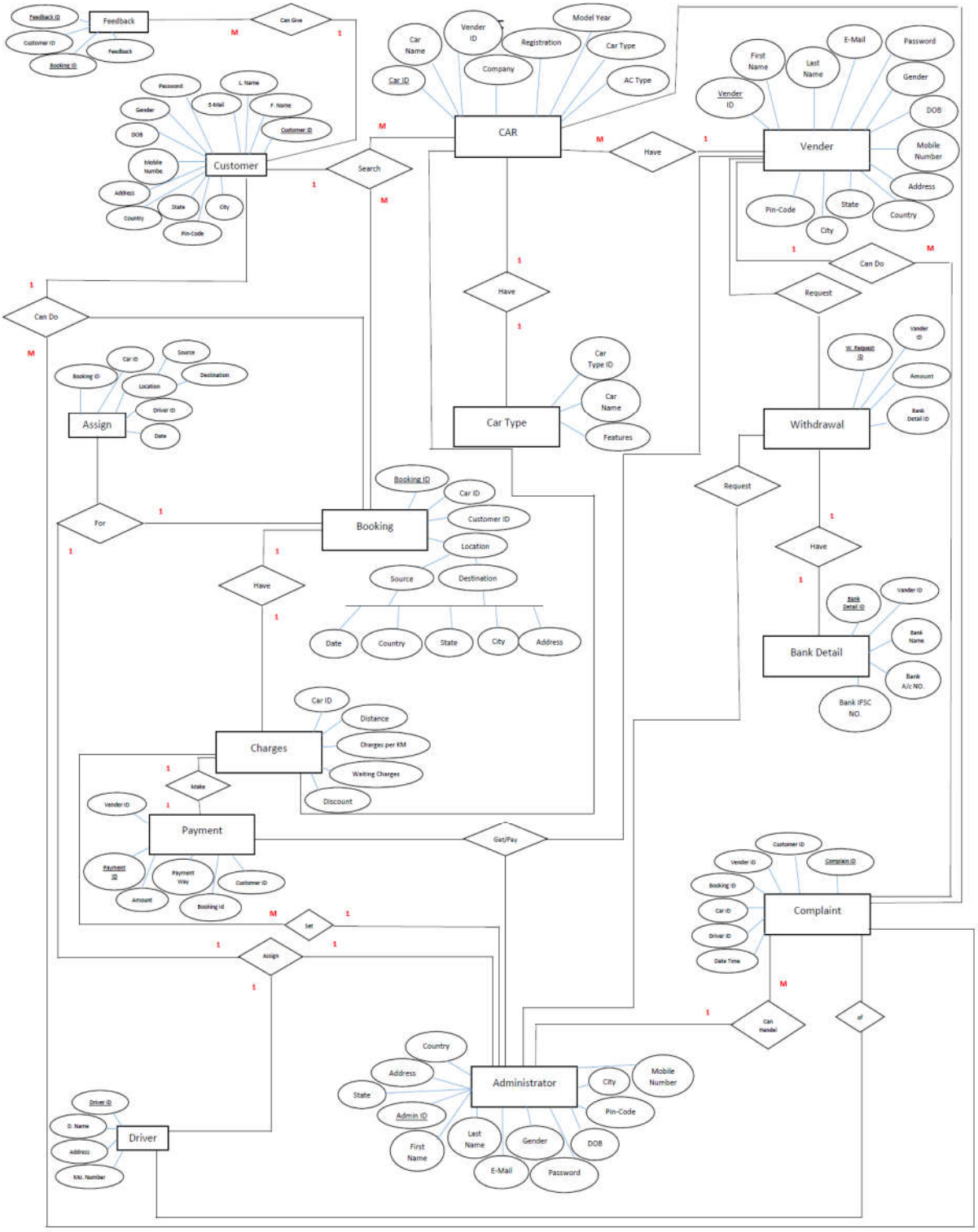

Figure 3.1.1 ER Diagram

3.1.2 Data Flow Diagram

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

There is a prominent difference between DFD and Flowchart. The flowchart depicts flow of control in program modules. DFDs depict flow of data in the system at various levels. DFD does not contain any control or branch elements.

Types of DFD

Data Flow Diagrams are either Logical or Physical.

- **Logical DFD** - This type of DFD concentrates on the system process, and flow of data in the system.For example in a Banking software system, how data is moved between different entities.
- **Physical DFD** - This type of DFD shows how the data flow is actually implemented in the system. It is more specific and close to the implementation.

DFD Components

DFD can represent Source, destination, storage and flow of data using the following set of components -

Figure 3.2.1

- **Entities** - Entities are source and destination of information data. Entities are represented by a rectangles with their respective names.

- **Process** - Activities and action taken on the data are represented by Circle or Round-edged rectangles.
- **Data Storage** - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.
- **Data Flow** - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

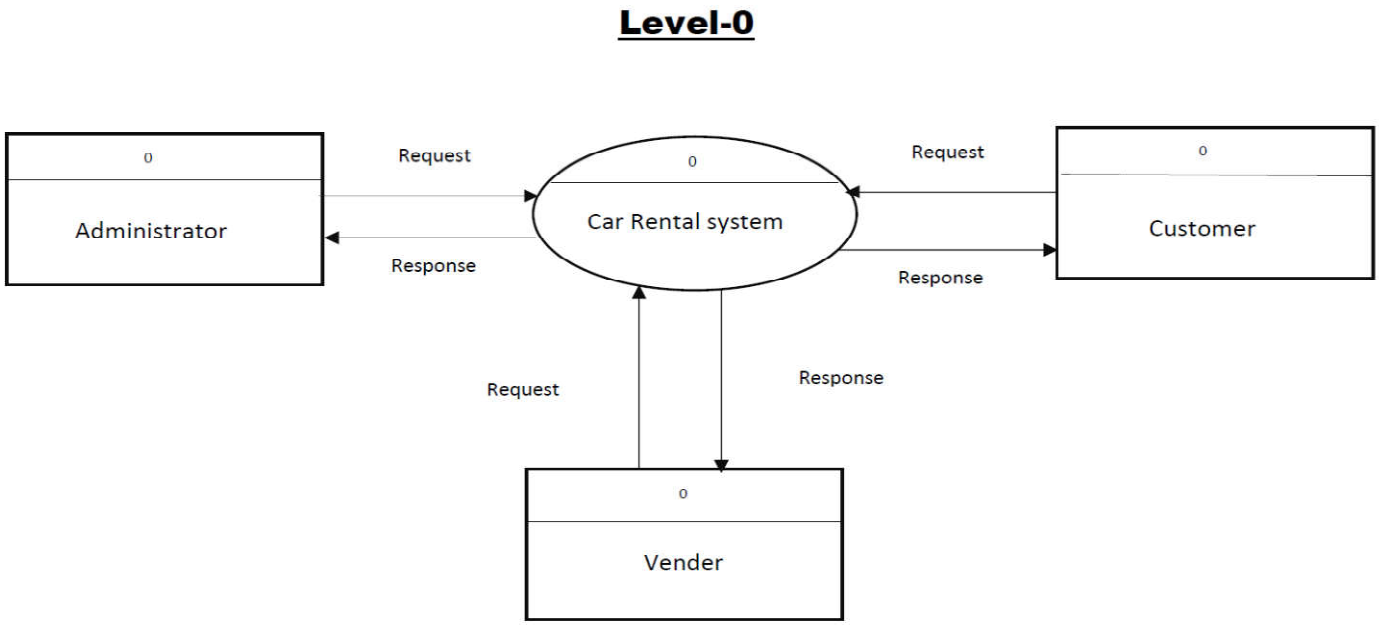

Figure 3.2.2

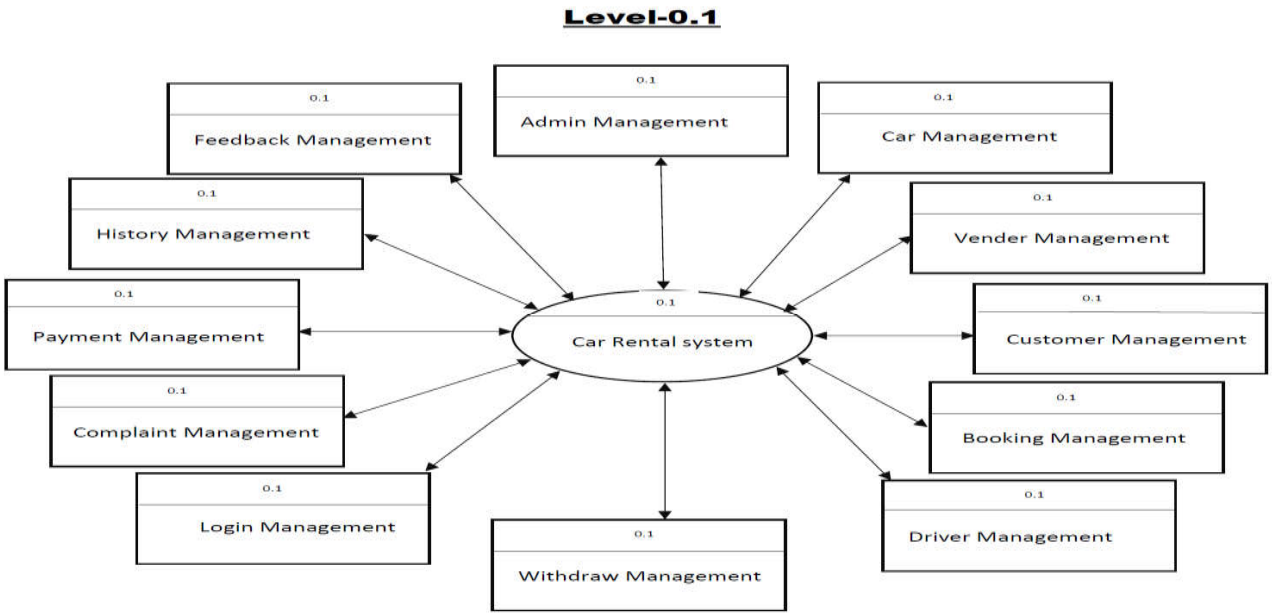
Level-0.1
0.1
Feedback Management
0.1
Admin Management
0.1
Car Management
0.1
History Management
0.1
Vender Management
0.1
Payment Management
0.1
Car Rental system
0.1
Customer Management
0.1
Complaint Management
0.1
Booking Management
0.1
Login Management
0.1
Withdraw Management
0.1
Driver Management

Figure 3.2.3

Level-1

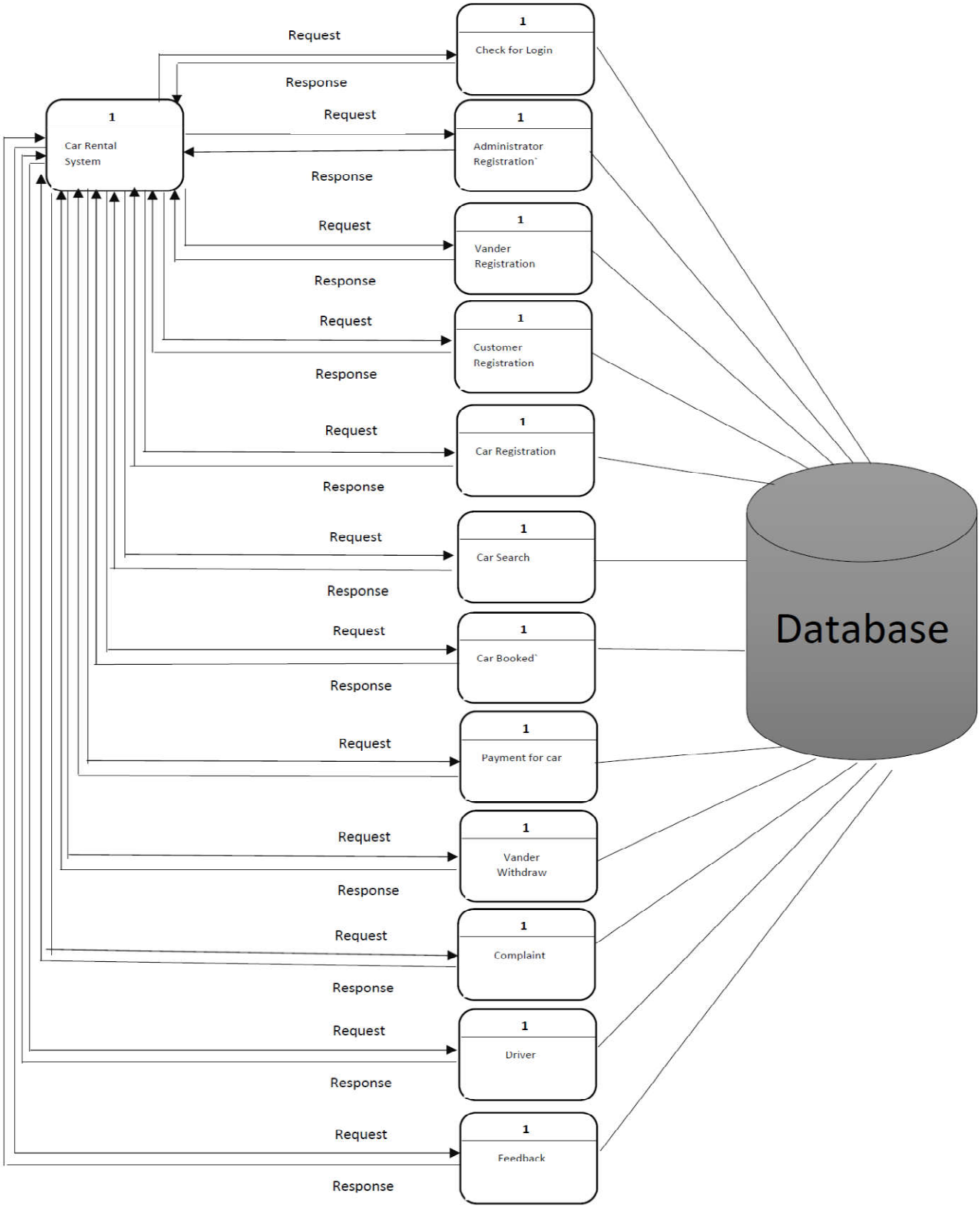

Figure 3.2.4

Level-1.1

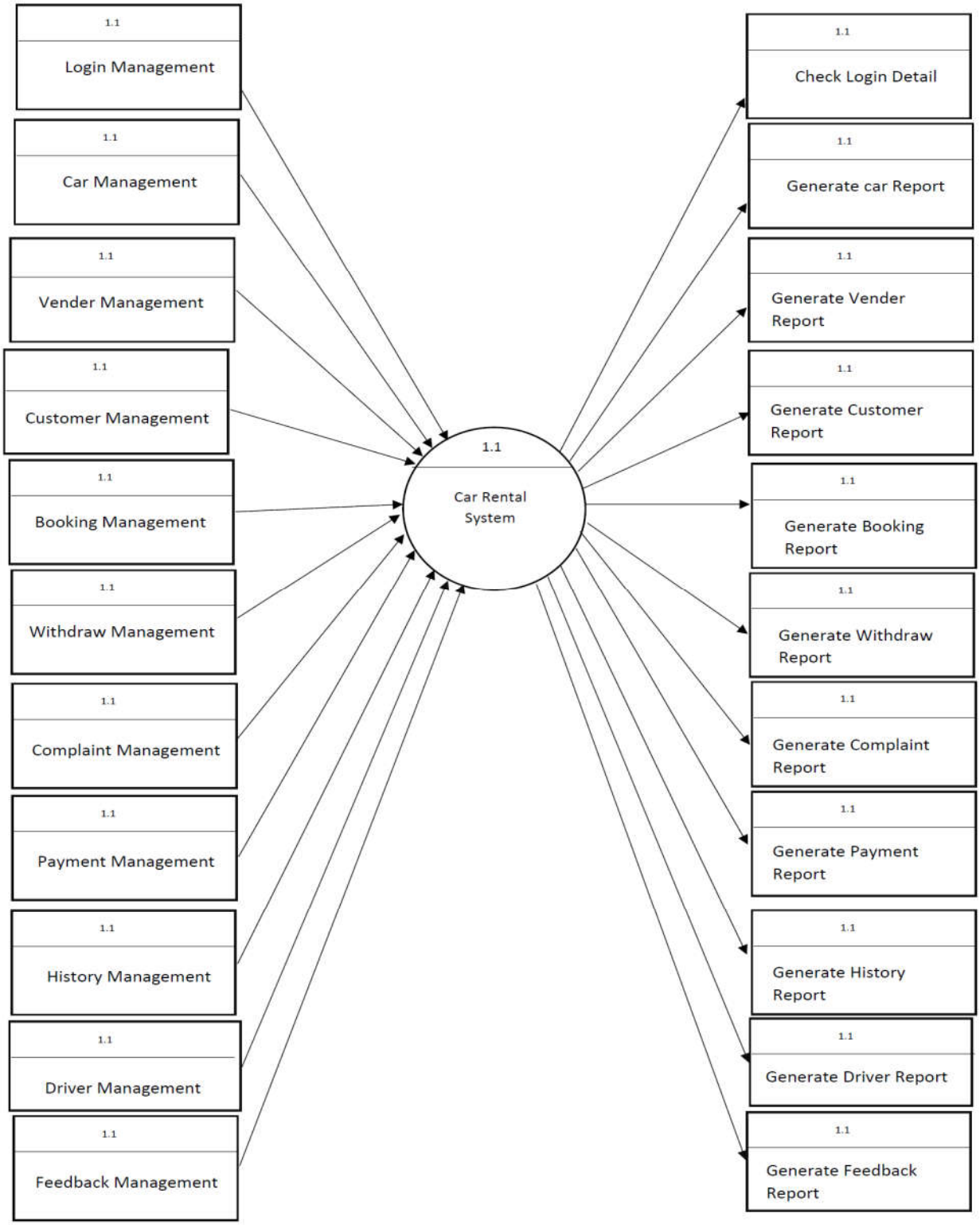

Figure 3.2.5

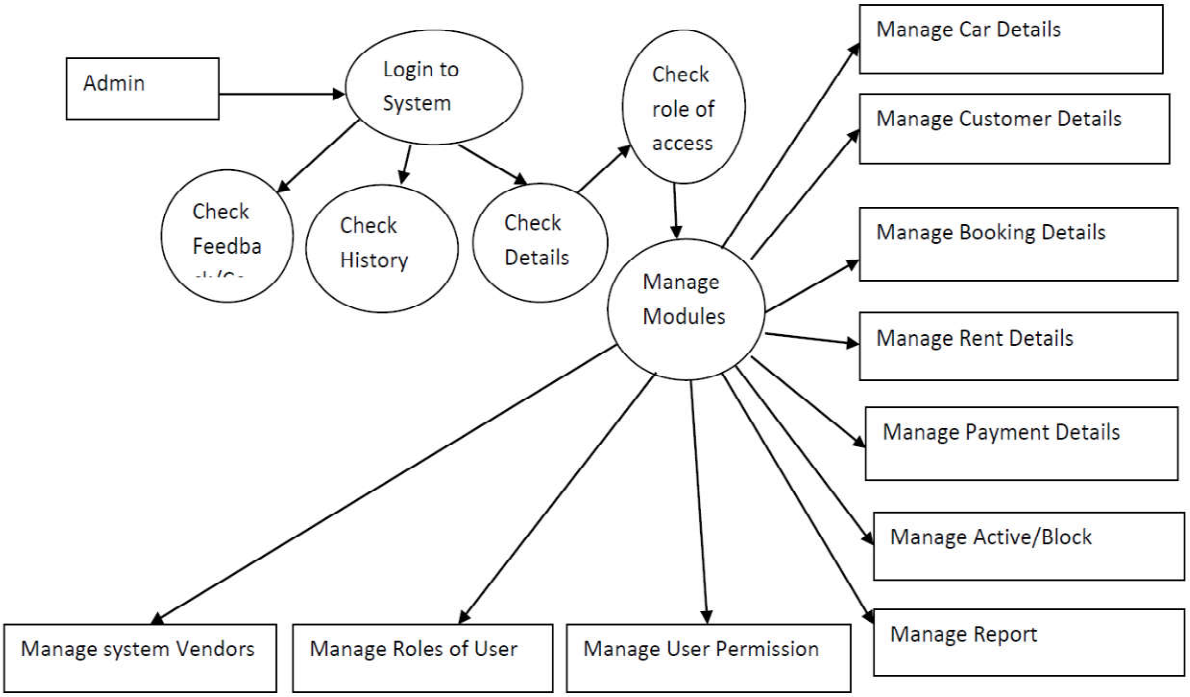
Admin
Login to System
Check Feedba
Check History
Check Details
Check role of access
Manage Modules
Manage Car Details
Manage Customer Details
Manage Booking Details
Manage Rent Details
Manage Payment Details
Manage Active/Block
Manage Report
Manage system Vendors
Manage Roles of User
Manage User Permission

Figure 3.2.6

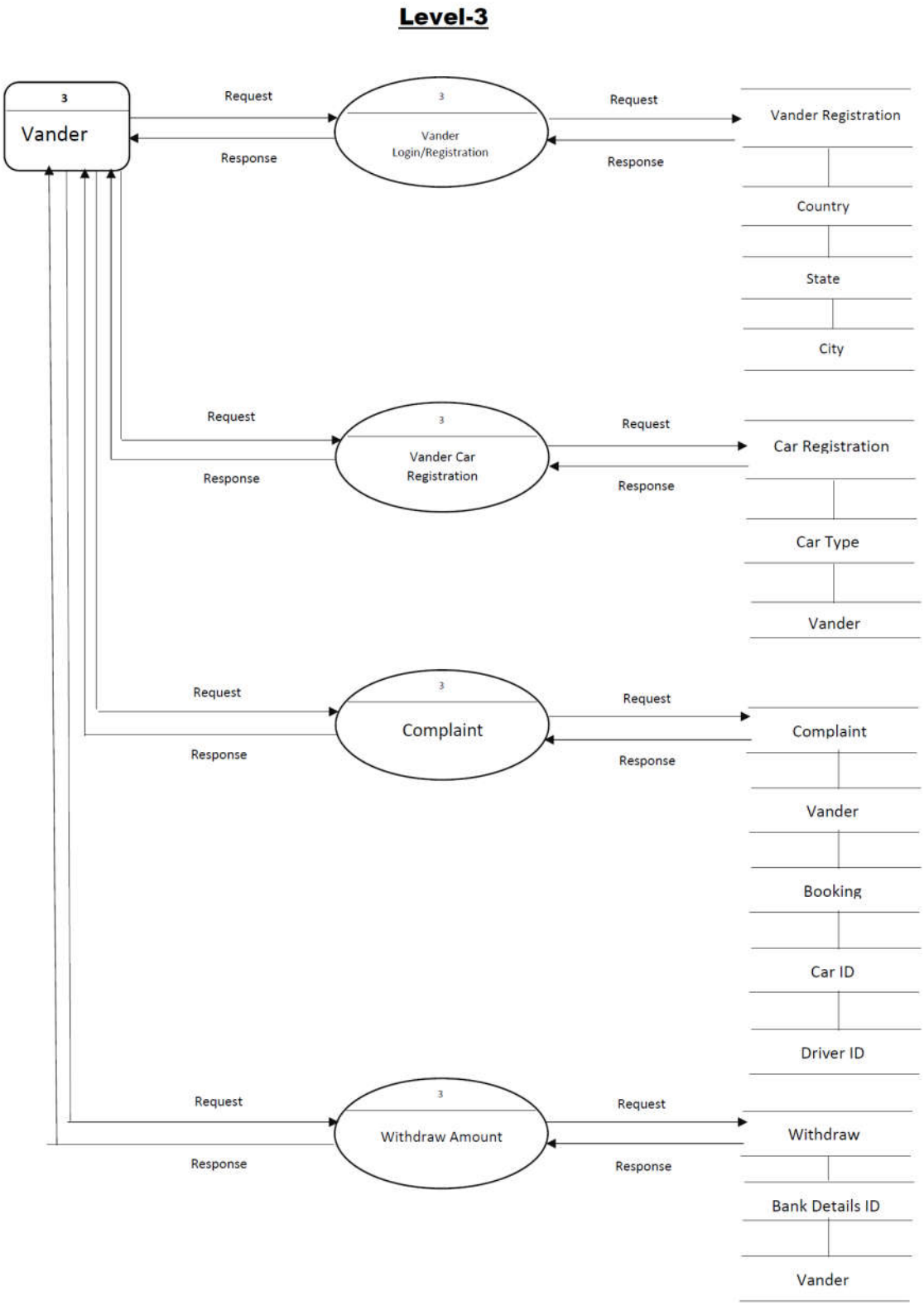
Level-3
3
Vander
Request
Response
3
Vander
Login/Registration
Request
Response
Vander Registration
Country
State
City
Request
Response
3
Vander Car
Registration
Request
Response
Car Registration
Car Type
Vander
Request
Response
3
Complaint
Request
Response
Complaint
Vander
Booking
Car ID
Driver ID
Request
Response
3
Withdraw Amount
Request
Response
Withdraw
Bank Details ID
Vander

Figure 3.2.7

Level-4

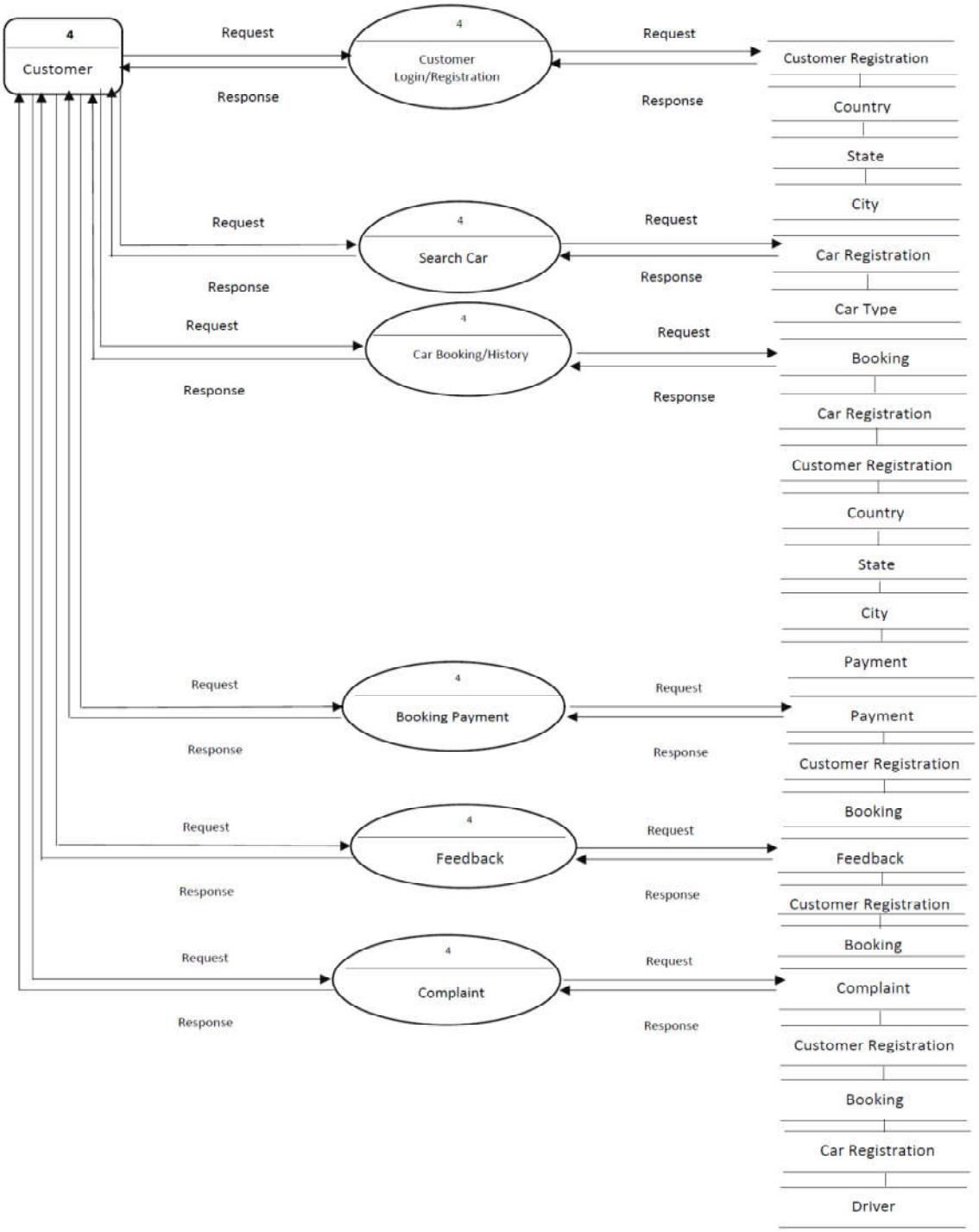

Figure 3.2.8

3.1.3 Data Dictionary

Data Dictionary for Customer

❖ **Customer Registration:-**

All information of vender are saved in this table, such as First Name, Last Name, Age, DOB, Gender, Address etc.

Filed	Data Type	Description
Customer ID	VARCHAR(50)	**(Primary Key)** Customer's ID that is use for login
First Name	VARCHAR(50)	Customer's First Name
Last Name	VARCHAR(50)	Customer's Last Name
Email ID	VARCHAR(50)	Customer's Email ID
Password	VARCHAR(50)	Customer's login password
Gender	VARCHAR(50)	Customer's gender type
DOB	Date	Customer's Date of Birth
Mobile Number	VARCHAR(50)	Customer's Contact Number
Address	VARCHAR(50)	Customer's Address
Country ID	VARCHAR(50)	(**Foreign Key**) is used to save Customer's Country details
State ID	VARCHAR(50)	(**Foreign Key**) is used to save Customer's State details
City ID	VARCHAR(50)	(**Foreign Key**) is used to save Customer's city details
Pin Code	INT (10)	Customer's area code
Status	VARCHAR(50)	Activate/Deactivate/Blocked status.
Date	Date	Registration Date

Table 1

❖ **Booking:-**

In that all Booking data is saved

Filed	Data Type	Description
Booking Id	VARCHAR(50)	(**Primary Key**) Get Booking Id after each booking
Car Id	VARCHAR(50)	(**Foreign Key**)It save Car Id
Customer Id	VARCHAR(50)	(**Foreign Key**)In that Customer Unique Id Save
Pick-Up Country Id	VARCHAR(50)	(**Foreign Key**)In that Customer Pick-Up Country Id save
Pick-Up Sate Id	VARCHAR(50)	(**Foreign Key**)In that Pick-Up State Id saved
Pick-Up City Id	VARCHAR(50)	(**Foreign Key**)In that Customer Pick-Up City Id saved
Pick-Up Address	VARCHAR(50)	In that Customer Pick Address save
Pick-Up DateTime	DateTime	In that Customer Pick Date and Time Saved
Drop Country Id	VARCHAR(50)	(**Foreign Key**)In that Customer Drop Country Id Saved
Drop State Id	VARCHAR(50)	(**Foreign Key**)In that Customer Drop State d Saved
Drop City Id	VARCHAR(50)	(**Foreign Key**)In that Customer Drop City Id Saved
Drop Address	VARCHAR(50)	In that customer Drop Address saved
Payment Id	VARCHAR(50)	(**Foreign Key**)In that payment Details Saved
Status	VARCHAR(50)	In that Booking Status Saved

Table 2

❖ **Payment:-**

In that all payment details are saved

Filed	Data Type	Description
Payment Id	VARCHAR(50)	(Primary Key) Payment Id is saved for each booking.
Customer Id	VARCHAR(50)	(**Foreign Key**)In that Customer Id save who booked car
Booking Id	VARCHAR(50)	(**Foreign Key**)In this Booking Id saved
Payment Way	VARCHAR(50)	In that payment way are save like Net Banking/Card Payment

Payment DateTime	Date	In that payment's date and time save
Status	VARCHAR(50)	In that payment Status saved

Table 3

❖ **Feedback:-**

In that all feedback data is saved that is given by customer

Field	Data Type	Description
Feedback Id	VARCHAR(50)	(Primary Key) Feedback is saved.
Customer Id	VARCHAR(50)	(**Foreign Key**) Customer Id saved in this field
Booking Id	VARCHAR(50)	(**Foreign Key**)In that booking saved
Feedback	VARCHAR(Max)	In this feedback save that is given by customer.
Feedback DateTime	DateTime	In that date and time save when feedback is given.

Table 4

Data Dictionary for Vender

❖ **Vender Registration:-**

All information of vender are saved in this table, such as First Name, Last Name, Age, DOB, Gender, Address etc.

Filed	Data Type	Description
Vender ID	VARCHAR(50)	(**Primary Key**) Vender's ID that is use for login
First Name	VARCHAR(50)	Vender's First Name
Last Name	VARCHAR(50)	Vender's Last Name
Email ID	VARCHAR(50)	Vender's Email ID
Password	VARCHAR(50)	Vender's login password
Gender	VARCHAR(50)	Vender's gender type

DOB	Date	Vender's Date of Birth
Mobile Number	VARCHAR(50)	Vender's Contact Number
Address	VARCHAR(50)	Vender's Address
Country ID	VARCHAR(50)	(**Foreign Key**) is used to save Vender's Country details
State ID	VARCHAR(50)	(**Foreign Key**) is used to save Vender's State details
City ID	VARCHAR(50)	(**Foreign Key**) is used to save Vender's city details
Pin Code	INT (10)	Vender's area code
Status	VARCHAR(50)	Activate/Deactivate/Blocked status.
Date	Date	Registration Date

Table 5

❖ **Country ID:-**

In that all country data is saved

Filed	**Data Type**	**Description**
Country Id	VARCHAR(50)	(**Primary Key**) Country Id is save for each country
Country Name	VARCHAR(50)	It save country name

Table 6

❖ **State ID:-**In that all state data is saved

Filed	**Data Type**	**Description**
State ID	VARCHAR(50)	(Primary Key) State Id is saved for each state
State Name	VARCHAR(50)	State names are saved.
Country ID	VARCHAR(50)	(**Foreign Key**)In which Country state comes.

Table 7

❖ **City ID:-**

In that all city data is saved

Field	Data Type	Description
City ID	VARCHAR(50)	(Primary Key) City Id is saved each city details.
City Name	VARCHAR(50)	City name are saved
State ID	VARCHAR(50)	(**Foreign Key**)State details are saved

Table 8

❖ **Car Registration:-**

All information of car are saving in this table, such as car model, car number, car registration number etc.

Filed	Data Type	Description
Car ID	VARCHAR(50)	**(Primary Key)** is used to save car details.
Car Name	VARCHAR(50)	Car Name
Vender Id	VARCHAR(50)	(**Foreign Key**)Car Honor Name
Car Company Name	VARCHAR(50)	Car belong to which Company
Car Registration Number	VARCHAR(50)	Car Unique ID
Car Model Date	Date	Car Model date
Car Type ID	VARCHAR(50)	(**Foreign Key**) is used to Car type like sports car or other
AC type	VARCHAR(50)	Car AC type like AC/ Non AC
Fuel Type	VARCHAR(50)	Car fuel type like petrol or other
Insurance Number	VARCHAR(50)	Car insurance Number
Car Image	Image	Car current Image
Date	Date	Registration date
Status	VARCHAR(50)	Activate/Deactivate/Blocked status.

Table 9

❖ **CAR Type:-**

In that car type details are saved likes sports cars.

Filed	Data Type	Description
Car Type ID	VARCHAR(50)	(**Primary Key**) is used to Car type like sports car or other
Car name	VARCHAR(50)	Car name
Car features	VARCHAR(50)	Car features details are save like sports car etc.

Table 10

❖ **Complaint**

In that all complaint are managed that are uploaded through vender.

Filed	Data Type	Description
Complaint ID	VARCHAR(50)	**(Primary Key)** is used to save Complaints.
Vender ID	VARCHAR(50)	(**Foreign Key**)Vender Id is used
Booking ID	VARCHAR(50)	(**Foreign Key**)Booking Id is used for complaint
Car ID	VARCHAR(50)	(**Foreign Key**)Car ID
Date-Time	DateTime	Current Date and time
Diver ID	VARCHAR(50)	(**Foreign Key**)Diver ID for diver information
Complaint	VARCHAR(MAX)	Complaint is save

Table 11

❖ **Withdrawal**

In that Vender withdrawal amount and request details are saved.

Filed	Data Type	Description
Withdrawal Request ID	VARCHAR(50)	(**Primary Key**) is used to save withdrawal request ID.
Vender ID	VARCHAR(50)	(**Foreign Key**)Vender id is used to take information about Vender

Amount	INT	Amount details are saved that Vender are withdraw.
Bank Details ID	VARCHAR(50)	(**Foreign Key**)In that bank details are saved those are help to withdraw amount directly in bank.
Status	VARCHAR(50)	Withdrawal status

Table 12

❖ **Bank Details**

In that Vender Bank Details are saved for transfer amount in Vender account directly.

Filed	**Data Type**	**Description**
Bank Details ID	VARCHAR(50)	(**Primary Key**) is used to Bank Detail ID
Vender ID	VARCHAR(50)	(**Foreign Key**)Vender details.
Bank Name	VARCHAR(50)	Vender's Bank Name
Bank Account No	VARCHAR(50)	Vender's Account No
Bank IFSC Code	VARCHAR(50)	Vender's Bank IFSC Code

Table 13

Chapter 4

Design

4.1. Modules Used

There are tree Module in this project.

1. Administrator
2. Vendor
3. Customer

1. <u>Administrator</u>

- **<u>Management web portal</u>**

- Login as Administrator
- My Account
- List of Customers
- List of Venders
- Active/Block Customer
- Active/Block Vender
- View Booking History Weekly, Monthly, Yearly by vender
- Payment History
- Accept Withdraw Request from Vender
- View complaints
- View reviews

- **My Account**

- Update profile
- Change password

- **List of Customer**

- Administrator can check list of customers.

- **List of vender**

- Administrator can check list of vender.

- **Active / Block Customer**

- Administrator can Active /Block Customer and Vender.

- **View Booking History**

- Administrator can see Weekly, Monthly & Yearly history.

- **Accept Withdraw Request from Vender**

- Administrator can check withdraw request and pay to vendor

- **View Complaint**

- Administrator can check complaint through customers, venders and drivers.

- **View reviews**

- Administrator can check reviews by customers.

2. <u>Vender</u>

- **<u>Login/signup</u>**

- Vender will be able to login/signup using email id and password.
- Vender can also use their Facebook/google account to login.
- Vender will be accept the trams and condition of the time of signup.
- Vender fill information includes first name, last name, Date of birth, age, email, phone/mobile number, password, conform password, Address.

- **<u>Forgot Password</u>**

- User will able to retrieve their password by clicking on forgot password link and entering their email id/mobile number

- **<u>My Account</u>**

Under My Account menu, vender will be able to browse the following details:

- My profile
- Dashboard
- Booking history
- Car upload
- Uploaded cars

- Sign-out

➢ **My Profile**

Under this Menu vender will be able to browse the following details

- Vender can update their profile, likes email, address, mobile number,
- Vender can able to change their password.

➢ **Dashboard**

Under this menu user can see following menus

- Car upload, delete, update.
- Complaints for driver.
- Withdraw.
- Withdraw history.

➢ **Booking history**

- Vender can see their car's booking history and Current booking status.

➢ **Car upload**

- Vender fill following information.
- Car name & model.
- Car number.
- Car features.
- Insurance number & expiry date.
- Car image more than one.
- Fuel type.

➢ **Uploaded cars**

- Vender can see uploaded cars history.

3. Users Features

- User able to search car for rent using basic details likes pick up place and drop place with date and time

- **Login/signup**

- User will be able to login/signup using email id and password.
- User can also use their Facebook/google account to login.
- User will be accept the trams and condition of the time of signup.
- User fill information includes first name, last name, Date of birth, age, email, phone/mobile number, password, conform password, Address.

- **Forgot Password**

- User will able to retrieve their password by clicking on forgot password link and entering their email id/mobile number

- **My Account**

Under My Account menu, user will be able to browse the following details:

- My profile
- Dashboard
- Booking history
- Feed Back
- Sign-out

- **My Profile**

Under this Menu user will be able to browse the following details

- User can update their profile, likes email, address, mobile number,
- User can able to change their password.

- **Dashboard**

Under this menu user can see following menus

- Car Search& book car
- Car cancel.
- Booking Payment.
- History of booked car.
- Post complaint.

- **Car Search**

- User will be able to search car for travel between to cities.
- User can also search car according to car category.

▪ **Car Cancel**

- In this menu user will able to cancel car booking.

▪ **Booking Payment**

- Under this user will able to pay online an average amount.

- **Post Complaints**
- Under this menu user will able to post complaints to admin.

➢ **Booking History**

- Under this menu user will be able to see their own booking status and booking.

➢ **Feed-Back**

- Under this menu user will able to give feedback to admin.

➢ **Informational Pages**

Users will be able to view information on following pages:

- About Us
- FAQ
- Term of Use
- Privacy Policy

4.2 Internal Data Structures

ASP.NET Web Forms Model

ASP.NET web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.

All client side user activities are forwarded to the server for stateful processing. The server processes the output of the client actions and triggers the reactions.

Now, HTTP is a stateless protocol. ASP.NET framework helps in storing the information regarding the state of the application, which consists of:

- Page state
- Session state

The page state is the state of the client, i.e., the content of various input fields in the web form. The session state is the collective information obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take an example of a shopping cart.

User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown on a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.NET session state and server side infrastructure keeps track of the information collected globally over a session.

The ASP.NET runtime carries the page state to and from the server across page requests while generating ASP.NET runtime codes, and incorporates the state of the server side components in hidden fields.

This way, the server becomes aware of the overall application state and operates in a two-tiered connected way.

The ASP.NET Component Model

The ASP.NET component model provides various building blocks of ASP.NET pages. Basically it is an object model, which describes:

- Server side counterparts of almost all HTML elements or tags, such as <form> and <input>.
- Server controls, which help in developing complex user-interface. For example, the Calendar control or the Grid view control.

ASP.NET is a technology, which works on the .Net framework that contains all web-related functionalities. The .Net framework is made of an object-oriented hierarchy. An ASP.NET web application is made of pages. When a user requests an ASP.NET page, the IIS delegates the processing of the page to the ASP.NET runtime system.

The ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework. Therefore, each ASP.NET page is an object and all its components i.e., the server-side controls are also objects.

Components of .Net Framework 3.5

Before going to the next session on Visual Studio.Net, let us go through at the various components of the .Net framework 3.5. The following table describes the components of the .Net framework 3.5 and the job they perform:

(1) Common Language Runtime or CLR

It performs memory management, exception handling, debugging, security checking, thread execution, code execution, code safety, verification, and compilation. The code that is directly managed by the CLR is called the managed code. When the managed code is compiled, the compiler converts the source code into a CPU independent intermediate language (IL) code. A Just In Time (JIT) compiler compiles the IL code into native code, which is CPU specific.

(2) .Net Framework Class Library

It contains a huge library of reusable types. Classes, interfaces, structures, and enumerated values, which are collectively called types.

(3) Common Language Specification

It contains the specifications for the .Net supported languages and implementation of language integration.

(4) Common Type System

It provides guidelines for declaring, using, and managing types at runtime, and cross-language communication.

(5) Metadata and Assemblies

Metadata is the binary information describing the program, which is either stored in a portable executable file (PE) or in the memory. Assembly is a logical unit consisting of the assembly manifest, type metadata, IL code, and a set of resources like image files.

(6) Windows Forms

Windows Forms contain the graphical representation of any window displayed in the application.

(7) ASP.NET and ASP.NET AJAX

ASP.NET is the web development model and AJAX is an extension of ASP.NET for developing and implementing AJAX functionality. ASP.NET AJAX contains the components that allow the developer to update data on a website without a complete reload of the page.

(8) ADO.NET

It is the technology used for working with data and databases. It provides access to data sources like SQL server, OLE DB, XML etc. The ADO.NET allows connection to data sources for retrieving, manipulating, and updating data.

(9) Windows Workflow Foundation (WF)

It helps in building workflow-based applications in Windows. It contains activities, workflow runtime, workflow designer, and a rules engine.

(10) Windows Presentation Foundation

It provides a separation between the user interface and the business logic. It helps in developing visually stunning interfaces using documents, media, two and three dimensional graphics, animations, and more.

(11) Windows Communication Foundation (WCF)

It is the technology used for building and executing connected systems.

(12) LINQ

It imparts data querying capabilities to .Net languages using a syntax which is similar to the tradition query language SQL.

ASP.NET life cycle specifies, how:

- ASP.NET processes pages to produce dynamic output
- The application and its pages are instantiated and processed
- ASP.NET compiles the pages dynamically

The ASP.NET life cycle could be divided into two groups:

- Application Life Cycle
- Page Life Cycle

ASP.NET Application Life Cycle

The application life cycle has the following stages:

- User makes a request for accessing application resource, a page. Browser sends this request to the web server.
- A unified pipeline receives the first request and the following events take place:
    - An object of the class Application Manager is created.
    - An object of the class Hosting Environment is created to provide information regarding the resources.
    - Top level items in the application are compiled.
- Response objects are created. The application objects such as Http Context, Http Request and Http Response are created and initialized.
- An instance of the Http Application object is created and assigned to the request.
- The request is processed by the Http Application class. Different events are raised by this class for processing the request.

ASP.NET Page Life Cycle

When a page is requested, it is loaded into the server memory, processed, and sent to the browser. Then it is unloaded from the memory. At each of these steps, methods and events are available, which could be overridden according to the need of the application. In other words, you can write your own code to override the default code.

The Page class creates a hierarchical tree of all the controls on the page. All the components on the page, except the directives, are part of this control tree. You can see the control tree by adding trace= "true" to the page directive. We will cover page directives and tracing under 'directives' and 'event handling'.

The page life cycle phases are:

- Initialization
- Instantiation of the controls on the page
- Restoration and maintenance of the state
- Execution of the event handler codes
- Page rendering

Understanding the page cycle helps in writing codes for making some specific thing happen at any stage of the page life cycle. It also helps in writing custom controls and initializing them at right time, populate their properties with view-state data and run control behavior code.

Following are the different stages of an ASP.NET page:

- **Page request** - When ASP.NET gets a page request, it decides whether to parse and compile the page, or there would be a cached version of the page; accordingly the response is sent.
- **Starting of page life cycle** - At this stage, the Request and Response objects are set. If the request is an old request or post back, the Is Post

back property of the page is set to true. The UI Culture property of the page is also set.

- **Page initialization** - At this stage, the controls on the page are assigned unique ID by setting the Unique ID property and the themes are applied. For a new request, post back data is loaded and the control properties are restored to the view-state values.
- **Page load** - At this stage, control properties are set using the view state and control state values.
- **Validation** - Validate method of the validation control is called and on its successful execution, the Is Valid property of the page is set to true.
- **Post back event handling** - If the request is a post back (old request), the related event handler is invoked.
- **Page rendering** - At this stage, view state for the page and all controls are saved. The page calls the Render method for each control and the output of rendering is written to the Output Stream class of the Response property of page.
- **Unload** - The rendered page is sent to the client and page properties, such as Response and Request, are unloaded and all cleanup done.

ASP.NET Page Life Cycle Events

At each stage of the page life cycle, the page raises some events, which could be coded. An event handler is basically a function or subroutine, bound to the event, using declarative attributes such as On click or handle.

Following are the page life cycle events:

- **PreInit** - PreInit is the first event in page life cycle. It checks the Is Post Back property and determines whether the page is a post back. It sets the themes and master pages, creates dynamic controls, and gets and sets

profile property values. This event can be handled by overloading the On PreInit method or creating a Page PreInit handler.

- **Init** - Init event initializes the control property and the control tree is built. This event can be handled by overloading the On Init method or creating a Page Init handler.
- **Init Complete** – Init Complete event allows tracking of view state. All the controls turn on view-state tracking.
- **Load View State** – Load View State event allows loading view state information into the controls.
- **Load Post Data** - During this phase, the contents of all the input fields are defined with the <form> tag are processed.
- **Pre Load** – Pre Load occurs before the post back data is loaded in the controls. This event can be handled by overloading the On Pre Load method or creating a Page Pre Load handler.
- **Load** - The Load event is raised for the page first and then recursively for all child controls. The controls in the control tree are created. This event can be handled by overloading the On Load method or creating a Page Load handler.
- **Load Complete** - The loading process is completed, control event handlers are run, and page validation takes place. This event can be handled by overloading the On Load Complete method or creating a Page Load Complete handler
- **Pre Render** - The Pre Render event occurs just before the output is rendered. By handling this event, pages and controls can perform any updates before the output is rendered.

- **Pre Render Complete** - As the Pre Render event is recursively fired for all child controls, this event ensures the completion of the pre-rendering phase.
- **Save State Complete** - State of control on the page is saved. Personalization, control state and view state information is saved. The HTML markup is generated. This stage can be handled by overriding the Render method or creating a Page Render handler.
- **Unload** - The Unload phase is the last phase of the page life cycle. It raises the Unload event for all controls recursively and lastly for the page itself. Final cleanup is done and all resources and references, such as database connections, are freed. This event can be handled by modifying the On Unload method or creating a Page Unload handler.

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as −

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,

Why SQL?

SQL is widely popular because it offers the following advantages −

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

A Brief History of SQL

- **1970** − Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
- **1974** − Structured Query Language appeared.

- **1978** − IBM worked to develop Codd's ideas and released a product named System/R.
- **1986** − IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

SQL Process

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

There are various components included in this process.

These components are −

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine handles all the non-SQL queries, but a SQL query engine won't handle logical files.

Following is a simple diagram showing the SQL Architecture −

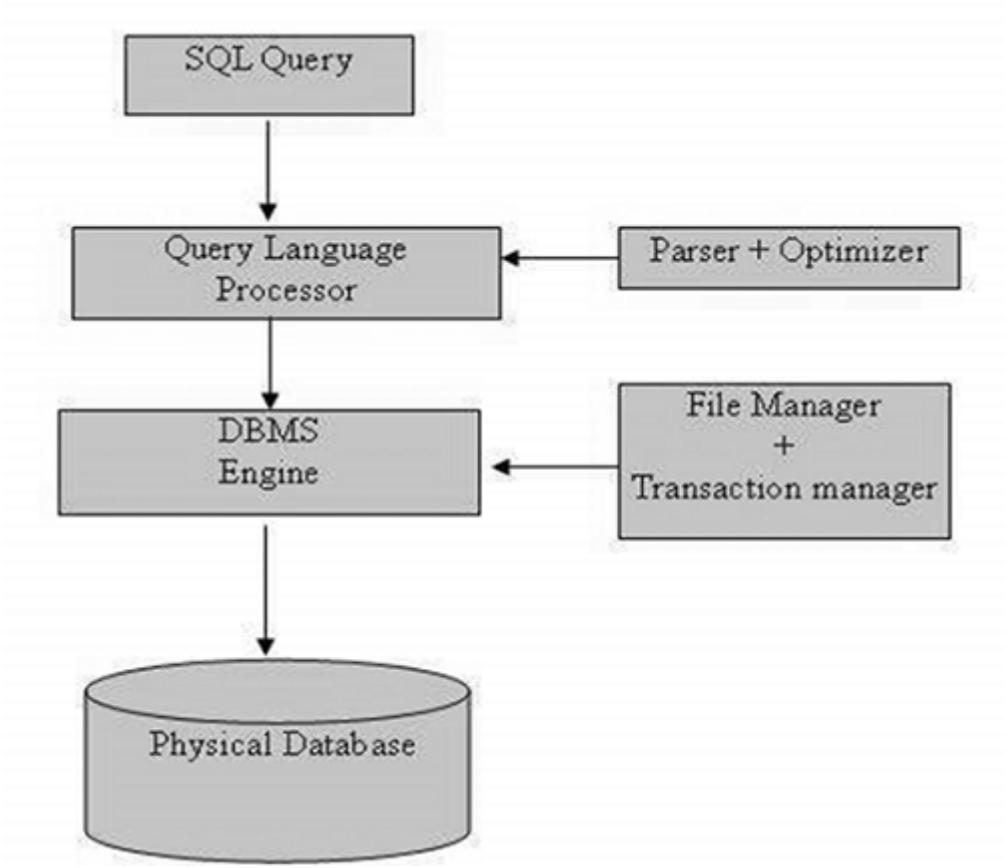

Figure 4.2 Query Processing

SQL Commands

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP.

4.3 Data Design

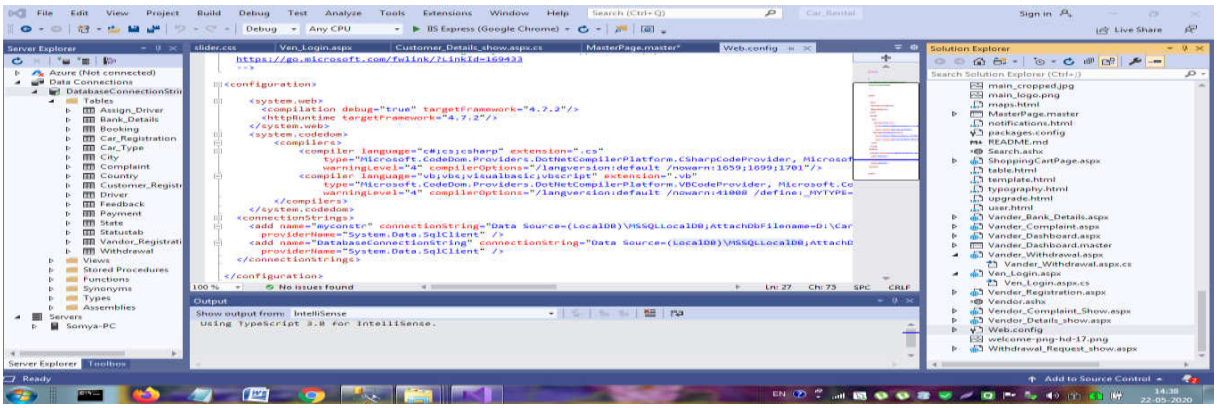

Figure 4.3.1 Code for Project

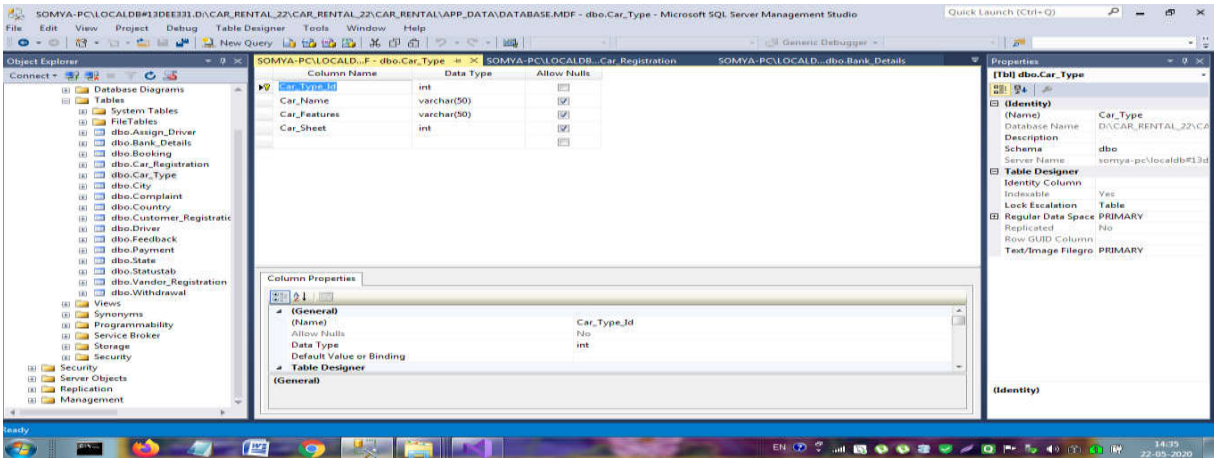

Figure 4.3.2 MSMS

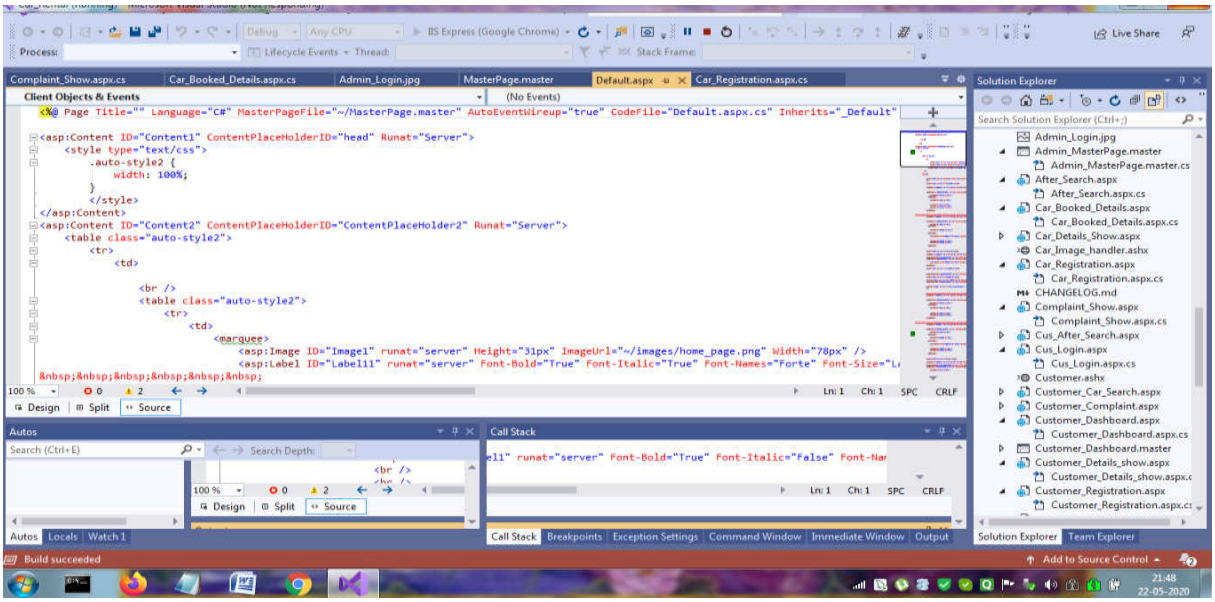

Figure 4.3.3 Code of Default Page

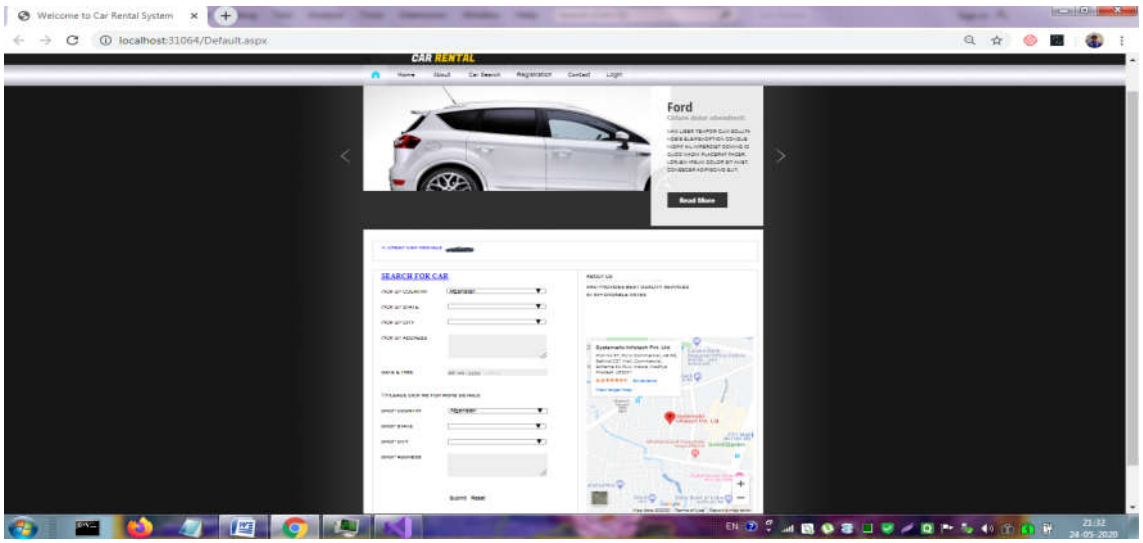

Figure 4.3.4 Home

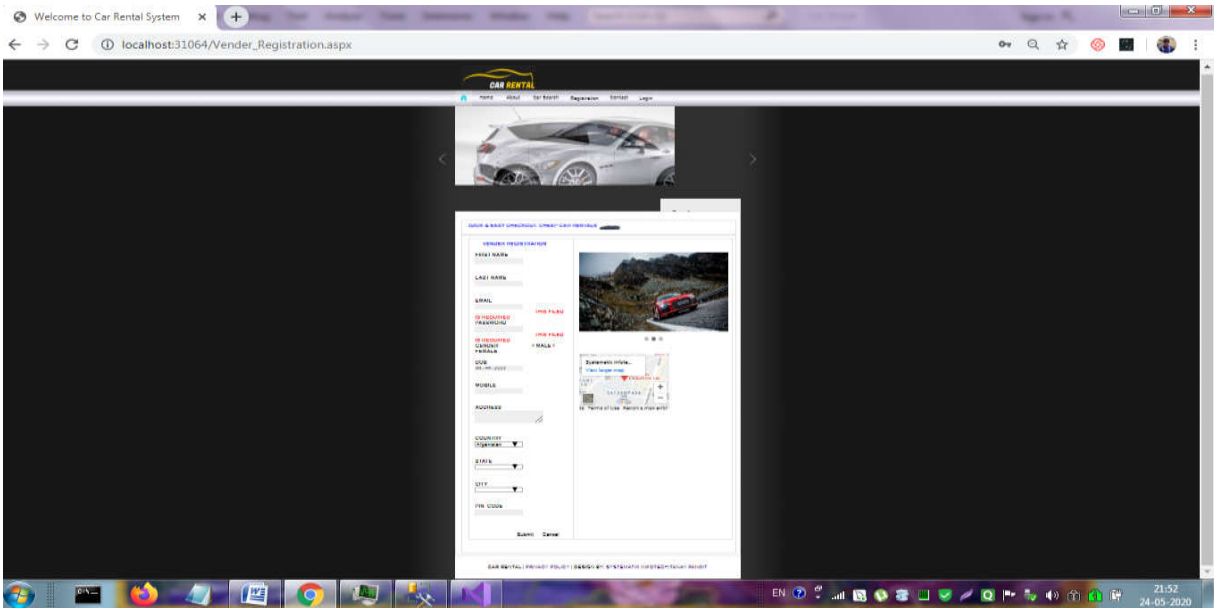

Figure 3.3.5 Vendor Registration

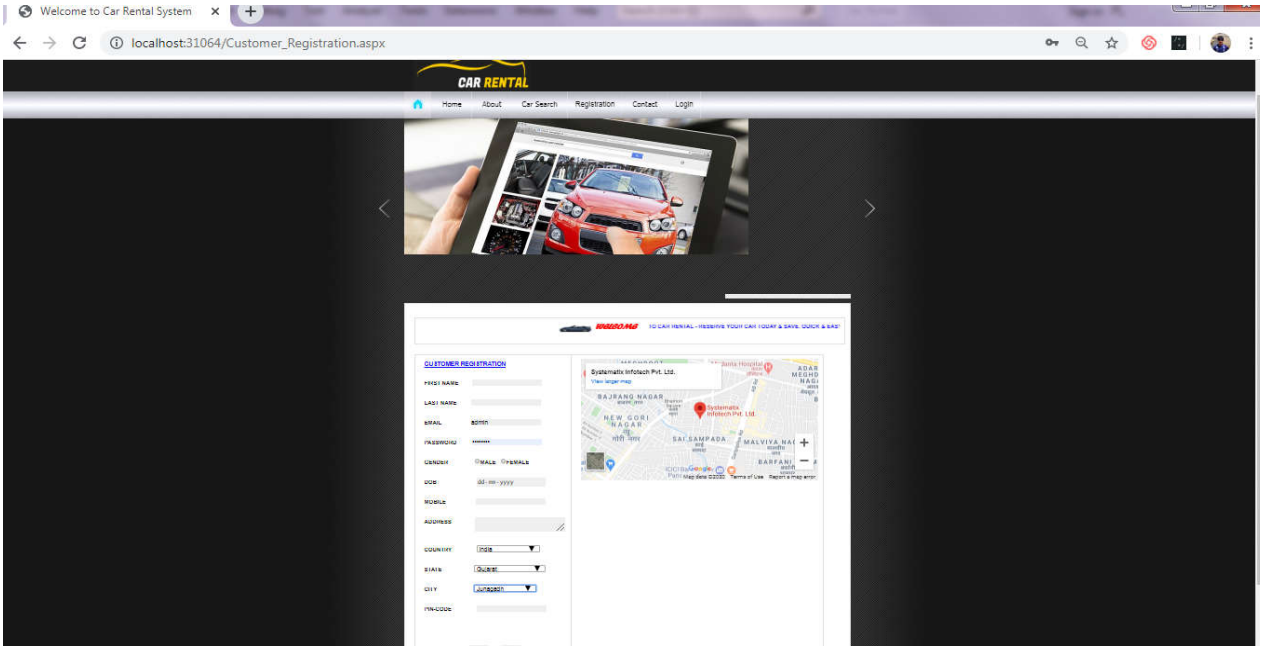

Figure 4.3.6 Customer Registration

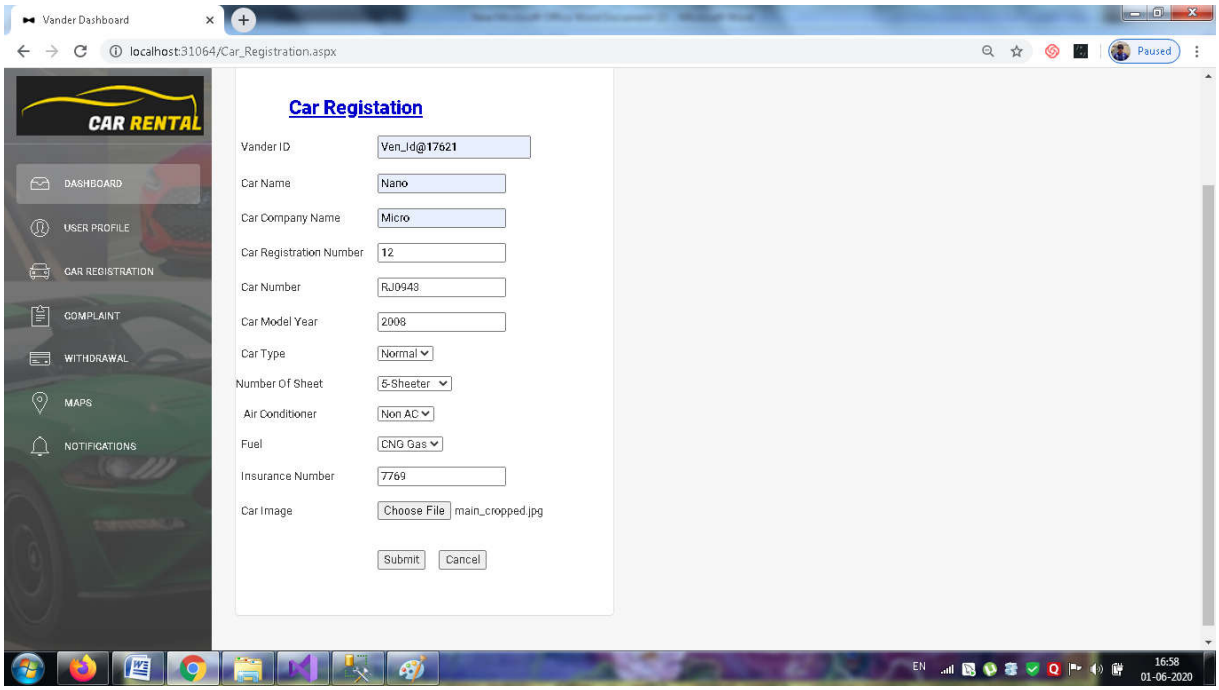

Figure 4.3.6 Vendor Dashboard

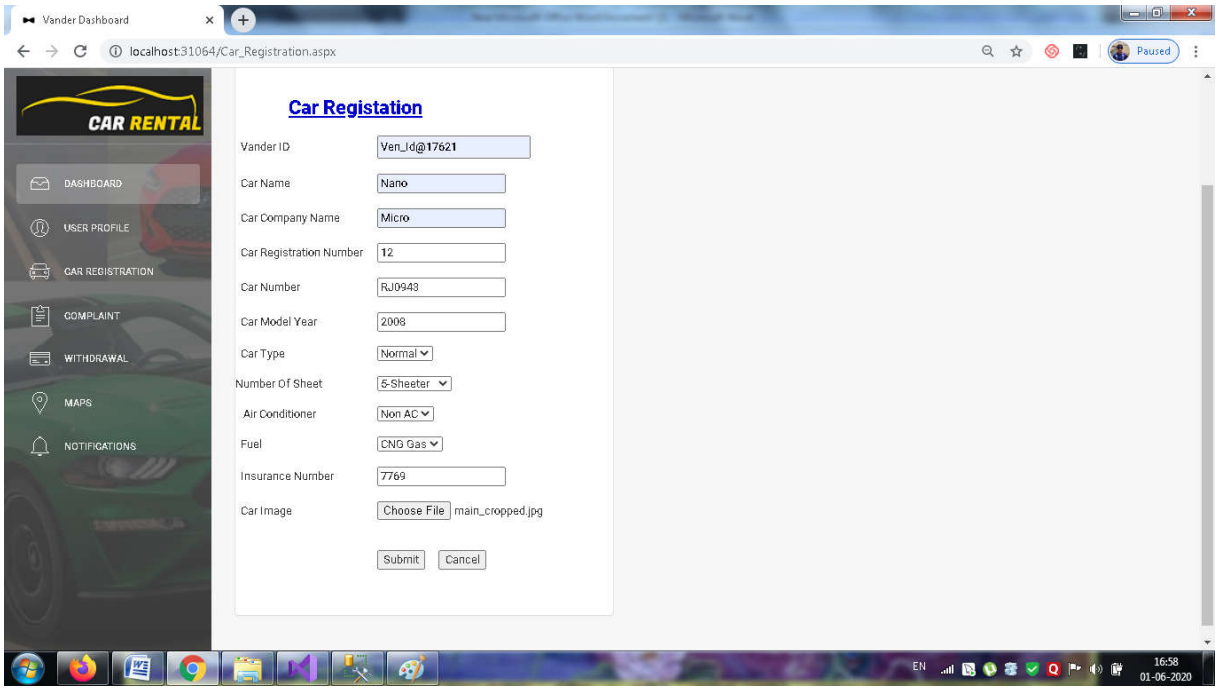

Figure 4.3.7 Car Registration

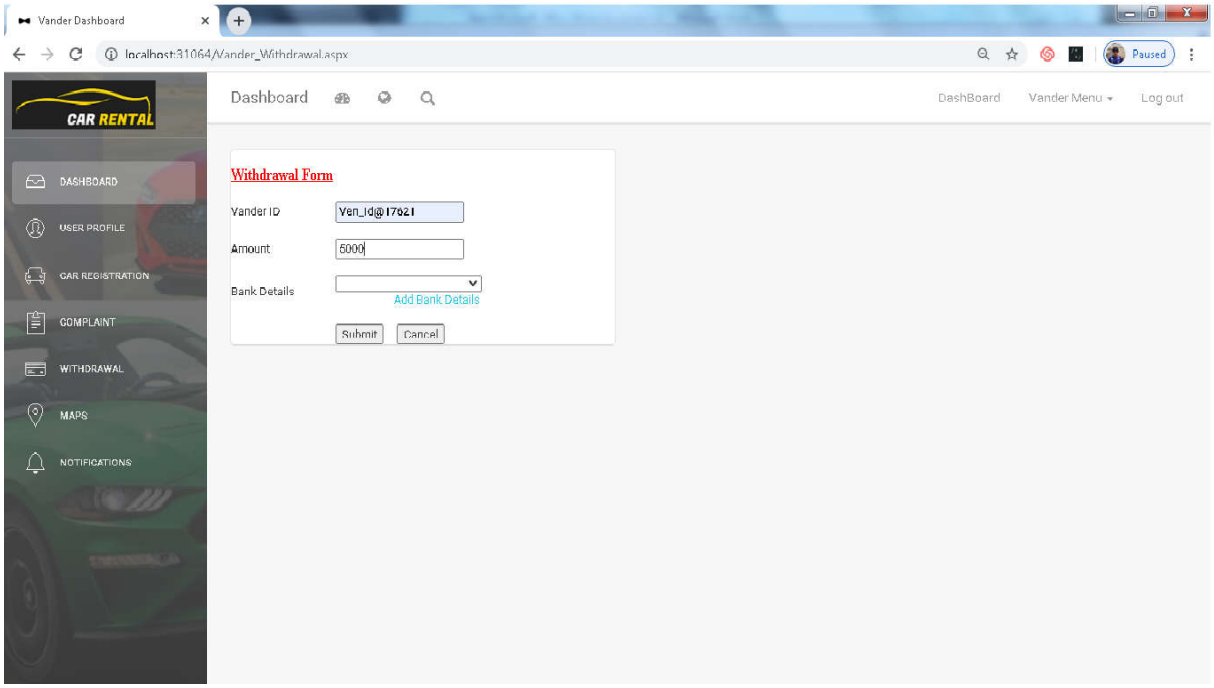

Figure 4.3.8 Withdrawal

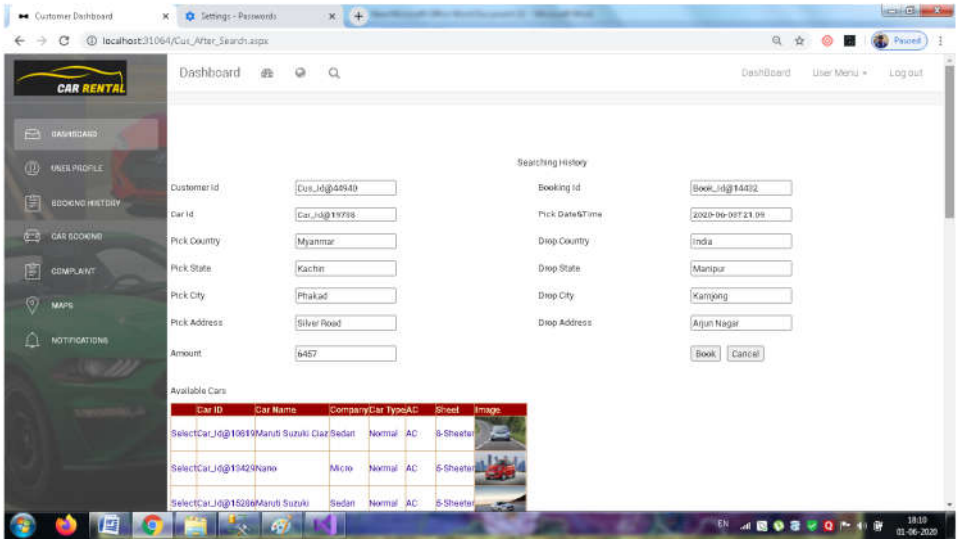

Figure 4.3.9 Car Booking

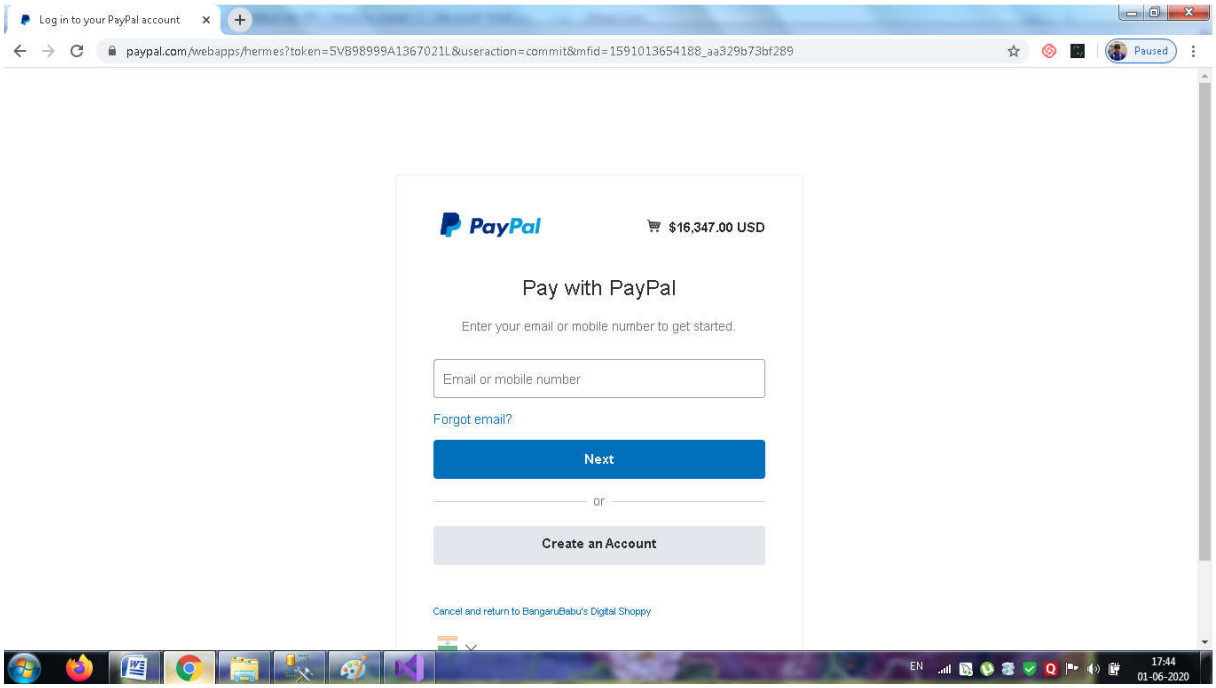

Figure 4.3.4 Payment

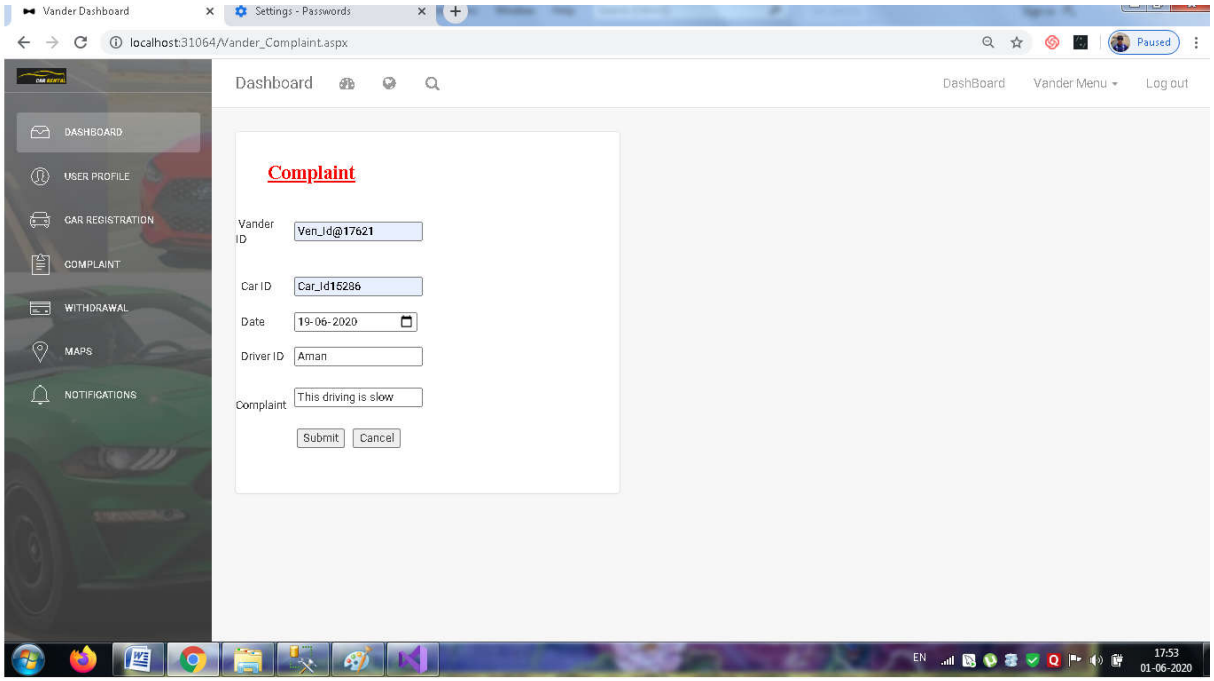

Figure 4.3.5 Complaint

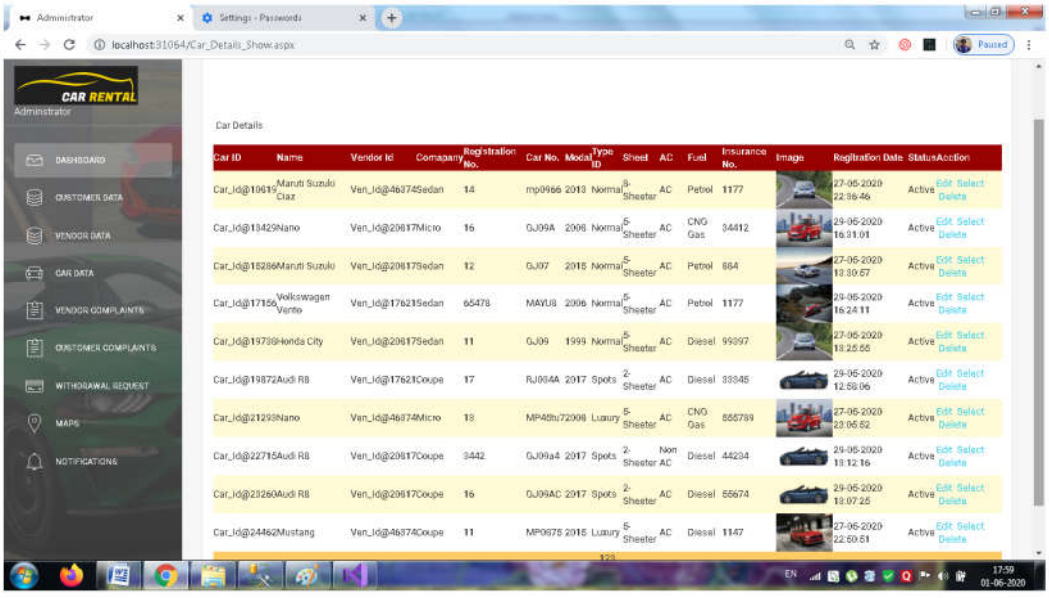

Figure 4.3.62 Car Data of Admin

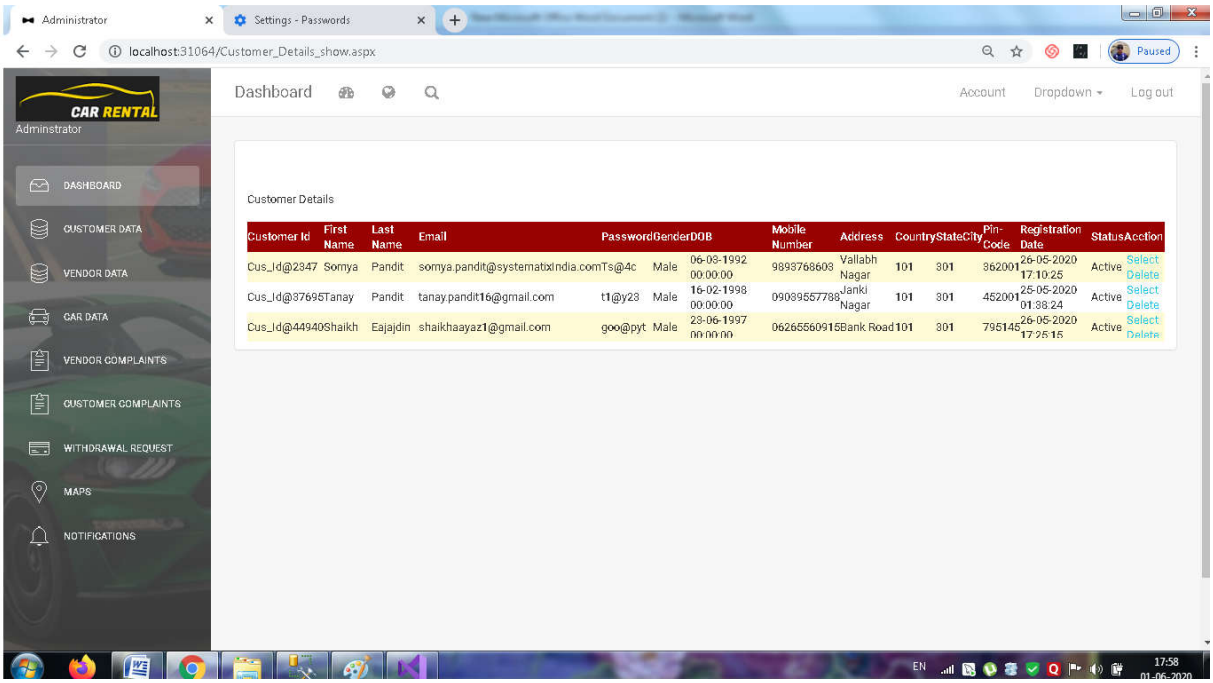

Figure 4.3.73 Customer Data

Chapter 5

Testing

5.1 Testing Objective:

- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an undiscovered error
- A successful test is one that uncovers an as-yet undiscovered error

5.2 Testing Scope:

Testing is the major quality control that can be used during software development. Its basic function is to detect the errors in the software. During requirement analysis and design, the output is a document that is usually textual and non-executable. After the coding phase, computer program is available that can be executed for testing purposes. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during previous phases. Thus the goal of the testing is to uncover requirement, design and coding errors in the program.An elaborate testing of data is prepared and the system is tested using that test date. Errors noted and corrections made during the testing. The corrections are also noted for future use. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully in future. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately before live operation commences.

5.3 Testing Principles:

- All tests should be traceable to customer requirements
- Tests should be planned long before testing begins
- Testing should begin “in the small” and progress toward testing “in the large”
- Exhaustive testing is not completely possible
- To be most effective, testing should be conducted by an independent third party

5.4 Testing Methods:

Software Testing Strategies

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. As important, a software testing strategy provides a road map. Testing is a set of activities that can be planned in advance and conducted systematically.

Various strategies are given below:

- Unit Testing
- Integration Testing
- Validation Testing

Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design of module. This is also known as "Module Testing". Acceptance of package is used for computerization of module. Machine Utilization was prepared and approved by the project leader.

In this testing step, each module is found to be working satisfactory as regards to the expected output from the module. The suggested changes were incorporated into the system. Here each module in the Machine Utilization has been tested.

Integration Testing

After the package is integrated, the user test version of the software was released. This testing consists of testing with live data and various stress tests and result were noted down. Then the corrections were made based on the users feedback. Integration testing is systematic testing for constructing the program structure, while at the same time conducting tests to uncover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. Thus the integration testing step, all the errors uncovered are corrected for the next steps.

Validation Testing

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected, and a final series of software tests - Validation testing - may begin.

Black Box Testing

Black box testing is carried out to check the functionality of the various modules. Although they are designed to uncover errors, black-box tests are used to demonstrate that software functions are operational; that input is properly accepted and output is correctly produced; and that the integrity of external information is maintained. A black-box test examines some fundamental aspect of the system with little regard for the internal logical structure of the software.

White Box Testing

White-box testing of software is predicated on close examination of procedural detail providing the test cases that exercise specific sets of conditions and, loops tests logical paths through the software. White-box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white-box testing methods, following test cases can be derived.

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- The errors that can be encountered while conducting white-box testing are Logic errors and incorrect assumptions.

5.5 Test Cases:

User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at time of development and making changes wherever required.

This is done in regard to the following points:

- Input Screen Design
- On-line Messages to guide the user
- Format of reports and other outputs

After performing all the above tests the system was found to be running successfully according to the user requirements i.e., (constraints)

System Testing

Software is only one element of a larger computer-based system.

Ultimately, software is incorporated with other system elements and a series of system integration and validation tests are conducted. The various types of system testing are:

- Recovery Testing: Many computer-based systems must recover from faults and resume processing within a pre specified time.
- Security Testing: Security testing attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration.

Chapter 6

Limitations

- User can’t be able to access website without internet connection.
- Car Owner Can’t be able to see their car live location.
- User can’t be able to open website without web browser.
- User can’t retrieve password.
- User can’t cancel booked car.

Chapter 7

Future Scope

Every Edition of an book comes with new topics and modifications if any errors are present. In the similar way, in near future ,our application will overcome the flaws if occurred, and attains new features offered to employees for the Flexible and easy Transportation. Following are the Enhancements to the application.

- Providing Good User Interface.
- Providing access permissions to the employees
- Try to Implement the GPS system in the Cabs.
- In future vendor will be able to see car's live location on map.
- In future user will be able to give rating for services.
- In future admin will be able to rating points in data chart.
- In future user gat retrieve their password if he forgot.
- In future user will be able to cancel car

Chapter 8

Learning From the training

Working on this project gave me deep insight about how things move in a professional software firm. While working in an IT company is fun but it equally challenging too.

While working on the .net and c sharp scripts of this project I got plenty of opportunities to improve my coding skills and use my knowledge to solve software engineering problems as well as learn new things that are very useful in any software development life cycle.

This project gave me deep insights about .net framework architecture, MVC architecture and SQL server. I learnt to work under pressure and how breaking up a project in modules helps in fast and accurate development of an application utilizing the talent of each member of the team to the best possible extent and deliver a quality project that meets the demand of the client in the best possible manner.

On the technical front this project taught me a great deal about .net framework and connecting SQL Databse with the application and also how to get confirmation email after successful project run.

Developing .net scripts in the project lessens coding and increases reuse of code.

While the technical learning greatly enhanced my skills I also got to learn to deal with ever changing client demands and to work under the pressure of a deadline. This proved to be a very new and different experience as the exposure to deal with people gave me a lot of confidence to deal with what would happen in a full time corporate job. Dealing with client demands and deadlines also taught me how to use various SDLC models to my benefit.

Chapter 9

Conclusion

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points…

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project.
- The description of Purpose, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.

Chapter 10

References

[1] R. Merchant and S. Saxena, *Engineering.* Chicago, IL: Britannica Educational Publishing, 2016.

[2] R. E. Ziemer and W. H. Tranter, *Principles of Communications: Systems, Modulation*, *and Noise*, 7th ed. Hoboken, NJ: Wiley, 2015, pp. 13–17.

[3] N. B. Vargafik, J. A. Wiebelt, and J. F. Malloy, "Radiative transfer,"

in *Convective Heat*. Melbourne: Engineering Education Australia, 2011, ch. 9, pp. 379–398.

[4] H. C. Hottel and R. Siegel, "Film condensation," in *Handbook of Heat Transfer,* 2nd ed. W. C. McAdams, Ed. New York: McGraw-Hill, 2011, ch. 9, pp. 78–99.

[5] W. M. Rohsenow, "Heat transmission," in *Thermal Radiation Properties,* vol. 3. M. W. Catton and J. P. Hartnett, Eds. New York: Macmillan, 2012, ch. 9, pp. 37–62.