ONLINE CAR PARKING SYSTEM

A PROJECT REPORT

Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

BY

JAISWAR VISHAL KUMAR

SEAT NUMBER: 4027961

Under the esteemed guidance of
Prof. KRUNALI PETKAR
Prof. PRIYA MAURYA
Prof. MUKESH SHARMA

DEPARTMENT OF INFORMATION TECHNOLOGY
SMT. PARMESHWARIDEVI DURGADUTT TIBREWALA
LIONS JUHU COLLEGE
OF ARTS, COMMERE AND SCIENCE
(Affiliated to University of Mumbai)
J.B. NAGAR, ANDHERI (E), MUMBAI-400059, MAHARASHTRA.
Academic Year 2018-19

PROFORMA FOR THE APPROVAL PEOJECT PROPOSAL

PNR No: 2017016402621402 **Roll No:** 343

1. Name of the Student

Vishal Kumar OM Prakash

2. Title of the Project

Online Car Parking System

3. Name of the Guide

Prof. Krunali Petkar & Prof. Mukesh Sharma &Prof. Priya Maurya

4. Teaching experience of the Guide: 04 Years

5. Is this your first submission? Yes □

Signature of the Student

Signature of the Guide

Date: …………………

Date: …………………

Signature of the Coordinator

Date: ……………

SMT. PARMESHWARIDEVI DURGADUTT TIBREWALA LIONS JUHU COLLEGE OF ARTS, COMMERE AND SCIENCE

Affiliated to University of Mumbai

J.B. NAGAR, ANDHERI (E), MUMBAI-400059

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **“ONLINE CAR PARKING SYSTEM”**, is bonafied work of **Mr. Jaiswar Vishal Kumar** bearing **Seat no.: 4027961** submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORATION TECHNOLOGY from University of Mumbai. Examination had not been submitted for any other examination and does not form of any other course undergone by the candidate. It is further certified that he has completed all required phases of the project.

Internal Examiner

External Examiner

Project Guide

Coordinator

College Stamp

ABSTRACT

- People face parking problems in most metropolitan area. Hence this project offers a web-based reservation system.

- where users can view various parking areas and select the space to view whether space is available or not. The difficulty roots from not knowing where the parking spaces are available at the given time, even if this is known; many vehicles may pursue a small number of parking spaces which in turn leads to serious traffic load. Users can even make payment online via credit card.

- After making payment users are notified about the booking via email along with unique parking number.

- Keywords: smart parking, modelling, qr (quick response) code, resource allocation, parking guidance and information (pgi).

- The purpose of Parking System is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same.

- The required software and hardware are easily available and easy to work with.

ACKNOWLEDGEMENT

Before we get into thick of things, we would like to add a few heartfelt words for the people who were part of this project in numerous ways, people who gave unending support right from the stage the project idea was conceived. A project report is such a comprehensive coverage, it would not have been materialized without they help of many.

The four things that go on to make a successful endeavor are dedication, hard word patience and correct guidance. Able and timely guidance not only helps in making an effort fruitful but also transforms the whole process of learning and implementing into an enjoyable experience.

In particular, I would like to thanks our principal **"Dr. (Mrs.) Trishla Mehta"** for her blessing and for being a constant source of inspiration to us and also grateful thank to our coordinator **"Prof.Mukesh Sharma"**. With immense gratitude, I would to give a very special honor and respect to our teacher, **"Prof. Krunali Petkar"** who took keep interest in checking the minute details of project work and guidance us throughout the same.

A sincere quote of thanks to the non-caching for providing us book with all the information we needed for this project, without which the successful completion of this project would not have been possible. I appreciate the outstanding cooperation by the non-teaching staff, especially for the long lab timing that could receive.

Last but not least I wish to avail myself of this opportunity. Express a sense of gratitude and love to my friend and my parents for their manual support, strength and help for everything.

Vishal Kumar

DECLARATION

I hereby declare that the project entitled, "**Online Car Parking System**" done at **SMT.PARMESHWARIDEVEI DURGADUTT TIBERWALA LIONS JUHU COLLEGE OF ARTS, COMMERECE & SCIENCE** of Information Technology, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Vishal Kumar

INDEX

CHAPTER-1

INTRODUCTION

CHAPTER-1
INTRODUCTION

Online Car parking system that is designed to make it easier for people to book parking spaces online. Our online reservation system to reserve parking spaces in the immediate parking, additional services and home purchase will increase your website by enabling customers to pay or go online. As they need, and to set the period of availability can add many types of vehicle seats as Online parking system administrator. It is designed to make it easier for people to book parking spaces online. Availability and prices can add up for a period of several vehicle types as vehicle parking space reservation system administrators as they need. In today parking lots there are no standard system to check for parking spaces.

Searching for a vacant parking space in a metropolitan area is the daily concern for most people and it is time consuming. The system heavily relies on human interaction with the physical space and entity. This leads to wastage of human manpower and also parking spaces at times. It commonly results more traffic load and air pollution in certain area only for an available parking 2 space. Previously, various techniques have been proposed to overcome such problems.

1.1) BACKGROUND

- As the society progresses and as technology develops, cars have become a human necessity. Presently, the automotive industry has grown to the extent that each family would own their own car.
- In addition, with the help from low-interest car loans, the number of cars for private use has increased drastically over the years.
- This prompted several commercial establishments to provide larger parking spaces which will accommodate the needs of the costumer.
- As a result, the management would have to provide more efficient quality service in their parking system.

❖ STATEMENT OF THE PROBLEM

- Present large-scale parking systems have become harder to manage, due to the accuracy, efficiency, and accessibility of the information.
- One error that usually occurs is human error. The employees at the entrance and exit of the parking may have the tendency to record the information incorrectly. This will lead to inaccurate information for the database. Furthermore, the database would be inefficient because of these inaccuracies. This could be solved through the use of cameras that could capture the image and store it. However, it is not advisable to use images since they consume too much memory space leading to more expense.
- Another factor that would affect car park expansions is the storage of information. Storage of a large amount of tickets containing the information about cars entering and leaving the premises is hard to manage. As a result, only the most recent information are filed and the logs of the parking premises. Aside from that, occurrence of natural disasters such as calamities, fires, or earthquakes may cause these files to be damaged or destroyed.
- On the other hand, accessibility of information should still be considered. Large amount of tickets piled up in one location would cause difficulty in searching for specific information. This would be unfortunate for people wanting to gain information about the cars entering and leaving the car park premises because they have to go through so much difficulty in looking for the information needed. In addition, unauthorized personnel may just break inside the office and go through the files without any restrictions.

1.2) OBJECTIVES

The main objective of this module is providing all the functionality related to car.

It tracks all the information of the car.

We have developed all type of CRUD (Create, Read, Update and Delete) operations of the car.

1.2.1) GENERAL OBJECTIVES

- To develop a car park system with automated log in and log out process.

1.2.2) SPECIFIC OBJECTIVES

- To develop the database system that will generate the time logs of cars entering the car park for computation of parking fee.
- To be able to print a ticket that includes a barcode that can be used to access the database.
- To use an LED display that will show the number of slots available per area.
- To obtain a 70% accuracy for the conversion process and 90% for the overall system.
- To develop a sensor system that will check the availability of parking slots.
- Admin can add new Car records Admin can see the list of car details.
- Only admin can edit and update the record of the car.
- Admin will be able to delete the records of the car. All car forms are validated on client side using JavaScript Parking

- **FRONT END: -**
- For this we have used Visual Studio 2010 Professional .
- Browser: Google chrome/ Mozilla Firefox
- Language: JavaScript, asp.net.
- Framework: React.

- **BACKEND: -**
- For this we have used SQL Server 2008 RC2.
- Server: Node.js, asp.net.
- Language: JavaScript, .net.
- Database: SQL Server 2008 RC2.
- Operating System: Windows 98, Windows XP, Windows7, Windows10.
- Browser: Mozilla. Opera, Chrome etc.
- Web Server: Tomcat 7.
- Software Development kit: asp.net

1.3) PURPOSE, SCOPE & APPLICABILITY

1.3.1) PURPOSE:

Traffic congestion is one of the biggest challenges faced in India due to limited parking spaces.

Growing global population and resultant increase in the number of vehicles on road are among the key factors responsible for traffic congestion.

Owing to the necessity of organized parking and reduction in traffic congestion, the concept of parking reservation system has been introduced. The system allows drives to obtain parking availability information

According to **ATCM, July 2014**, result from the survey confirm that drivers use a wide range of criteria to choose where to park. Unsurprisingly, their overriding concern is ‘location’, in other words, proximity of the car park to the amenity or location which represents the very purpose of their trip. Their preference is online car park reservation system which allow them to book a parking place easily where they can easily find a space that comfortably accommodates their vehicle.

The main purpose for developing this project Car Parking System in C# and asp is to managing entry and exits of cars from parking area. This project can also help to manage paid parking facility where the amount of parking gets deducted automatically whenever the card is swiped and the available number of car parking are displayed.

This project is very useful in this modern world, where space has become a very big problem and in the area of miniaturization it's become a very crucial necessity to avoid the wastage of space in modern, big companies and apartments etc. In space where more than 200 cars need to be parked, it's a very tough task to do and also to reduce the wastage of area, this system can be used. It saves our time and money This project provides a lot of features to manage in very well manner.

1.3.2) SCOPE

The scope of this study will only include the standard car plates that use a font color of green and have a format of three letters and three numbers respectively. Plates that use a different font color will not be included in the study. In addition, plates that are covered in glass or plates that are glossy are also not included in the system due to the reflection that they may incur.

The study will also not include specially-made plates such as government plates and euro plates into the system. Furthermore, the study will not include motorcycle plates. Vehicles with specially-made plates that wish to enter the parking system would have to manually enter their plate number through the use of a keyboard.

This study will also be limited to cars that have their plates located on the bumper or on the lower middle side of the car's front part because the camera of the system will be on a fixed position. Plates that are not located on the indicated position would result to the system encountering an error.

In cases wherein the system fails to recognize the plate number, the driver would be asked to input the plate number manually using the keyboard. The keyboard would serve as an alternative solution and compensation for the limitation of the system.

The study will consider testing parameters for checking the efficiency of the system.

The parameters included in our study are:

- number of plates,
- camera angle at the entrance and exit stations of the parking lot,
- car’s position,
- camera’s distance from the car,
- lighting condition of the environment, and
- response time of the system.

The study has delimited the use of the specially made plates, government plates, dirty plates and wet plates, however, the study will use these plates for the purpose of testing the system and check how the system will react from it.

1.3.3) APPLICABILITY

- The Project can be implemented in commercial areas for employee parking.
- The System can be used in public places for public parking like a Shopping Malls, Stations, Restaurants.
- This System can also be used in Universities, Big Colleges, Big Societies.

CHAPTER -2

SURVEY OF TECHNOLOGIES

CHAPTER -2

SURVEY OF TECHNOLOGIES

❖ Technologies Used

➢ **ASP.NET**:

ASP.NET was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

ASP.NET's successor is ASP.NET Core. It is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open-source .NET Compiler Platform (codename "Roslyn") and is cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) have merged into a unified MVC 6.

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, web applications and web services.

- **SQL Server 2008 RC2:**

Microsoft SQL Server 2008 R2 Express with Service Pack 2 is a free, feature-rich edition of SQL Server that is ideal for learning, developing, powering desktop, web & small server applications, and for redistribution by ISVs.

Key Features Offered by SQL Server 2008 R2 SP2 Express:

- Supports stored procedures, triggers, functions, and views
- Store all kinds of business data with native support for relational data, XML, FILESTREAM and spatial data
- Improved performance, usability, visualization, in addition to integration with the Microsoft 2007 Office System in SQL Server Reporting Services
- Simplify development efforts by leveraging existing T-SQL skills, ADO.NET Entity Framework and LINQ
- Closely integrated with Visual Studio and Visual Web Developer

SQL Server 2008 R2 SP2 Express Editions:

- **SQL Server 2008 R2 Express with Tools**

  Core edition of Express that supports development. Includes SQL Server 2008 Database Engine and SQL Server Management Studio Express

- **SQL Server 2008 R2 Express with Advanced Services**

Extends SQL Server 2008 Express with tools to include support for Integrated Full-text Search and Reporting Services

- **SQL Server 2008 R2 Express (Runtime Only)**

  Includes the SQL Database Engine only and is designed for deployments and ISV redistribution

➢ Visual Studio:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

➢ .NET Framework:

.NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability across several programming languages.

.NET Framework executes in a software environment named Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling.

- **StarUML:**

StarUML is an open source software modelling tool that supports UML (Unified Modelling Language). It is based on UML version 1.4, provides eleven different types of diagram and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing to generate code for multiple languages.

CHAPTER-3

REQUIREMENTS AND ANALYSIS

CHAPTER-3

REQUIREMENTS AND ANALYSIS

3.1) PROBLEM DEFINITION

People’s had challenges concerning its safety of data in the store since they currently use paper-based system, physical struggle for parking by drivers, wastage of time, congestion and collision.

There was also a problem of monitoring the profit made for the company where by the company was losing money to its workers who receive the money (fraud).

This system majorly solved the congestion, collision and save time during parking activities.

3.1.1) Existing System

- In Existing System, the exams are done manually but in Proposed System we have to computerize the exams using this application.
- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials.

3.1.2) Proposed System

- The aim of proposed system is to develop a system of improved facilities.
- The proposed system can overcome all the limitations of the existing system.
- The system provides proper security and reduce the manual work.
- Security of data.
- Ensure data accuracies.
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimize time needed for the various processing.
- Greater efficiency.
- Better services.
- User friendliness and interactive.

3.2) REQUIREMENT SPECIFICATION

3.2.1) System Requirement

- For the application to run on computer, the device is expected to meet the following system requirements. The system requirements were categorized into hardware and software requirements as shown in the table-1 and table-2.

- **Table-1: Minimum Hardware Requirements**

Hardware	Minimum Requirement	Reason
Processor speed	1.6GHZ	Accommodate most PCs
Memory of user PC	512MB RAM	Relatively fast
Disk Space of user PC	5GB	Adequate Storage capacity
Memory of server Pc	8GB	Fast
Bandwidth (network connection)	15Mbps	Relatively Good
Disk space of server	200GB	Adequate Storage for database and application

- A fast processor is required because there is need to handle large amounts of data queries. 8 GB memory is required on the server for faster performance because it runs many processes simultaneously, while the memory in the computer should be relatively fast so as to run the processes required.

- 200 GB storage capacity in the server's hard disk is necessary for storage of huge amounts of data while the disk space in a user computer should be modest enough. Good network connection is vital because the application is majorly internet based and there is need for faster communication and retrieval of information.

- **Table-2: Minimum Software Requirements**

Software	**Minimum requirement**	**Reason**
Operating System for computer	Window 10 or Window 7	Globally distributed and widely accessed
Database Management System	Microsoft SQL Server 2008 Express Edition RC2 update	Easy to use and scalable
Browser	Opera, Google Chrome	Standard Browser

- The software requirements for the web application that define the prerequisites needed for the optimal functioning of the web application.
- Each of the following operating system can handle the application, windows 7, windows 10, those operating systems were chosen because they are affordable and readily available. MySQL was used in the development of the databases and is relatively cheap, easy to use and scalable.

- Browsers: any browser including opera, Google chrome, Microsoft Internet explorer.

3.2.2) User Requirements

- The following user requirements were attained:
- The system allows drivers/clients to create accounts on it.
- The system allows the system administrator to book the parking place, printing their receipt, managing the client and parking lot information (such as deleting, updating, adding viewing the client information and viewing different type of parking lot status).
- The system allows drivers to locate and reserve a parking place online through accessing it on web platform.
- The system allows the client and the system administrator to view the parking status (either available or already reserved).
- The system allows authentication of registered users.
- The system is easy to use and learn.
- The system allows the clients to view their account info (booking status) and also to print receipt.

3.2.3) Functional Requirements

- The web application displays the availability of parking lot
- The web application enables employees to set the reaching date and time for the car also the departure date and time.
- The web application enables employees to cancel a parking place.
- The web application enables drivers to book parking place.

3.2.4) Non-Functional requirements

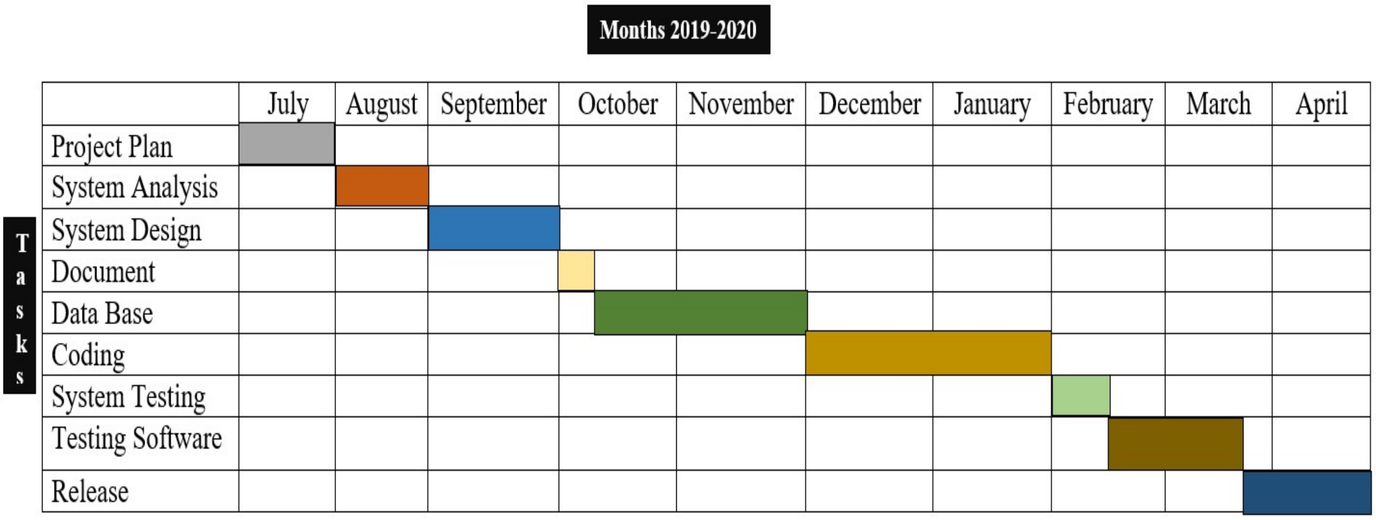
- The designed system should have little or no down time. It should always be up and running.
- The system should have a fast response time. System should not take more than 30 seconds minus loading.
- The system should be secure. User should fill in his/her email address and password so as to be authenticated to the system.
- The system Should allow the customer to park without making a reservation.
- The system should be scalable. Even with an increasing number of users, system should be able to perform effectively.
- The system should be user friendly with ability to show users where they are in the system and guide them on some processes through programmed controls.
- The system should be reliable. In case of system failure, the system should be able to recover quickly and continue working normally.

3.3) PLANNING AND SCHEDULING

- The activities and considerations included in planning and scheduling a project are intended to provide the Project Manager and the project team members with a systematic approach to organizing, defining, scheduling, tracking and managing a project.

3.3.1) Gantt Chart: -

- A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows you to see at a glance.



Tasks	Starting Date	Duration	Completing Date
Project Plan	1-July-2019	31 Days	31-July-2019
System Analysis	1-Aug-2019	31 Days	31-Aug-2019
System Design	01- Sep-2019	30 Days	30-Sep-2019
Document	01- Oct-2019	09 Days	09-Oct-2019
Data Base	10-Oct-2019	51 Days	30-Nov-2019
Coding	01-Dec-2019	62 Days	31-Jan-2020
System Testing	01-Feb-2020	15 Days	15-Feb-2020
Testing Software	16-Feb-2020	33 Days	20-Mar-2020
Release	21-Mar-2020	40 Days	30-April-2020

3.4) SOFTWARE AND HARDWARE REQUIREMENTS

3.4.1) Software Requirements

❖ **FRONT END: -**

For this we have used Visual Studio 2010.

➢	Browser	:	Google chrome/ Mozilla Firefox
➢	Language	:	JavaScript, ASP.NET.
➢	Framework	:	React.

❖ **BACKEND: -**

For this we have used Microsoft SQL Server 2008 Express Edition RC2 update.

➢	Server	:	Node.js, ASP.NET.
➢	Language	:	JavaScript, .NET.
➢	Database	:	Microsoft SQL Server 2008 Express Edition RC2.
➢	Operating System	:	Windows 98, Windows XP, Windows7, Linux.
➢	Browser	:	Mozilla. Opera, Chrome etc.
➢	Web server	:	Tomcat 7.
	Software Development kit	:	Java JDK 1.7, ASP.NET

❖ WINDOWS 10:

- Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Wind family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and broadly released for retail sale on July 29, 2015.
- Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10 which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

- Windows 10 makes its user experience and functionality more consistent between different classes of device, and addresses shortcomings in the user interface that were introduced in Windows 8. Windows 10 also allows web apps and desktop software (using either Win32 or .NET Framework) to be packaged for distribution on Microsoft Store.

- To reduce the storage footprint of the operating system, Windows 10 automatically compresses system files. The system can reduce the storage footprint of Windows by approximately 1.5 GB for 32-bit systems and 2.6 GB for 64-bit systems. The level of compression used is dependent on a performance assessment performed during installations or by OEMs.

- **ASP.NET**:

- ASP.NET was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

- ASP.NET's successor is ASP.NET Core. It is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open-source .NET Compiler Platform (codename "Roslyn") and is cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) have merged into a unified MVC 6.

- 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, web applications and web services.

- ASP.NET supports a number of programming models for building web applications: -

- **ASP.NET Web Forms:** A framework for building modular pages out of components, with UI events being processed server-side.
- **ASP.NET MVC:** Allows for building web pages using the model–view–controller design pattern.

- **ASP.NET Web Pages:** A lightweight syntax for adding dynamic code and data access directly inside HTML markup.
- **ASP.NET Web API:** A framework for building Web APIs on top of the .NET Framework.
- **ASP.NET Webhooks:** Implements the Webhook pattern for subscribing to and publishing events via HTTP.
- **Signal R**: A real-time communications framework for bi-directional communication between client and server.
- **ASP.NET Handler:** Are components that implement the System. Web. HTTP Handler interface. Unlike ASP.NET Pages, they have no HTML markup file, no events and other supporting. All they have is a code-file (written in any .NET-compatible language) that writes some data to the server HTTP response. HTTP handlers are similar to ISAPI extensions.
- **ASP.NET AJAX:** An extension with both client-side as well as server-side components for writing ASP.NET pages that incorporate Ajax functionality.
- **ASP.NET Dynamic Data:** A scaffolding extension to build data driven web applications.

- **Features of ASP.NET:**

- We will be using ASP.NET technology for developing our system which will allow us to develop dynamic web-application and rich user interface.
- The framework technology reduces coding time. (If you are building large web applications, you’re definitely going to appreciate this framework technology because it has the ability to reduce the amount of time needed to code.)

- The applications that you build on this framework are secure.

  (In truth, because of the per-application configuration and built-in Windows

  authentication, your programs have never been more safe and secure. And this is a huge advantage if you’re worried about potential security issues in the future.)

- The framework for ASP.NET has a complementary design and rich toolbox in the form of Visual Studio. (Some excellent features include automatic deployment, WYSIWYG editing, and drag and drop server controls to name just a few of the amazing features provided by this incredible tool).

- ASP.NET provides continuous monitoring. (Another reason to love this framework is because of the constant and continuous monitoring. It will continue to monitor applications, components, and pages that it is running. And even better, if he programs happens to notice that there are illegal activities taking place like infinite loops and memory leaks, it will immediately destroy all of these activities that shouldn’t be happening and then it will actually restart itself.)

- Deployment is easier than ever with ASP.NET. (For starters, mainly, the biggest reason why deployment is much easier is because you

  no longer need to register components. Instead, the configuration info is already built into the system, which is going to make your life a heck of a lot easier.)

- ASP.NET is server-side technology. (Well, since this technology is purely server-side, the code has the ability to execute on the server. This is good because it actually executes before it’s actually sent to the browser.)

- **.NET Framework:**

- .NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability across several programming languages.
- .NET Framework executes in a software environment named Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling.
- Programmers produce software by combining their source code with .NET Framework and other libraries. The framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.
- People prefer searching in internet for different NET Framework led to a family of .NET platforms targeting mobile computing, embedded devices, alternative operating systems, and web browser plug-ins. A reduced version of the framework, .NET Compact Framework, is available on Windows CE platforms, including Windows Mobile devices such as smartphones.
- .NET Micro Framework is targeted at very resource-constrained embedded devices. Silverlight was available as a web browser plugin. Mono is available for many operating systems and is customized into popular smartphone operating systems (Android and iOS) and game engines. .NET Core targets the Universal Windows Platform (UWP), and cross platform and cloud computing workloads.

❖ SQL EXPRESS:

- Microsoft SQL Server Express is a version of Microsoft's SQL Server relational database management system that is free to download, distribute and use. It comprises a database specifically targeted for embedded and smaller-scale applications. SQL
  Server Express is a free version of Microsoft’s primary relational database management system (RDBMS) – the SQL Server. Essentially, the SQL Server is a database management system that can be used to store and access the information stored in many different databases. SQL Server comes with an impressive range of features like business intelligence, reporting, and in-depth advanced analytics.

- The enterprise edition of SQL Server competes against enterprise-oriented systems like Oracle Database (DB) and MySQL. SQL Server Enterprise comes loaded with features and can be too expensive for smaller sized companies to maintain. SQL Server Express is the most basic offering available. It is a full database engine you can deploy to a server or embed into an application. Express is free and comes with many of the same features as the enterprise edition. SQL Server Express is probably most suited to supporting production applications for smaller to midsize businesses. typical SQL Server Express use case would be a deployment by developers who do not want to create applications with a database hosted on a server. Using Express, they would be able to develop apps with their SQL Server database.

- **SQL Server Express Benefits:**

➔ Some benefits come with an SQL Server Express deployment: -

- **Free**: One huge advantage of SQL Server Express is that it is free. Your only outlay is the time investment you make downloading and setting up the system. If you only want to learn how to use SQL Server, then Express is for you. There is nothing to lose by downloading the system and getting used to how it works.
- **Scalability**: SQL Server Express is an ideal starting point for smaller independent software vendors (ISVs) since it can be used with any smaller application. The licensing allows Express to be included as part of an app or product. While there are limitations on memory and socket usage, they are not as restrictive as some might think. Express is not limited to a single user which is a commonly held misconception. There is a 10GB database restriction, but that is a maximum size per database meaning you can have multiple databases that store up to 10GB of data. If you are an ISV and your company experiences high growth rate resulting in increased database demands, then you can only upgrade to a paid version of SQL Server.
- **Security:** Within SQL Server Express there is the option of free online backup that will help to protect your valuable business data if anything goes wrong. Administrators should still follow security best practices like restricting access to backup folders and following Windows password policies.

- **CSS:**

- Cascading Style Sheets (CSS) is a stylesheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

  CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content.

- Separation of formatting and content also makes it feasible to present the same mark-up page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech- based browser or screen reader), and on Braille-based tactile devices.

- CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

❖ **C#:**

➢ C# is pronounced as "C-Sharp". It is an object-oriented programming language provided by Microsoft that runs on .Net Framework. By the help of C# programming language, we can develop different types of secured and robust applications: -

1) Window applications.
2) Web applications.
3) Distributed applications.
4) Web service applications.
5) Database applications etc.

➢ C# is approved as a standard by ECMA and ISO. C# is designed for CLI (Common Language Infrastructure). CLI is a specification that describes executable code and runtime environment. C# programming language is influenced by C++, Java, Eiffel, Modula-3, Pascal etc.

❖ **Visual Studio:**

➢ Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

➢ Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level - including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

- Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language- specific service exists. Built-in languages include: -
    - C/C++
    - Visual Basic.Net
    - C#
    - F#
    - Java Script
    - Type-Script
    - XML
    - XSLT
    - HTML
    - CSS
    - JSP

3.4.2) Hardware Requirements

Processor	INTEL (core i3)
RAM	4GB
HARD DISK	8GB
NETWORK	LAN
MONITOR	15” COLOR MONITOR
KEYBOARD	108 KEYS
MOUSE	OPTICAL MOUSE

3.5) Preliminary Product Description

Preliminary study is problem solving activity that requires intensive communication between the system user and system developers. It does various feasibility studies. In these studies, a rough figure of system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Analysis is a detailed of the various operation performed by the system and the relationship exist between the system. In our project we analyzed the relationship that we will be using in our project. In analysis, we also decided how many modules we will be including in our project.

Also, the brief overview of how our project will look like was done in the analyses. We also decided what type of connectivity we will be providing in our project was done. Also, we referred some of the books of VB so that which will be useful for writing the code of the project. In analysis, also we analyses that in how many days we will be completing our project so that it will be submitted in correct time as per the given schedule.

The cost of the project is less as compared to the other project. Also, it is cheaper as compared to another project. Here only software is used so there is no need of extra cost that the project might be damaged.

It is user friendly and can be interacted with another person. Also, it can be used by any person if that person knows VB language then can be easily handled. Also, there is no need of having any high range pc it can work with normal configuration pc.

❖ Feasibility Study: -

Feasibility Study is the test of the system proposal according to its work ability, impact on the current system, ability to meet the needs of the current users and effective use of the resources.

Its main objective is not to solve the problem, but to acquire its scope. It focuses on following:

- Meet user requirements
- Best utilization of available resources
- Develop a cost-effective system
- Develop a technically feasible system

➢ There are three aspects in the feasibility study:

- Technical Feasibility
- Economical Feasibility
- Operational Feasibility

❖ Technical Feasibility: -

➢ The technical issues usually raised during the feasibility stage of the investigation include following:

- The necessary technology must be existed to do what is suggested.
- The proposed equipment must have the technical capacity to hold the data required to use the new system.
- There must be technical guarantees of accuracy, reliability ease of access and data security.
- All the functional and non-functional requirements of our Online Learning System is developed in the J2EE (NetBeans 7.0/ASP.NET as the front-end and MySQL as back-end).
- The development of the system is technically feasible as the various technological needs for the development and deployment are fulfilled.

❖ Economical Feasibility: -

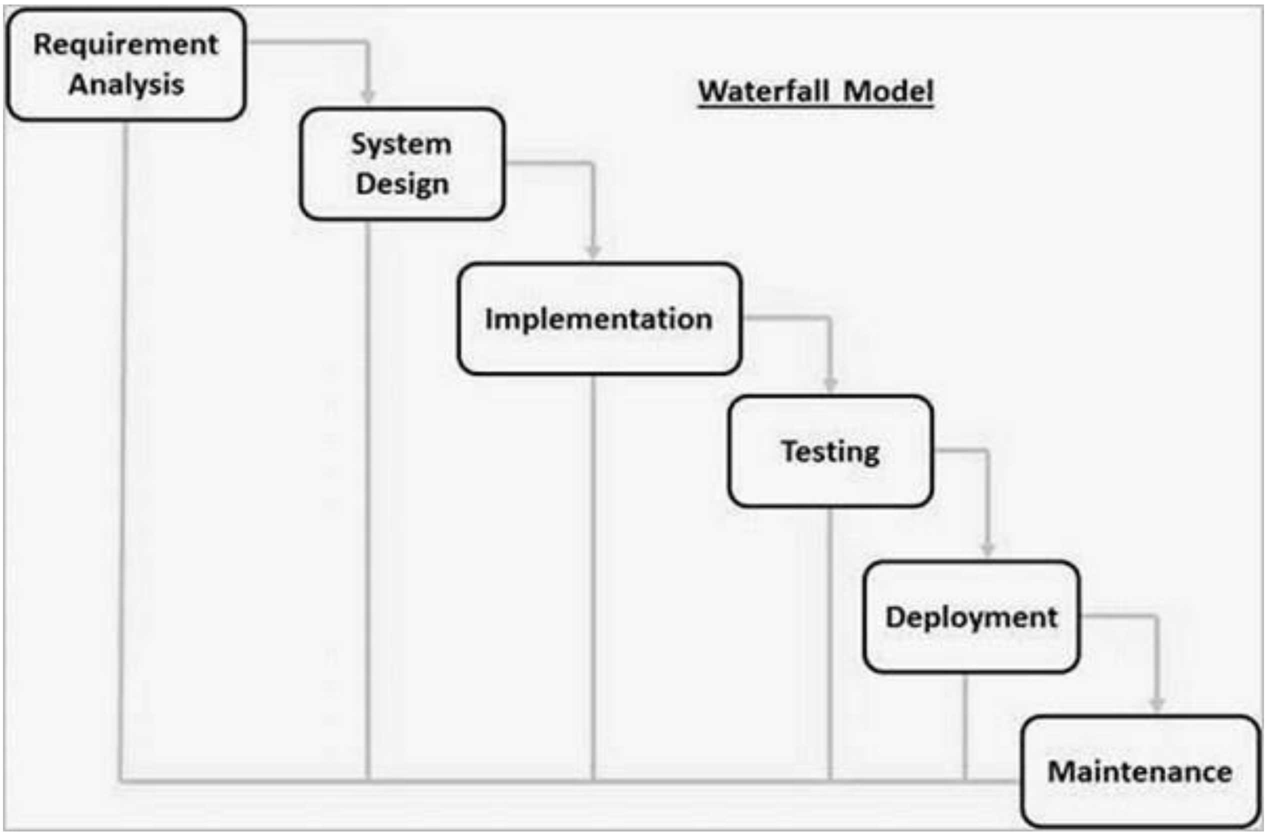
- Issues to be studied are, whether the new system is cost effective or not? The benefits in the form of reduced cost?
- This Web based Virtual Education is economically feasible. As their hardware was installed from quite beginning, the cost on project of hardware is low.
- Similarly, the software loaded for this project was used for many other applications. The software cost was under budget.
- As student trainees were developing the application, there were no major personnel costs associated. Moreover, the technical requirements were already available so there was no further expenditure for buying software packages.
- Here, this system is beneficial for the organization as work cost will be decreased by developing the project.

3.6) CONCEPTUAL MODELS:

3.6.1) WATERFALL MODEL:

- Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

- The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

❖ Waterfall Model - Advantages

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables
- and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.

❖ Waterfall Model - Disadvantages

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long projects.

CHAPTER- 4
SYSTEM DESIGN

CHAPTER-4

SYSTEM DESIGN

- In this phase, a logical system is built which fulfills the given requirements.
- Design phase of software development deals with transforming the client's requirements into a logically working system.
- Normally design is performed in the following two steps.

1) Primary Design Phase:

- In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

2) Secondary Design Phase:

- In the secondary phase the detailed design of every block is performed.

➔ **The general tasks involved in the design process are the following:**

- Design various blocks for overall system processes.
- Design smaller, compact and workable modules in each block.
- Design various database structures.
- Specify details of program to achieve desired functionality.
- Design the form of inputs and outputs of the system.
- Perform documentation of the design.
- System reviews.

4.1) BASIC MODULES

4.1.1) Admin Module:

- Guard will see reserved slots only for the today/current date.
- Once any vehicle leaves from parking lot, guard can vacate that slot.
- Once guard will vacate slot, it will become available for new user to book that slot.

4.1.2) User Module:

- User can make reservation on site signing up and logging into the site. Once user reserve the parking slot for the desired date.
- User will get confirmation email from site confirming reservation is complete successfully. Once user book the slot for particular date it will become unavailable for other users. Means no other user can book that slot for that particular date.

4.1.3) Payment Module:

- The payment module consists of payment gateway which facilities the users to make payment for slot confirmation.

4.1.4) Car Module:

- The main objective of this module is providing all the functionality related to car. It tracks all the information of the car. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the car.

❖ Features of Car Module:

- Admin can add new Car records
- Admin can see the list of car details
- Only admin can edit and update the record of the car
- Admin will be able to delete the records of the car
- All car forms are validated on client side using JavaScript

4.1.5) Parking Module:

The main aim for developing this module is to man- age the parking. This parking is the important module in this project Online Car Parking System which has been developed on C #, ASP and MySQL. So, all parking will be managed by admin.

❖ Features of Parking Module:

- Admin can manage the parking
- Admin can edit/delete the parking
- Admin can see the list of all parking
- User can see his/her parking

4.1.6) Parking Slots Module:

The main objective for developing this module is to manage the parking slots. So, all parking slots will be managed by admin. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the parking slots.

✓ Features of parking slots Module:

- Admin can manage the parking slots
- Admin can edit/delete the parking slots
- Admin can see the list of all parking slots
- User can see his/her parking slots

4.2) DATA DESIGN: -

• A database is a system intended to organize, store, and retrieve large amounts of data easily. It consists of an organized collection of data for one or more uses, typically in digital form.

• Database design is the process of producing a detailed data model of a database.

• This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database.

• A fully attributed data model contains detailed attributes for each entity

• Data design is the first design activity, which results in less complex, modular and efficient program structure.

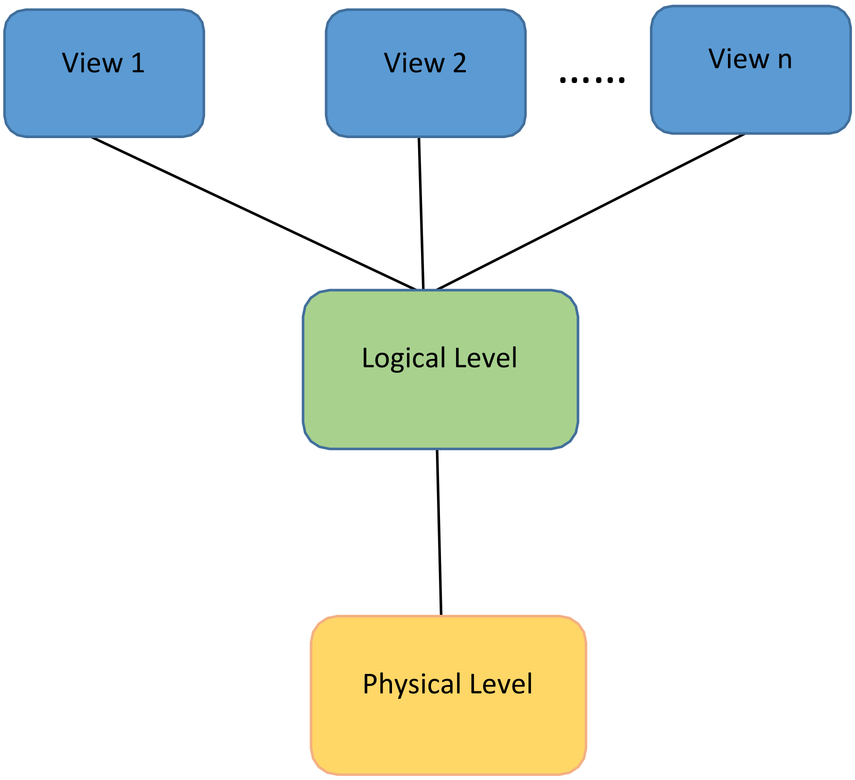
• The information domain model developed during analysis phase is transformed into data structures needed for implementing the software

4.2.1) SCHEMA DESIGN

A schema can be defined as the design of a database.

The overall description of the database is called the database schema. It can be categorized into three parts. These are:

- Physical Schema
- Logical Schema
- View Schema



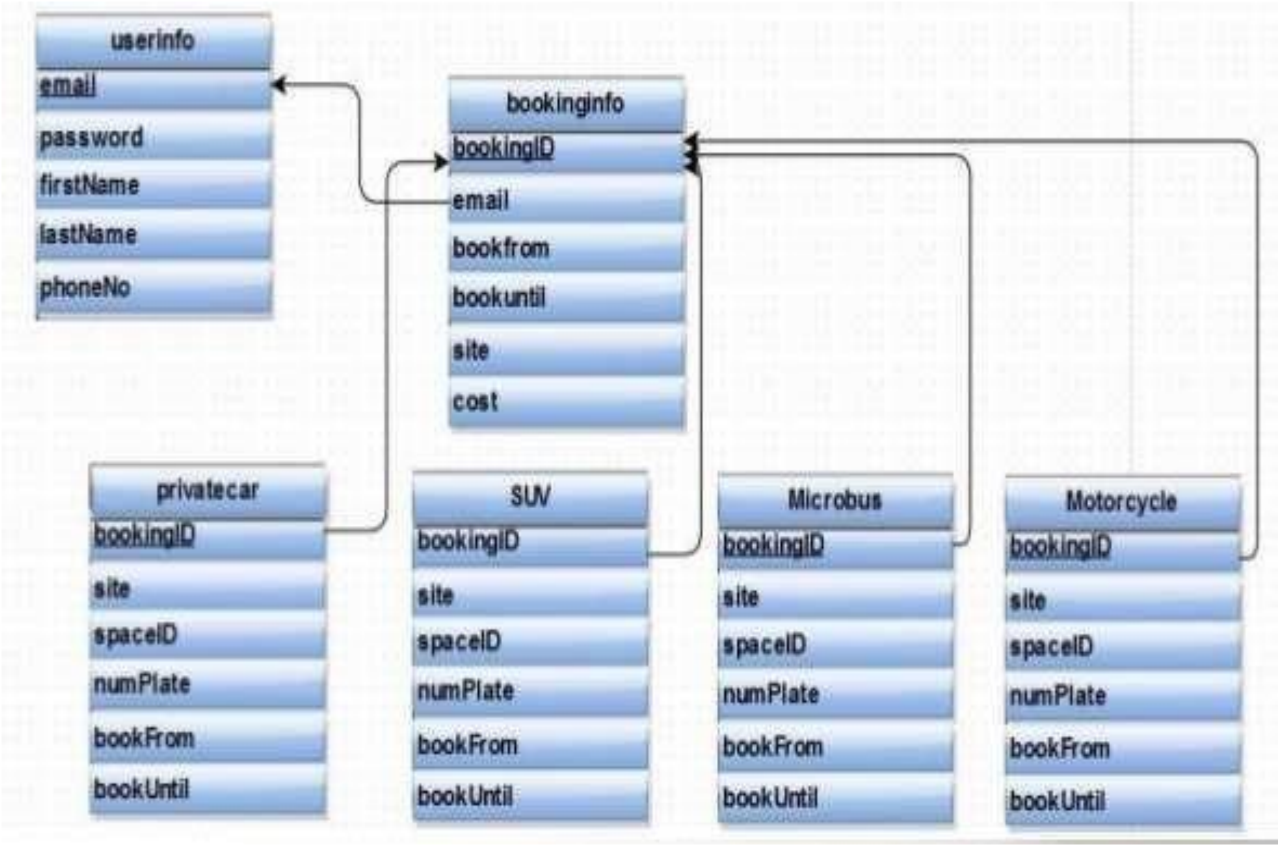
A physical schema can be defined as the design of a database at its physical level. In this level, it is expressed how data is stored in blocks of storage.

A logical schema can be defined as the design of the database at its logical level.

In this level, the programmers as well as the database administrator (DBA) work.

At this level, data can be described as certain types of data records which can be stored in the form of data structures. However, the internal details (such as an implementation of data structure) will be remaining hidden at this level.

View schema can be defined as the design of the database at view level which generally describes end-user interaction with database systems.



4.2.2) DATA ITEGRITY AND CONSTRAINTS

- **The Primary-Key Constraint: -**

The primary key constraint designates a column, or a combination of columns, as the primary for the table. This action enforces entity integrity, which requires each row to have a unique identifier, so that data modifications or queries always refer to a specific row without ambiguity. When you place a primary key constraint on a column, you're requiring each row in that column to have a unique value, which can't be NULL. A primary key constraint on multiple columns requires the combination of the values in those columns to be unique. Examples of primary key include invoice number, employee ID, purchase order number and item or part number.

- **The Foreign-Key Constraint: -**

The Foreign-key constraints defines the relationship between a column or combination of columns in the current table and a column or combination of columns in another table. In other words, it enforces referential integrity. This relationship might be one to one, such as in the case of an employee in the payroll table who must already exist in the employees table. Or it could be a many to one relationship. A typical example too many to one foreign-key relationship is the Customer ID in the voices table.

Below are the tables contained in the Online Parking system database:

Table-1: Describe the Parking Table

Column name	**Data type**	**Description**	**Size**	**Allow null value**
Park_id	Int	Primary key for the table	15	no
Park	Text	Name of the parking	Default	no
Plot	Text	Name the parking space number	Default	no
Num_plate	Int	Indicate the vehicle number plate	9	No
Payment	Float	Indicate the amount paid	Default	No
Time_A	Timestamp	Date and time of the client arrive in parking space	Default	No
Time_D	Timestamp	Date and time of the client departure from parking lot	Default	No

Table-2: Describe the User Table

Column name	**Data Type**	**Description**	**Size**	**Allow null value**
User_id	Int	Primary key for the user	20	no
Fname	Text	The first name and the last name of the user	default	no
Email	Text	The email of the user	default	no
Password	Text	Password of the user	default	no
Phone_num	Int	The phone number of the user	12	no
Gender	Varchar	The gender of the user	6	no

Table-3: Describe the Admin Table

Column name	Data Type	Description	Size	Allow null value
Admin_id	Int	Primary key for the admin	12	no
User_name	Text	Username of the admin	default	no
Password	Text	Password of the admin	default	no
Phone_num	Int	The phone number of the admin	12	no
DOB	Date	Date of Birth of the admin	default	no

Table-4: Describe the Contact Table

Column name	Data Type	Description	Size	Allow null value
Contact_id	Int	Primary key for the contact	12	no
Name	Text	The name of the client or the user who want to send a feedback.	default	no
Email	Text	The email of the user who want to send a feedback.	default	no
Purpose	Text	The reason of sending a feedback	Default	no
Message	Text	The message which needed to be delivered	default	no

4.2.3) ENTITY RELATIONSHIP DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of the training, Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

Connectivity and Cardinality

The basic types of connectivity for the relations are: one-to-one, one-to-many and many-to-many. A **one-to-one** (1:1) relationship is when at most instance of an entity A is associated with one instance of entity B. For example, “employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A **one-to-many** (1: N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1: N relationships is a department has many employees each employee is assigned to one department.

A **many-to-many** (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one or many instances of entity B and for one instance of the entity B there are zero, one or many instances of entity A. the connectivity of the relationship describes the mapping of associated.

ER Notation: -

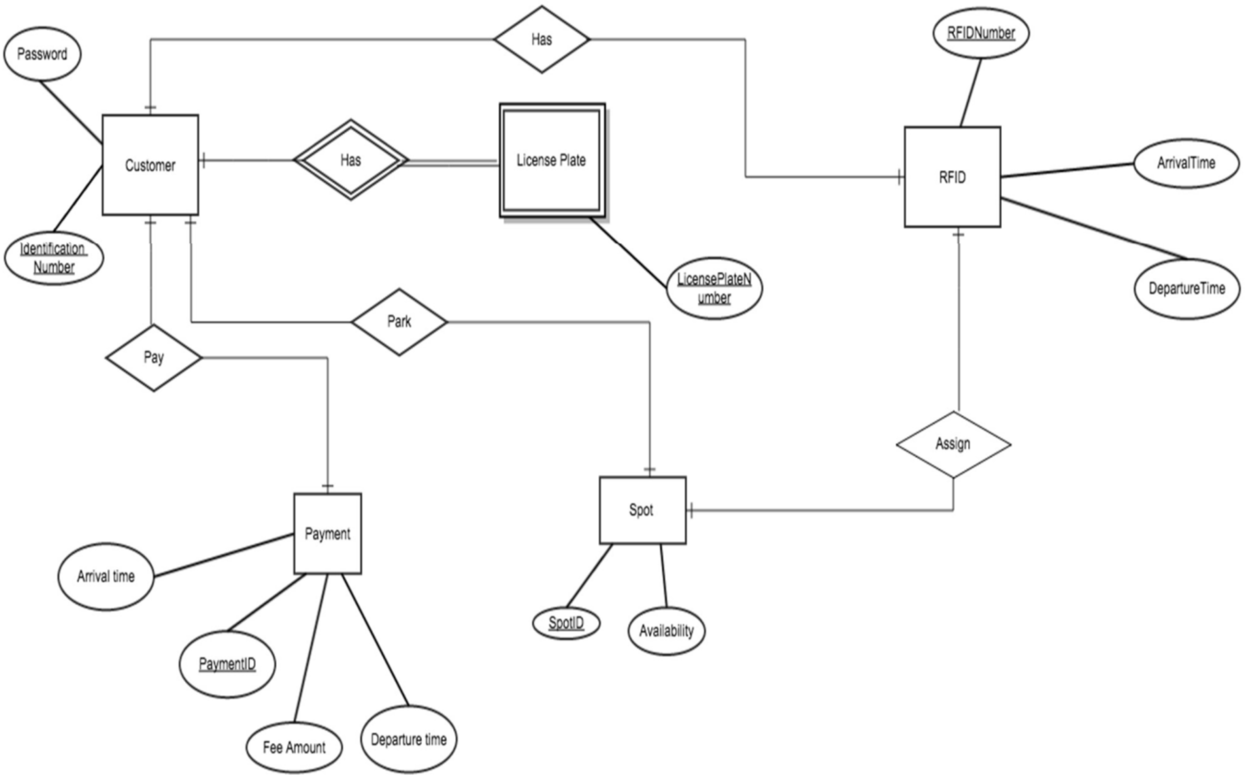
There is no standard for representing the data objects in ER diagrams. Each modelling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either case tools or publications by non-common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entices as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

- **Entities** are the represented by labelled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- **Relationships** are represented by a solid line connecting two entities. The name of the relationship is of the relationship is written above the line. Relationship names should be verbs.

- **Attributes** when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- **Cardinality** of many is represented by a line ending in a crow's foot. If the crow's is omitted, the cardinality is one.

Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.



4.2.4) DATAFLOW DIAGRAM

Flow diagrams in general are usually designed using simple symbols such as a rectangle, an oval or a circle depicting a process, data stored or an external entity, and arrows are generally used to depict the data flow from one step to another.

These symbols can be explained in detail as follows: External entities (source/destination of data) are represented by squares; Processes (input-processing-output) are represented by rectangles with rounded corners; Data Flows (physical or electronic data) are referred to by arrows; and finally, Data Stores (physical or electronic like XML files) are presented by open-ended rectangles.

Basic Dataflow Diagram Notations and Symbols:

- **External Entity**

  An external entity is a source or destination of a data flow which is outside the area of study. Only those entities which originate or receive data are represented on a business process diagram. The symbol used is an oval containing a meaningful and unique identifier.

- **Process**

  A process shows a transformation or manipulation of data flows within the system. The symbol used is a rectangular box which contains 3 descriptive elements:

  Firstly, an identification number appears in the upper left-hand corner. This is allocated arbitrarily at the top level and serves as a unique reference.

Secondly, a location appears to the right of the identifier and describes where in the system the process takes place. This may, for example, be a department or a piece of hardware. Finally, a descriptive title is placed in the center of the box. This should be a simple imperative sentence with a specific verb, for example 'maintain customer records' or 'find driver'.

- **Data Flow**

A data flow shows the flow of information from its source to its destination. A data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of it123wxs contents.

- **Data Store**

A data store is a holding place for information within the system:

It is represented by an open-ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example, batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.

✓ Online Car Parking System Dataflow Diagram:

Online Car Parking System Data Flow Diagram is often used as a preliminary step to create an overview of the Car Parking without going into great detail, which can later be elaborated. It normally consists of overall application dataflow and processes of the Car Parking process.

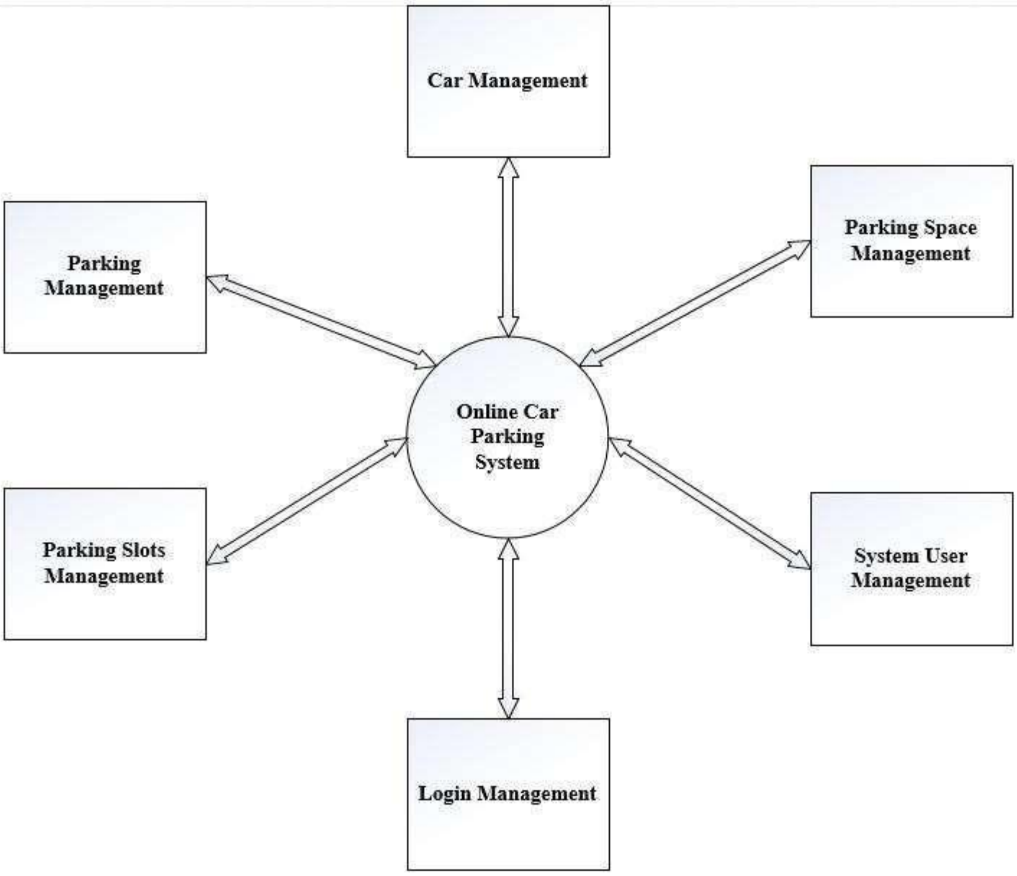
It contains all of the user flow and their entities such all the flow of Car, Parking, Parking Space, Parking Slots, Parking Fess, Car Owner, Car Number.

All of the below diagrams have been used for the visualization of data processing and structured design of the Car parking process and working flow**.**

✓ Zero Level Data Flow Diagram (0 Level DFD) of Online Car Parking System:

➢ **High Level Entities and process flow of Online Car Parking System:**

- Managing All the Car
- Managing All the Parking
- Managing All the Parking Space
- Managing All the Parking Slots
- Managing All the Parking Fees
- Managing All the Car Owner
- Managing All the Car Number

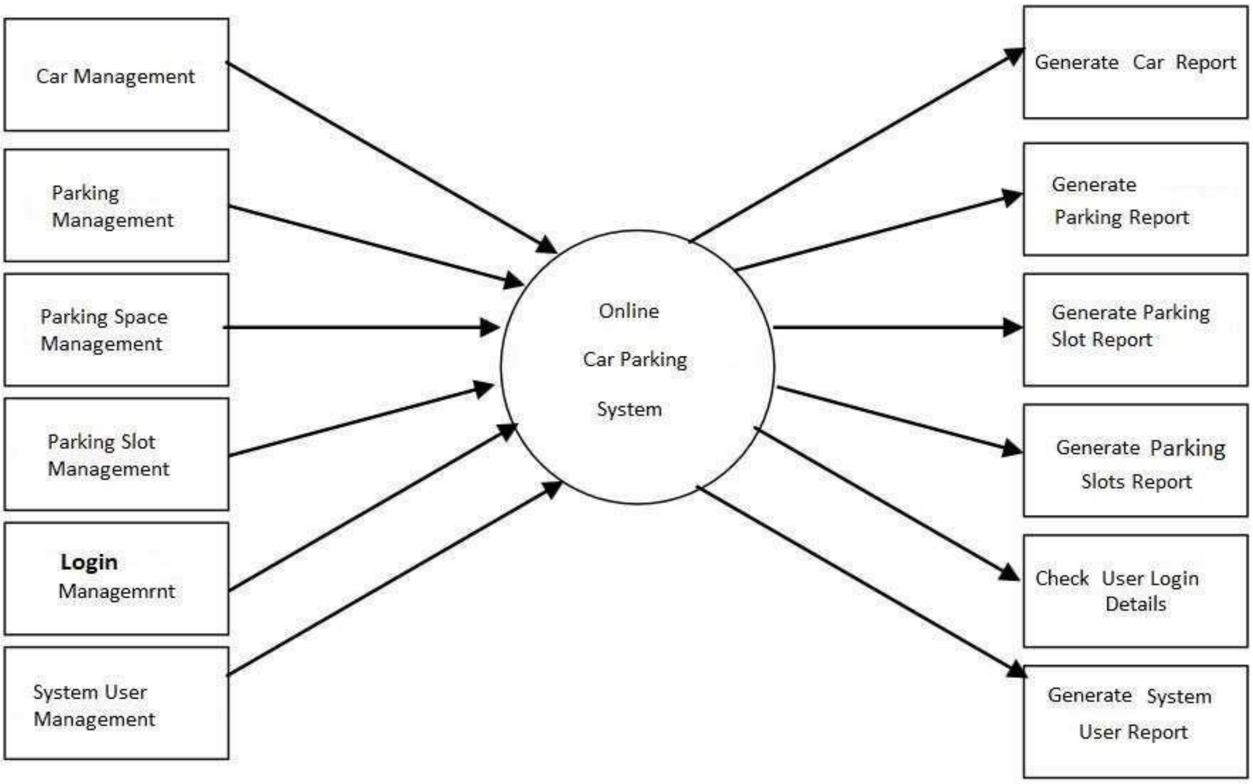


Zero Level DFD- Online Car Parking System

✓ First Level Data Flow Diagram (1st Level DFD) of Online Car Parking System:

➢ **Main entities and output of First Level DFD (1st Level DFD):**

- Processing Car records and generate report of all Car
- Processing Parking records and generate report of all Parking
- Processing Parking Space records and generate report of all Parking Space
- Processing Parking Slots and generate report of all Parking Slots
- Processing Parking Fees records and generate report of all Parking Fees
- Processing Car Owner records and generate report of all Car Owner
- Processing Car Number and generate report of all Car Number

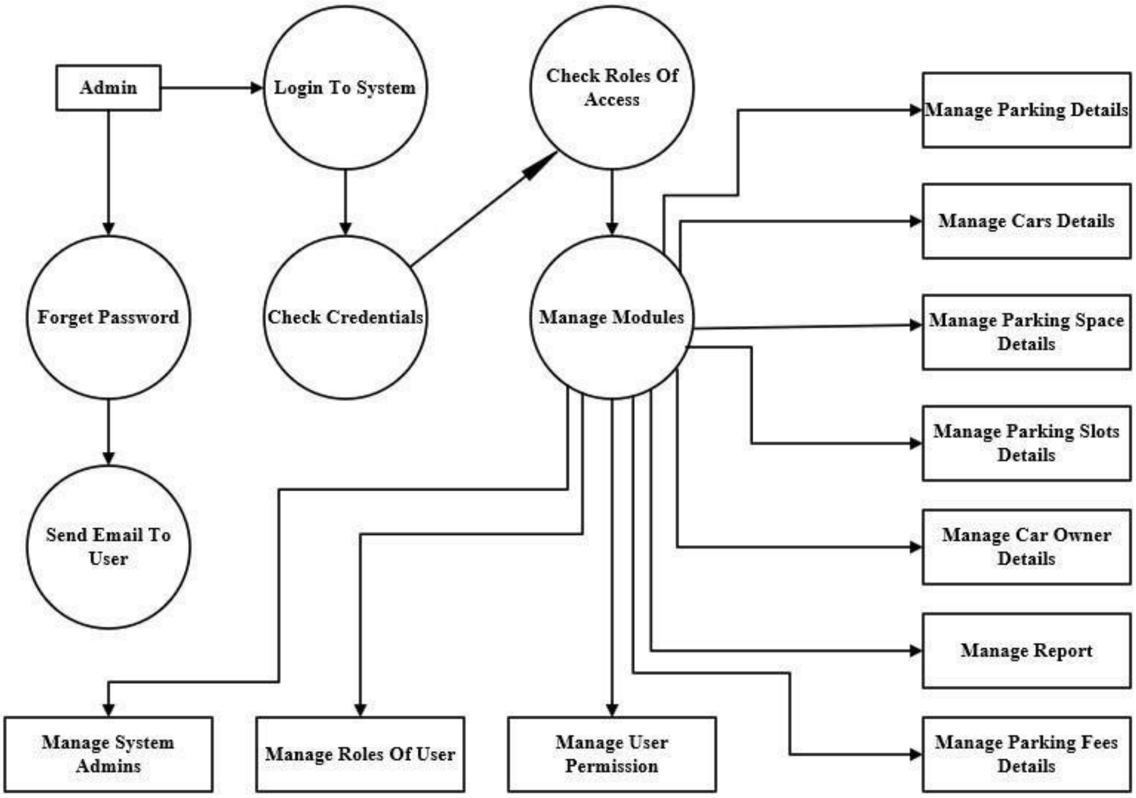


- ✓ First Level Data Flow Diagram (1st Level DFD) of Online Car Parking System:

DFD level 2 then goes one step deeper into parts of level 1 of Car Parking. It may require more functionalities of Car Parking to reach the necessary level of detail about the Car Parking functioning. First Level DFD of Online Car Parking System shows how the system is divided into sub systems (processes). The 2nd Level DFD contains more details of Car Number, Car Owner, Parking Fees, Parking Slots, Parking Space, Parking Car.

Low level functionalities of petrol pump management system

- Admin logins to the system and manage all the functionalities of Online Parking System.
- Admin can add, edit, delete and view the records of car, Parking Space, Parking Fees, Car Number.
- Admin can manage all the details of Parking, Parking Slots, Car Owner.
- Admin can apply different level of filters on report of Car, Parking Slots, Parking Fees
- Admin can track the detailed information of Parking, Parking Space, Parking Slots, Parking Fees.



4.3) PROCEDURAL DESIGN

Component level design also called procedural design occurs after data, architectural, and interface design have been established.

What is it? Info, architectural, and interface style should be translated into operational software. To accomplish this, the design must be represented at a level of abstraction that is close to code.

Component level design establishes: -

1. The algorithmic detail required to manipulate data structures.

2. Effect communication between software components via their interfaces, and

3. Implement the processing algorithms allocated to each component.

Who does it? A software program engineer performs component level design.

Why is it important? You have to be able to determine whether the program will work before you build it. The component level style represents the software in a way that allows you to review the details of the design for correctness and regularity with earlier design representations. It provides a means intended for assessing whether data constructions, interface and algorithms will continue to work.

What are the steps? Design representations of the data, architecture, and interfaces form the foundation for component level style. The digesting narrative for each component is converted into a procedural design model using a set of structured programming constructs. Graphical, tabular or text-based notation is used to represent the look.

Approach is to represent the procedural style using some intermediate representation that can be translated easily into source code.

4.3.1) ALGORITHMS DESIGN

Login into the system (Admin/Users): -

- Start-up system
- Enter username and password
- On clicking the login button
- Connect to database
- Query database to know whether user credentials are correct
- Deny access and return login page with an error message

If correct

- Check if credentials are for administrator

- If yes Allow login
    - Set admin session
    - Redirect administrator to admin home page
- If no Allow login
    - Set user session
    - Redirect user to user home page.

Add new users: -

- Check if administrator is logged in If correct
- Check if all fields entered are correct o If not
- System message: please enter all fields o If correct

Registration of new user successful

4.4) UML DIAGRAM:

StarUML is an open source software modelling tool that supports UML (Unified Modelling Language). It is based on UML version 1.4, provides eleven different types of diagram and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing to generate code for multiple languages.

4.4.1) USE CASE DIAGRAM

Use Case diagrams are defined as diagrams that capture the system's functionality and requirements in UML. Use-cases are the core concepts of Unified Modelling language modelling.

A Use Case consists of use cases, persons, or various things that are invoking the features called as actors and the elements that are responsible for implementing the use cases.

Use case diagrams capture the dynamic behavior of a live system.

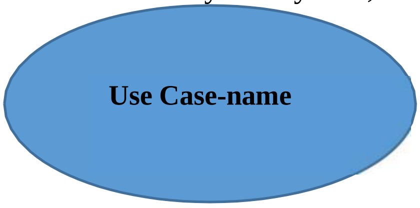
It models how an external entity interacts with the system to make it work. Use case diagrams are responsible for visualizing the external things that interact with the part of the system.

Use-case diagram notations

Following are the common notations used in a use case diagram:

Use-case:

Use cases are used to represent high-level functionalities and how the user will handle the system. A use case represents a distinct functionality of a system, a component, a package, or a class. It is



denoted by an oval shape with the name of a use case written inside the oval shape. The notation of a use case in UML is given below:

Actor:

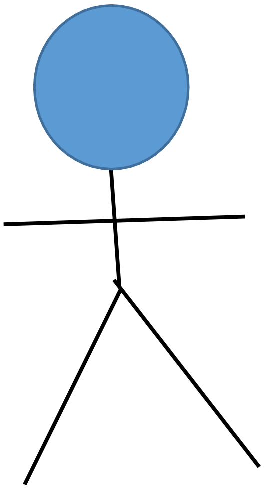
It is used inside use case diagrams.

The actor is an entity that interacts with the system.

A user is the best example of an actor. An actor is an entity that initiates the use case from outside the scope of a use case.

It can be any element that can trigger an interaction with the use case. One actor can be associated with multiple use cases in the system.

The actor notation in UML is given below.



Actor-name

- This Use Case Diagram is a graphic depiction of the interactions among the elements of Online Car Parking System.

- It represents the methodology used in system analysis to identify, canty, and organize system requirements of Online Car Parking System.

- The main actors of Online Car Parking System in this Use Case Diagram are Super Admin, System User, Customer, Agents, who perform the different type of use cases such as Manage Car, Manage Parking Manage Parking Slots, Manage Parking Space, Manage Parking Fees, Manage Car Owner, Manage Car Number, Manage Users and Full Car Parking System Operations.

- Major elements of the UML use case diagram of Online Car Parking System are shown on the picture below.

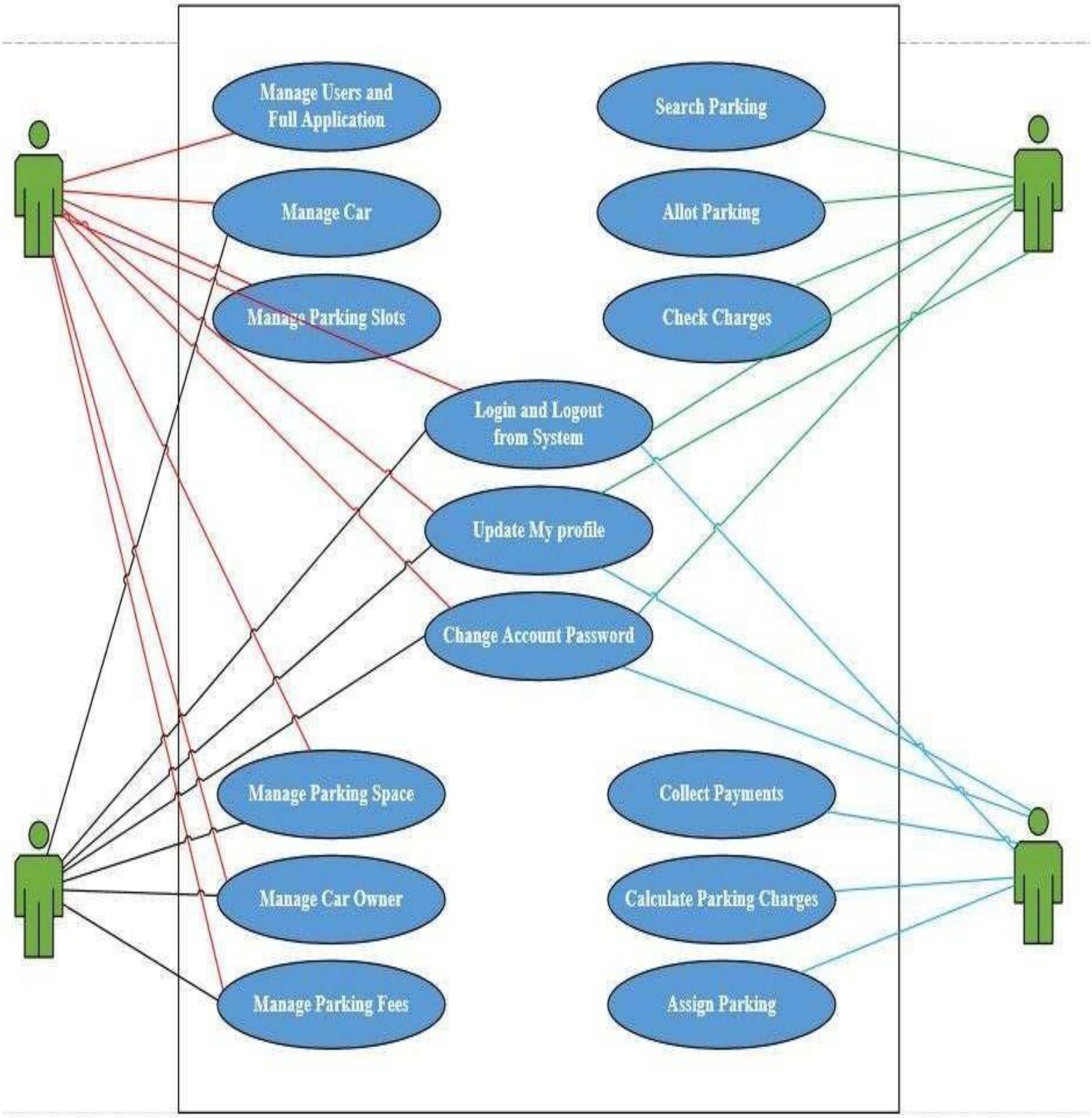
❖ The relationships between and among the actors and the use cases of Online Car Parking System:

- **Super Admin Entity:** Use cases of Super Admin are Manage Car, Manage Parking, Manage Parking Slots, Manage Parking Space, Manage Parking Fees, Manage Car Owner, Manage Car Number, Manage Users and Full Car Parking System Operations.

- **System User Entity:** Use cases of System User are Manage Car, Manage Parking Manage Parking Slots, Manage Parking Space, Manage Parking Fees, Manage Car Owner, Manage Car Number.

- **Customer Entity:** Use cases of Customer are Search Parking. Allot Parking Check Charges, Make Payments.

- **Agents Entity:** Use cases of Agents are Collect Payments, Calculate Parking Charges, Assign Parking.

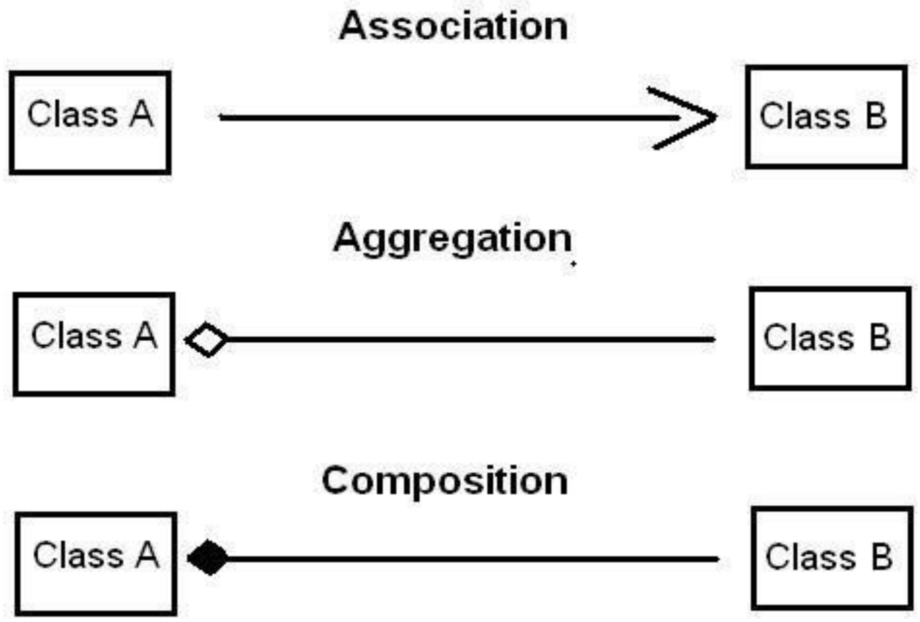

Manage Users and Full Application
Search Parking
Manage Car
Allot Parking
Manage Parking Slots
Check Charges
Login and Logout from System
Update My profile
Change Account Password
Manage Parking Space
Collect Payments
Manage Car Owner
Calculate Parking Charges
Manage Parking Fees
Assign Parking

4.4.2) CLASS DIAGRAM

- Class diagram is the main building block in object-oriented modeling.
- They are being used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code.
- The classes in a class diagram represent both the main objects and interactions in the application.
- The Class diagrams are used to identify and classify the objects which constitute a system.
- It also includes the important attributes of the objects which must be captured.

Class Diagram Notations: -

- In Class diagram, rectangles represent Class and the lines connecting the rectangle show the relationships among Classes.



Online Car Parking System Class Diagram describes the structure of a Car Parking System classes their attributes, operations (or methods), and the relationships among objects the main classes of the Car Parking System are Car, Parking, Parking Slots, Parking Space, Parking Fees, Car Owner.

❖ Classes of Online Car Parking System Class Diagram:

- **Car Class:** Manage all the operations of Car.
- **Parking Class:** Manage all the operations of Parking.
- **Parking Slots Class:** Manage all the operations of Parking Slots.
- **Parking Space Class:** Manage at the operations of Parking Space.
- **Parking Fees Class:** Manage all the operations of Parking Fees.
- **Car Owner Class:** Manage all the operations of Car Owner.

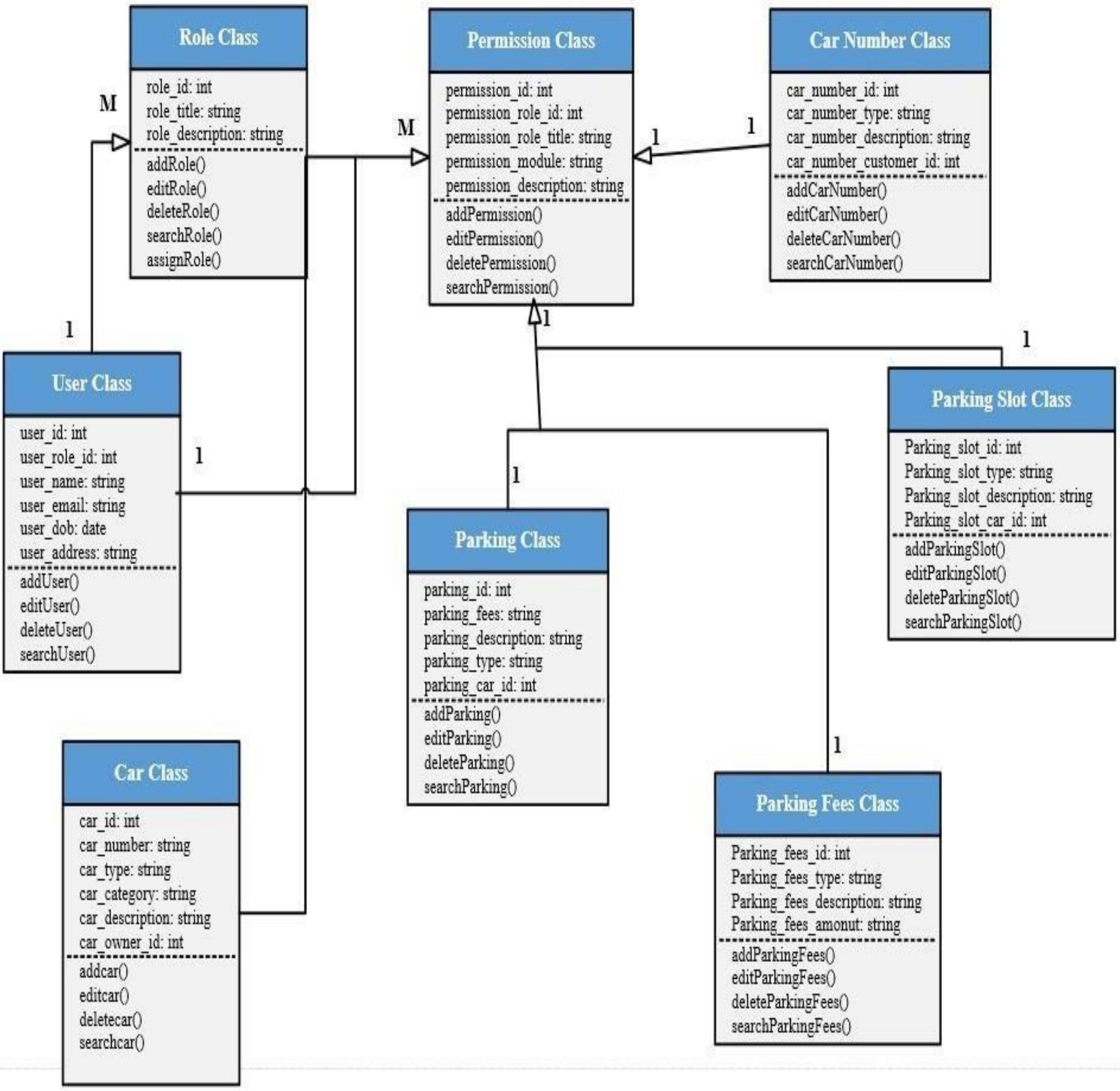
❖ Classes and their attributes of Online Car Parking System Class Diagram:

- **Car Attributes:** car_id, car_owner_id, car_number, car_company, car_type, car_description.
- **Parking Attributes:** parking_id, parking_car_id, parkang_fees, parking_type, parking_description.
- **Parking Slots Attributes:** parking_slot_id, parking_slot_car_id, parking_slot_type, parking_slot_description.
- **Parking Space Attributes:** parking_space_id, parking_space_car_id, parking_space_type, parking_space_description.
- **Parking Fees Attributes:** parking_fees_id, parking_fees_amount, parking_fees_type, parking_fees_description.

- **Car Owner Attributes:** car_owner_id, car_owner_name, car_owner_mobile, car_owner_email, car_owner_username, car_owner_password, car_owner_address.

- Classes and their methods of Online Car Parking System Class Diagram:

- **Car Methods:** addCar (), editCar (), deleteCar (), updateCar (), saveCar (), searchCar ().
- **Parking Methods:** addParking (), editParkng (), deleteParking (), updateParking (), savePaking (), searchParking ().
- **Parking Slots Methods:** addParking Slots (), editParking Slots (), deleteParking Slots (), updateParking Slots (), saveParking Slots (), searchParking Slots (). Parking Space Methods: addParkng Space (), editParking Space (), deleteParking Space (), updateParking Space (), saveParking Space (), searchParking Space ().
- **Parking Fees Methods:** addParking Fees (), editParking Fees (), deleteParking Fees (), updateParking Fees (), saveParking Fees (), searchParking Fees ().
- **Car Owner Methods:** addCar Owner (), editCar Owner (), deleteCar Owner (), updateCar Owner (), saveCar Owner (), searchCar Owner ().


Role Class
role_id: int
role_title: string
role_description: string
addRole()
editRole()
deleteRole()
searchRole()
assignRole()
M
M
Permission Class
permission_id: int
permission_role_id: int
permission_role_title: string
permission_module: string
permission_description: string
addPermission()
editPermission()
deletePermission()
searchPermission()
1
1
Car Number Class
car_number_id: int
car_number_type: string
car_number_description: string
car_number_customer_id: int
addCarNumber()
editCarNumber()
deleteCarNumber()
searchCarNumber()
1
1
1
User Class
user_id: int
user_role_id: int
user_name: string
user_email: string
user_dob: date
user_address: string
addUser()
editUser()
deleteUser()
searchUser()
1
Parking Slot Class
Parking_slot_id: int
Parking_slot_type: string
Parking_slot_description: string
Parking_slot_car_id: int
addParkingSlot()
editParkingSlot()
deleteParkingSlot()
searchParkingSlot()
1
Parking Class
parking_id: int
parking_fees: string
parking_description: string
parking_type: string
parking_car_id: int
addParking()
editParking()
deleteParking()
searchParking()
1
Car Class
car_id: int
car_number: string
car_type: string
car_category: string
car_description: string
car_owner_id: int
addcar()
editcar()
deletecar()
searchcar()
Parking Fees Class
Parking_fees_id: int
Parking_fees_type: string
Parking_fees_description: string
Parking_fees_amonut: string
addParkingFees()
editParkingFees()
deleteParkingFees()
searchParkingFees()

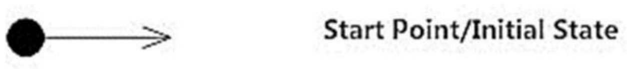
4.4.3) ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

Basic Activity Diagram Notations and Symbols:

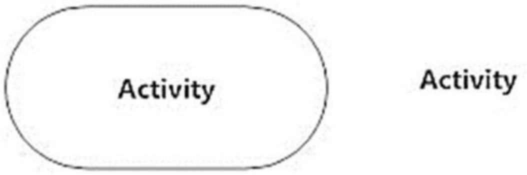
- **Initial State or Start Point:**

  A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swim lanes, make sure the start point is placed in the top left corner of the first column.



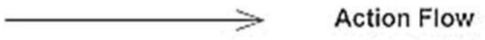
- **Activity or Action State:**

  An action state represents the non-interruptible action of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.



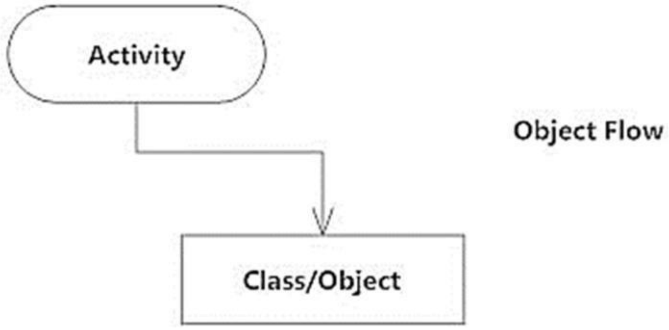
- **Action Flow:**

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.



- **Object Flow:**

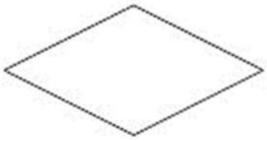
Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



- **Decisions and Branching:**

A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression.

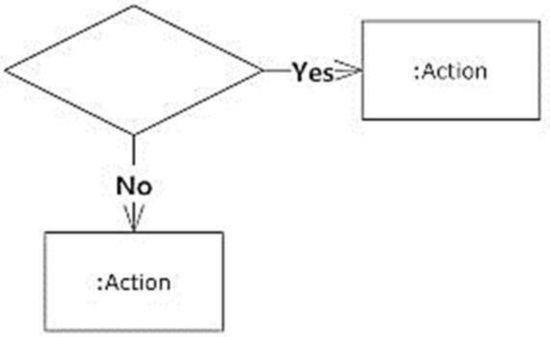
You can also label one of the paths "else."



Decision Symbol

- **Guards:**

In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.



Guard Symbols

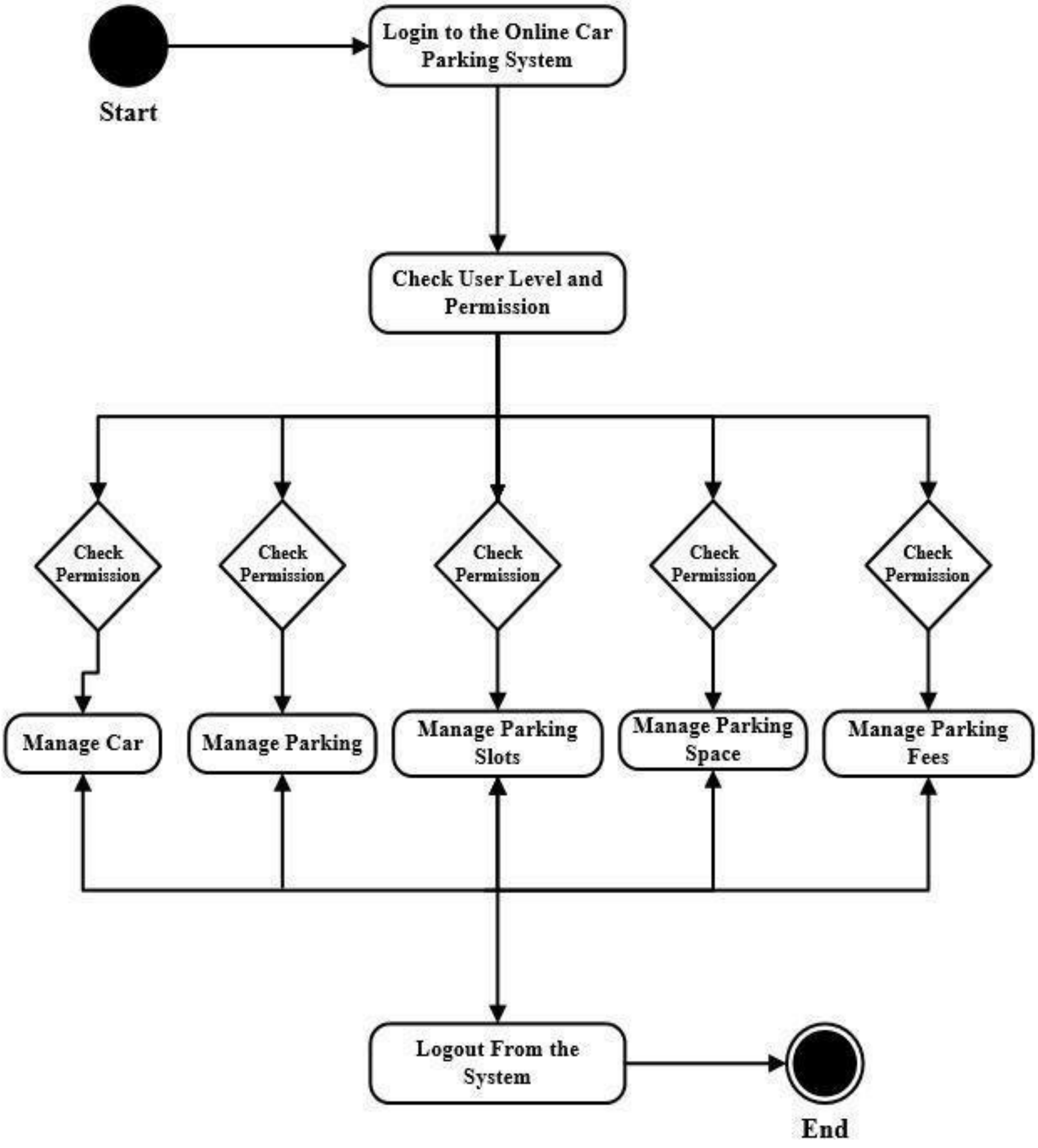
Online Car Parking System Activity Diagram:

This is the Activity UML diagram of Car Parking System which shows the flows between the activity of Car Number, Car, Car Owner, Parking Fees, Parking Space. The main activity involved in this UML Activity Diagram of Car Parking System are as follows:

- Car Number Activity
- Car Activity
- Car Owner Activity
- Parking Fees Activity
- Parking Space Activity

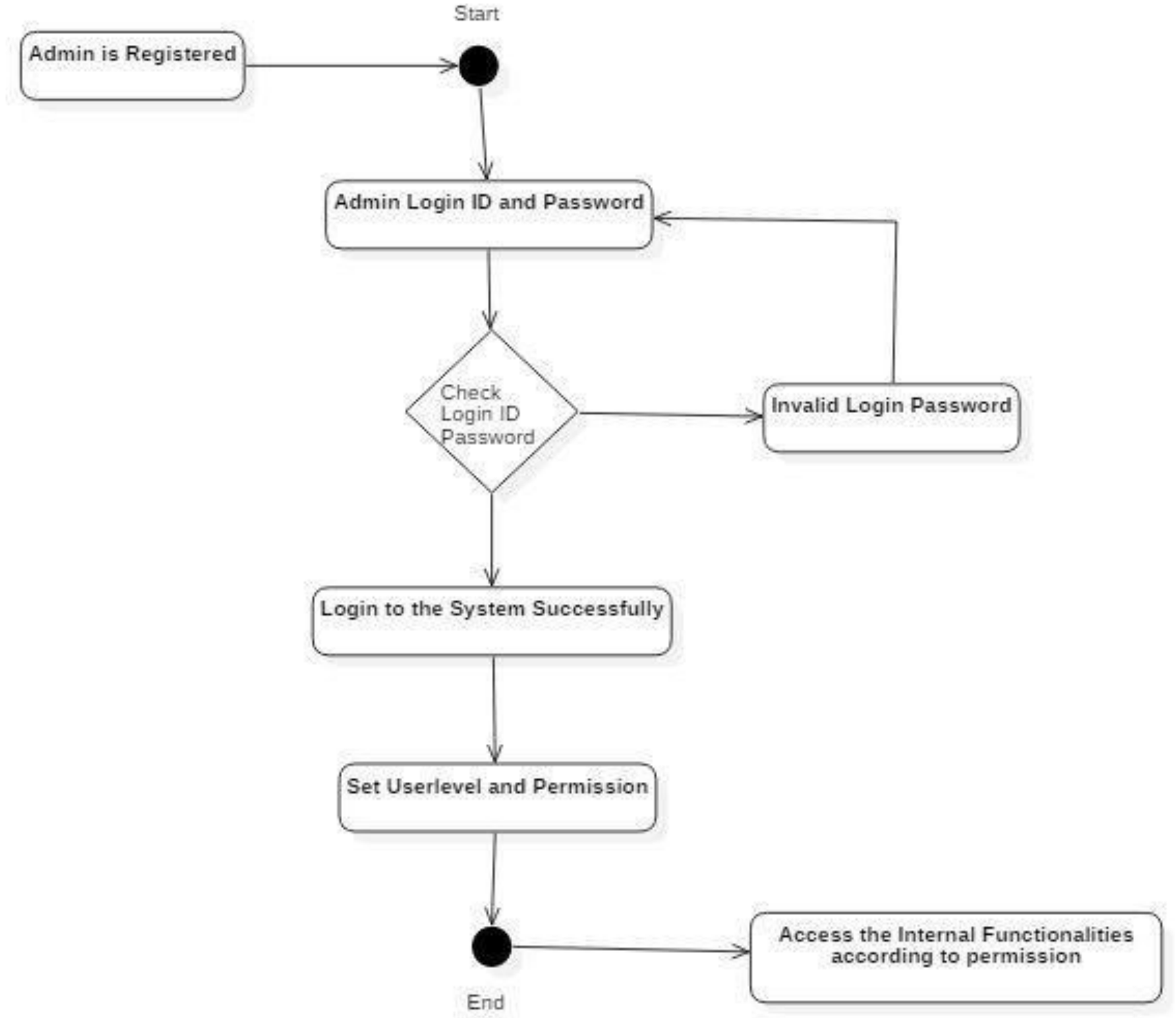
❖ Features of The Activity UML Diagram of Online Car Parking System:

- Admin User can search Car Number, view description of a selected Car Number, add Car Number, update Car Number and delete Car Number.
- Its shows the activity flow of editing, adding and updating of Car.
- User will be able to search and generate report of Car Owner, Parking Fees, Parking Space.
- All objects such as (Car Number, Car, Parking Space) are interlinked.
- Its shows the full description and flow of Car Number, Parking Fees, Parking Space, Car Owner, Car.


Start
Login to the Online Car Parking System
Check User Level and Permission
Check Permission
Check Permission
Check Permission
Check Permission
Check Permission
Manage Car
Manage Parking
Manage Parking Slots
Manage Parking Space
Manage Parking Fees
Logout From the System
End

❖ Login Activity Diagram 0f Online Car Parking System:

This is the Login Activity Diagram of Car Parking System, which shows the flows of Login Activity, where admin will be able to login using their username and password After login user can manage all the operations on Car Owner, Car Number, Car, Parking Space, Parking Fees. All the pages such as Car, Parking Space, Parking Fees are secure and user can access these pages after login. The diagram below helps demonstrate how the login page works in a Car Parking System. The various objects in the Parking Space, Car Owner, Car Number, Car, and Parking Fees page-interact over the course of the Activity, and user will not be able to access this page without verifying their identity.


Start
Admin is Registered
Admin Login ID and Password
Check Login ID Password
Invalid Login Password
Login to the System Successfully
Set Userlevel and Permission
End
Access the Internal Functionalities according to permission

4.4.4) SEQUENCE DIAGRAM

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams.

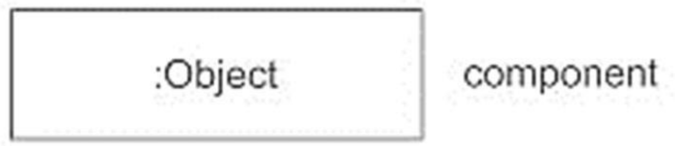
A sequence diagram is a good way to visualize and validate various runtime scenarios.

These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

❖ **Basic Sequence Diagram Notations:**

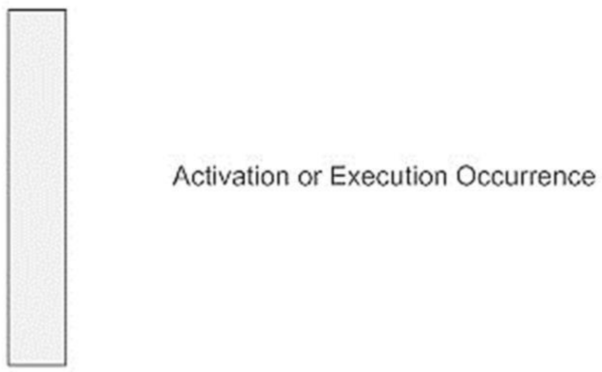
Class Roles or Participants

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.



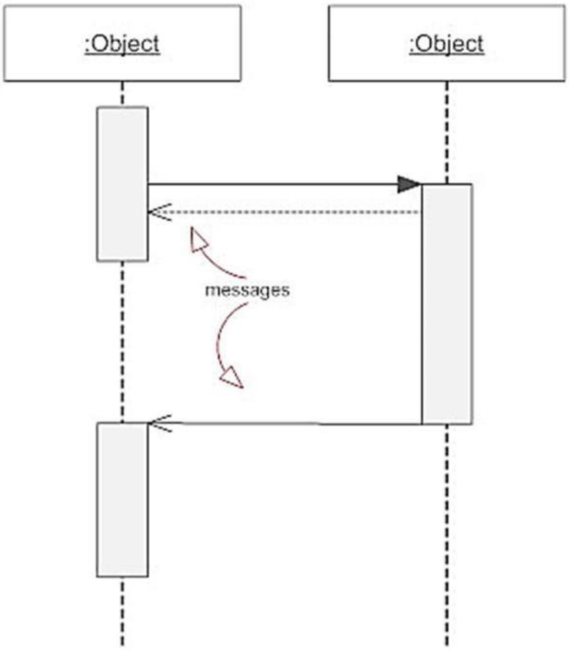
Activation or Execution Occurrence:

Activation boxes represent the time an object needs to complete a task.
When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.



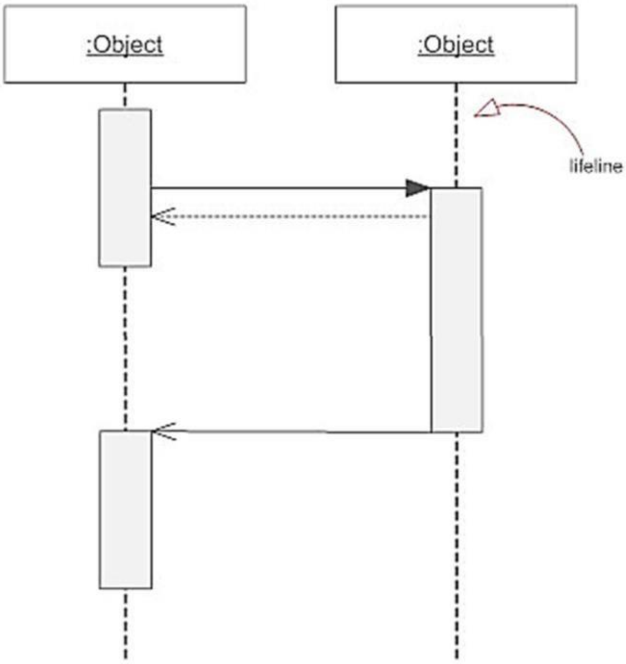
Messages:

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks. For message types, see below.



Lifelines:

Lifelines are vertical dashed lines that indicate the object's presence over time.



❖ Online Car Parking System Sequence Diagram:

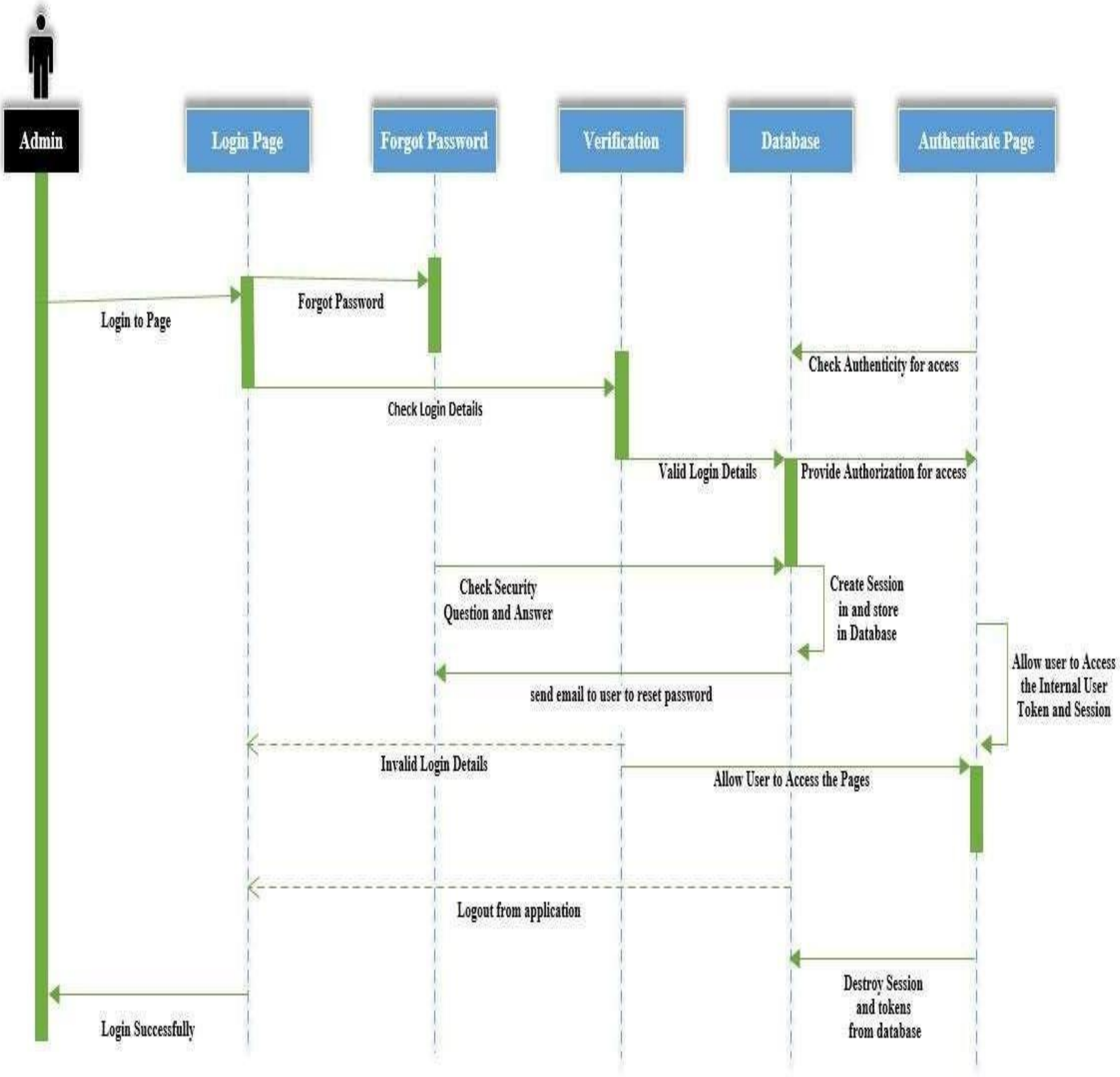
This the UML sequence diagram of Online Car Parking System which shows the interaction between the objects of Parking Fees, Parking Space, Car Number, Parking Slots, Car The instance of class objects involved in this UML Sequence Diagram of Car Parking System are as follows:

- Parking Fees Object
- Parking Space Object
- Car Number Object
- Parking slots Object
- Car Object

❖ Login Sequence Diagram of Online Car Parking System:

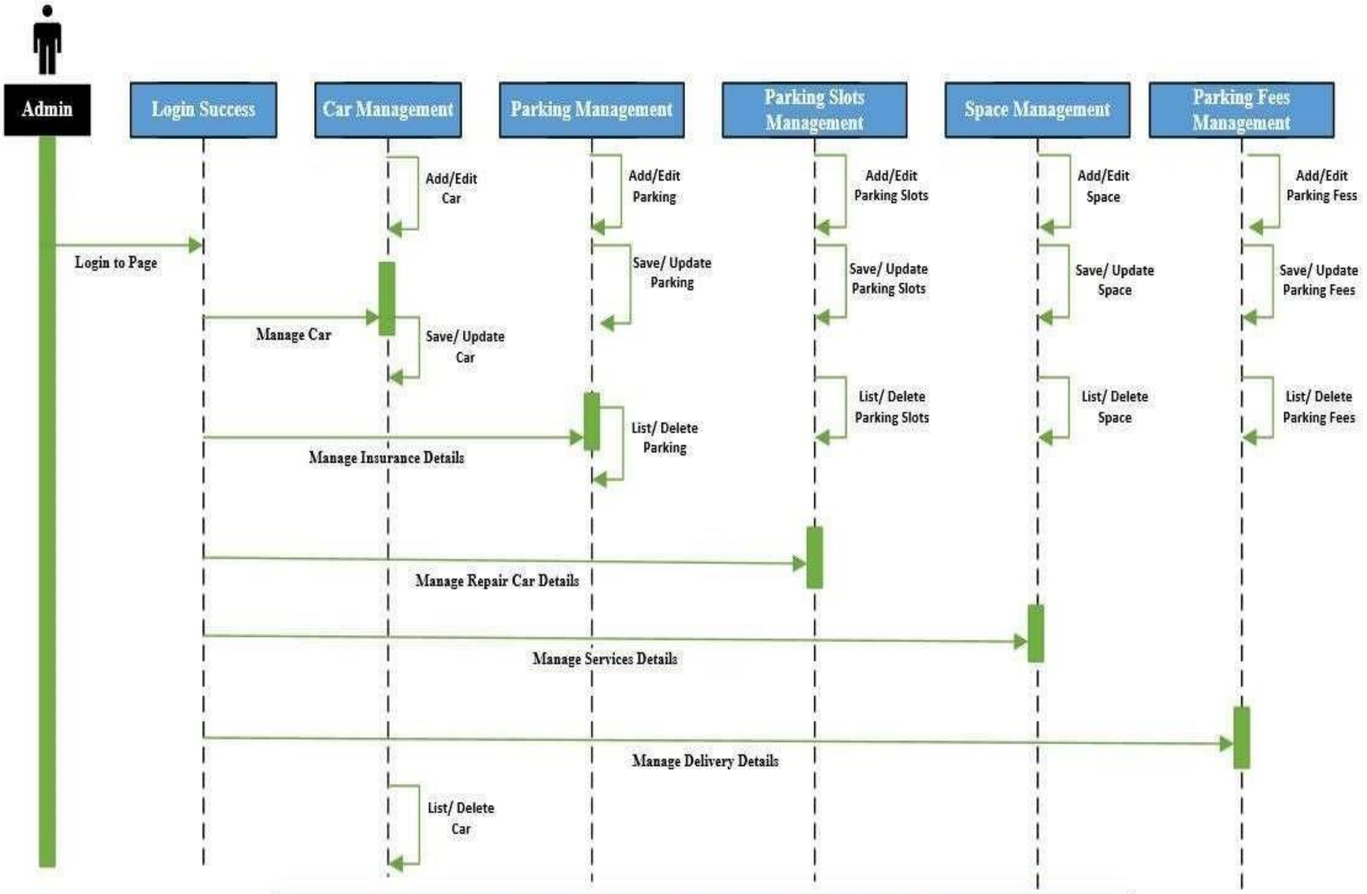
The is the Login Sequence Diagram of Car Parking System, where admin will be able to login in their account using their credentials. After login user manage all the operations on Car Number Parking Fees Parking Space, Car, Parking Slots All the pages such as Parking Space, Car, Parking Slots are secure and user can access these pages after login.

The diagram below helps demonstrate how the login page works in a Car Parking System. The various objects in the Car, Car Number, Parking Fees, Parking Space, and Parking Slots page-interact over the course of the sequence, and user will not be able to access this page without verifying their identity.


Admin
Login Page
Forgot Password
Verification
Database
Authenticate Page
Login to Page
Forgot Password
Check Authenticity for access
Check Login Details
Valid Login Details
Provide Authorization for access
Check Security
Question and Answer
Create Session
in and store
in Database
Allow user to Access
the Internal User
Token and Session
send email to user to reset password
Invalid Login Details
Allow User to Access the Pages
Logout from application
Destroy Session
and tokens
from database
Login Successfully

This is the UML sequence diagram of Car Parking System which shows the interaction between the objects of Parking Fees, Parking Space, Car Number, Parking Slots, Car. The instance of class objects involved in this UML Sequence Diagram of Car Parking System are as follows:

- Parking Fees Object
- Parking Space Object
- Car Number Object
- Parking Slots Object
- Car Object



** Sequence Diagram of Online Car Parking System **

4.4.5) COMPONENT DIAGRAM:

This is a Component diagram of Online Car Parking System which shows components, provides and required interfaces, ports, and relationships between the Parking Slots, Car, Parking Fees, Parking and Car Number.

This type of diagrams is used in Component- Based Development (CBD) to describe systems with Service – Oriented Architecture (SOA). Online Car Parking System UML component diagram, describes the organization and writing of the physical components in a system.

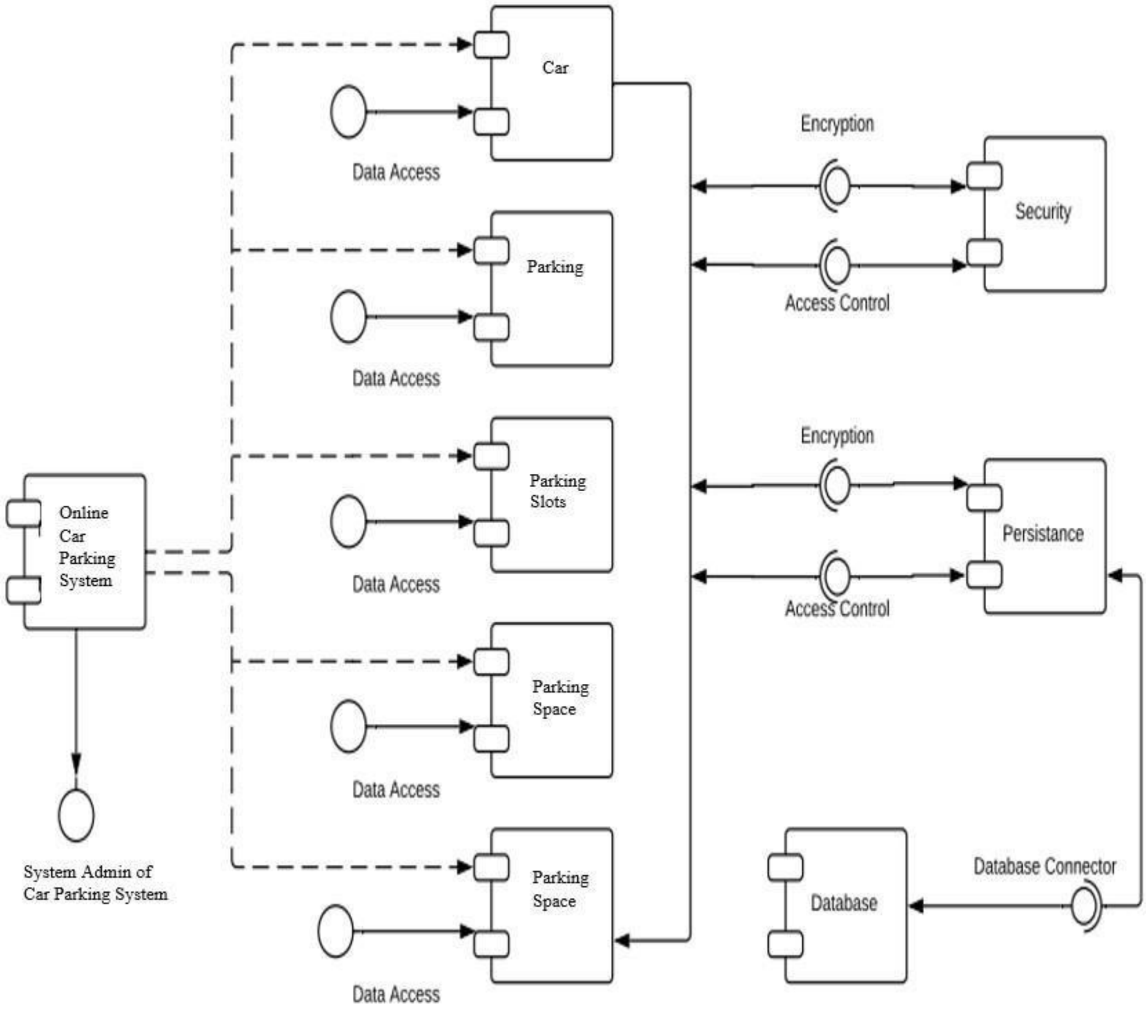
Components of UML Component Diagram of Online Car Parking System

- Parking Slots Component.
- Car Component.
- Parking Fees Component.
- Parking Component.
- Car Number Component.

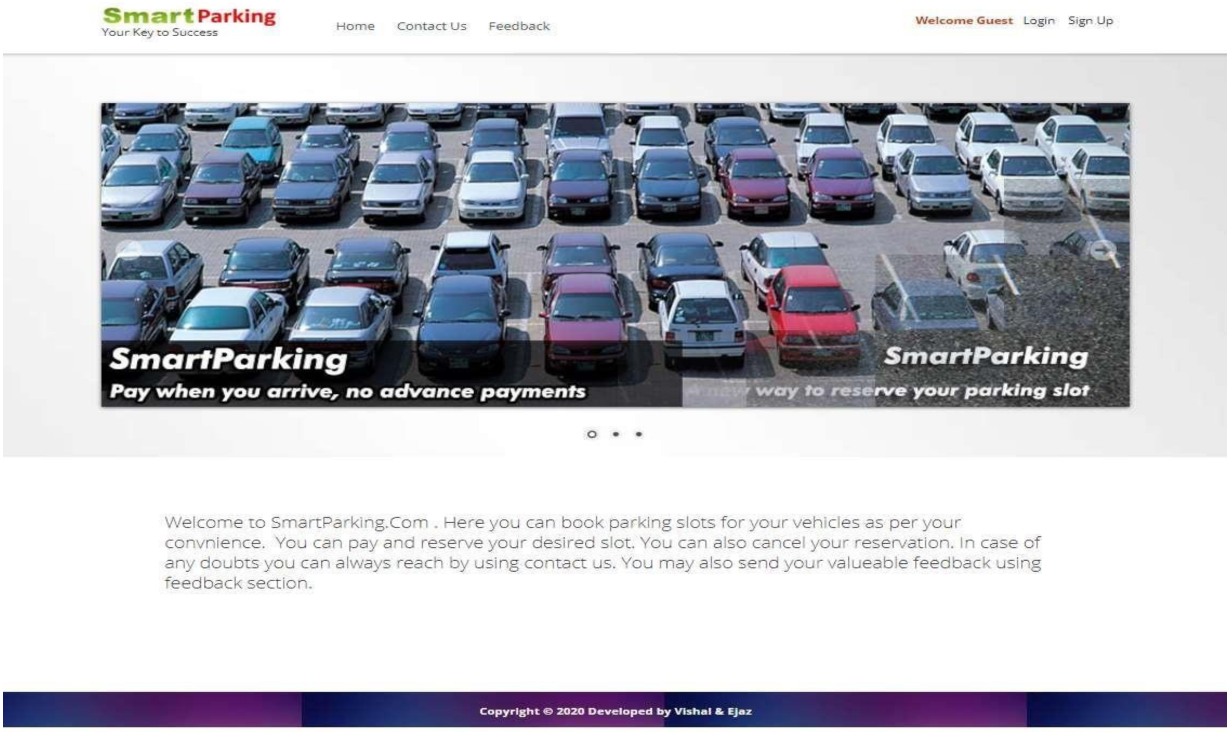
Features of Online Car Parking System Component Diagram:

- You can show the models the components of Online Car Parking System.
- Model the database schema of Online Car Parking System.
- Model the executables of an application of Online Car Parking System.

- Model the system's source code of Online Car Parking System.



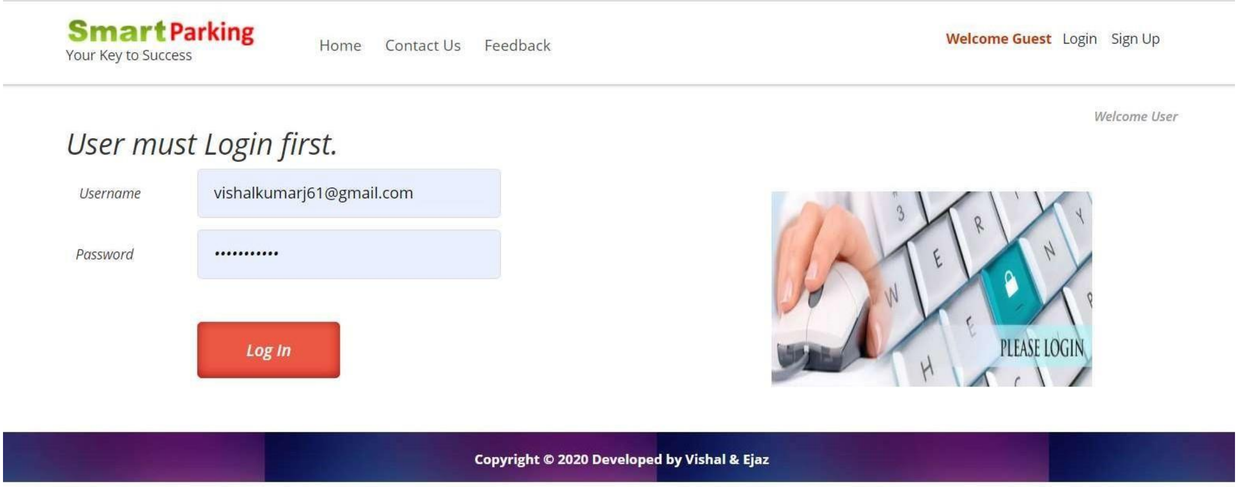
4.5) USER INTERFACE DESIGN



Welcome to SmartParking.Com . Here you can book parking slots for your vehicles as per your convnience. You can pay and reserve your desired slot. You can also cancel your reservation. In case of any doubts you can always reach by using contact us. You may also send your valueable feedback using feedback section.

Copyright © 2020 Developed by Vishal & Ejaz

Fig. 4.5.1 – Homepage UI



User must Login first.

Username vishalkumarj61@gmail.com

Password •••••••••••

Copyright © 2020 Developed by Vishal & Ejaz

Fig. 4.5.2 – Login UI

4.6) SECURITY ISSUES

Introduction:

The protection of IT resources that includes hardware, software, data, procedure and people against unauthorized or natural use

The disaster is known as system security.

System security can be divided into four related topics:

- Security
- Integrity
- Privacy
- Confidentiality

- SYSTEM SECURITY refers to technical innovations and procedures applied to hardware and operating systems to protect against intentional or accidental damage from a defined threat.
- DATA SECURITY is the protection of data against loss, disclosure, modification and destruction.
- SYSTEM INTEGRITY refers to the energetic operation of hardware and programs, adequate physical security and protection against external threats, such as illegal tapping and telephone tapping.
- PRIVACY defines the rights of the user or organization to determine what information they are willing to share or accept from others and how the organization can be protected from unwanted, unjust, or excessive disclosure of information about it.

- CONFIDENTIALITY is a special status given to confidential information in a database to minimize any violation of privacy. It is an attribute of information that characterized your need of protection.

- All actions require a session-based authorization code.
- Administrative actions require the user’s password (and are not based on cookies).

System security refers to various data validation in the from of controls and controls to prevent system malfunction. It is always important to ensure that only valid data is entered and that only valid operations are performed on the system. The system uses two types of checks and checks:

Client-side validation

Several validations are used on the client side to ensure that only valid data is entered on the client side. Client-side validation saves time on the server and loads invalid data. Some tax checks are:

- Java Scripts is used to ensure that mandatory fields in only with appropriate data. The maximum lengths of the module fields are correctly defined.
- Forms can not be sent without completing required data, so manual errors of empty fields that are required can be resolved on the client side to save time and load the server.

4.7) TEST CASE DESIGN

The importance of software testing and its impact can ‘not be underestimated.

Software Testing is a key component of software quality assurance and is a revision of specification, testing and coding. The increased visibility of the software system and the code associated with software failure are motivating factors for planning, through testing. It is not uncommon for a software organization to spend 40% of its efforts on experimentation.

Test of the white box

The white box test is a test case design approach that employs the procedural design control architecture to produce test case. Using white box test approaches, software engineering can produce test cases.

1. Ensure that all independent routes in a module have been exercised at least once.
2. Exercise all logical decisions.
3. Execute all loops within their limits and their operational limits. Exercise internal data structures to maintain their validity.

- **Integration test**

Integration tests ensure that software and subsystem work together. Try the interface of all the modules to make sure the modules behave correctly when they are integration together.

- **System test**

It involves internal tests of the entire system before delivery to the user. Its purpose is to satisfy the user, the meets all the requirement of customer specifications.

- **Acceptance test**

It is a pre-delivery test in which the entire system is tested on the customer's site in real data to find errors.

- **Validation**

The system has been tested and implemented successfully, and therefore made sure that all the requirements listed in the specification of the software requirements are fully met. In case of incorrect entry, the corresponding error messages are displayed.

- **Compilation test**

It was a good idea to make our stress tests from beginning because it gave us time to solve some of the unexpected crashes and stability issues only occurred when the components were exposed to very high transaction volumes.

- **Execution proof**

This program has been loaded and executed successfully. Due to the good programming, no execution errors occurred.

CHAPTER-5

IMPLEMENTATION AND TESTING

5.1) IMPLEMENTATION APPROACHES:

The technology used for development of the system ‘Online Parking System (Smart Parking)’ are as follows: -

• Tools and technology Tools and Techniques used:

1. StarUml
2. Visual Studio 2010 Professional
3. Windows 10
4. SQL Server 2008 RC2

STARUML: -

StarUml is a software modelling platform that supports UML (Unified Modelling Language). It is based on UML version 1.4 and provides eleven different types of diagram, and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept. StarUml excels in customizability to the user’s environment and has a high extensibility in its functionality. Using StarUml, one of the top leading software modelling tools, will guarantee to maximize the productivity and quality of your software projects

Features of star UML: -

- Supports most of the diagrams specified in UML 2.0.

- Very rich feature set and formatting options.

- Ability to generate source code from the UML diagram.

- Reverse engineer the existing code into UML diagrams.

- Supported languages: C, C# and Java.

- Fast load time/execution time compared with other UML tools.

- Familiar Visual Studio like user interface.

- Supports exporting diagrams into JPG format

VISUAL STUDIO 2010 PR0FESSIONAL:

The Microsoft Visual Studio 2010 Professional development system is an integrated environment that simplifies creating, debugging, and deploying applications.

Unleash your creativity and bring your vision to life with powerful design surfaces and innovative collaboration methods for developers and designers. Work within a personalized environment by targeting a growing number of platforms, including Microsoft SharePoint and cloud applications, and accelerate the coding process by using your existing skills.

Integrated support for test-first development and new debugging tools let you find and fix bugs quickly and easily to ensure high quality solutions.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level - including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include: -

- C/C++
- Visual Basic.Net
- C#
- F#
- Java Script
- Type-Script
- XML

- XSLT
- HTML
- CSS

WINDOWS 10:

Windows 10 is a personal computer operating system developed and released by Microsoft as part of the Windows NT family of operating systems. It was officially unveiled in September 2014 following a brief demo at Build 2014. The first version of the operating system entered a public beta testing process in October 2014, leading up to its consumer release on July 29, 2015, and its release to volume licensing on August 1, 2015.

Windows 10 introduces what Microsoft described as "universal apps"; expanding on Metro- style apps, these apps can be designed to run across multiple Microsoft product families with nearly identical code—including PCs, tablets, smartphones, embedded systems, One, Surface and Windows Holographic. The Windows user interface was revised to handle transitions between a mouse-oriented interface and a touchscreen-optimized interface based on available input devices—particularly on 2-in-1 PCs; both interfaces include an updated Start menu which incorporates elements of Windows 7's traditional Start menu with the tiles of Windows 8. The first release of Windows 10 also introduces a virtual desktop system, a window and desktop management feature called Task View, the Microsoft Edge web browser, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX 12 and WDDM 2.0 to improve the operating system's graphics capabilities for games.

Windows 10 received mostly positive reviews upon its original release in July 2015; critics praised Microsoft's decision to downplay user-interface mechanics introduced by Windows 8 (including the full screen apps and Start screen) in non-touch environments to provide a desktop-oriented interface in line with previous versions of Windows, although Windows 10's touch-oriented user interface mode was panned for containing regressions upon the touch- oriented interface of Windows 8. Critics also praised the improvements to Windows 10's bundled software over 8.1, Xbox Live integration, as well as the functionality and capabilities of Cortana personal assistant and the replacement of Internet Explorer with Edge—although the browser was criticized for being a work in progress-- that was not yet feature complete.

MICROSOFT SQL SERVER 2008 R2:

Microsoft SQL Server 2008 R2 Express with Service Pack 2 is a free, feature-rich edition of SQL Server that is ideal for learning, developing, powering desktop, web & small server applications, and for redistribution by ISVs.

Key Features Offered by SQL Server 2008 R2 SP2 Express:

- Supports stored procedures, triggers, functions, and views
- Store all kinds of business data with native support for relational data, XML, FILESTREAM and spatial data
- Improved performance, usability, visualization, in addition to integration with the Microsoft 2007 Office System in SQL Server Reporting Services
- Simplify development efforts by leveraging existing T-SQL skills, ADO.NET Entity Framework and LINQ

- Closely integrated with Visual Studio and Visual Web Developer

SQL Server 2008 R2 SP2 Express Editions:

- **SQL Server 2008 R2 Express with Tools**

  Core edition of Express that supports development. Includes SQL Server 2008 Database Engine and SQL Server Management Studio Express

- **SQL Server 2008 R2 Express with Advanced Services**

  Extends SQL Server 2008 Express with tools to include support for Integrated Full-text Search and Reporting Services

- **SQL Server 2008 R2 Express (Runtime Only)**

  Includes the SQL Database Engine only and is designed for deployments and ISV redistribution

5.2) CODING DETAILS AND EFFICIENCY:

5.2.1) Coding Details: -

• User Registration Code: -

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="signup.aspx.cs" Inherits="signup" %>

<asp: Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp: Content>
<asp: Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <table cellspacing="0" class="style4" >
        <tr>
            <td width="15%">
                <h5 class="style26">
                    Name</h5>
            </td>
            <td width="50%">
                <h1>
                <em>
                <asp: TextBox ID="TextBox1" runat="server" CssClass="style8">
</asp: TextBox>
                </em>
                </h1>
            </td>
            <td class="style9">
                <em>
                <asp: RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
                    ControlToValidate="TextBox1" ErrorMessage="This information is
required"
                    CssClass="style15" ForeColor="Red"></asp: RequiredFieldValidator>
                </em>
            </td>
            <td rowspan="11" valign="middle">
                <asp: Image ID="Image2" runat="server" Height="177px"
                    ImageUrl="~/images/sign-up.jpg" Width="284px"/>
            </td>
        </tr>

        <tr>
            <td class="style19">
                <h5 class="style15">
                <em>Address</em></h5>
            </td>
            <td class="style7">
                <em>
                <asp: TextBox ID="TextBox2" runat="server" CssClass="style8"
                    TextMode="MultiLine"></asp: TextBox>
                </em>
            </td>
            <td class="style9">
                <em>
                <asp: RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
                    ControlToValidate="TextBox2" ErrorMessage="This information is
required"
                    CssClass="style15" ForeColor="Red"></asp: RequiredFieldValidator>
                </em>
            </td>
        </tr>

        <tr>
            <td class="style19">
                <h5 class="style15">
                <em>State</em></h5>
            </td>
            <td class="style7">
                <em>
                <asp: TextBox ID="TextBox4" runat="server" CssClass="style8">
</asp: TextBox>
                </em>
            </td>
            <td class="style9">
                <em>
                <asp: RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
                    ControlToValidate="TextBox4" ErrorMessage="This information is
required" CssClass="style15"
                    ForeColor="Red"></asp: RequiredFieldValidator>
                </em>
            </td>
        </tr>

        <tr>
            <td class="style19">
                <h5 class="style15">
                <em>Mobile No.</em></h5>
            </td>
            <td class="style7">
                <em>
                <asp: TextBox ID="TextBox5" runat="server" CssClass="style8">
</asp: TextBox>
                </em>
            </td>
            <td class="style9">
                <em>
                <asp: RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server"
                    ControlToValidate="TextBox5" ErrorMessage="Phone no. should be of 10
digits"
                    ValidationExpression="\d {10}" CssClass="style15"
ForeColor="Red"></asp: RegularExpressionValidator>
                </em>
            </td>
        </tr>

        <tr>
            <td class="style19">
                <em><strong>Gender</strong></em></td>
            <td class="style7">
                <em>
                <asp: RadioButtonList ID="RadioButtonList1" runat="server"
                    RepeatDirection="Horizontal" style="height: 27px; font-size: large;"
                    ForeColor="Black">
                    <asp: ListItem>Male</asp: ListItem>
                    <asp: ListItem>Female</asp: ListItem>
                </asp: RadioButtonList>
                </em>
            </td>
            <td class="style9">
                &nbsp;</td>
        </tr>

        <tr>
            <td class="style19">
                <h5 class="style15">
                <em>Email (Username)</em></h5>
            </td>
            <td class="style7">
                <em>
                <asp: TextBox ID="TextBox6" runat="server" CssClass="style8"></asp:
TextBox>
                </em>
            </td>
            <td class="style9">
                <asp: RequiredFieldValidator ID="RequiredFieldValidator5" runat="server"
                    ControlToValidate="TextBox6" ErrorMessage="This information is
required"
                    ForeColor="Red"></asp: RequiredFieldValidator>
                <em>
                <asp: RegularExpressionValidator ID="RegularExpressionValidator2"
runat="server"
                    ControlToValidate="TextBox6" ErrorMessage="Please enter in correct
format"
                    ValidationExpression="\w+ ([-+.'] \w+) *@\w+ ([-.] \w+) *\.\w+ ([-.]
\w+) *"
                    CssClass="style15" ForeColor="Red"></asp: RegularExpressionValidator>
                </em>
            </td>
        </tr>

        <tr>
            <td class="style19">
                <h5 class="style15">
                <em>Password</em></h5>
            </td>
            <td class="style7">
                <em>
                <asp: TextBox ID="TextBox7" runat="server" CssClass="style8"
TextMode="Password"></asp: TextBox>
                </em>
            </td>
            <td class="style6">
                <asp: RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"
                    ControlToValidate="TextBox7" ErrorMessage="RequiredFieldValidator"
```

```
                        ForeColor="Red">This information is required</asp:
RequiredFieldValidator>
                </td>
        </tr>

        <tr>
                <td class="style19">
                    <h5 class="style15">
                    <em>Confirm Password</em></h5>
                </td>
                <td class="style7">
                    <em>
                    <asp: TextBox ID="TextBox8" runat="server" CssClass="style8"
TextMode="Password"></asp: TextBox>
                    </em>
                </td>
                <td class="style9">
                    <asp: RequiredFieldValidator ID="RequiredFieldValidator7" runat="server"
                        ControlToValidate="TextBox8" ErrorMessage="This information is
required"
                        ForeColor="Red"></asp: RequiredFieldValidator>
                    <asp: CompareValidator ID="CompareValidator1" runat="server"
                        ErrorMessage="Password doesn't match" ForeColor="Red"
                        ControlToCompare="TextBox7" ControlToValidate="TextBox8"></asp:
 CompareValidator>
                 </td>
        </tr>

        <tr>
                <td class="style11">
                    <h5 class="style15">
                        &nbsp;</h5>
                </td>
                <td class="style12">
                    &nbsp;</td>
                <td class="style13">
                    </td>
        </tr>

        <tr>
                <td class="style10">
                    &nbsp;</td>
                <td class="style24">
                    <em>
                    <asp: Button ID="Button1" runat="server" CssClass="style8" Text="Sign Up"
                        onclick="Button1_Click" />
                    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
                    </em>
                </td>
                <td class="style6">
                    &nbsp;</td>
        </tr>

        <tr>
                <td class="style21">
                    &nbsp;</td>
                <td class="style22">
                    <em>
                    <asp: Label ID="Label1" runat="server" CssClass="style15" 
ForeColor="Black"></asp: Label>
                    </em>
                </td>
                <td class="style23">
                    &nbsp;</td>
        </tr>
    </table>
</asp: Content>
```

Functionality of User Registration Code: -

• A new user has to register into the system in-order to access the system.

• In the registration page, the user has to enter the following 8 details:

1. Name

2. Address

3. State

4. Mobile Number

5. Gender

6. Email Id (Username)

7. Password

8. Confirm Password

• All the fields are mandatory.

• Every user must have a unique Username.

• The user must enter the password in the ‘Confirm Password’ column in-order to successfully continue his/her registration.

5.2.2) Code Efficiency: -

Code efficiency is a broad term used to depict the reliability, speed and programming methodology used in developing codes for an application. Code efficiency is directly linked with algorithmic efficiency and the speed of runtime execution for software. It is the key element in ensuring high performance.

Code efficiency plays a significant role in applications in a high-execution-speed environment where performance and scalability are paramount. One of the recommended best practices in coding is to ensure good code efficiency. Well-developed programming codes should be able to handle complex algorithms.

Code Optimization: -

In computer science, program optimization or software optimization is the process of modifying a software system to make some aspect of it work more efficiently or use fewer resources. In general, a computer program may be optimized so that it executes more rapidly, or to make it capable of operating with less memory storage or other resources, or draw less power. Optimization can occur at a number of levels

- **Design level: -**

At the highest level, the design may be optimized to make best use of the available resources, given goals, constraints, and expected use/load.

• **Algorithms and data structures: -** Given an overall design, a good choice of efficient algorithms and data structures, and efficient implementation of these algorithms and data structures comes next.

• **Source code level: -** Beyond general algorithms and their implementation on an abstract machine, concrete source code level choices can make a significant difference.

• **Build level:** - Between the source and compile level, directives and build flags can be used to tune performance options in the source code and compiler respectively, such as using pre-process or defines to disable unneeded software features, optimizing for specific processor models or hardware capabilities, or predicting branching, for instance.

• **Compile level: -** Use of an optimizing compiler tends to ensure that the executable program is optimized at least as much as the compiler can predict.

• **Assembly level: -** At the lowest level, writing code using an assembly language, designed for a particular hardware platform can produce the most efficient and compact code if the programmer takes advantage of the full repertoire of machine instructions**.**

Reasons for the selection of technology: -

ASP.NET:

ASP.NET may be easier to use and maintain because of its class library system.

ASP.NET’s flexibility and object-oriented features are its main advantages. ASP.NET got its name from Microsoft’s old ASP technology. However, the new .NET Framework and CLR provides for seamless interface with other programming languages, such as Visual Basic.NET and C#. This means the programming language of a project can be changed mid-project without the need to rewrite everything. It also means multiple developers can work on the same project using other languages, such as Visual Basic.NET or C#.

Another advantage is that ASP.NET’s library is task-based. Because the library is organized into inheritable classes related to a given task, such as XML or image editing, developers can save time on many common development tasks. Visual Studio.NET also saves developers time, provided it is installed on a computer fast enough to allow the program to function at its peak. This massive development IDE is equipped with built-in debugging functions. It also incorporates IntelliSense, an auto-completion feature that eliminates the need for developers to memorize methods and variables.

5.3 TESTING APPROACHES:

5.3.1 Unit Testing: -

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

Unit Testing Benefits: -

• Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.

• Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.

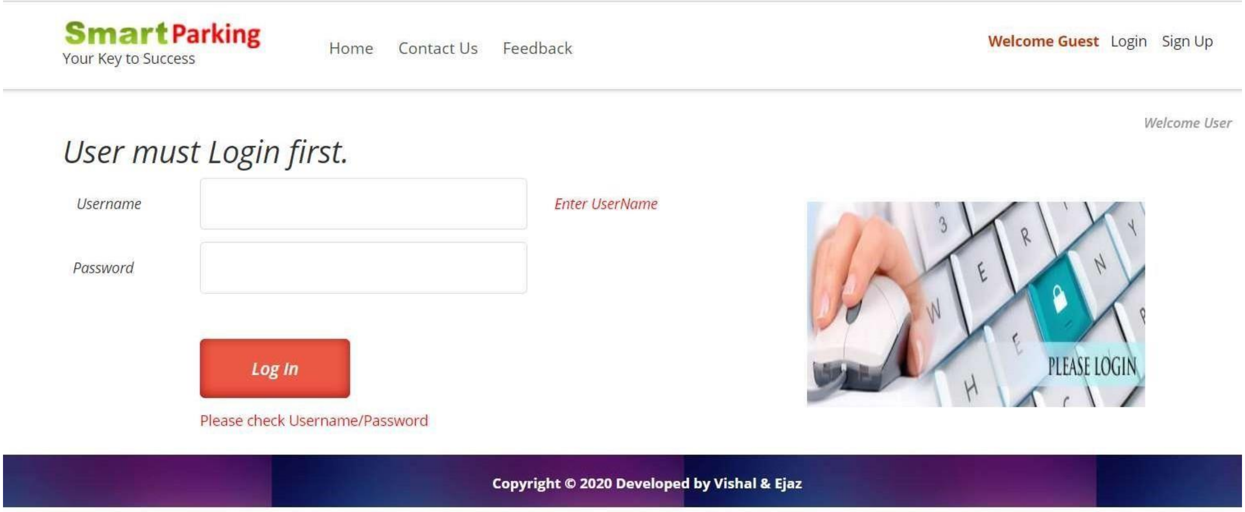
• Development is faster. How? If you do not have unit testing in place, you write your code and perform that fuzzy ‘developer test’ (You set some breakpoints, fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) But, if you have unit testing in place, you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than 'developer tests'. Development is faster in the long run too. How? The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix defects found during system testing or acceptance testing.

- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the cost (time, effort, destruction, humiliation) of a defect detected during acceptance testing or when the software is live.

- Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span of several days/weeks/months need to be scanned.

'User Login' Page Unit Testing: -

- Invalid Login Attempt: -



User must Login first.

Username

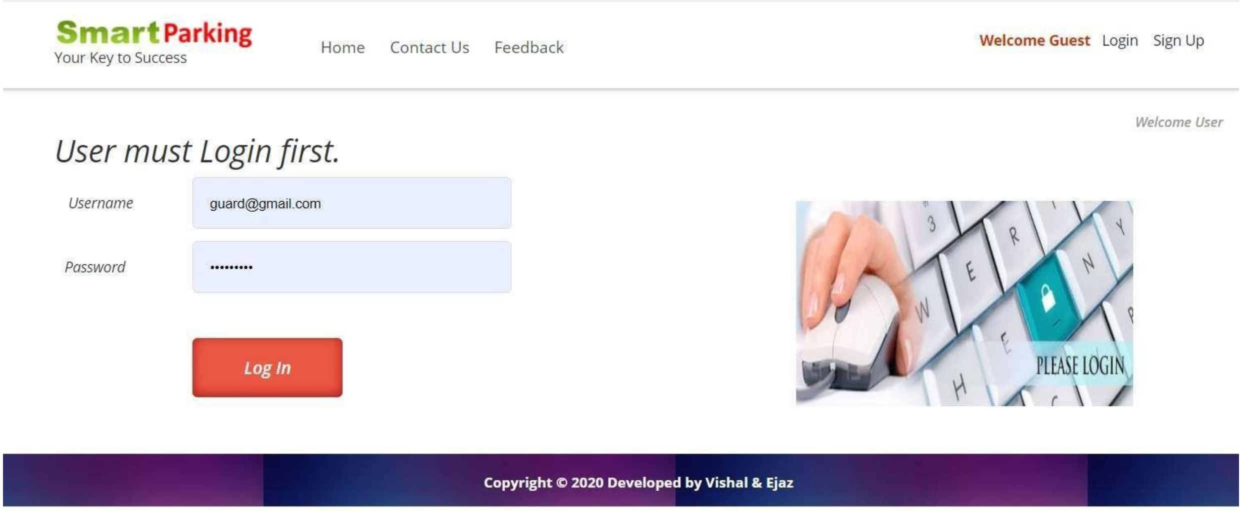
Enter UserName

Password

Please check Username/Password

Copyright © 2020 Developed by Vishal & Ejaz

- Valid Login Attempt: -



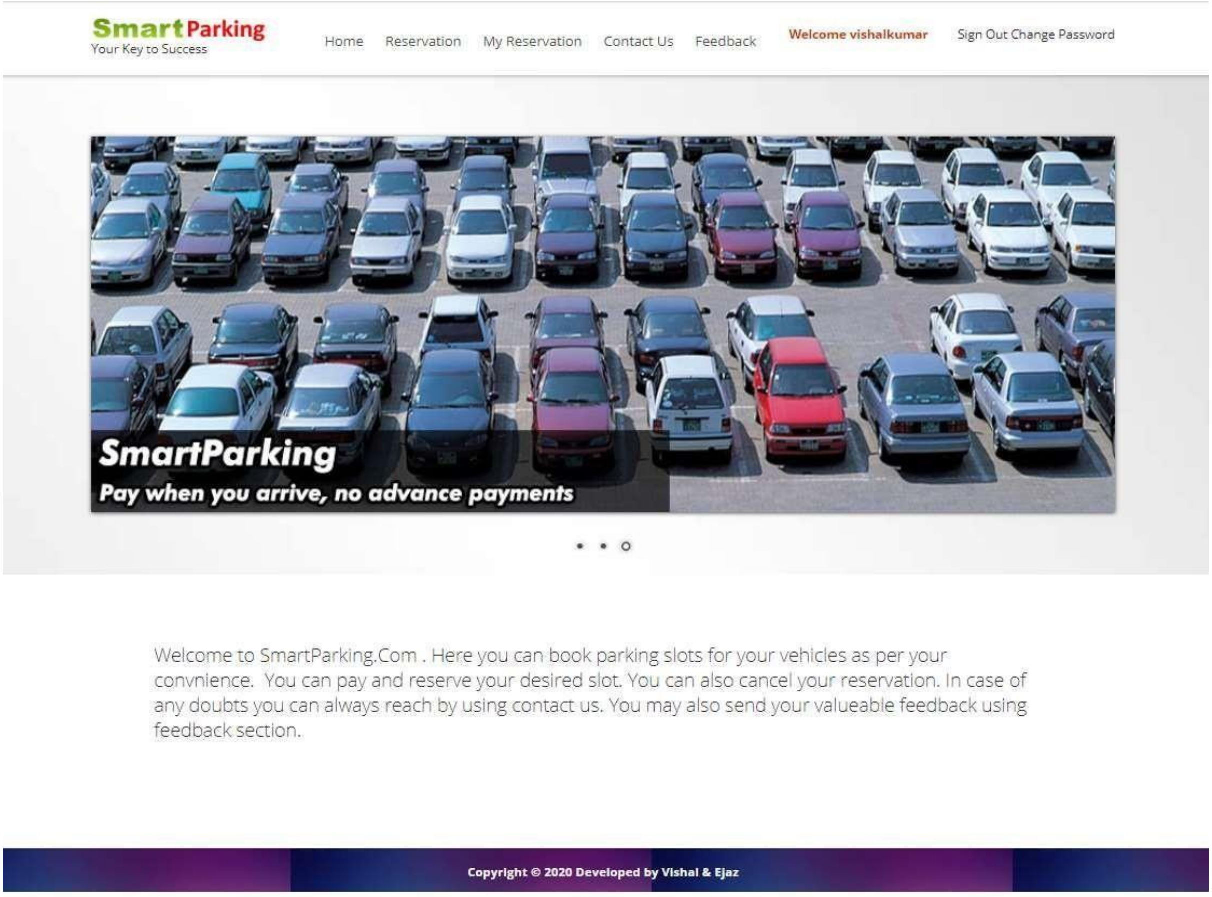
User must Login first.

Username guard@gmail.com

Password ·········

Copyright © 2020 Developed by Vishal & Ejaz

- After Successful User Login: -



Welcome to SmartParking.Com . Here you can book parking slots for your vehicles as per your convnience. You can pay and reserve your desired slot. You can also cancel your reservation. In case of any doubts you can always reach by using contact us. You may also send your valueable feedback using feedback section.

Copyright © 2020 Developed by Vishal & Ejaz

5.3.2) Integration Testing:

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

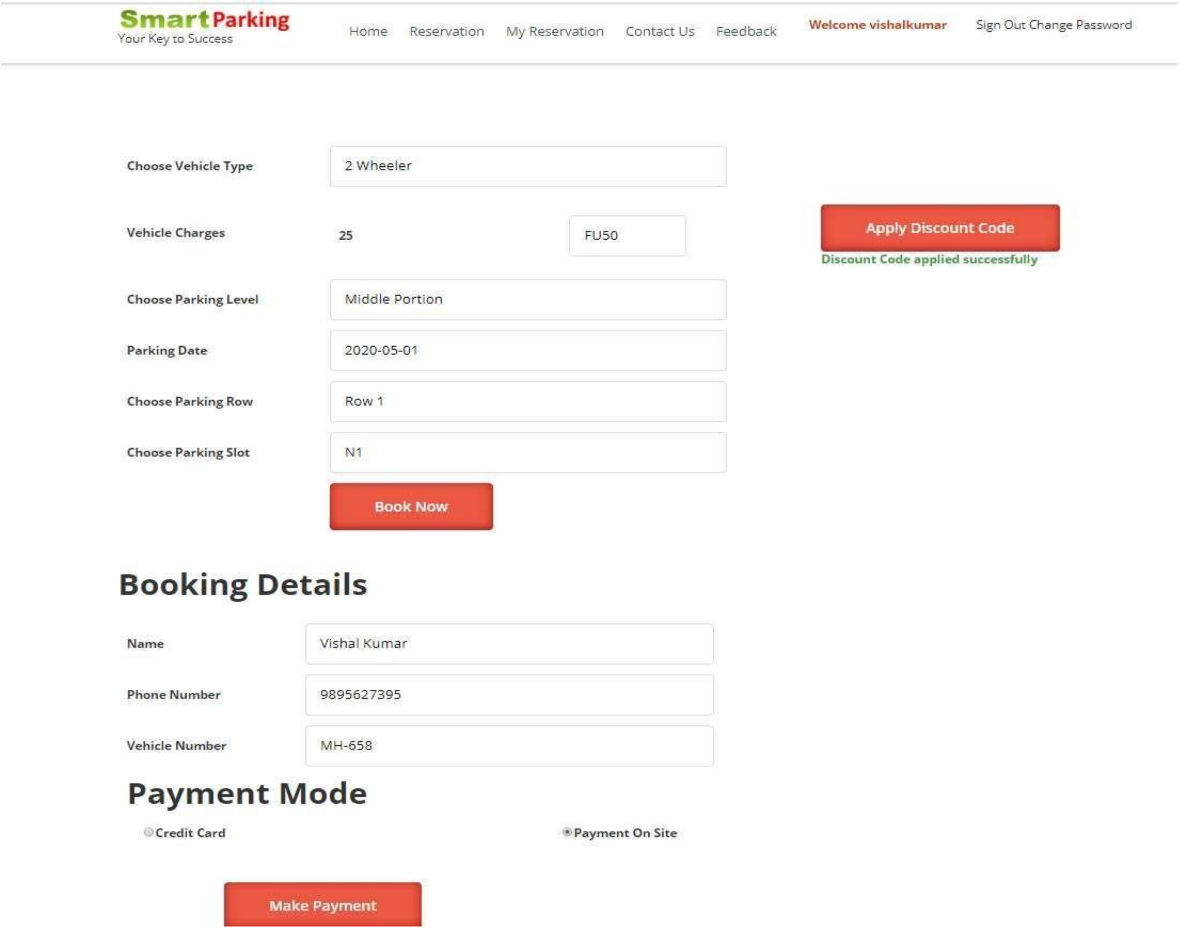
• Top Down is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that. This approach is taken when top-down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.

• Bottom Up is an approach to Integration Testing where bottom level units are tested first and upper-level units' step by step after that. This approach is taken when bottom-up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.

• Big Bang is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle.

• Hybrid is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

- Form-1: - User Reservation Form:



Choose Vehicle Type	2 Wheeler	
Vehicle Charges	25	FU50
Choose Parking Level	Middle Portion	
Parking Date	2020-05-01	
Choose Parking Row	Row 1	
Choose Parking Slot	N1	

Discount Code applied successfully

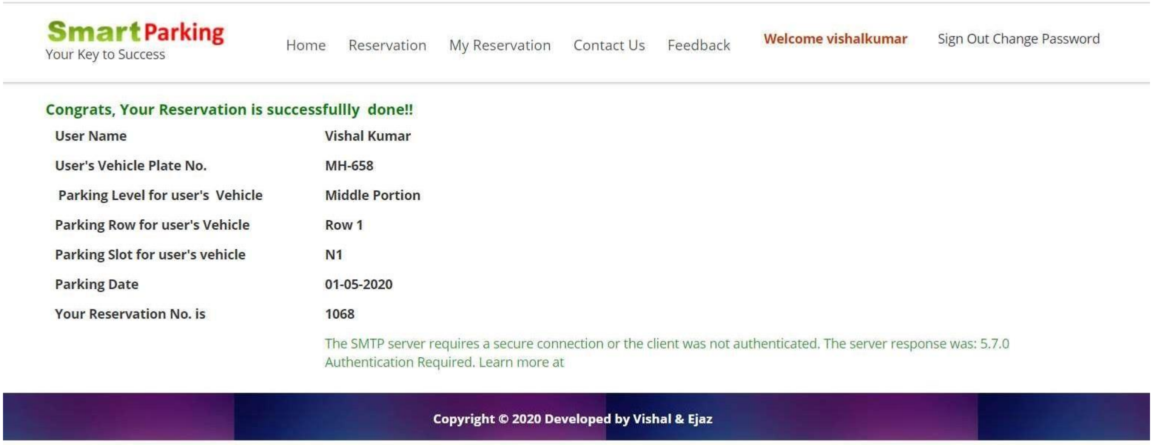
Booking Details

Name	Vishal Kumar
Phone Number	9895627395
Vehicle Number	MH-658

Payment Mode

○ Credit Card ● Payment On Site

- Form-2: - After Successfully Reservation



Congrats, Your Reservation is successfullly done!!

User Name	Vishal Kumar
User's Vehicle Plate No.	MH-658
Parking Level for user's Vehicle	Middle Portion
Parking Row for user's Vehicle	Row 1
Parking Slot for user's vehicle	N1
Parking Date	01-05-2020
Your Reservation No. is	1068

The SMTP server requires a secure connection or the client was not authenticated. The server response was: 5.7.0 Authentication Required. Learn more at

5.3.3 Beta Testing: -

Beta Testing is one of the Acceptance Testing types, which adds value to the product as the end user (intended real user) validates the product for functionality, usability, reliability, and compatibility. Inputs provided by the end-users helps in enhancing the quality of the product further and leads to its success. This also helps in decision making to invest further in the future products or the same product for improvisation. Since Beta Testing happens at the end user's side, it cannot be the controlled activity.

Beta Testing is one of the Customer Validation methodologies to evaluate the level of customer satisfaction with the product by letting it to be validated by the end users, who actually use it, for over a period of time. Product experience gained by the end users are asked for feedback on design, functionality, and usability and this helps in assessing the quality of the product.

Purpose of beta testing: -

The points mentioned below can even be considered as the objectives for Beta Test and are very much required to produce far better results for a product-

1. Beta Test provides a complete overview of the true experience gained by the end users while experiencing the product.

2. It is performed by a wide range of users and the reasons for which the product is being used varies highly. Marketing managers focus on target market's opinion on each and every feature,

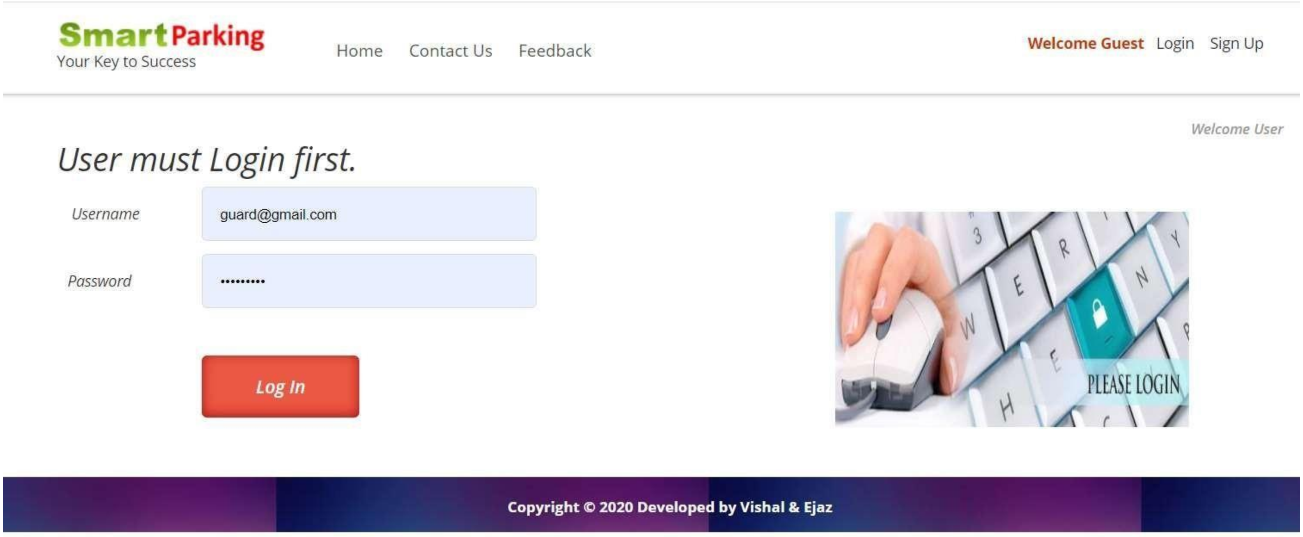
while a usability engineer / common real user focus on product usage and easiness, technical users focus on installation and uninstallation experience, etc.

3. Real world compatibility for a product can be ensured to a greater extent through this testing, as a great combination of real platforms is used here for testing on a wide range of devices, OS, Browsers, etc.

4. As a wide range of platforms which the end users are actually using, might not be available to the internal testing team during the QA, this testing also helps to uncover the hidden bugs and gaps in the final product.

5. Few specific platforms will cause the product to fail with showstopper bug which was not covered during QA. And this helps in improvising/fixing the product to be a compatible one with all possible platforms.

‘User Login’ Form Beta Testing: -



SmartParking
Your Key to Success

Home Contact Us Feedback

Welcome Guest Login Sign Up

Welcome User

User must Login first.

Username: guard@gmail.com

Password: •••••••••

Copyright © 2020 Developed by Vishal & Ejaz

CHAPTER – 6

RESULTS AND DISCUSSION

6.1 TEST REPORTS:

• Home Page: -

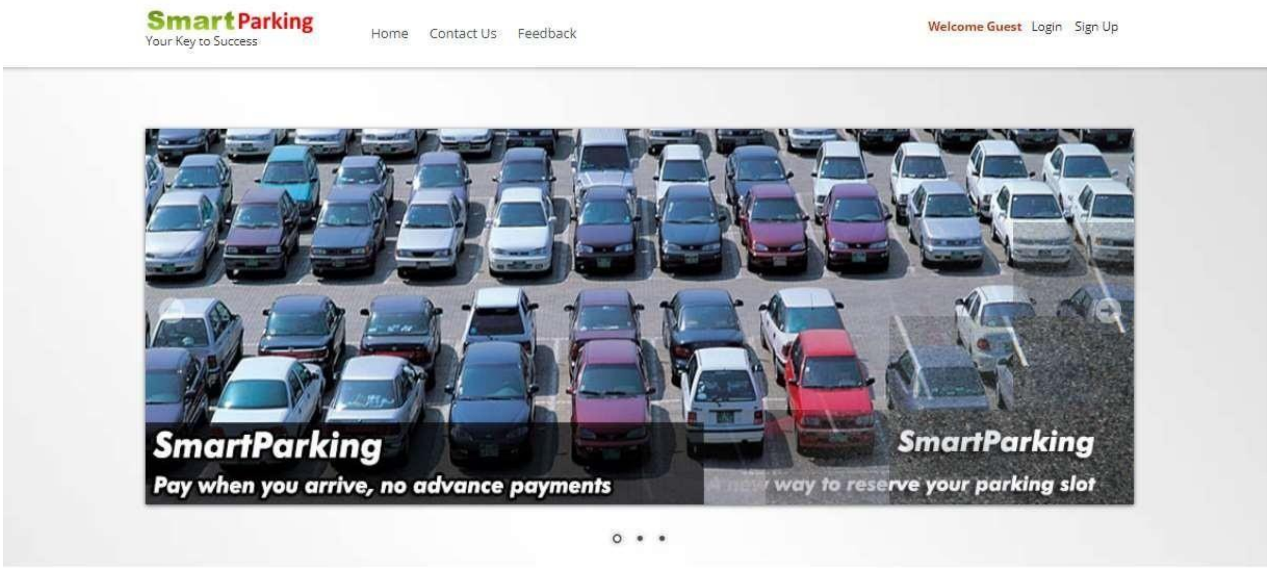
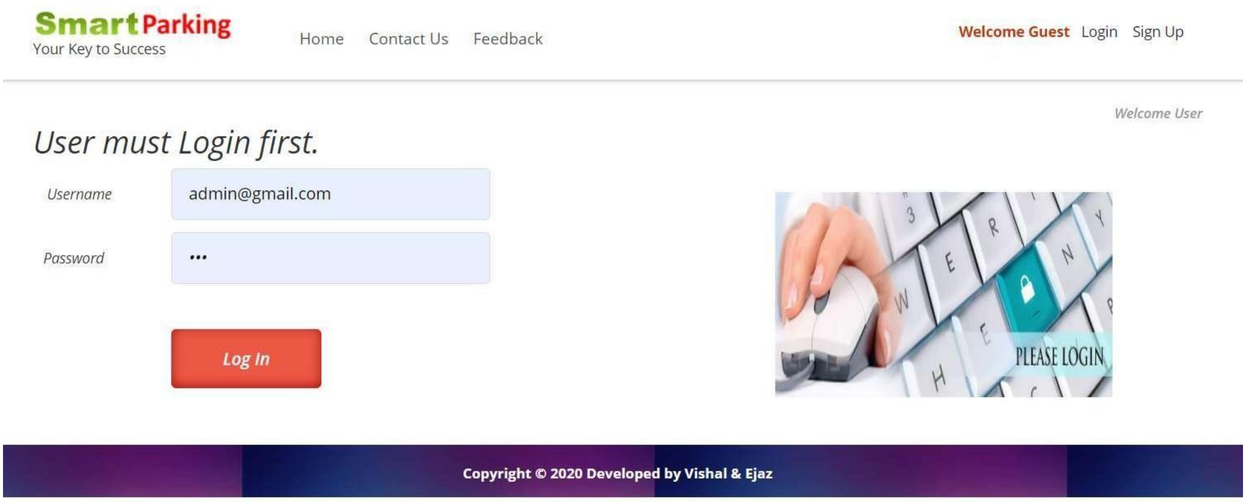


Fig. 6.1 – Homepage

- **Admin Module: -**



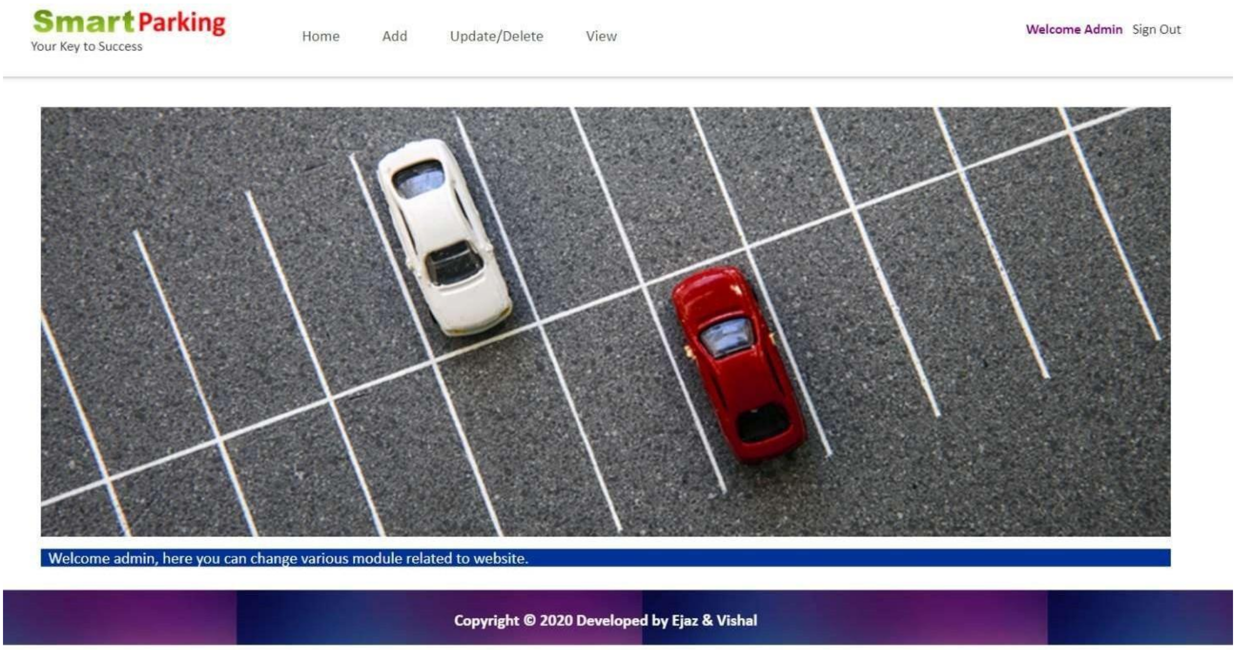
User must Login first.

Username admin@gmail.com

Password •••

Copyright © 2020 Developed by Vishal & Ejaz

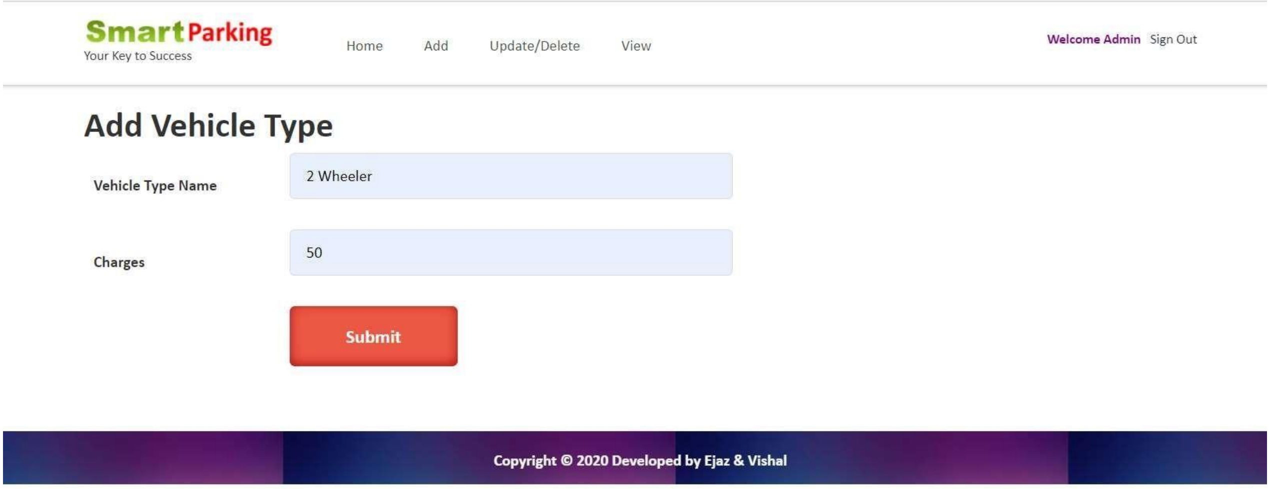
Fig. 6.2 - Admin Login



Welcome admin, here you can change various module related to website.

Copyright © 2020 Developed by Ejaz & Vishal

Fig. 6.3 - Admin Homepage

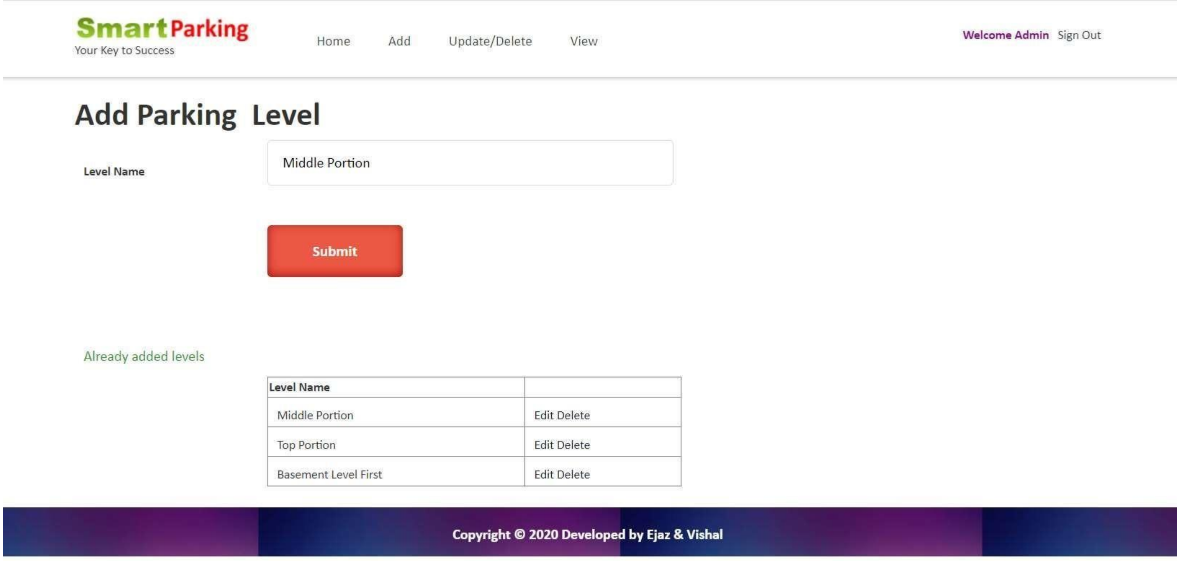


Add Vehicle Type

Vehicle Type Name: 2 Wheeler

Charges: 50

Fig 6.4 – Add Vehicle Type



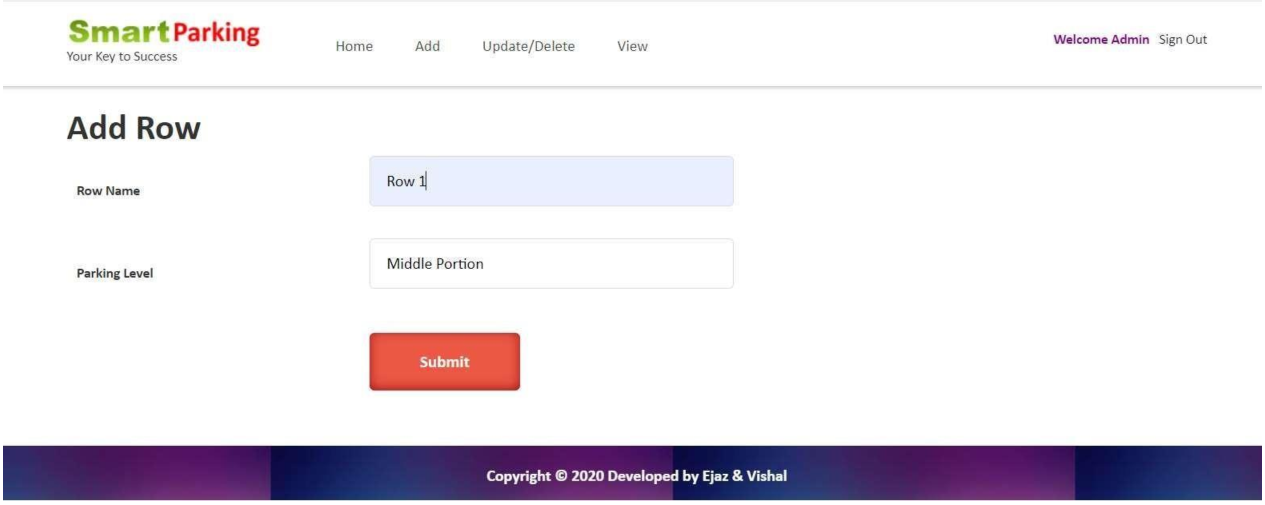
Add Parking Level

Level Name: Middle Portion

Already added levels

Level Name	
Middle Portion	Edit Delete
Top Portion	Edit Delete
Basement Level First	Edit Delete

Fig 6.5 – Add Parking Level

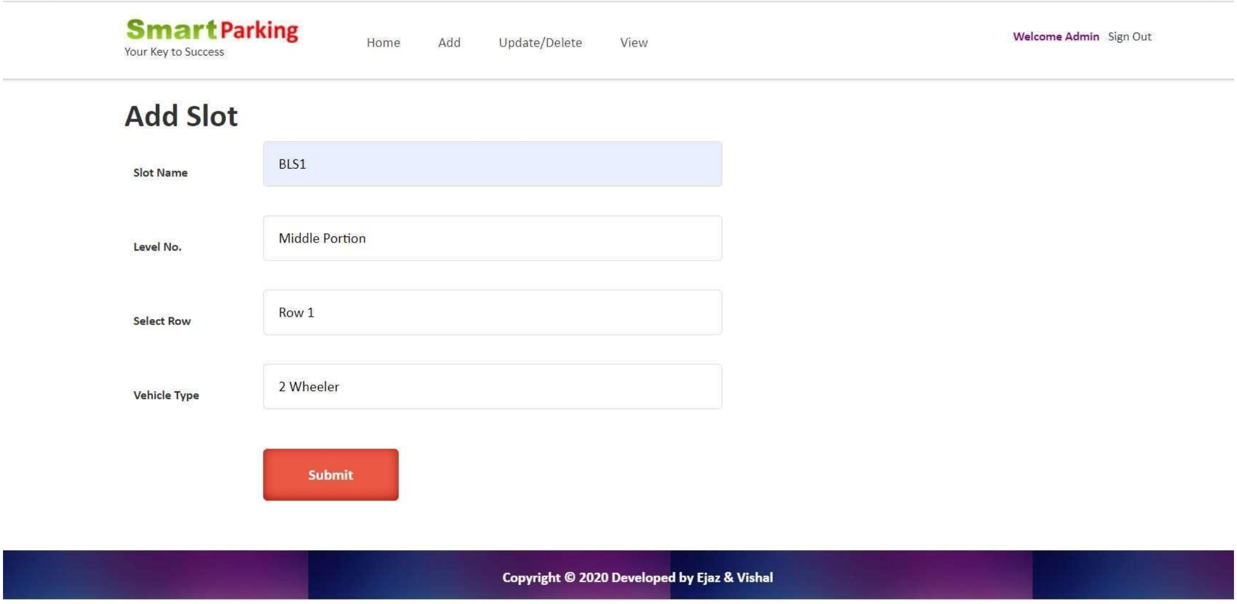


Add Row

Row Name: Row 1

Parking Level: Middle Portion

Fig 6.6 – Add Row



Add Slot

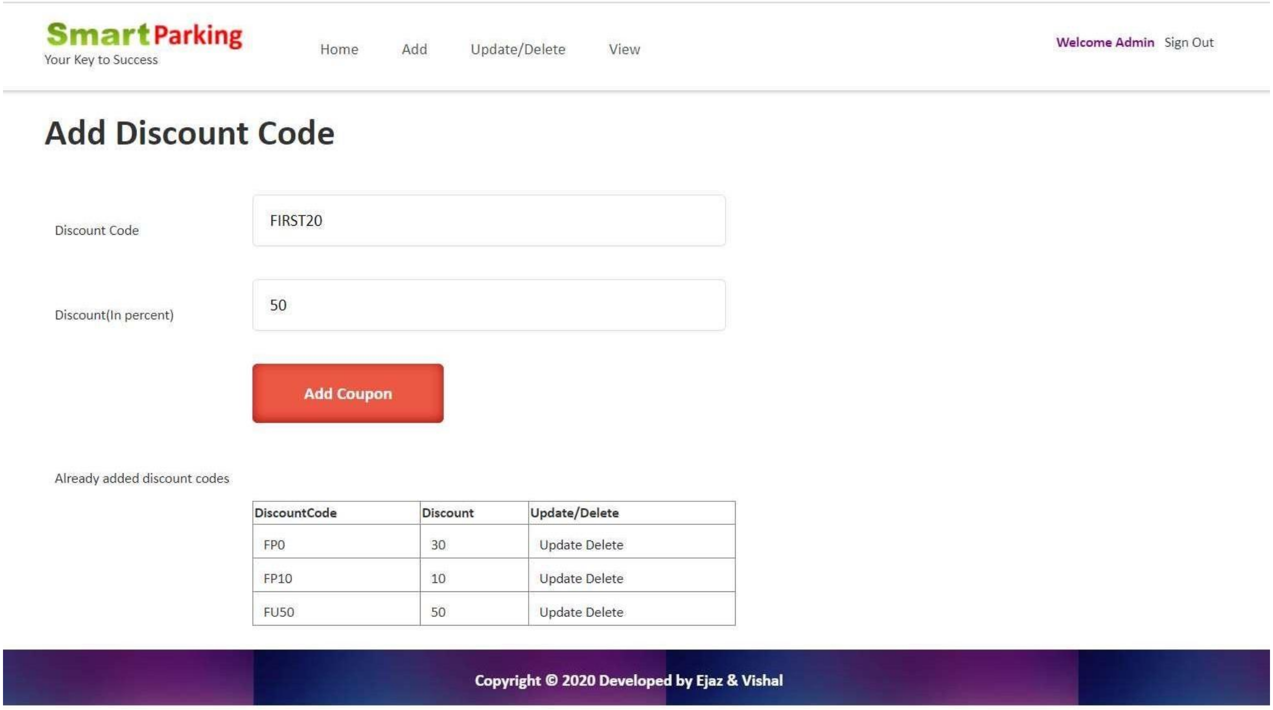
Slot Name: BLS1

Level No.: Middle Portion

Select Row: Row 1

Vehicle Type: 2 Wheeler

Fig 6.7 – Add Slot



Add Discount Code

Discount Code: FIRST20

Discount(In percent): 50

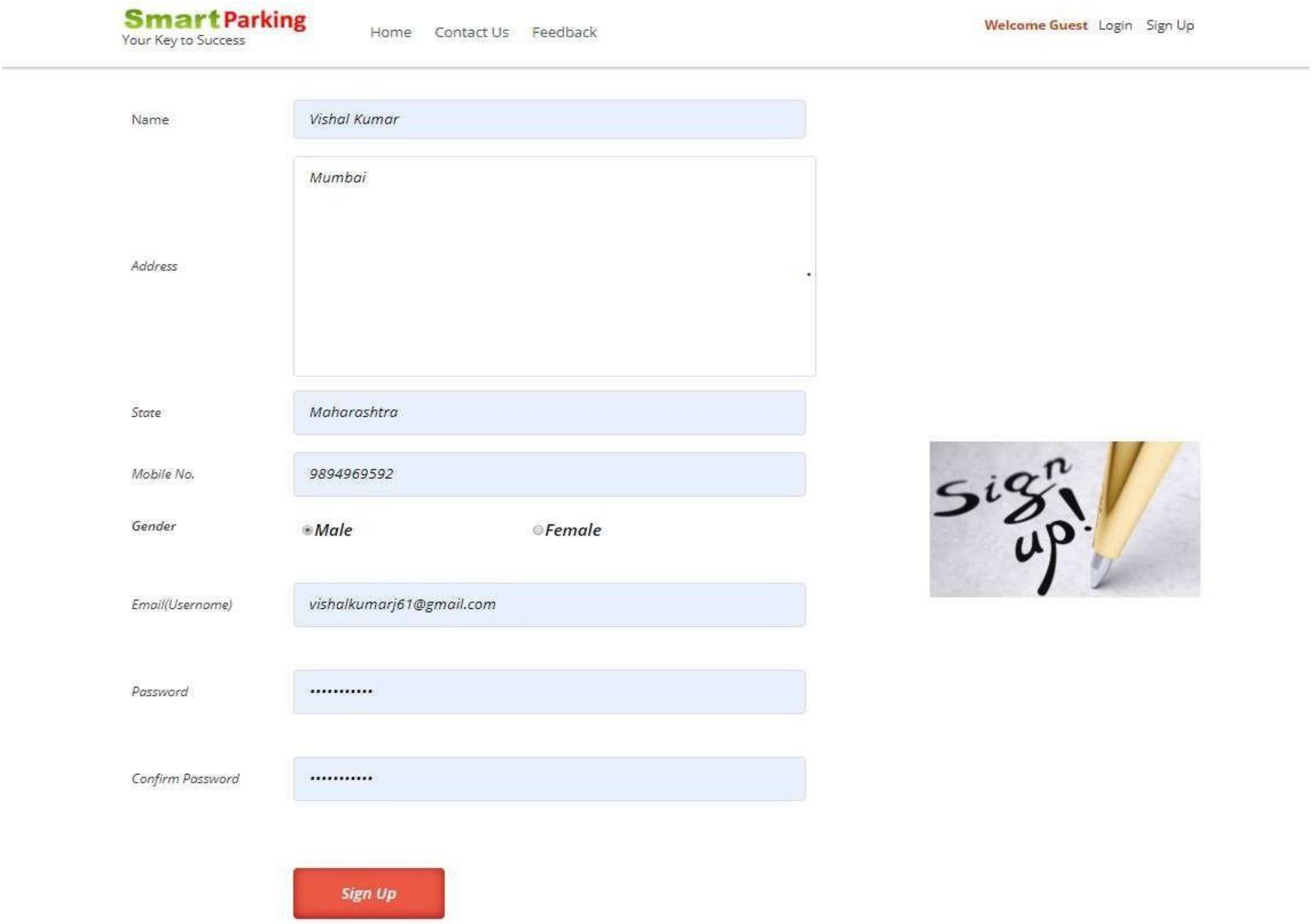
Already added discount codes

DiscountCode	Discount	Update/Delete
FP0	30	Update Delete
FP10	10	Update Delete
FU50	50	Update Delete

Copyright © 2020 Developed by Ejaz & Vishal

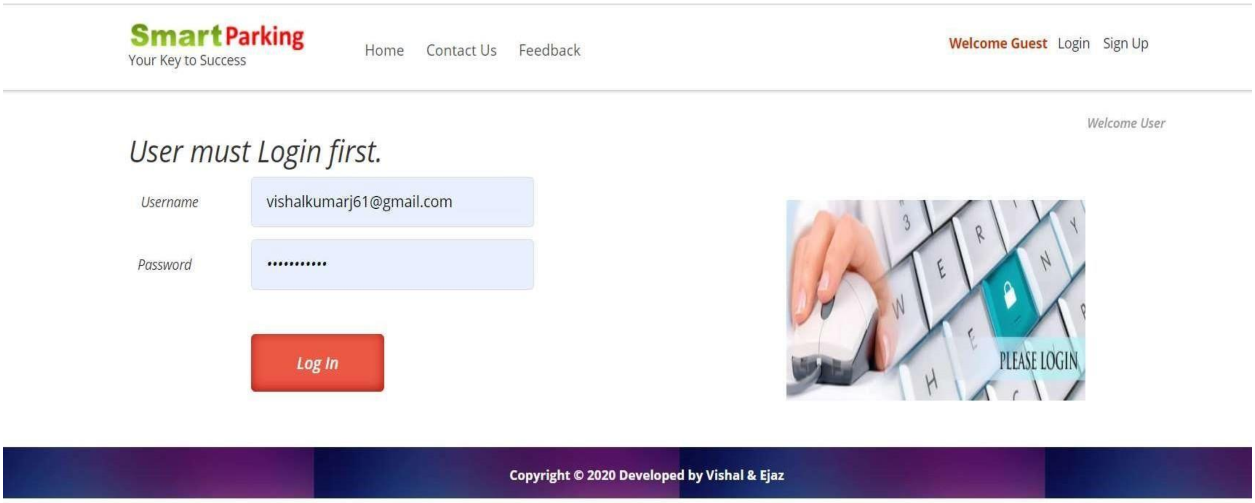
Fig 6.8 – Add Discount Code

User Module: -



Name	Vishal Kumar
Address	Mumbai
State	Maharashtra
Mobile No.	9894969592
Gender	● Male ○ Female
Email(Username)	vishalkumarj61@gmail.com
Password	•••••••••••
Confirm Password	•••••••••••

Fig. 6.9 – User Registration Page



User must Login first.

Username	vishalkumarj61@gmail.com
Password	•••••••••••

Copyright © 2020 Developed by Vishal & Ejaz

Fig. 6.10 – User Login Page

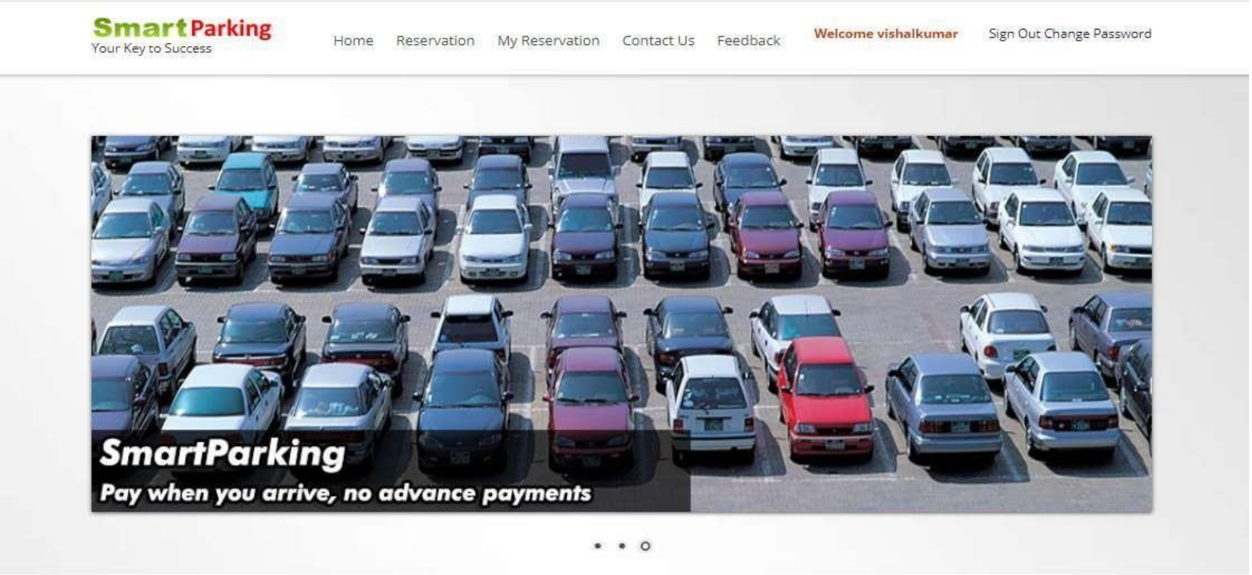
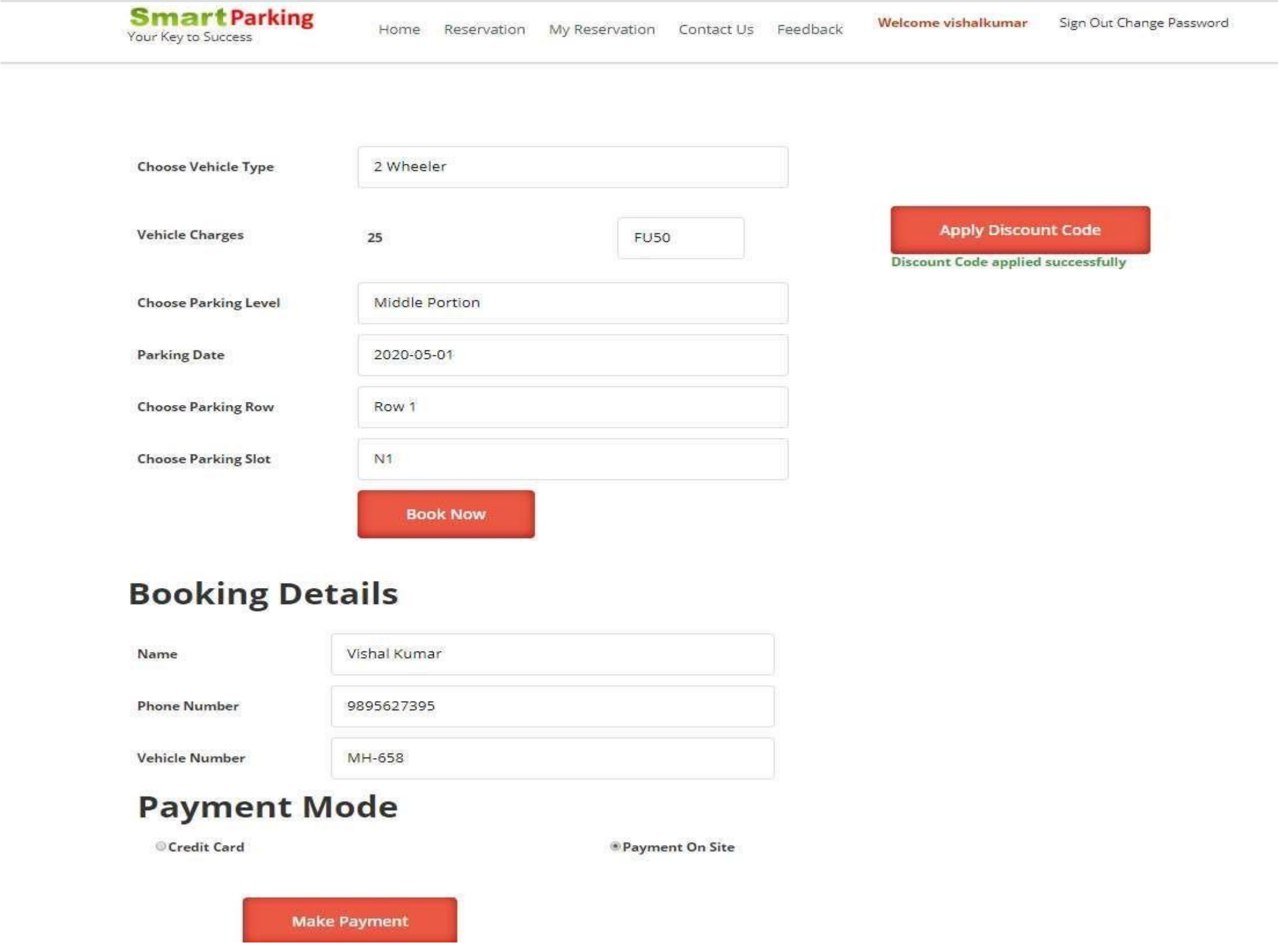


Fig. 6.11 – User Home Page



Choose Vehicle Type	2 Wheeler	
Vehicle Charges	25	FU50
Choose Parking Level	Middle Portion	
Parking Date	2020-05-01	
Choose Parking Row	Row 1	
Choose Parking Slot	N1	

Discount Code applied successfully

Booking Details

Name	Vishal Kumar
Phone Number	9895627395
Vehicle Number	MH-658

Payment Mode

○ Credit Card ● Payment On Site

Fig. 6.12 – Reservation Page

6.2 USER DOCUMENTATION:

User documentation refers to the documentation for a product or service provided to the end users. The user documentation is designed to assist end users to use the product or service. This is often referred to as user assistance. The user documentation is a part of the overall product delivered to the customer.

User documentation is important because it provides an avenue for users to learn: -

1. how to use your software

2. features of your software

3. tips and tricks of your software

4. how to resolve common problems with your software

User Documentation part consist of the description about all the forms that the web application contains, they are as follows: -

Fig. 6.1 – Homepage: - Home page consist of the different tabs such as 'Login', 'User Registration', 'Feedback', 'Contact Us', 'Sign Out'.

Fig. 6.2 - Admin Login Page: - To login successfully into the system as an administrator one has to enter a valid 'Username' and 'Password'.

Fig. 6.3 - Admin Homepage: - After successfully logging in as a system admin, admin can access the Admin Homepage. The Admin Homepage consists of tabs such as 'Home', 'Add', 'Update/Delete' and 'View '.

Fig. 6.4 – Add Vehicle Type: - In the 'Add Vehicle Type Page, admin can add different Vehicle with its description whenever required helping the users to find their favorable Vehicle more efficiently.

Fig. 6.5 – Add Parking Level: - In 'Add Parking Level' Page admin can add different Parking level such as Middle Portion, Top Portion, etc.

Fig. 6.6 – Add Row: - In the 'Add Row' Page admin can add different Row such as Row 1, Row 2, etc.

Fig. 6.7 – Add Slot: - In the 'Add Slot' Page admin can add different Slots such as TLS1, BLS1, etc.

Fig. 6.8 – Add Discount Code: - In the 'Add Discount Code' Page admin can add different Slots such as TLS1, BLS1, etc.

Fig. 6.9 – User Registration Page: - In the registration page, the user has to enter the following 8 details:

1. Name

2. Address

3. State

4. Mobile Number

5. Gender

6. Email Id (Username)

7. Password

8. Confirm Password

- All the fields are mandatory.

- Every user must have a unique Username.

- The user must enter the password in the ‘Confirm Password’ column in-order to successfully continue his/her registration.

Fig. 6.10 – User Login Page: - In the registration page, the user has to enter the following 2 details:

1.Username

2.Password

Fig. 6.11 – User Homepage: - After successfully logging in as an user, the User Homepage is displayed. It consists tabs such as ‘Home’, ’Reservation’, ’My Reservation’, ‘Contact Us’, ‘Feedback’, ’Sign Out’ and ’Change Password’.

Fig. 6.12 – Reservation Page: - In the Reservation Page, the user can book the reservation for his/her Vehicle.

CHAPTER-7:
CONCLUSION & REFERENCES

7.1 : - CONCLUSION:

This Project is minimizing the task of parking a vehicle by paying and saying some details about customer and vehicle to save data. In this the vehicle is parked as a safe and secure.

I also conclude that this project has helped us gain more knowledge about the topic that we are indulged ourselves into "Visual Studio". I would be glad to enhance and promote this project if given chance and help ourselves and society in the near future.

The developed application is tested with sample inputs and outputs obtained in according to the requirement. Even though I have tried our level best to make it a dream project. Due to time constraints I could not add more facilities to it.

The efficiency of the developed system can be enhanced with some minor modifications. Future development can be made in proposed system by integration more services like:

- It can be implemented through web pages.
- New effectives modules can be added time to time

7.2 : - FUTURE WORK:

This is the modern age. Many people have vehicles. Vehicle is now a basic need. Every place is under the process of urbanization. There are many corporate offices and shopping centers etc. There are many recreational places where people used to go for refreshment. So, all these places need a parking space where people can park their vehicles safely and easily. Every parking area needs a system that records the detail of vehicles to give the facility. These systems might be computerized or non-computerized. With the help of computerized system, we can deliver a good service to customer who wants to park their vehicle into the any organization's premises.

Enhancement to create a Bigger and Better System

These enhancements deal with what would be required in a new improved, bigger and better system

- In future if when a vehicle enters into the parking area there should be one sensor in which the user can easily identify from outside only Is there parking is full or empty or space is allocated.
- In future the vehicle can be parked by machines

7.3 : - REFERENCES:

1) FOR .NET INSTALLATION:

Install Microsoft Visual Web Express 2010 Edition or Visual Studio 2010.

https://visualstudio.microsoft.com/vs/older-downloads/

2) FOR SQL:

Update your database by installing Microsoft SQL Server 2008 Express Edition RC2 update .
http://download.microsoft.com/download/0/4/B/04BE03CD-EAF3-4797-9D8D-2E08E316C998/SQLEXPR_x64_ENU.exe

3) https://www.google.com

4) https://www.wikipedia.com

5) https://www.youtube.com/