A PROJECT REPORT ON

**Manan Car PARKING SYSTEM**

Submitted

By

**Mr. Manan Prakash Datta**

**(Enrollment No.03052101388)**

In fulfillment for the award of the degree

Of

**Bachelor of Computer Application**

Guided by

**Mr. Thakrar Zalak**

Shri V J Modha College of IT and Management – Porbandar

Bhakta Kavi Narsinh Mehta University, Junagadh

**Academic Year**

**2021 - 2024**



**Acknowledgement**

During the preparation of the project, we have a good fortune of receiving support, in various ways, from several personal, numerous to mention here. We owe a debt of gratitude to all of them.

It is our privilege to express our sincerest regards to our project coordinator, Prof. ZALAK THAKRAR for their valuable inputs, able guidance, Encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project.

It is our great pleasure to represent our project as one desktop application titled “MANAN CAR PARKING SYSTEM” and which we conceived in the 5th semester of BCA affiliated with BKNMU (Bhakt Kavi Narsinh Mehta University).

We are also thankful to the BKNMU (Bhakt Kavi Narsinh Mehta University) for including this project development subject in our syllabus. We got a golden opportunity to test and implement our creativity and programming skill simultaneously. Lastly, we would like to extend our sincere thanks to our advisors, classmates as well as all the books and websites who have directly or indirectly helped us.

**Preface**

This Desktop Application Provide efficient, reliable way to manage the parking, manage parking slots.

* Manage Parking Space
* Manage Parking Slots
* Assing slots
* Remove Slots
* Reports

 This Software made with C# Language and Ms Access Database.

 This Software has only Access Right to ADMIN.

 Customer Role is to Receive Receipt.

**INDEX**

Contents

[Chapter 1 Introduction 8](#_Toc144475275)

[ 1.1 Purpose 9](#_Toc144475276)

[ 1.2 Scope 10](#_Toc144475277)

[ 1.3 Technology & Literature Review 11](#_Toc144475278)

[Chapter: 2 System Analysis 13](#_Toc144475279)

[ 2.1 Problem Definition 14](#_Toc144475280)

[ 2.2 Process Model 15](#_Toc144475281)

[ 2.3 Requirement Analysis 17](#_Toc144475283)

[ 2.3.1 Hardware 17](#_Toc144475284)

[ 2.3.2 Software Requirement 18](#_Toc144475285)

[ 2.4 SRS 19](#_Toc144475286)

[ 2.5 Grant Chart 26](#_Toc144475287)

[Chapter 3 System Desing 27](#_Toc144475288)

[ 3.1 DFD Diagram 28](#_Toc144475289)

[ 3.1.1 Level 0 28](#_Toc144475290)

[ 3.1.2 Level 1 29](#_Toc144475292)

[ 3.2 ER Diagram 30](#_Toc144475293)

[ 3.3 Use Case Diagram 31](#_Toc144475294)

[Chapter 4 Data Dictionary 32](#_Toc144475295)

[ 4.1 Data Dictionary 33](#_Toc144475296)

[ 4.1.1 Login 34](#_Toc144475297)

[ 4.1.2 Car Parking Report 35](#_Toc144475298)

[ 4.1.3 Space List 36](#_Toc144475299)

[Chapter 5 Input & Output Design 37](#_Toc144475300)

[ 5.1 Admin Layout 38](#_Toc144475301)

[ 5.1.1 Loading 38](#_Toc144475302)

[ 5.1.2 Login 39](#_Toc144475303)

[ 5.1.3 Home Page 42](#_Toc144475305)

[ 5.1.4 Parking Space 45](#_Toc144475306)

[ 5.1.5 Level 0 47](#_Toc144475307)

[ 5.1.6 Level 1 49](#_Toc144475308)

[ 5.1.7 Assing Car Parking 51](#_Toc144475309)

[ 5.1.8 Assing Car Parking Exit 55](#_Toc144475310)

[ 5.1.9 Receipt 60](#_Toc144475311)

[ 5.1.10 Space List 63](#_Toc144475312)

[ 5.1.11 Parking Report 65](#_Toc144475314)

[Chapter 6 Limitation & Future Enhancement 67](#_Toc144475315)

[ 6.1 Limitation 68](#_Toc144475316)

[ 6.2 Future Enhancement 69](#_Toc144475317)

[Chapter 7 Conclusion 70](#_Toc144475318)

[ 7.1 Conclusion 71](#_Toc144475319)

[ 7.2 Advantages 72](#_Toc144475320)

[Chapter 8 Bibliography 73](#_Toc144475321)

[ 8.1 Bibliography 74](#_Toc144475322)

[Chapter 9 Reference 75](#_Toc144475323)

[ 9.1 References 76](#_Toc144475324)

# 

# Chapter 1 Introduction

**1.1Purpose**

**1.2 Scope**

* 1. **Technology and Literature Review**

## 1.1 Purpose

* The main purpose for creating this is to managing the entry & exits of car from parking

area. This project is helpful for the company who provide paid parking facility.

* This project can also help to manage paid parking facility.
* This project contains many modules which makes backend powerful.
* This project helps the administrator to administrate the parking easily.
* This Software Make an Easy way to Manage Parking, Car Information, Parking Space,

Parking Slots.

* This software provides a receipt to the user.

## 1.2 Scope

* The scope of this software is to provide an easy option for the client

who is willing to Digitalis Management of their Parking.

* It saves their Time.
* This software can be accessed from desktop & laptop, thus providing

client’s comfort.

* Considering the benefits of the client, the software has also some

additional features.

* **The goals of the software are:**
* To provide a better management system then manual management system.
* To handle the parking system digitally.
* To reduce manual work.
* To provide digitalis management system.

## 1.3 Technology & Literature Review

* **Front End:**
* **C# .NET**
* C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run in the .NET ecosystem.
* The .NET ecosystem is composed of all the implementations of .NET, including both but not limited to .NET Core, and .NET Framework.
* You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more.
* **Backend:**
* **MS ACCESS**
* Microsoft Access, often abbreviated as MS Access, is a relational database management system (RDBMS) developed by Microsoft. It is a part of the Microsoft Office suite of applications and is used to create and manage databases. Access allows users to store, organize, and manipulate large amounts of data in a structured manner.
* With Access, you can design tables to store your data, create forms for data entry and display, design queries to extract specific information, and generate reports for presenting data in a formatted way. It provides a user-friendly interface that doesn't require extensive programming knowledge, making it accessible to a wide range of users.
* Access uses the Structured Query Language (SQL) to interact with databases, which enables users to perform operations like retrieving, updating, and deleting data. It's often used for smaller-scale applications, departmental solutions, and projects that require a relatively simple database setup and management. For more complex and enterprise-level applications, organizations might opt for more robust database systems like Microsoft SQL Server or Oracle.
* MS Access is a database management system developed by Microsoft. It allows users to create and manage databases, queries, forms, reports, macros, and modules. Some details on MS Access are:
* MS Access is part of the Microsoft Office suite of applications, along with Word, Excel, PowerPoint, and Outlook.
* MS Access uses the Jet Database Engine to store data in a relational format. It also supports SQL queries and VBA programming.
* MS Access can connect to various data sources, such as Excel files, text files, ODBC data sources, SharePoint lists, and web services.
* MS Access can create desktop applications or web applications that can be published to SharePoint or Azure websites.
* MS Access has various features that help users design and customize databases, such as templates, wizards, forms, reports, charts, and controls.

# Chapter: 2 System Analysis

**2.1 Problem Definition**

**2.2 Process Model**

**2.3Requirement Analysis**

**2.4 SRS**

**2.5 Grant Chart**

## 2.1 Problem Definition

* A parking system is a solution that manages the allocation, availability, and payment

for parking spaces in a controlled area, such as a parking spaces in a controlled area,

such as a parking lot or garage. It typically involves technologies like sensors, cameras,

and software to streamline the parking process. The problem definition could focus on

on optimizing space utilization, minimizing congestion, enhancing user experience,

and facilitating efficient payment methods. Do you need more specific details or

assistance with anything else related to this?

## 2.2 Process Model

**Iterative Waterfall Model**

## 

## 

 Advantages of Iterative Waterfall Model

 Simple and Easy to Understand and Each Phase has well Defined Input And Output

 It Work well for Smaller Project Where Requirement Are Clear And very well understood

 It Divide complex task into more manageable works.

 Application of Iterative Waterfall Model

1. This Model is used When Requirements are clear And Fix

2. Product Definition is Stable & Technology is understood & it used when Project is short.

 Why Iterative Waterfall Model??

* Online Photography registering website is a large system with all functionality and specification.
* ITERATIVE WATERFALL Model is used for development process of online Photography registering website.
* The incremental Model is an evolution of the waterfall model, where the waterfall model incrementally applied

## 2.3 Requirement Analysis

### 2.3.1 Hardware

* + Operating System: 64bit
  + RAM: 4 GB

### 2.3.2 Software Requirement

* + Front End Tool: c# .NET
  + Back End Tool: MS ACCESS (.accdb)
  + Development Tool: Visual Studio 2010 (ultimate)
  + Supported Operating Systems:

 Windows 7 (32-bit/64-bit)

 Windows 8 (32-bit/64-bit)

 Windows 10 (32-bit/64-bit).

## 2.4 SRS

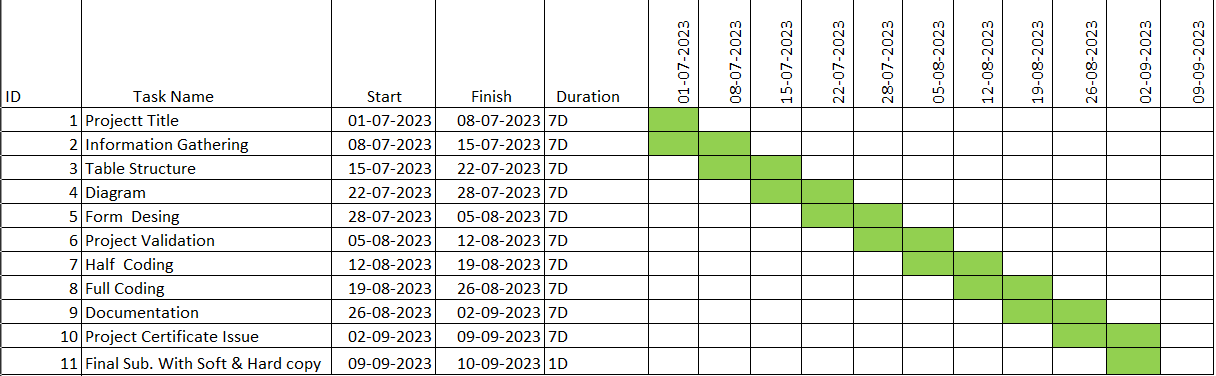
* R1: Login

* R2: Parking Space
* R3: Parking Slots
* R4: Allot Parking
* R5: Space Report
* R6: Parking Report
* R1:LOGIN
* This is Login Form.
* Through this Admin can login into software.
* I have put authentication in this form.
* Admin has to enter User Name & Password.
* R2: Parking Space
* This page is for displaying the available spaces.
* And also be able to display the capacity of the parking space.
* This form shows the levels of the parking.
* This helps to indicate the floor in which the car is to be park.
* R3: Parking Slots
* This form shows the parking slots of all the level of parking space.
* This for selecting the slot for the parking.
* This form allow admin to know which slot is empty or occupied.
* R4: Allot Parking
* This is page is to allot the parking slot.
* This page is for registering the vehicle entry date & time of the car.
* This is to insert the data of the client.
* R5: Space Report

* This is used to display the parking space list.
* This is parking space report page.
* This displays the data in Data Grid View.
* R6: Parking Report

* This is used to display the whole parking slots list.
* This is parking report page.
* This displays the data in Data Grid View.

## 2.5 Grant Chart



# Chapter 3 System Desing

**3.1 DFD Diagram**

**3.2 ER Diagram**

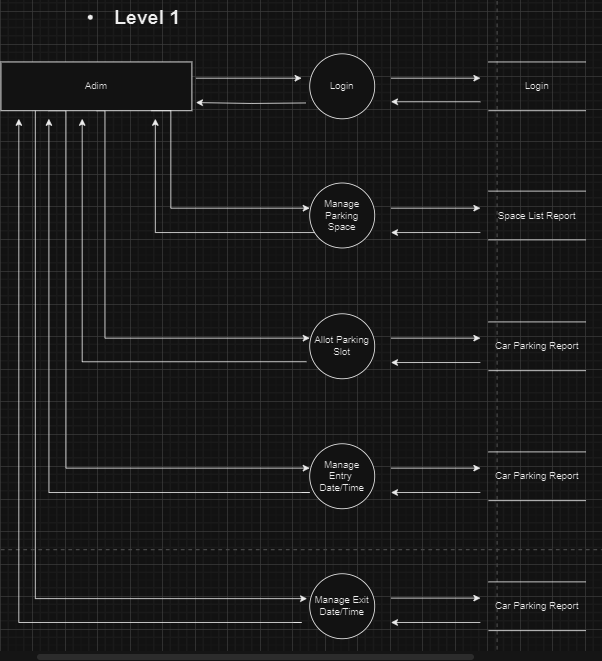
**3.3 Use Case**

## 3.1 DFD Diagram

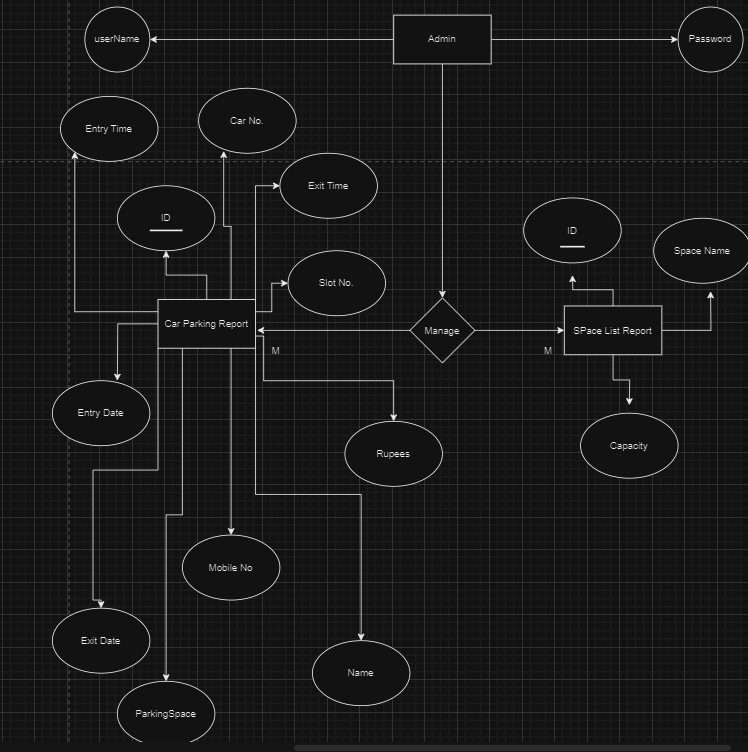
### 3.1.1 Level 0

## 

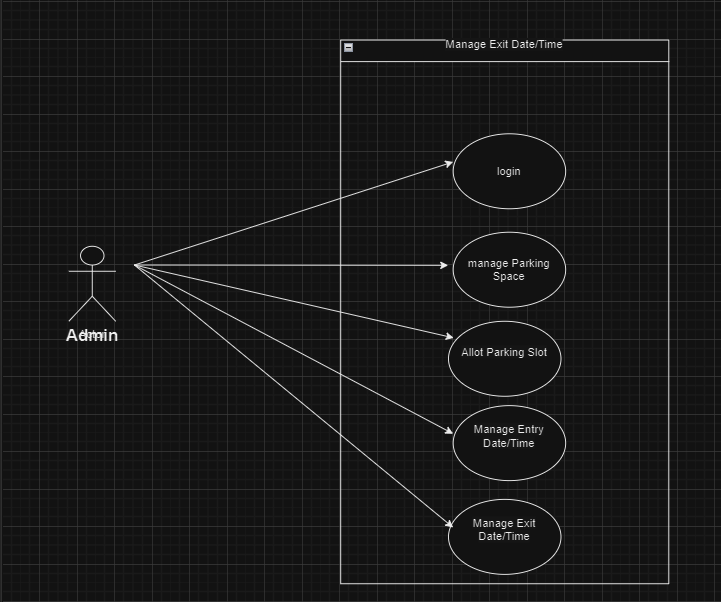
### 3.1.2 Level 1



## 3.2 ER Diagram



## 3.3 Use Case Diagram



# Chapter 4 Data Dictionary

**4.1 Data Dictionary**

**4.1.1 Login**

**4.1.2 Car Parking Report**

**4.1.3 Space List**

## 4.1 Data Dictionary

* DATABASE: ACCESS
* DATABASE NAME: PROJECT
* NO OF TABLES: 3
* TABLE NAME:
* Login
* Car Parking Report
* Space List

### 4.1.1 Login

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| ID | AUTONUMBER | This field is use to indicate the serial no. |
| USER NAME | TEXT | This field is use to display the user - name. |
| PASSWORD | TEXT | This field is use to display the password of the user. |

* This is a login table.
* This table stores the data of User Name and Password.

### 

### 4.1.2 Car Parking Report

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| ID | AUTONUMBER | This field is use to indicate the serial no. |
| PARKING SPACE | TEXT | This field is use to  Indicate the parking space. |
| CAR NO | NUMBER | This field is use to display the car number. |
| ENTRY DATE | DATE/TIME | This field is use to display the entry date and time. |
| EXIT DATE | DATE/TIME | This field is use to display the exit date and time. |
| SLOT NO | NUMBER | This field is use to display the slot no. |
| NAME | TEXT | This field is use to display the name of the owner. |
| CONTACT NO | NUMBER | This field is use to display the mobile no of the owner |
| ENTRY TIME | DATE/TIME | This field is use to display the entry date and time. |
| EXIT TIME | DATE/TIME | This field is use to display the exit time. |
| RUPEES | NUMBER | This field is for Rupees. |

* This is a Car Parking Report table.
* This table stores the data of car which is stored in parking lot.

### 4.1.3 Space List

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| ID | AUTONUMBER | This field is use to indicate the serial no. |
| SPACE NAME | TEXT | This field is use to display the car parking space name. |
| CAPACITY | NUMBER | This field is use to display the capacity of the parking spaces. |

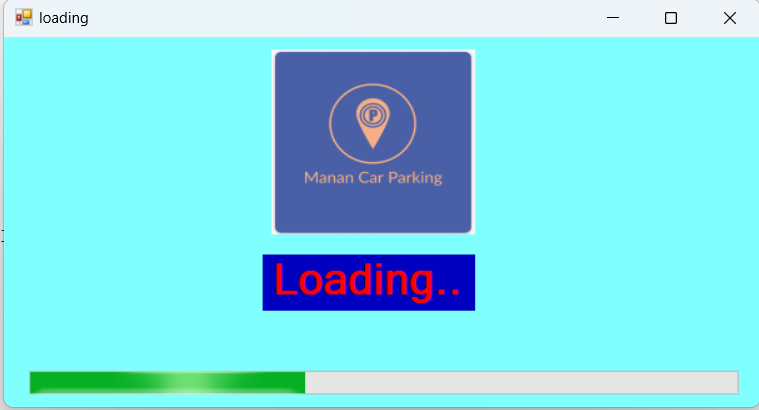
* This is a Space List table.
* This table stores the data of car which is stored in parking space.

# Chapter 5 Input & Output Design

**5.1 Admin Layout**

## 5.1 Admin Layout

### 5.1.1 Loading



* This form shows the progress bar.
* This is the first form of when the program starts.

### 5.1.2 Login

## 

* This is the login form.
* This form is for the login process of the admin.

private void button1\_Click(object sender, EventArgs e)

{

OleDbConnection connect = new OleDbConnection();

connect.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\PDF\B.C.A-5\PROJECT\Project.accdb";

connect.Open();

OleDbCommand command = new OleDbCommand();

command.Connection = connect;

command.CommandText = "select UserName,Password from Login";

OleDbDataReader reader = command.ExecuteReader();

while (reader.Read())

{

if (textBox1.Text == reader[0].ToString() && textBox2.Text == reader[1].ToString())

{

this.Hide();

Homepage hp = new Homepage();

hp.Show();

break;

}

else if (textBox1.Text == "" && textBox2.Text == "")

{

MessageBox.Show("Enter Data", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Your user ID or password is incorrect.");

textBox1.Clear();

textBox2.Clear();

}

}

count = count + 1;

if (count > 3)

{

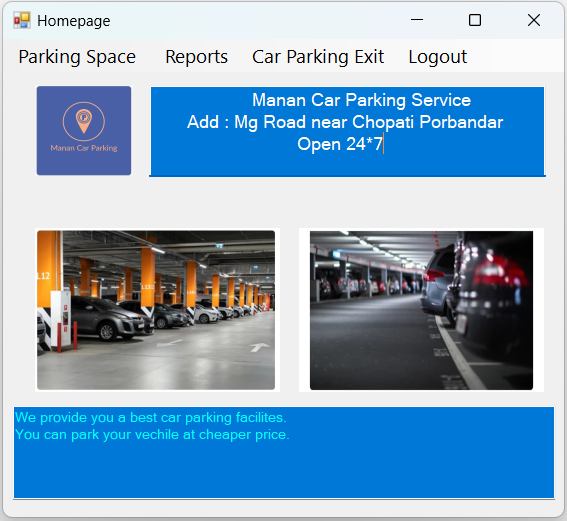
MessageBox.Show("System has been blocked");

Application.Exit();

}

}

### 5.1.3 Home Page



* This the homepage.
* From this form we can access all other forms.

private void allotParkingToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

ParkingSpace ps = new ParkingSpace();

ps.Show();

}

private void spaceReportToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

SpaceReport sr = new SpaceReport();

sr.Show();

}

private void parkingReportToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

ParkingReport pr = new ParkingReport();

pr.Show();

}

private void carParkingExitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

AssingCarParkingExit acpe = new AssingCarParkingExit();

acpe.Show();

}

private void logoutToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Login l = new Login();

l.Show();

}

### 5.1.4 Parking Space



* This form is of parking space.
* This shows the information about the levels of parking space available.

private void label4\_Click(object sender, EventArgs e)

private void label4\_Click(object sender, EventArgs e)

{

this.Hide();

Level0 lo = new Level0();

lo.Show();

Level0.lo.tbx.Text = label4.Text;

}

private void label6\_Click(object sender, EventArgs e)

{

this.Hide();

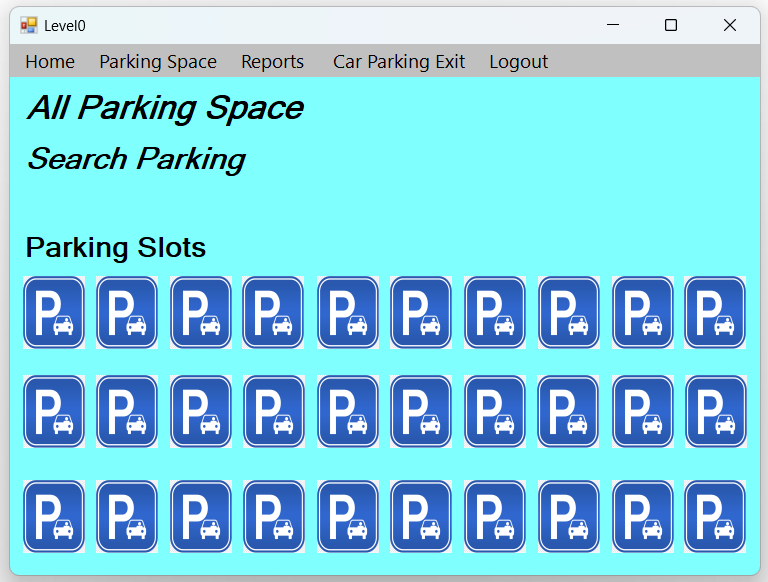
Level1 ll = new Level1();

ll.Show();

Level1.lo.tbx.Text = label6.Text;

}

### 5.1.5 Level 0



* This form shows the parking slots of the level 0.
* This is for selecting the parking slots available in that floor.

private void pictureBox1\_Click(object sender, EventArgs e)

{

this.Hide();

AssingCarParking acp = new AssingCarParking();

acp.Show();

AssingCarParking.acp.tbx.Text = textBox1.Text;

AssingCarParking.acp.t.Text = label3.Text;

pictureBox1.Enabled = false;

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

this.Hide();

AssingCarParking acp = new AssingCarParking();

acp.Show();

AssingCarParking.acp.tbx.Text = textBox1.Text;

AssingCarParking.acp.t.Text = label5.Text;

pictureBox2.Enabled = false;

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

this.Hide();

AssingCarParking acp = new AssingCarParking();

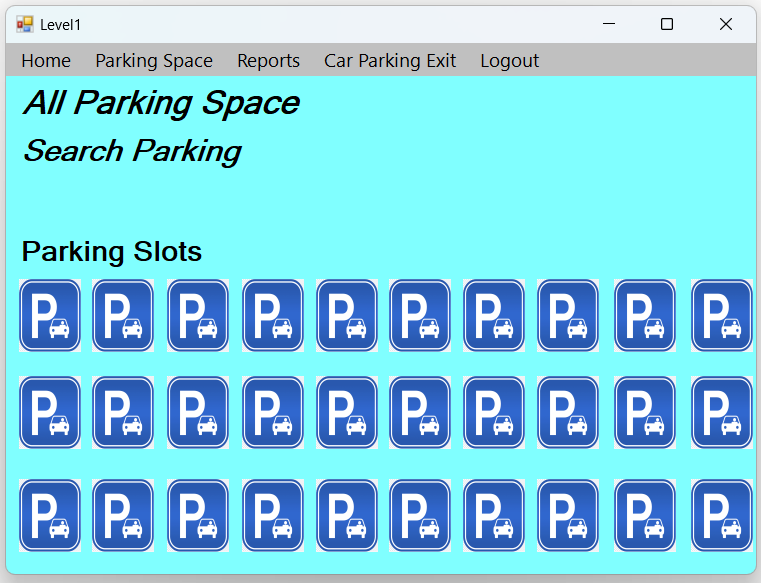
acp.Show();

AssingCarParking.acp.tbx.Text = textBox1.Text;

AssingCarParking.acp.t.Text = label6.Text;

pictureBox3.Enabled = false;)}

### 5.1.6 Level 1



* This form shows the parking slots of the level 1.
* This is for selecting the parking slots available in that floor.

private void pictureBox1\_Click(object sender, EventArgs e)

{

this.Hide();

AssingCarParking acp = new AssingCarParking();

acp.Show();

AssingCarParking.acp.tbx.Text = textBox1.Text;

AssingCarParking.acp.t.Text = label3.Text;

pictureBox1.Enabled = false;

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

this.Hide();

AssingCarParking acp = new AssingCarParking();

acp.Show();

AssingCarParking.acp.tbx.Text = textBox1.Text;

AssingCarParking.acp.t.Text = label5.Text;

pictureBox2.Enabled = false;

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

this.Hide();

AssingCarParking acp = new AssingCarParking();

acp.Show();

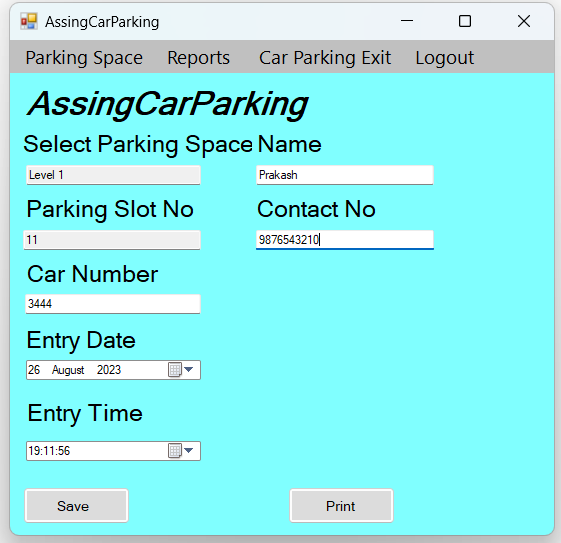
AssingCarParking.acp.tbx.Text = textBox1.Text;

AssingCarParking.acp.t.Text = label6.Text;

pictureBox3.Enabled = false;

}

### 5.1.7 Assing Car Parking



* This form is for assigning car parking slots.
* This form takes the information of the customer and maintain the data of entry date/time.

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" && textBox2.Text == "" && textBox4.Text == "" && textBox5.Text == "" && textBox6.Text == "")

{

MessageBox.Show("Blank text box not allowed", "ERROR", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

}

else

{

try

{

OleDbConnection con = new OleDbConnection();

con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\PDF\B.C.A-5\PROJECT\Project.accdb";

con.Open();

query = "INSERT INTO CarParkingReport(CarNo,EntryDate,EntryTime,SlotNo,Name,MobileNo,ParkingSpace) values ('" + textBox2.Text + "','" + dateTimePicker1.Text + "','" + dateTimePicker2.Text + "','" + textBox1.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "')";

OleDbCommand cmd = new OleDbCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = query;

cmd.Connection = con;

cmd.ExecuteNonQuery();

MessageBox.Show("Record Save ");

textBox1.Clear();

textBox2.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n\n" + query);

}

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

string text = textBox1.Text;

if (!Regex.Match(text, "^[0-9]\*$").Success)

{

MessageBox.Show("Enter Number ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Clear();

}

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

string text = textBox4.Text;

if (!Regex.Match(text, "^[A-Z||a-z]\*$").Success)

{

MessageBox.Show("Enter Name ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox4.Clear();

}

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

string text = textBox5.Text;

if (!Regex.Match(text, "^[0-9]\*$").Success )

{

MessageBox.Show("Enter Number ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox5.Clear();

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

string text = textBox2.Text;

if (!Regex.Match(text, "^[0-9]\*$").Success)

{

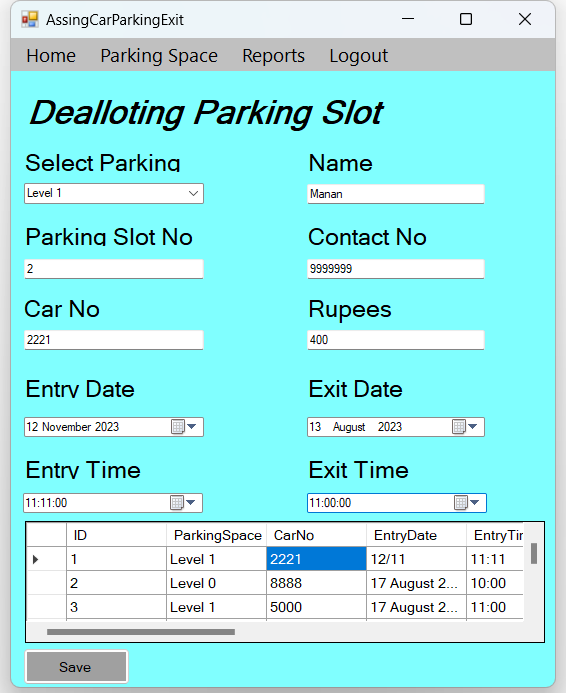
MessageBox.Show("Enter Number ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox2.Clear();

}

}

### 5.1.8 Assing Car Parking Exit



* This form is for removing the assigned car parking slots.
* This form maintains the data of exit date/time.

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" && textBox2.Text == "" && dateTimePicker3.Text == "" && dateTimePicker4.Text == "" && textBox5.Text == "" && comboBox1.Text == "" && textBox7.Text=="" )

{

MessageBox.Show("Blank text box not allowed", "ERROR", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

}

else

{

try

{

OleDbConnection con = new OleDbConnection();

con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\PDF\B.C.A-5\PROJECT\Project.accdb";

con.Open();

string query = "update CarParkingReport SET ExitDate='"+dateTimePicker2.Text+"',ExitTime='"+dateTimePicker4.Text+"',Rupees='"+textBox7.Text+"' where id="+id;

OleDbCommand cmd = new OleDbCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = query;

cmd.Connection = con;

cmd.ExecuteNonQuery();

MessageBox.Show("Record Update ");

dataGridView1.Refresh();

comboBox1.Text = null;

textBox1.Clear();

textBox2.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n\n" + query);

}

try

{

OleDbConnection con = new OleDbConnection();

con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\PDF\B.C.A-5\PROJECT\Project.accdb";

con.Open();

query = "select \* from CarParkingReport";

OleDbDataAdapter da = new OleDbDataAdapter(query, con);

DataSet ds = new DataSet();

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n\n" + query);

}

}

}

private void AssingCarParkingExit\_Load(object sender, EventArgs e)

{

try

{

OleDbConnection con = new OleDbConnection();

con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\PDF\B.C.A-5\PROJECT\Project.accdb";

con.Open();

query = "select \* from CarParkingReport";

OleDbDataAdapter da = new OleDbDataAdapter(query, con);

DataSet ds = new DataSet();

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n\n" + query);

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

id = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());

comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();

dateTimePicker1.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();

dateTimePicker3.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();

dateTimePicker2.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();

dateTimePicker4.Text = dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();

textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[7].Value.ToString();

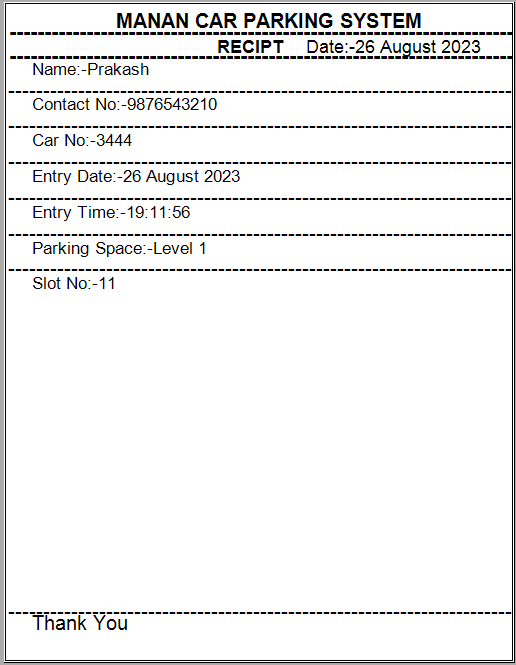
textBox5.Text = dataGridView1.Rows[e.RowIndex].Cells[8].Value.ToString();

textBox6.Text = dataGridView1.Rows[e.RowIndex].Cells[9].Value.ToString();

textBox7.Text = dataGridView1.Rows[e.RowIndex].Cells[10].Value.ToString();

}

### 5.1.9 Receipt



* This form shows the Receipt of the allotted parking.
* This shows the details of the customer and allotted parking slot.

private void button2\_Click(object sender, EventArgs e)

{

printPreviewDialog1.Document = printDocument1;

printPreviewDialog1.ShowDialog();

}

private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

string d = "-------------------------------------------------------------------------------------------------";

e.Graphics.DrawString("MANAN CAR PARKING SYSTEM ", new Font("Arial", 24, FontStyle.Bold),Brushes.Black,new Point(180,10));

e.Graphics.DrawString(d, new Font("Arial", 26, FontStyle.Bold), Brushes.Black, new Point(0, 25));

e.Graphics.DrawString("RECIPT", new Font("Arial", 22, FontStyle.Bold), Brushes.Black, new Point(350, 55));

e.Graphics.DrawString("Date:-" + dateTimePicker1.Text, new Font("Arial", 22, FontStyle.Regular), Brushes.Black, new Point(500, 55));

e.Graphics.DrawString(d , new Font("Arial", 26, FontStyle.Bold), Brushes.Black, new Point(0,65));

e.Graphics.DrawString("Name:-"+textBox4.Text,new Font("Arial",20,FontStyle.Regular),Brushes.Black,new Point(40,95));

e.Graphics.DrawString(d, new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(0,125));

e.Graphics.DrawString("Contact No:-" + textBox5.Text, new Font("Arial", 20, FontStyle.Regular), Brushes.Black, new Point(40, 155));

e.Graphics.DrawString(d, new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(0, 185));

e.Graphics.DrawString("Car No:-" + textBox2.Text, new Font("Arial", 20, FontStyle.Regular), Brushes.Black, new Point(40, 215));

e.Graphics.DrawString(d, new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(0, 245));

e.Graphics.DrawString("Entry Date:-" + dateTimePicker1.Text, new Font("Arial", 20, FontStyle.Regular), Brushes.Black, new Point(40,275));

e.Graphics.DrawString(d, new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(0, 305));

e.Graphics.DrawString("Entry Time:-" + dateTimePicker2.Text, new Font("Arial", 20, FontStyle.Regular), Brushes.Black, new Point(40, 335));

e.Graphics.DrawString(d, new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(0, 365));

e.Graphics.DrawString("Parking Space:-" + textBox6.Text, new Font("Arial", 20, FontStyle.Regular), Brushes.Black, new Point(40, 395));

e.Graphics.DrawString(d, new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(0, 425));

e.Graphics.DrawString("Slot No:-" + textBox1.Text, new Font("Arial", 20, FontStyle.Regular), Brushes.Black, new Point(40, 455));

e.Graphics.DrawString(d, new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(0,1000));

e.Graphics.DrawString("Thank You", new Font("Arial", 24, FontStyle.Regular), Brushes.Black, new Point(40,1020));

}

### 5.1.10 Space List

## 

* This form is about the Parking Space List.
* This form shows the data of available all parking space.

private void SpaceReport\_Load(object sender, EventArgs e)

{

try

{

OleDbConnection con = new OleDbConnection();

con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\PDF\B.C.A-5\PROJECT\Project.accdb";

con.Open();

query = "select \* from SpaceListing ";

OleDbDataAdapter da = new OleDbDataAdapter(query, con);

DataSet ds = new DataSet();

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n\n" + query);

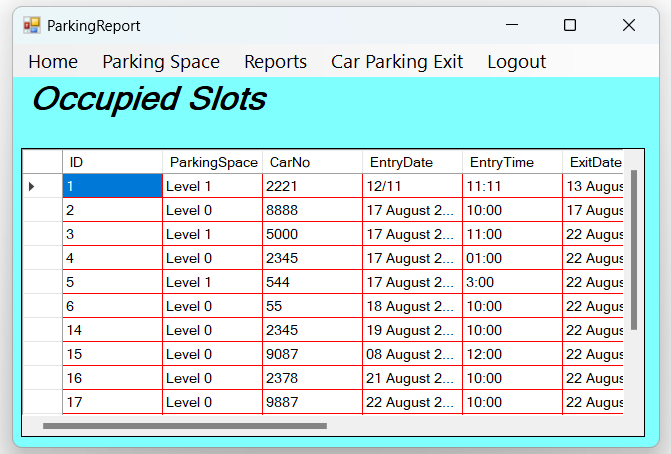
}

// static ID=0 ;

//ID=int.Parse(dataGridView1.Rows[e.Rowindex].Cells[0].Value.ToString();

}

### 5.1.11 Parking Report



* This form is about to show the slots which are occupied.
* This form can also display all the data of customer and allotted slots.

private void ParkingReport\_Load(object sender, EventArgs e)

{

try

{

OleDbConnection con = new OleDbConnection();

con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\PDF\B.C.A-5\PROJECT\Project.accdb";

con.Open();

query = "select \* from CarParkingReport";

OleDbDataAdapter da = new OleDbDataAdapter(query, con);

DataSet ds = new DataSet();

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message + "\n\n" + query);

}

}

# Chapter 6 Limitation & Future Enhancement

**6.1 Limitation**

**6.2 Future Enhancement**

## 6.1 Limitation

* In this project there is not payment method available.
* I will make the software more responsive.
* No backup and restore utilities are incorporated.
* Only work in Windows OS.
* No remote access.

## 6.2 Future Enhancement

* We will try to provide a web site for the customer for payment, receipt.
* We try for online payment system to be accepted.

# Chapter 7 Conclusion

**7.1 Conclusion**

**7.2 Advantages**

## 7.1 Conclusion

* There are some disadvantages of parking management systems, but the good side can’t be overlooked.
* Parking management systems have become a necessity in the parking industry.
* So, if you are a car park owner, it’s high time you should integrate innovative parking solutions into your parking space to make it more efficient, safer, and organised.

## 7.2 Advantages

* The application makes easy to use and it reduce the manual work.
* It has fast access to database.
* Reliable and efficient.
* Easy to manage the information.

# Chapter 8 Bibliography

**8.1 Bibliography**

## 8.1 Bibliography

* Introduction to .NET framework - Wrox publication.
* C# 5.0 and .NET 4.5 Framework (By: Andrew Troelsen )

# Chapter 9 Reference

**9.1 Reference**

## 9.1 References

* We are really thankful to our guider Prof. ZALAK THAKRAR to guide us and inspire

us. We also Thankful to the Prof. JAYDIP RATHOD & Prof. JYOTSNA SALET to

gives us a huge support in our project.

* Other References
* [www.freeprojectz.com/paid-projects/online-car-parking-system](http://www.freeprojectz.com/paid-projects/online-car-parking-system)
* [www.youtube.com/watch?v=9smsepM8mMU&list=PPSV](https://www.youtube.com/watch?v=9smsepM8mMU&list=PPSV)
* [www.youtube.com/watch?v=osI7P1YdL3o&t=111s](https://www.youtube.com/watch?v=osI7P1YdL3o&t=111s)
* [www.youtube.com/watch?v=1vbAcBNzFcE&t=144s](http://www.youtube.com/watch?v=1vbAcBNzFcE&t=144s)