



UNIVERSITY OF KELANIYA

**DEPARTMENT OF STATISTICS & COMPUTER
SCIENCE**

ACADEMIC YEAR 2021/2022

FINAL PROJECT REPORT

VEHICLE SALE MANAGEMENT SYSTEM

GROUP 19

ACKNOWLEDGEMENT

We extend our heartfelt gratitude and appreciation to all the team members whose contributions and support have made the development and documentation of the Vehicle Sale Management System (VSMS) possible. We acknowledge the dedication, expertise, and tireless efforts of the team who have worked diligently to design, code, test, and deliver this robust C#-based VSMS. Their technical skills and commitment to excellence have been instrumental in shaping this project. We would like to express our sincere appreciation to our lecturer Miss.Vindya Senanayake who imparted her knowledge and expertise during our Visual Programming course. Your guidance, mentorship, and passion for teaching have played a pivotal role in equipping us with the skills necessary for the successful development of this system. Special acknowledgment is extended to the contributors to this project document, including all the team members. Your dedication to documenting project details, requirements, and processes has ensured transparency and clarity in project communication

Contents

1. INTRODUCTION AND PROJECT BACKGROUND	5
1.1 Problems and Requirements.....	5
1.2 Objectives.....	6
2.FEASIBILITY REPORT	6
2.1 Technical Feasibility	6
2.2 Economic Feasibility	6
3. ANALYSIS REPORT	7
3.1 About the System	9
4. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	10
4.1 Functional Requirements:	10
4.2 Non-Functional Requirements	11
5. APPLICATION DESIGN	13
5.1 System Structure	13
5.2 Flow Charts	14
5.3 Use case Diagram	18
5.4 Class Diagram	19
5.5 Sequence Diagram	20
6. USER ROLES , INTERFACES AND IMPLEMENTATION	29
6.1 Login Form	29
6.2 Admin Dashboard	33
6.3 Staff Dashboard.....	38
6.4 Appointment.....	42
6.5 Customer	50

6.6 Inventory	59
6.7 Add or Edit Vehicle	65
6.8 Add or Edit Employee.....	80
6.9 Employee's Salary Calculator	91
6.10 About Us Web Page	95
6.10.1 car.html file	96
6.10.2 stylesheet2.css file.....	97
7. DATA TABLE STRUCTURE	103
7.1 Inventory	103
7.2 Employee	104
7.3 Customer	105
105	
7.4 Appointment.....	105
105	
8. TOOLS.....	106
9. WORK CONTRIBUTION	107
9.1 Individual Work Contribution	107
9.2 DETAILS OF GROUP MEMBERS	107
9.3 Challenges.....	108
10. FUTURE ENHANCEMENT OF THE PROJECT.....	109
11. REFERENCES	111

1. INTRODUCTION AND PROJECT BACKGROUND

The main purpose of this project is to develop a real time vehicle sale management system with multiple branches. The primary requirement of this system is to maintain a detailed vehicle inventory along with customer details, employee details and their monthly salary details.

The Vehicle Sale Management System represents a pivotal advancement in the realm of vehicle dealership operations. In an ever-evolving automotive industry, where competition is fierce and customer expectations are continually rising, this comprehensive software application serves as a beacon of innovation and efficiency. Developed using the C# programming language, it stands as a testament to modern technology's capacity to transform and elevate the vehicle sales and management experience. In an era where digitalization and automation are becoming increasingly indispensable, this system emerges as a crucial tool for vehicle dealerships, providing a dynamic and user-friendly interface that caters to both dealership staff and discerning customers. It is poised to revolutionize the way vehicle businesses operate, simplifying complex processes and enhancing productivity across the board.

1.1 Problems and Requirements

- An application which operator can add, update, edit, search and delete data record within a single software solution was proposed.
- Rather than memorizing or dairy entries of appointments, the appointments should be visible in the application.
- Maintaining customer details and employee details is another issue. Hence, an application which both administrator and staff can access customer details and administrator can maintain employee details was proposed.
- The salaries of employees should be calculated in a simplified method.

1.2 Objectives

Our main objective is to develop a more efficient and productive system to fulfill the basic requirements of our system properly and effectively within the given period. Therefore, we must do more research in many areas to pass this goal. In this case we can split our objectives into two main parts.

Team Objectives	Development Objectives
Learning how to browse and add pictures to application.	Manage vehicle data.
Learning how to browse and add pictures to application.	Manage employee data with images.
Learning How to import an image from Database to application.	Allowing access to two parties (Admin and User) via a Log-in Form.
Learning how to build an in-built web browser to the application.	Maintaining a cleaner user interface throughout the application.
Time management.	Creating a User-manual.

2.FEASIBILITY REPORT

The main objective of preparing this document is to give clear ideas about the features of this system. This system brings many long-term advantages, and this system could be used for quite a long time without any modification by who concern.

2.1 Technical Feasibility

- Saves time.
- Ability to calculate employee salaries efficiently.
- Easy access to live auction via web.
- Security features via a login system.
- Increase organizational flexibility.

2.2 Economic Feasibility

- Saves money for stationery.
- New market niche and increasing sales.

3. ANALYSIS REPORT

Since there was no physical client on this case to get the requirements for the developing stage, we used Internet as our main source of reference and as the main fact-finding technique. We used our own personal computers and laptops to create the system with the use of Visual Studio 2022, MS SQL Server Management Studio and coded with .Net C#. Since this was a group project all the members took the responsibility with each part of the system development. After Unit testing, we have implemented each unit into a single system and did the final system testing. For design user friendly interfaces as form designing stage, we used in-built visual studio tools mentioning below,

- **Labels:** Labels are used to display text or descriptions on your forms. They are typically used for headings, field names, or instructional text.
- **Buttons:** Buttons are used for triggering actions or operations within your application. For example, you can use buttons for saving data, submitting forms, or navigating to other screens.
- **Text Boxes:** Text boxes allow users to input text or data. You can use them for fields like vehicle registration numbers, driver names, or search boxes.
- **ComboBoxes:** ComboBoxes provide a dropdown list of options for users to select from. They are useful for selecting vehicle types, statuses, or categories.
- **Checkboxes:** Checkboxes are used for binary choices, like marking a vehicle as active or inactive.
- **Radio Buttons:** Radio buttons are used for selecting one option from a list of mutually exclusive options. For example, you can use radio buttons to specify the condition of a vehicle (e.g., good, fair, poor).
- **DataGridView (or DataGrid):** This control allows you to display tabular data. You can use it to show lists of vehicles, maintenance records, or reports.
- **GroupBoxes:** GroupBoxes are used for grouping related controls together on a form, providing a visual organization of data or options.
- **PictureBox:** PictureBoxes are used for displaying images, which can be handy for showing vehicle images or logos.

- **DateTimePicker:** DateTimePickers enable users to select dates and times. They are useful for specifying vehicle purchase dates or maintenance schedules.
- **TabControl:** TabControls allow you to organize your UI into multiple tabs, making it easier to manage and navigate through different sections of your application.
- **ListView:** A ListView can be used to display items in a list or details view. It's suitable for showing a list of vehicles and their details.
- **MenuStrip and ToolStrip:** These controls are used for creating menus and toolbars in your application, which can be helpful for navigation and actions.
- **StatusBar:** A StatusBar can display information or status messages at the bottom of your application's window.
- **Chart Controls:** Visual Studio includes chart controls for displaying data in various chart formats, such as bar charts or pie charts, which can be useful for displaying statistics or reports.
- **RichTextBox:** A RichTextBox control allows for formatted text input, which can be useful for adding detailed notes or descriptions.
- **FlowLayoutPanel or TableLayoutPanel:** These layout controls can help you arrange other controls in a more structured manner, aligning them horizontally or in a grid, respectively.

3.1 About the System

- We introduce this system to enable the owner to make it economically & technically flexible. Administrators can add new vehicles to the inventory where we gave each vehicle a three-digit inventory number, add customers' details, employee's details and maintain an appointment list. Also, the user can navigate through the application easily.
- The main interface of admin is made from three selection tabs. The first one shows the appointment list and option to edit them, second one shows the quick action buttons and last tab shows the in-built web browser for checking live auction making it very user friendly, simple & easily adaptable.
- There is username and password for both parties to login. Therefore, this system cannot be used by anyone else. It is a step taken to ensure the security of the software.
- The main function is managing vehicle details. When inventory number is typed, chassis number, make of the vehicle, model, color, manufacture year, transmission, fuel type, registration status, engine capacity, in which branch is the vehicle available and finally the price is displayed on the screen along with an image of the vehicle.
- When the employee number is given, all his details will appear on the screen. Same for the customer details as well. An employee salary calculator is also added to the system.

4. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

4.1 Functional Requirements:

User Authentication and Authorization: The system must have user login functionality. Different user roles (e.g., admin, staff) with specific permissions must be defined.

Inventory Management: Users should be able to add new vehicle listings with details such as make, model, year, vregno, price, and availability. Existing listings can be edited, and sold or outdated vehicles should be removable from the system. Data related to each vehicle can be uploaded.

Employee Detail Management: is a core functionality within the Vehicle Sale Management System (VSMS) that enables the efficient and organized handling of employee information and roles within the dealership. This feature is designed to streamline the management of dealership personnel, ensuring that staff details are accurately recorded, roles are defined, and access permissions are appropriately allocated.

Make Appointments: is a vital functionality within the Vehicle Sale Management System (VSMS) that facilitates the scheduling of appointments for various dealership activities. This feature is designed to enhance organization, customer service, and operational efficiency by allowing users to schedule appointments for sales consultations, vehicle test drives, service appointments, and other dealership-related tasks.

Customer Relationship Management (CRM): Maintain a customer database with detailed profiles and contact information. Record customer interactions, including inquiries, purchases, and service requests. Send automated notifications and updates to customers. Utilize customer data to provide personalized recommendations and promotions.

Employee Salary Calculator: is a critical functionality within the Vehicle Sale Management System (VSMS) that automates the process of calculating employee salaries, ensuring accuracy, transparency, and efficiency in compensation management. This feature plays a pivotal role in managing the financial aspects of the dealership's workforce.

4.2 Non-Functional Requirements

1) Performance:

- The system should be responsive and capable of handling a large number of concurrent users.
- Response times for common operations should be within acceptable limits.
- Database queries and transactions must be optimized for efficiency.

2) Security:

- User data, including login credentials, must be securely stored and encrypted.
- Role-based access control should be robust to prevent unauthorized access.
- Data access should be protected against SQL injection and other security threats.

3) Scalability:

- The system should be designed to accommodate future growth in terms of users and data volume.
- Consider cloud-based hosting options for scalability.

4) Usability:

- The user interface should be intuitive and user-friendly.
- Provide user training and documentation to ensure effective system usage.

5) Reliability:

- Ensure system availability with minimal downtime for maintenance and updates.
- Implement regular data backups and disaster recovery procedures.

6) Compatibility:

- The system should be compatible with common web browsers and operating systems.
- Mobile responsiveness should be considered for users accessing the system on smartphones and tablets.

7) Data Integrity:

- Data integrity constraints should be enforced to prevent data inconsistencies and errors.
- Regular data validation and verification processes should be in place.

8) Regulatory Compliance:

- Ensure that the system complies with relevant legal and industry-specific regulations, such as data protection laws.

9) Documentation:

- Maintain comprehensive documentation, including user manuals, system architecture diagrams, and code documentation.

10) Performance Testing:

- Conduct performance testing to ensure the system can handle expected loads and perform optimally under stress.

11) Accessibility:

- Ensure that the system is accessible to users with disabilities in compliance with accessibility standards (e.g., WCAG).

12) Cost:

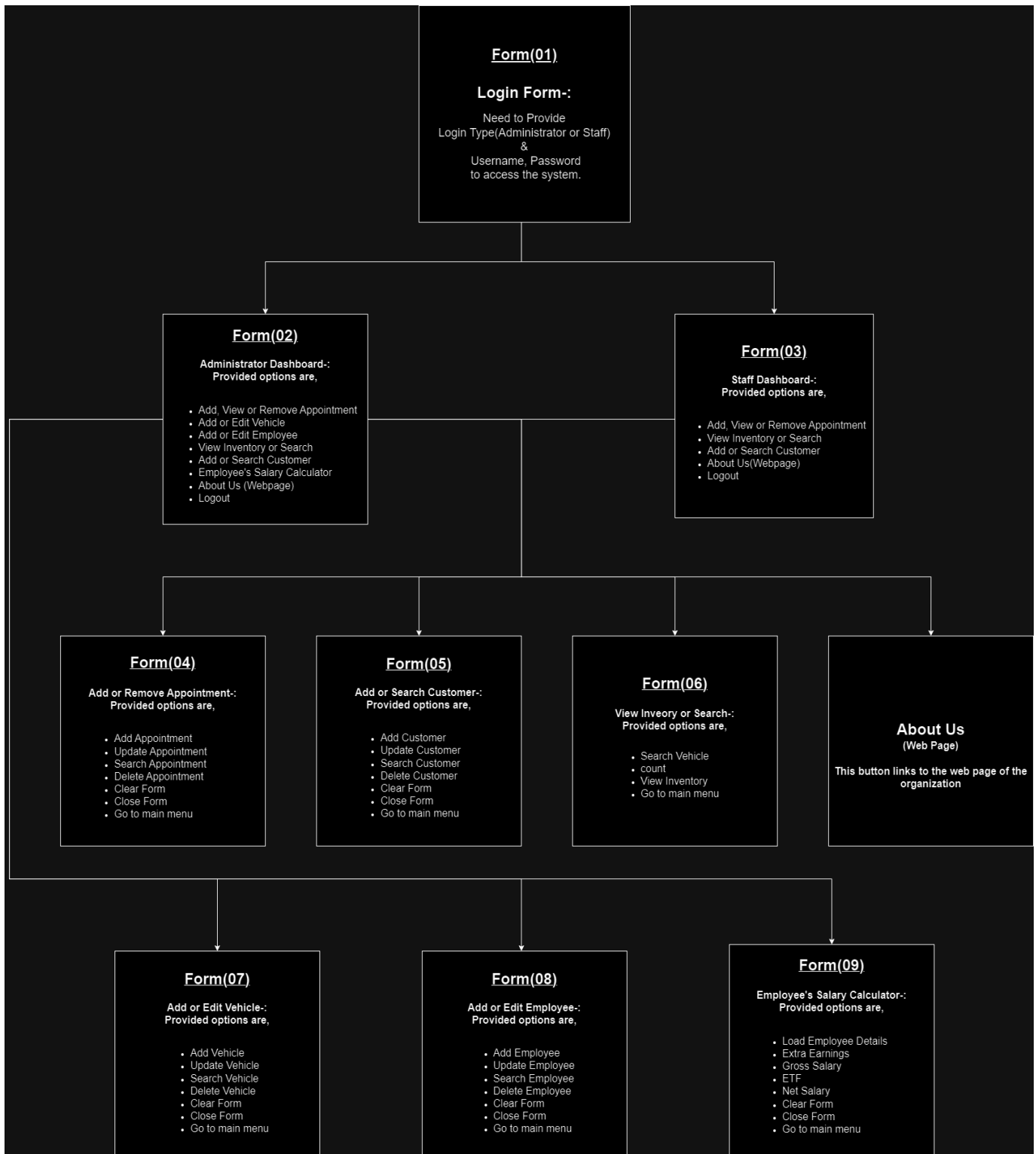
- Manage infrastructure and operational costs effectively, considering factors like hosting fees and licensing.

13) Backup and Recovery:

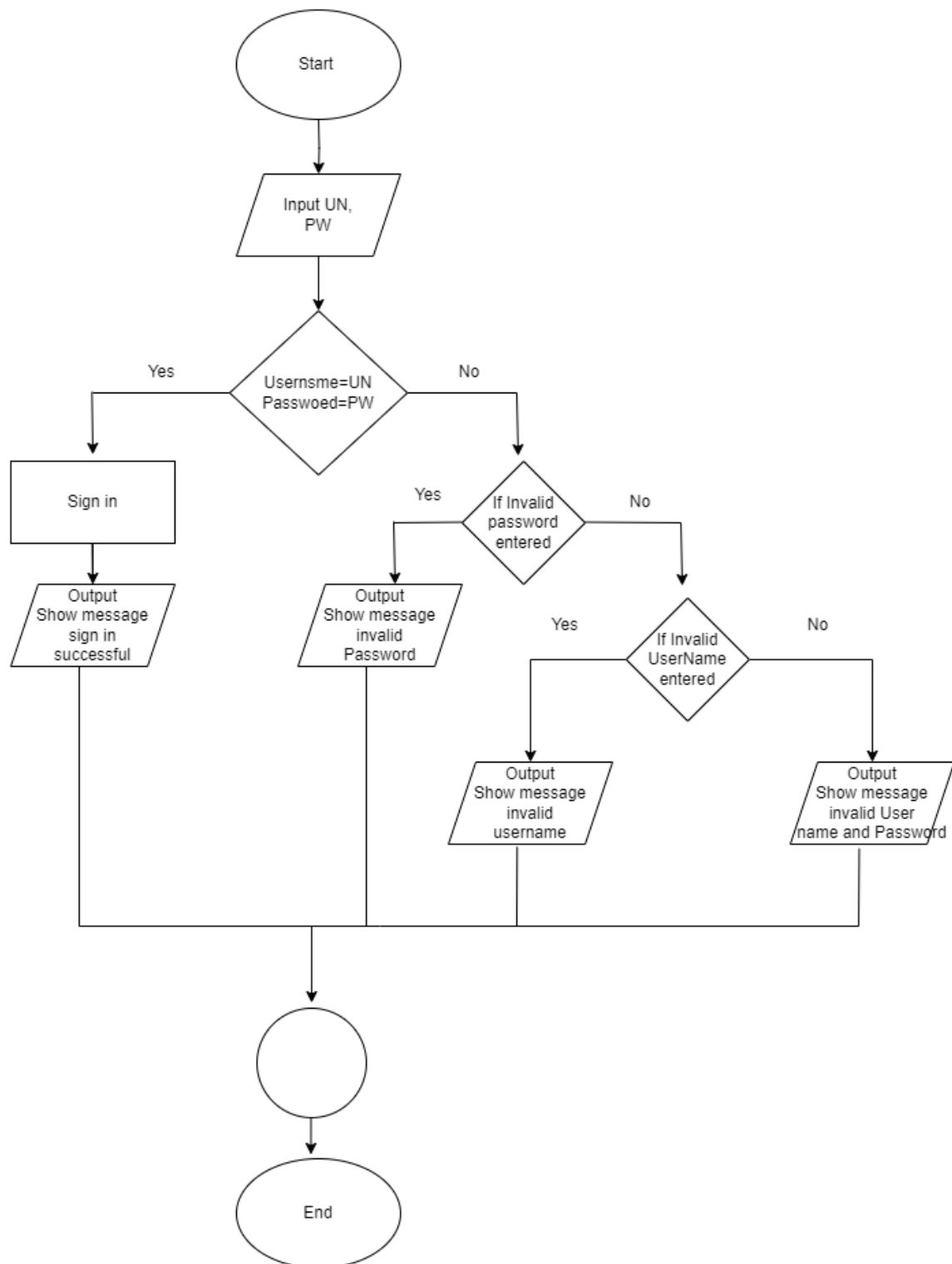
- Implement regular automated backups of the system data and establish recovery procedures in case of data loss or system failure.

5. APPLICATION DESIGN

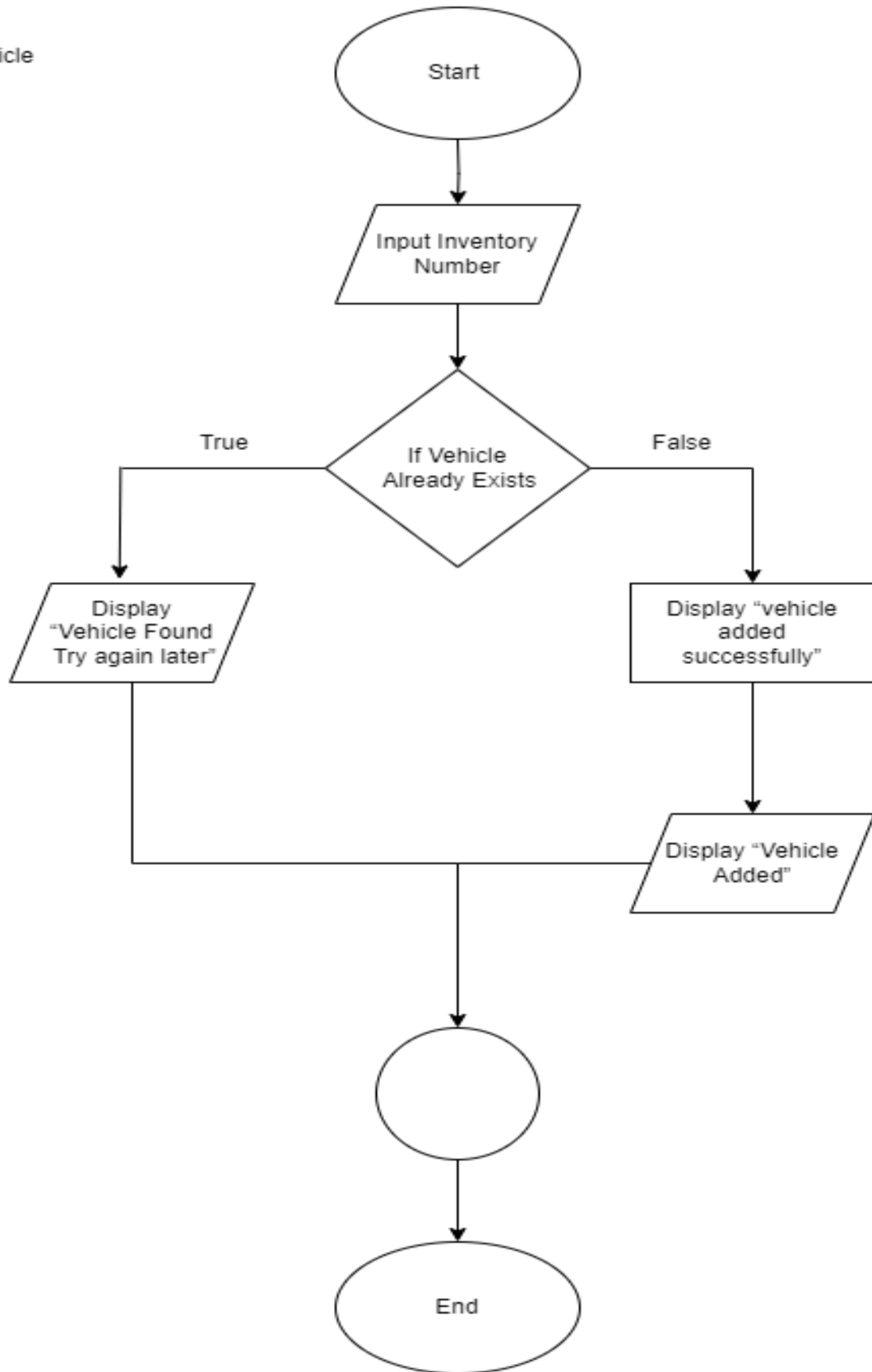
5.1 System Structure



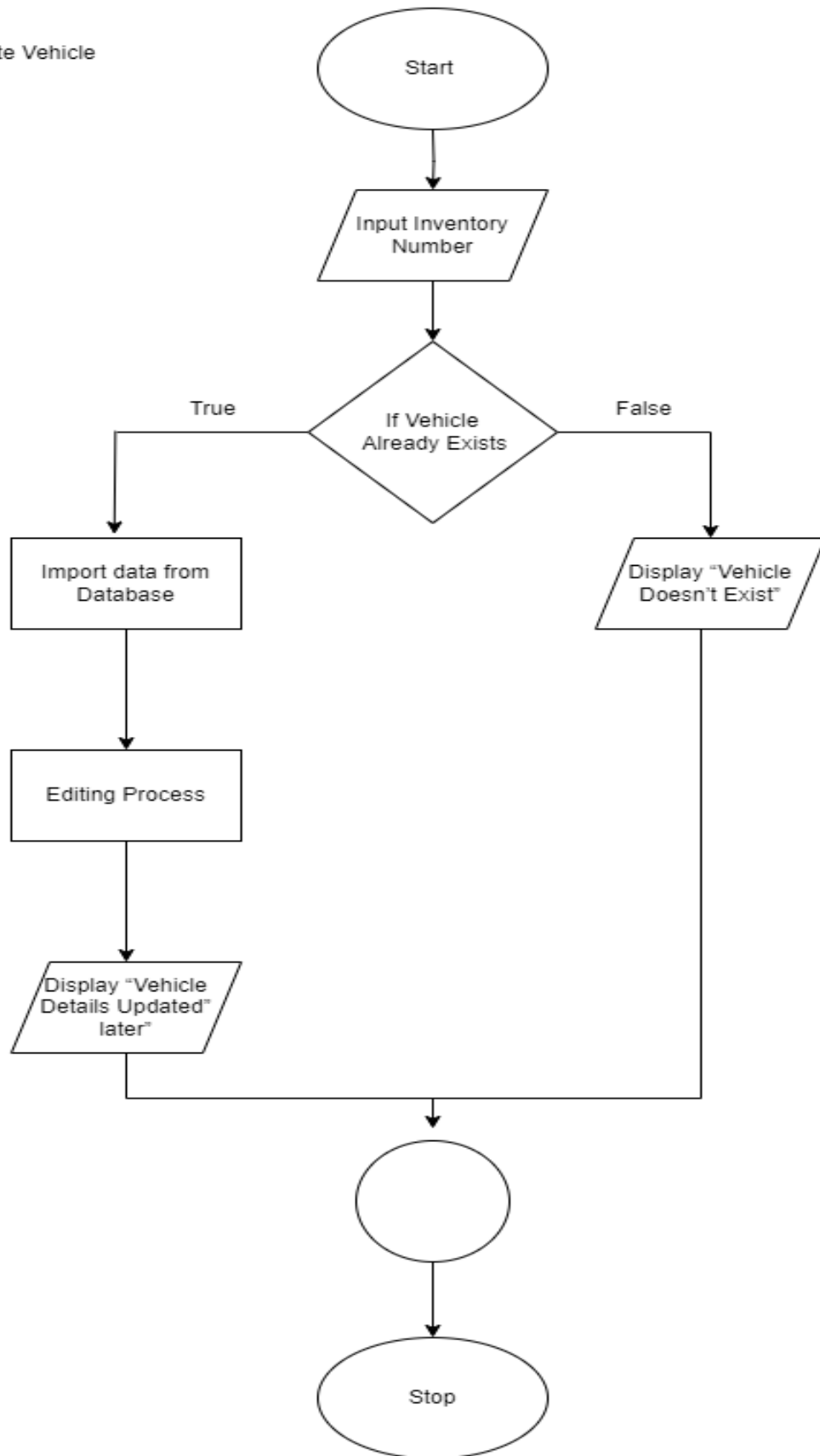
5.2 Flow Charts



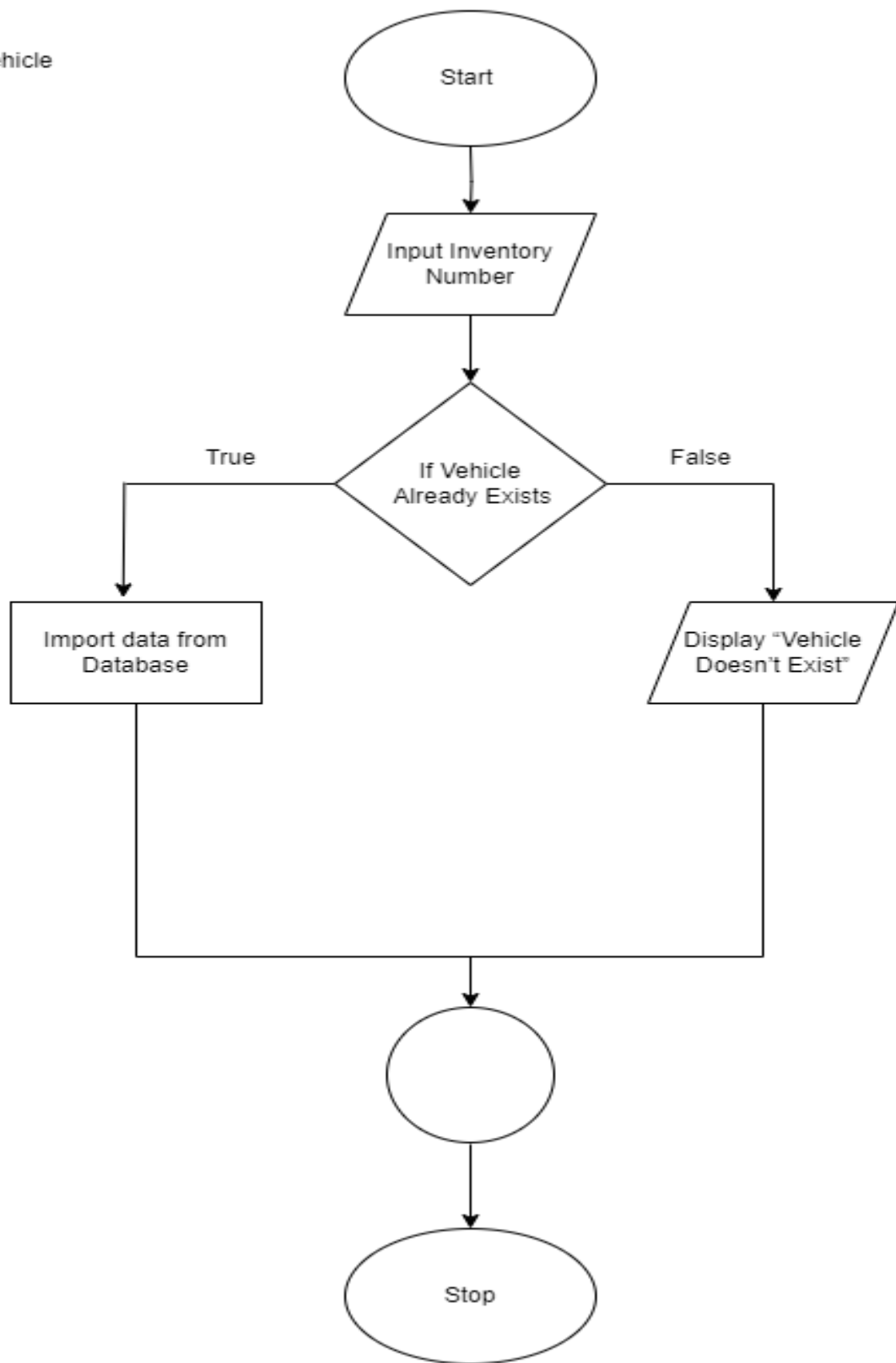
Add Vehicle



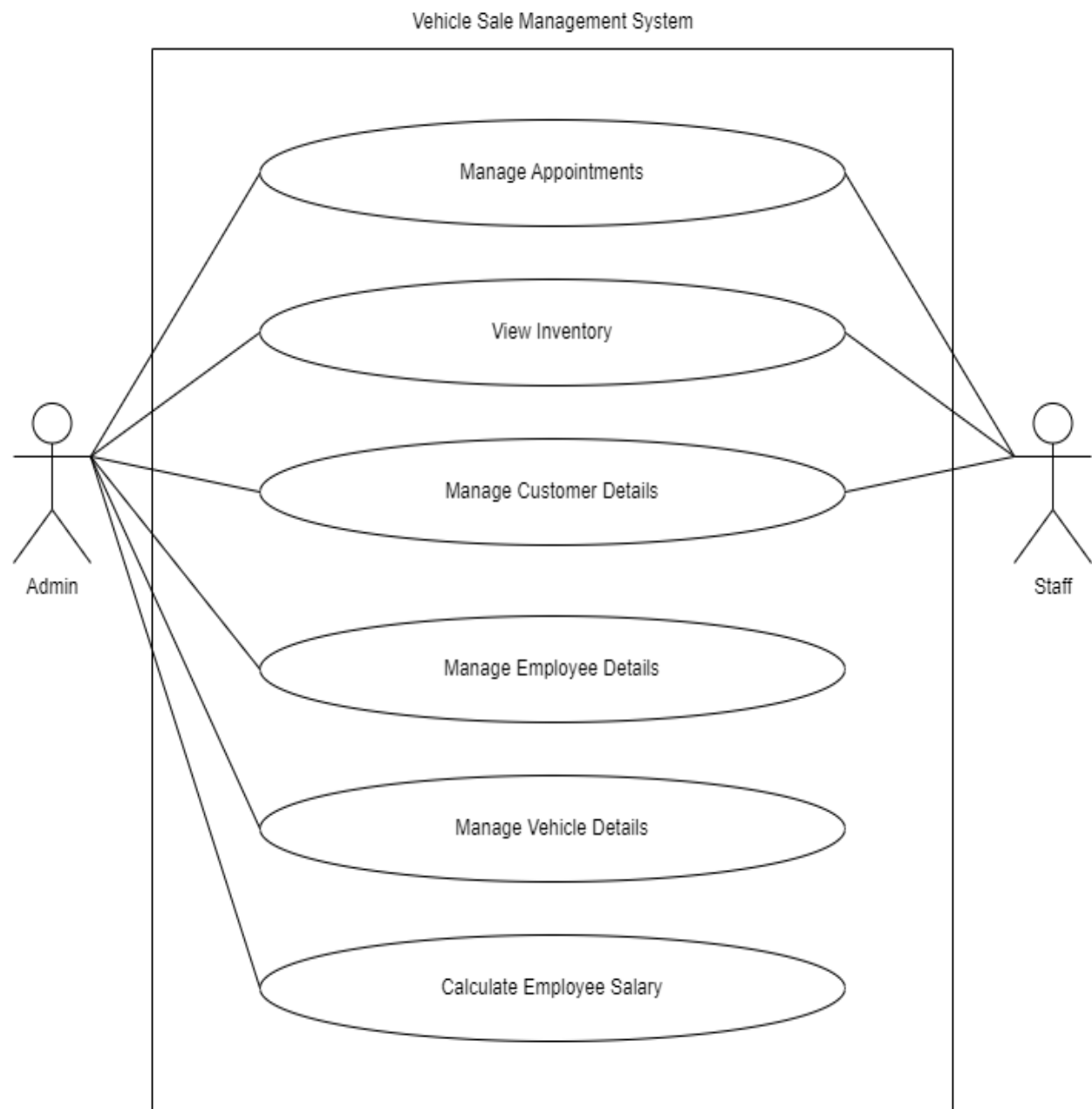
Edit and Update Vehicle



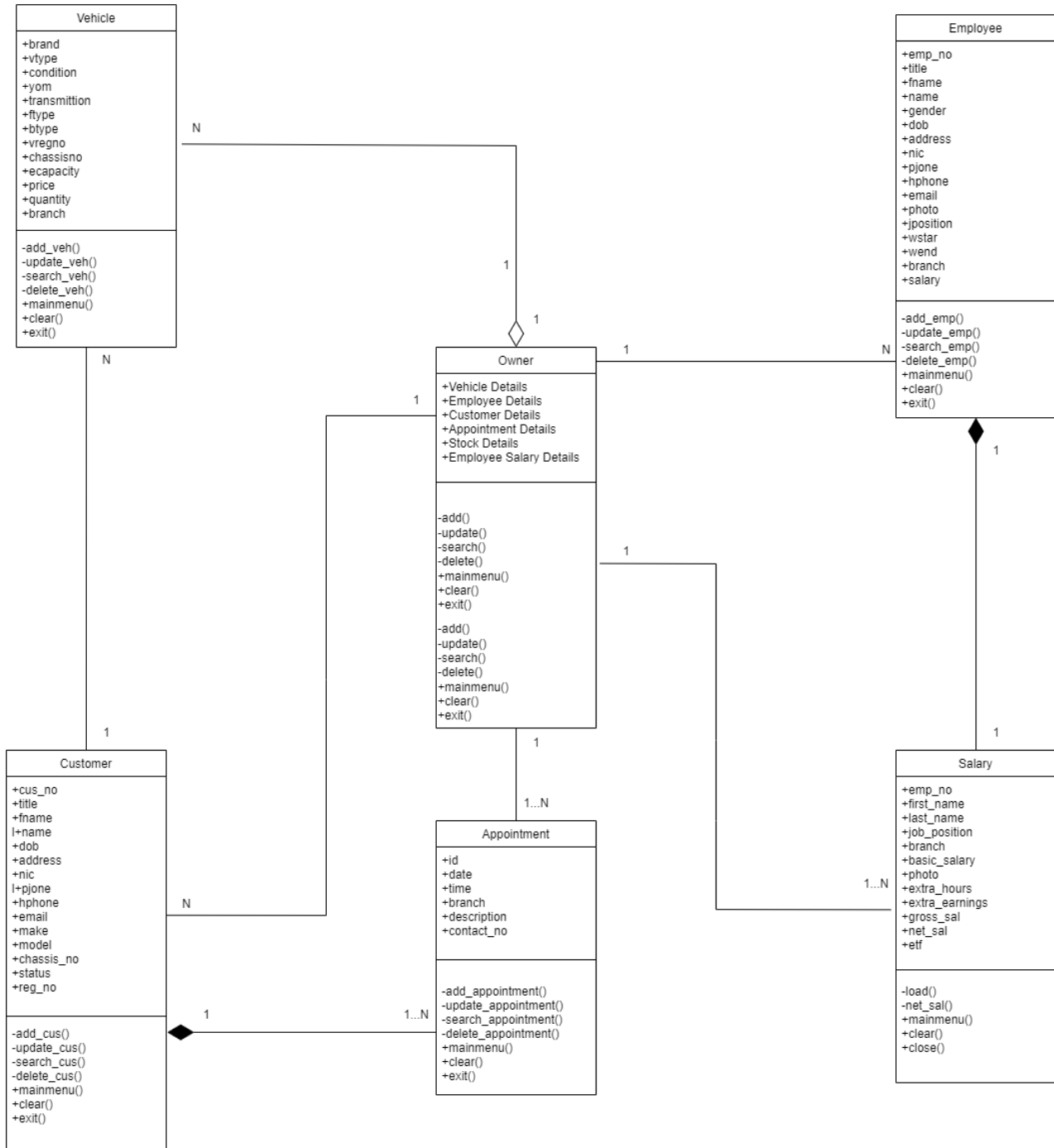
Search Vehicle



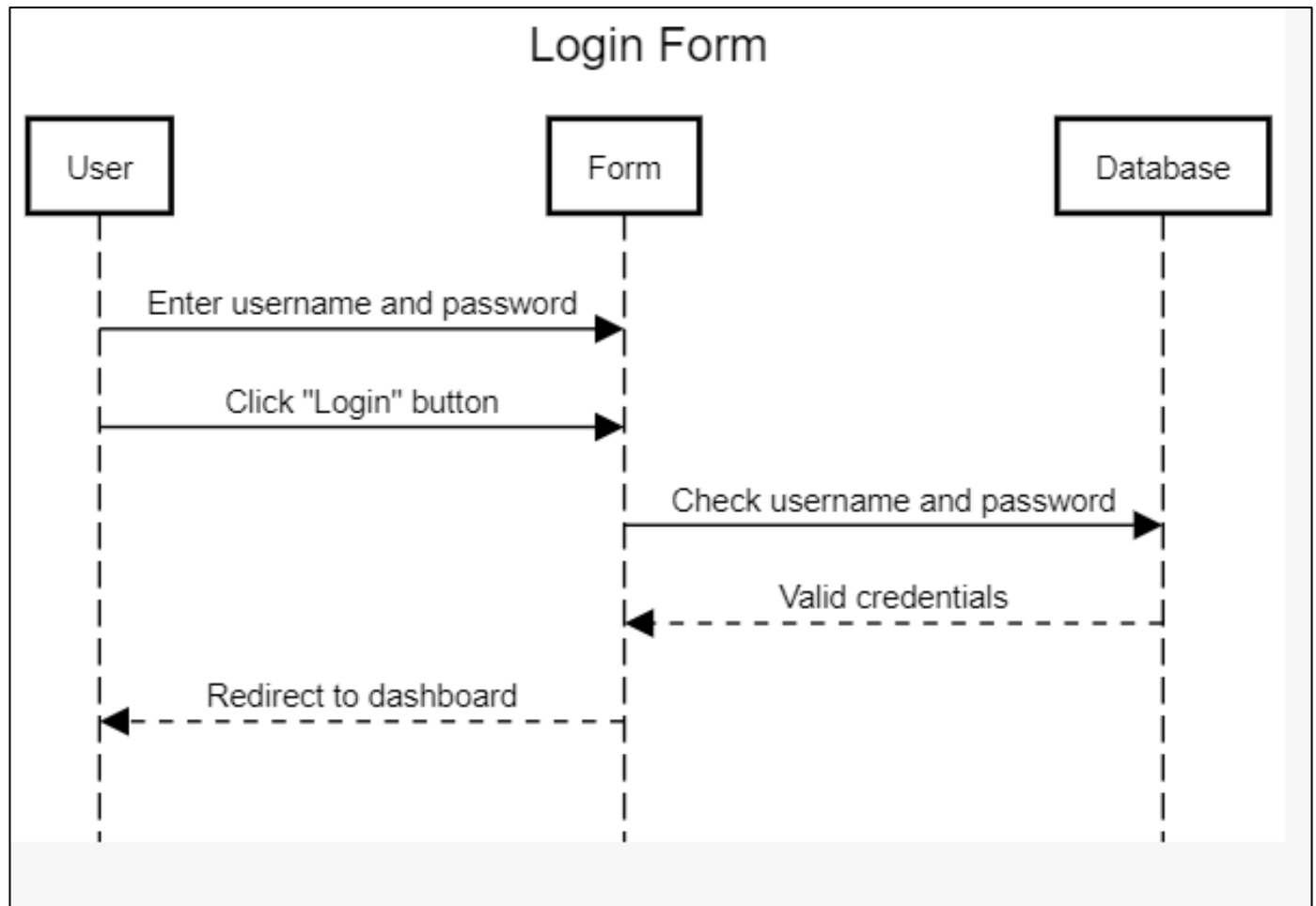
5.3 Use case Diagram

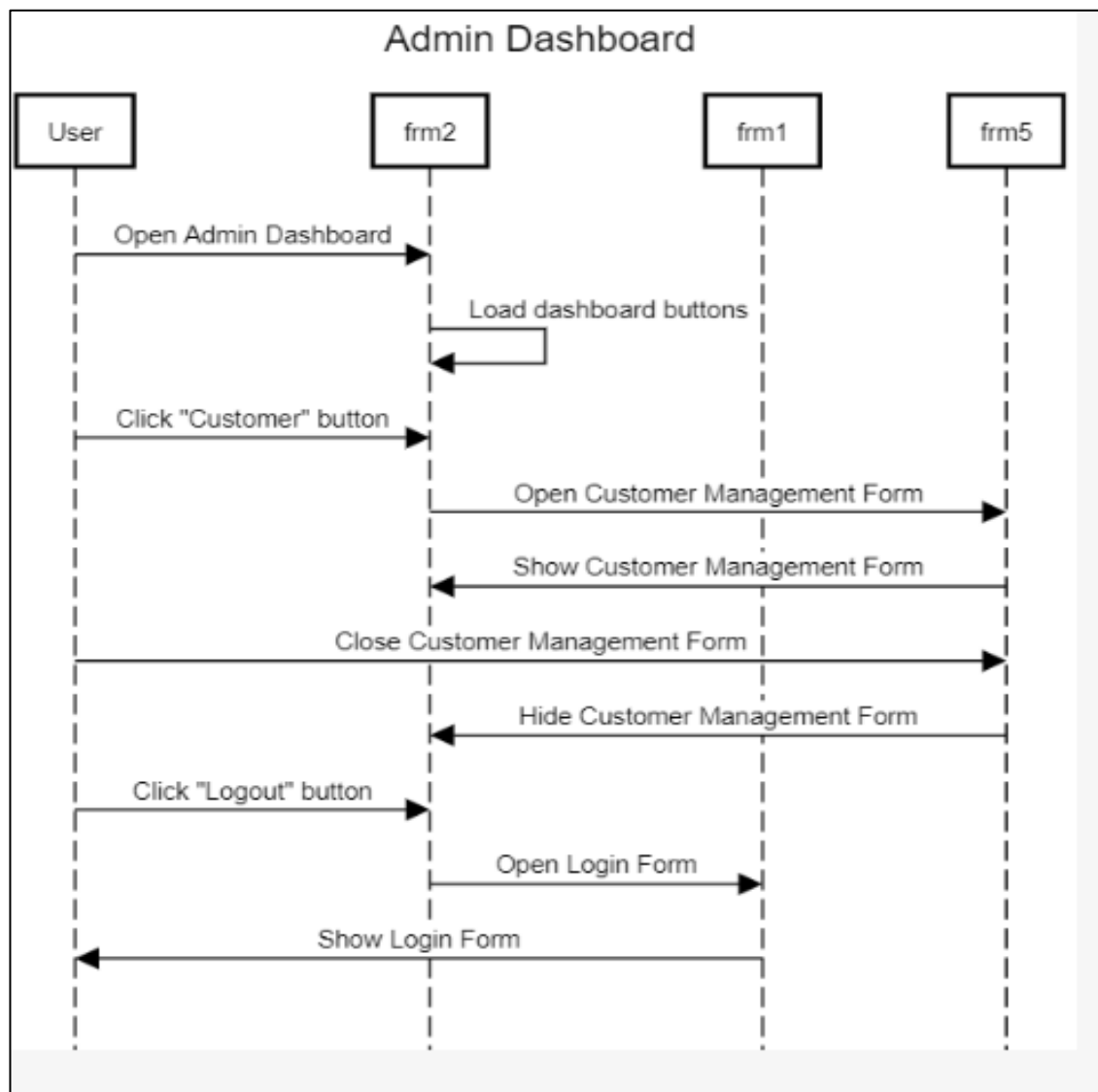


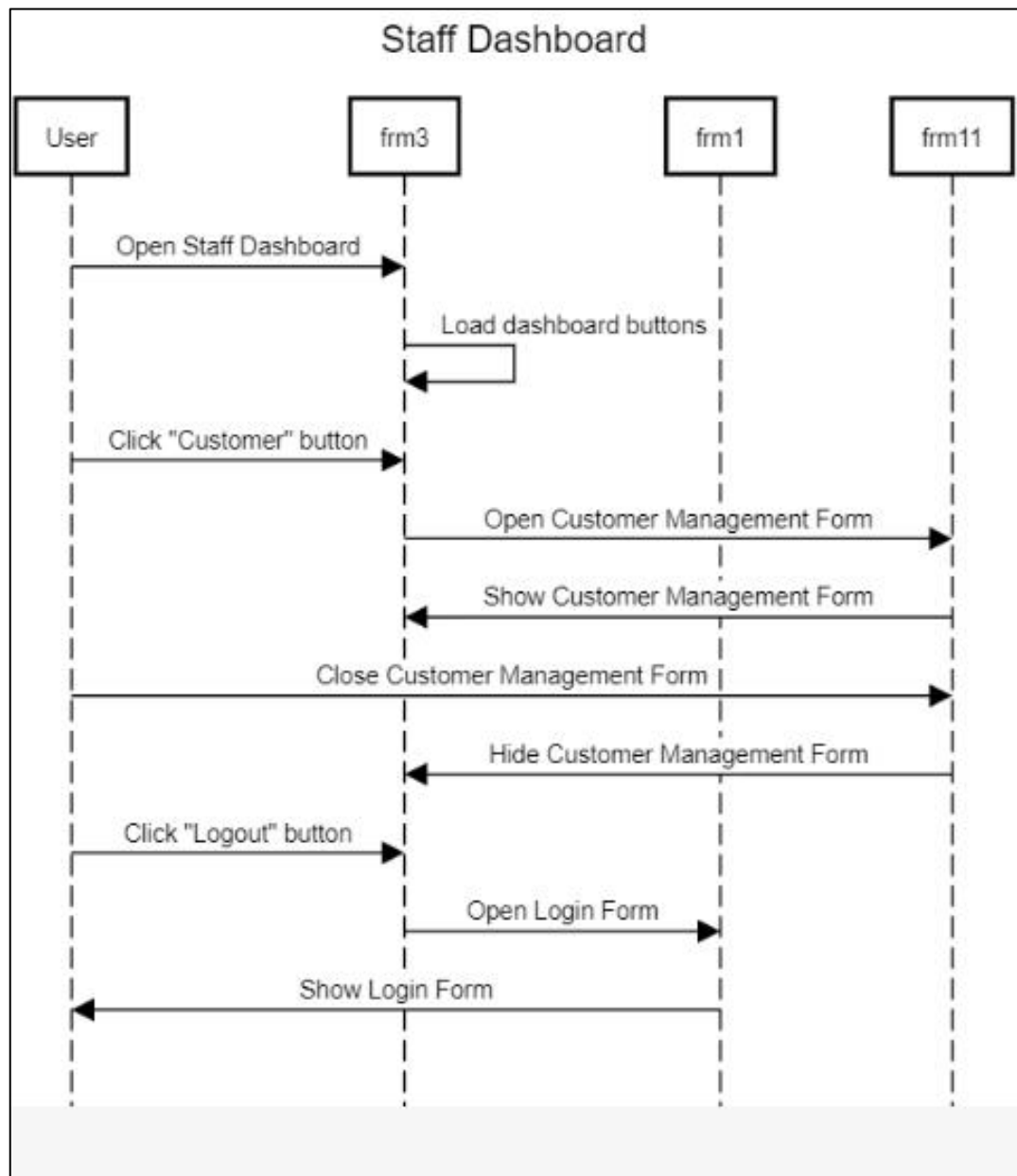
5.4 Class Diagram

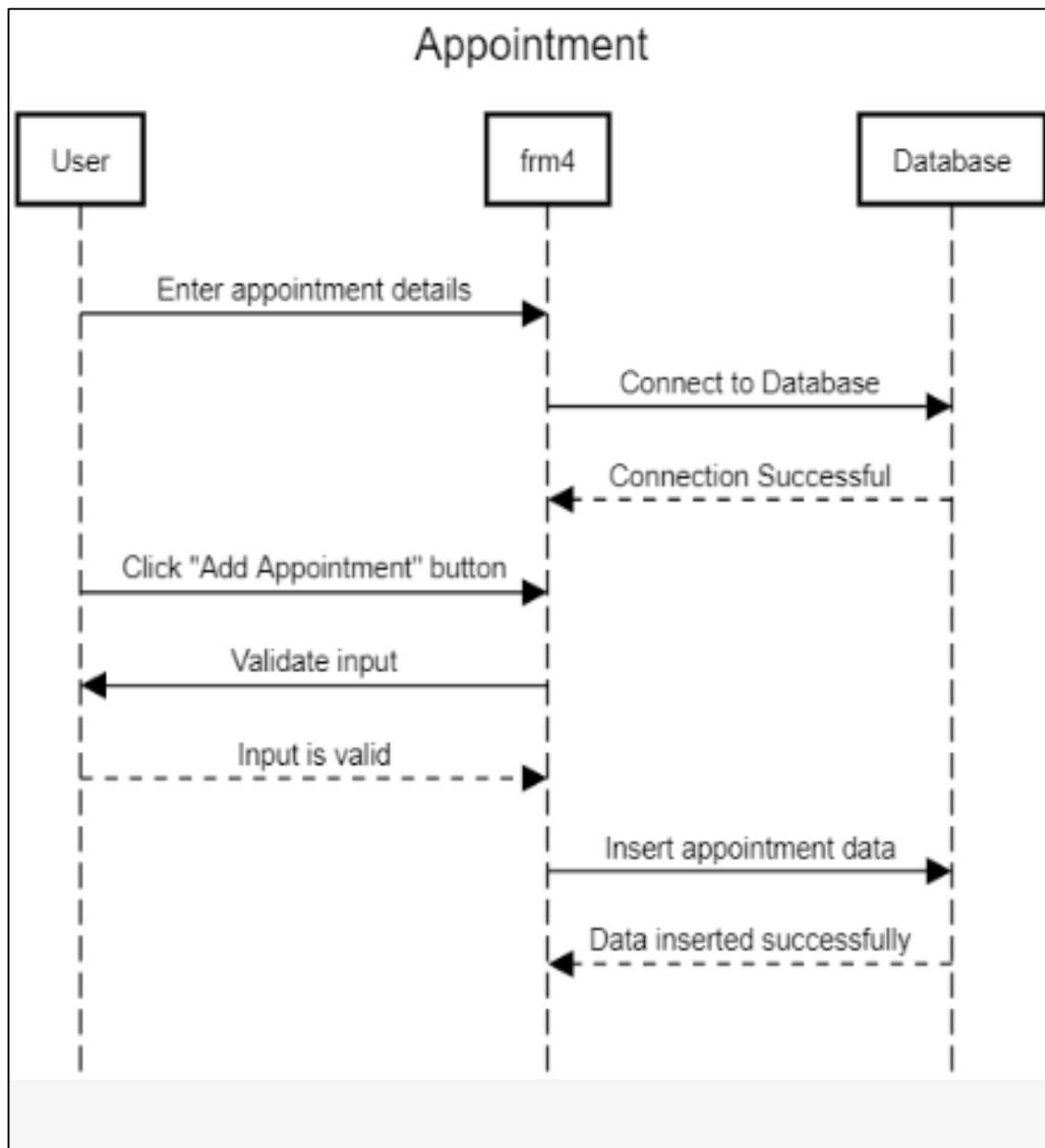


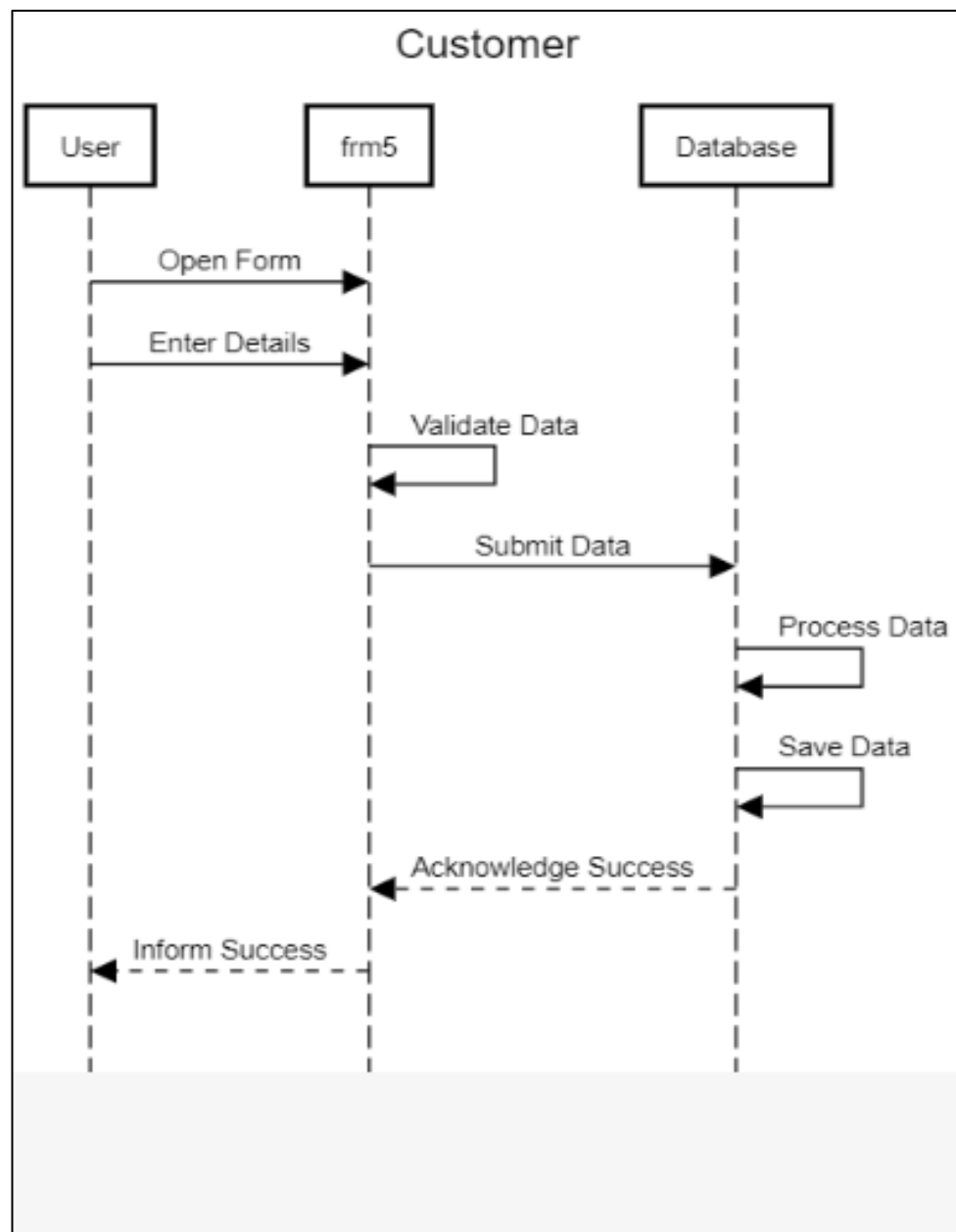
5.5 Sequence Diagram

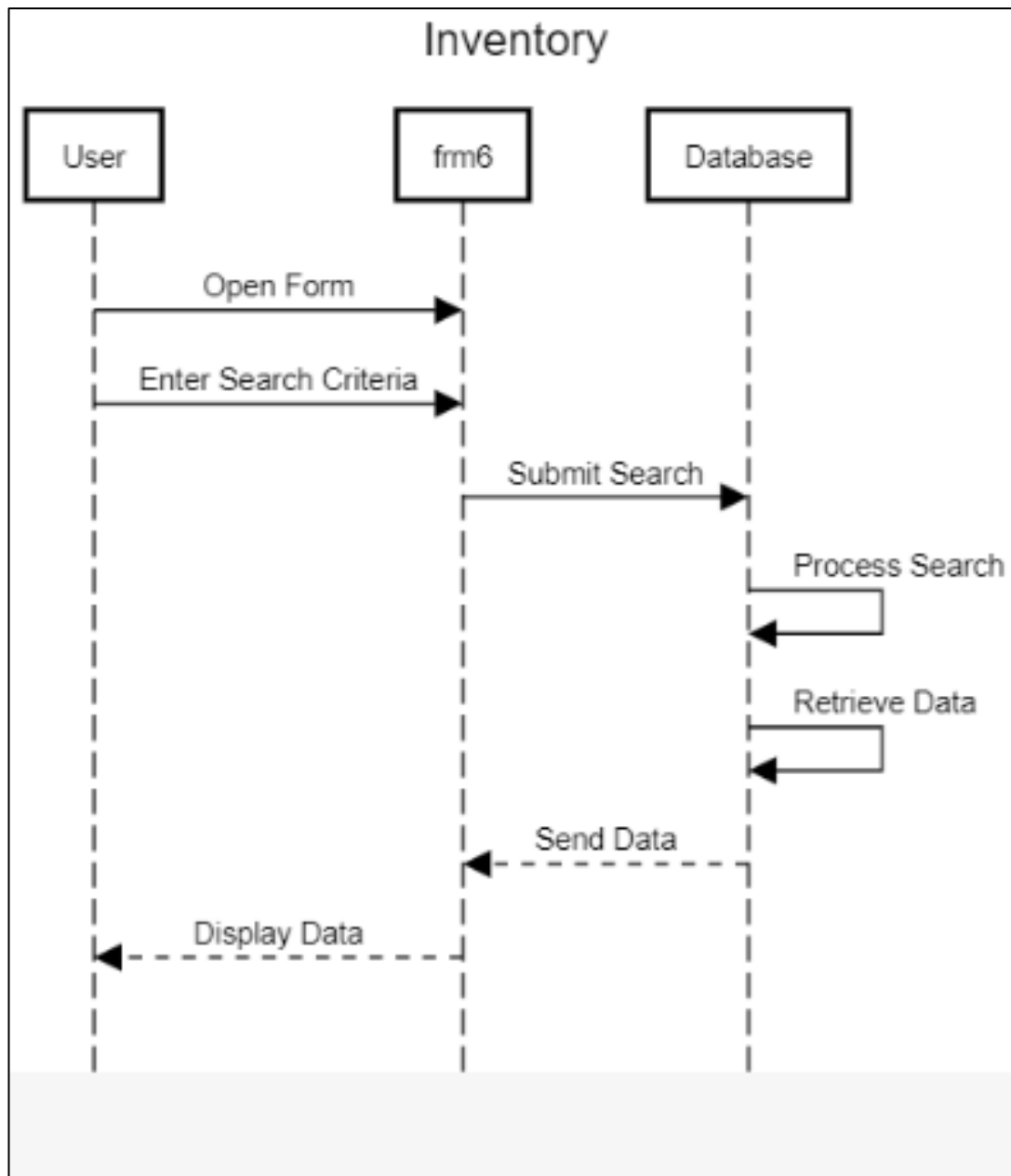


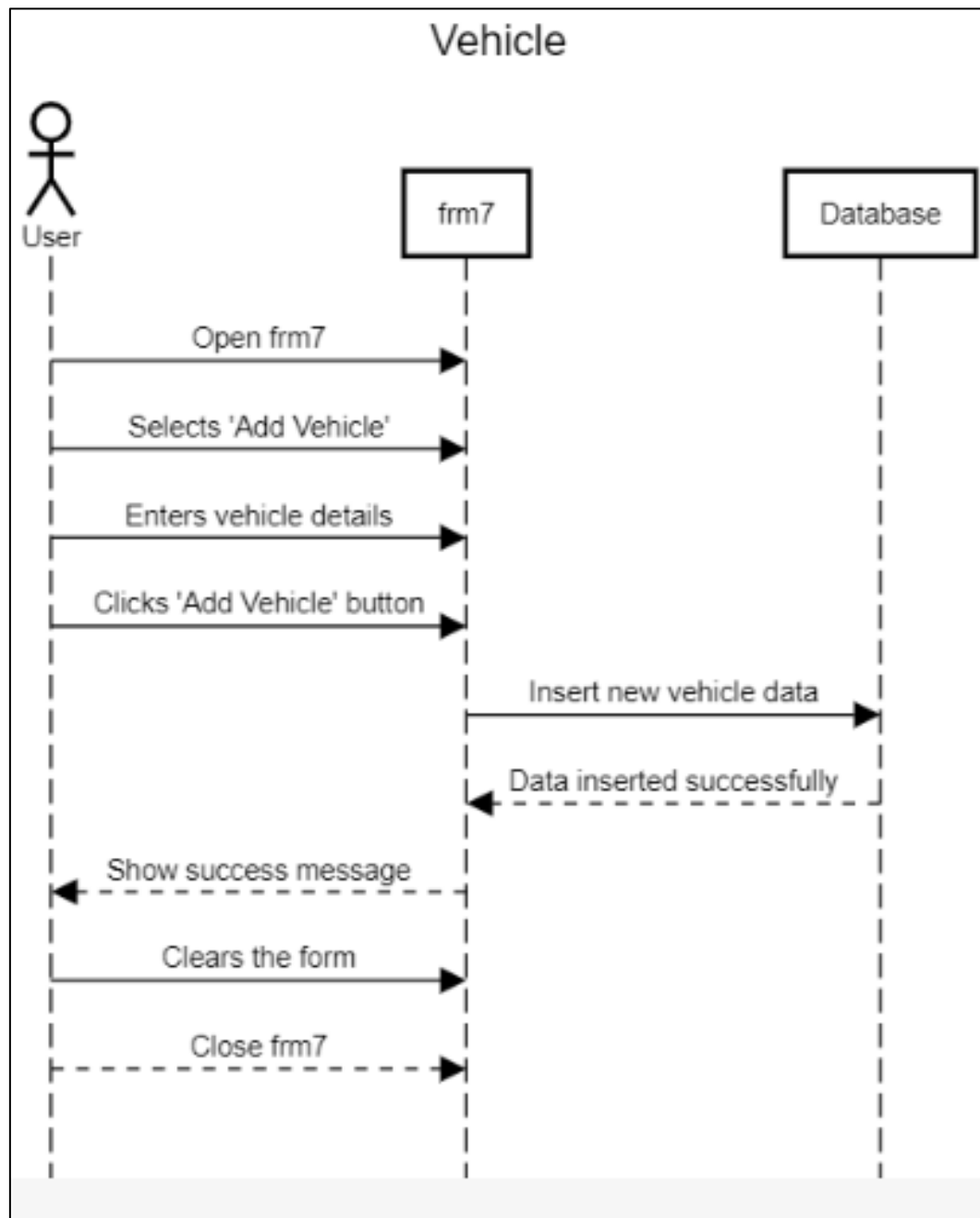


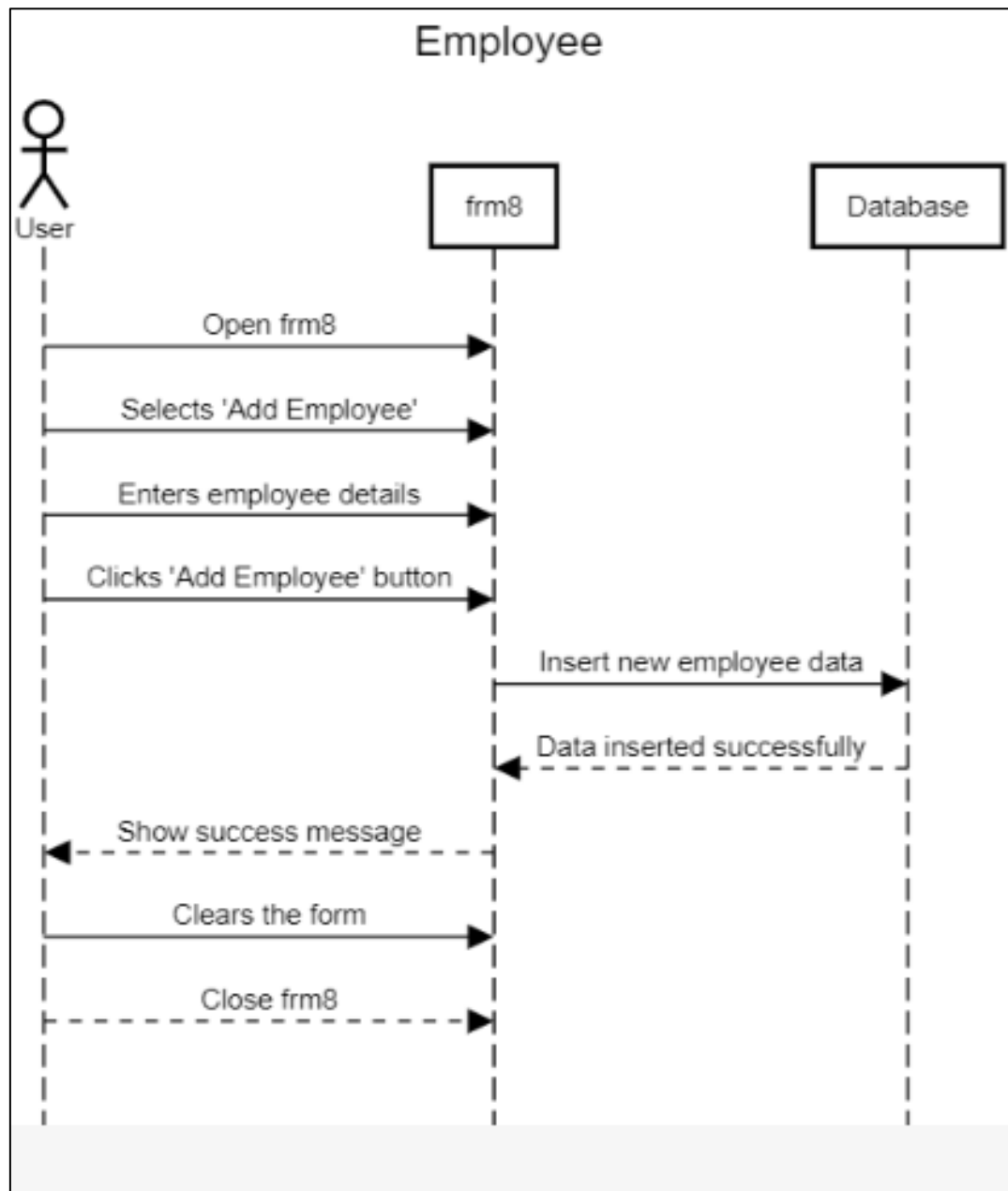




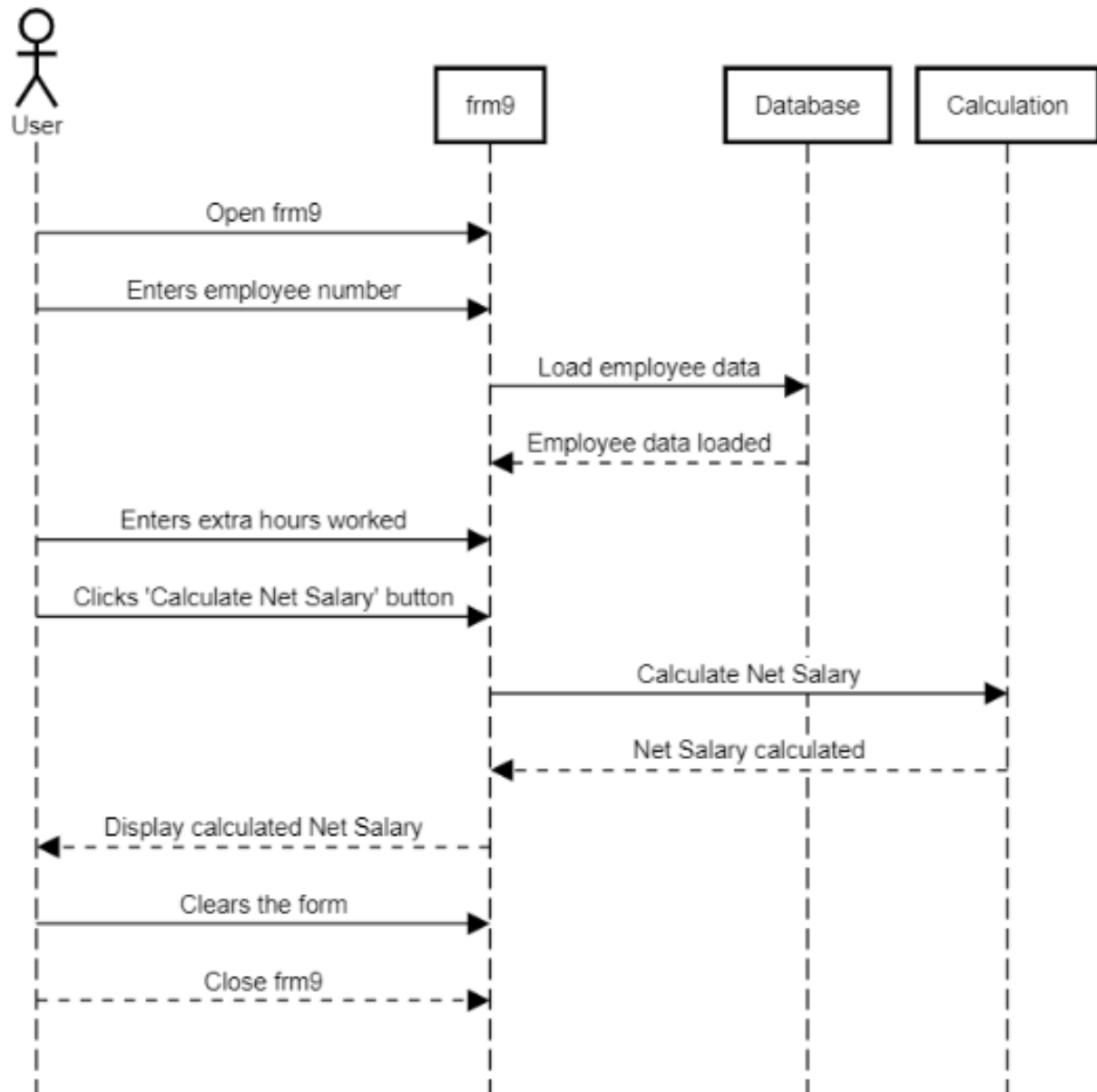








Employee Salary Calculator

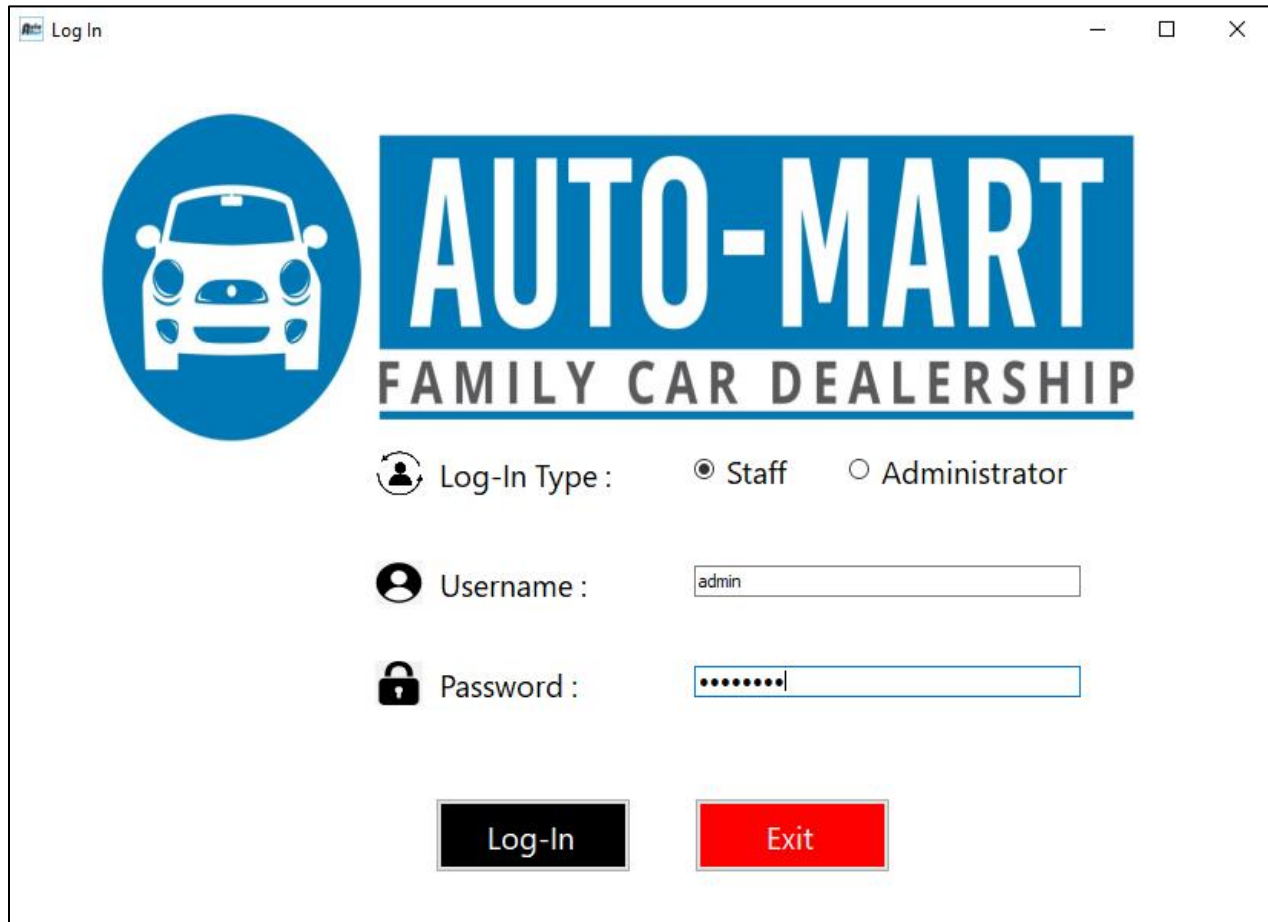


6. USER ROLES , INTERFACES AND IMPLEMENTATION

6.1 Login Form

Functionality: User Authentication

Description: Allows users to log in to the system securely by providing their credentials (e.g., username and password). It verifies user identity and grants access to the system based on the provided information.



The screenshot shows a web browser window titled "Log In". The page features the "AUTO-MART FAMILY CAR DEALERSHIP" logo, which consists of a blue circle containing a white car icon, followed by the text "AUTO-MART" in large white letters on a blue background, and "FAMILY CAR DEALERSHIP" in smaller grey letters below it. Below the logo, there are three login fields: "Log-In Type" with radio buttons for "Staff" (selected) and "Administrator"; "Username" with a text box containing "admin"; and "Password" with a text box containing seven dots. At the bottom, there are two buttons: a black "Log-In" button and a red "Exit" button.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;
using System.Data.SqlClient;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm1 : Form
    {
        public frm1()
        {
            InitializeComponent();
        }

        string aun = "admin";
        string apw = "admin123";
        string sun = "staff";
        string spw = "staff123";
        private void btnregister_Click(object sender, EventArgs e)
        {

        }

        private void btnlogin_Click(object sender, EventArgs e)
        {

        }

        private void btnexit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void frm1_Load(object sender, EventArgs e)
        {
            lbllogintype.BackColor = Color.Transparent;
            lblun.BackColor = Color.Transparent;
            lblpw.BackColor = Color.Transparent;

```

```

        rdbtnadministrator.BackColor = Color.Transparent;
        rdbtnstaff.BackColor = Color.Transparent;
        pictureBox1.BackColor = Color.Transparent;
        pictureBox2.BackColor = Color.Transparent;
    }

    private void btnlogin_Click_1(object sender, EventArgs e)
    {
        if (txtun.Text == aun && txtpw.Text == apw && rdbtnadministrator.Checked == true)
        {
            frm2 obj = new frm2();
            obj.Show();
            this.Hide();
            MessageBox.Show("Login Successfull!", "Welcome To Admin Dashboard",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else if (txtun.Text != aun && txtpw.Text != apw && rdbtnadministrator.Checked ==
        true)
        {
            MessageBox.Show("Invalid username & password", "Information",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else if (txtun.Text == aun && txtpw.Text != apw && rdbtnadministrator.Checked ==
        true)
        {
            MessageBox.Show("Invalid password", "Information", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }
        else if (txtun.Text != aun && txtpw.Text == apw && rdbtnadministrator.Checked ==
        true)
        {
            MessageBox.Show("Invalid username", "Information", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }
        else if (txtun.Text == sun && txtpw.Text == spw && rdbtnstaff.Checked == true)
        {
            frm3 obj = new frm3();
            obj.Show();
            this.Hide();
            MessageBox.Show("Login Successfull!", "Welcome To Staff Dashboard",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else if (txtun.Text != sun && txtpw.Text != spw && rdbtnstaff.Checked == true)
        {
            MessageBox.Show("Invalid username & password", "Information",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }

```

```

    }
    else if (txtun.Text == sun && txtpw.Text != spw && rdbtnstaff.Checked == true)
    {
        MessageBox.Show("Invalid password", "Information", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    else if (txtun.Text != sun && txtpw.Text == spw && rdbtnstaff.Checked == true)
    {
        MessageBox.Show("Invalid username", "Information", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("Please select the login type and enter the username and
        password");
    }
}

private void btnexit_Click_1(object sender, EventArgs e)
{
    this.Close();
}

private void btnlogin_MouseEnter(object sender, EventArgs e)
{
    btnlogin.BackColor = Color.RoyalBlue;
}

private void btnlogin_MouseLeave(object sender, EventArgs e)
{
    btnlogin.BackColor = Color.Black;
}

private void btnexit_MouseEnter(object sender, EventArgs e)
{
    btnexit.BackColor = Color.RoyalBlue;
}

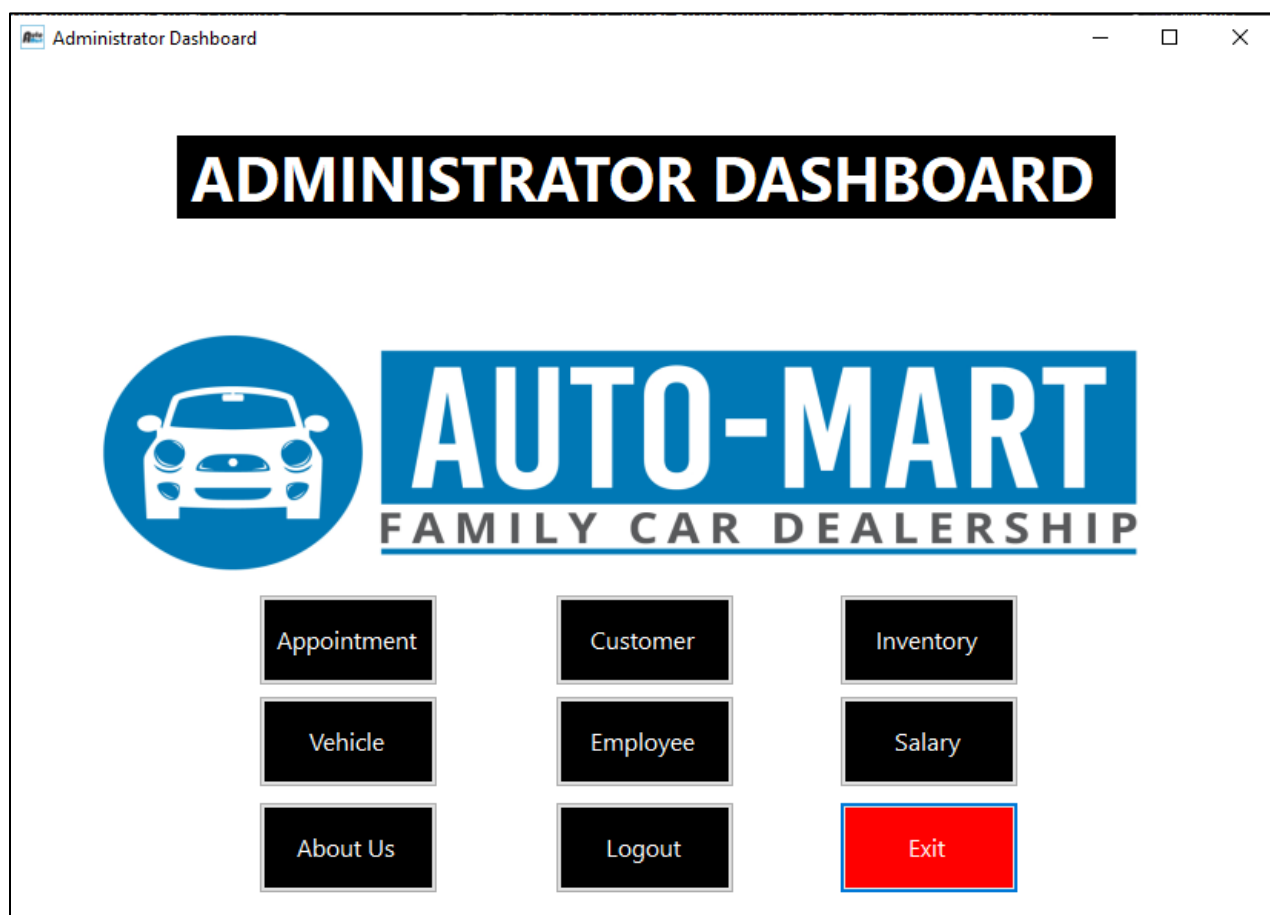
private void btnexit_MouseLeave(object sender, EventArgs e)
{
    btnexit.BackColor = Color.Red;
}
}
}

```


6.2 Admin Dashboard

Functionality: System Administration

Description: Provides administrators with a centralized dashboard to manage the entire VSMS. It includes features for user management, role assignments, system settings, and other administrative tasks.



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm2 : Form
    {
        public frm2()
        {
            InitializeComponent();
        }

        private void btncustomer_Click(object sender, EventArgs e)
        {
            frm5 obj = new frm5();
            obj.Show();
            this.Hide();
        }

        private void btnappointment_Click(object sender, EventArgs e)
        {
            frm4 obj = new frm4();
            obj.Show();
            this.Hide();
        }

        private void btnvehicle_Click(object sender, EventArgs e)
        {
            frm7 obj = new frm7();
            obj.Show();
            this.Hide();
        }

        private void btnemployee_Click(object sender, EventArgs e)
        {
            frm8 obj = new frm8();

```

```

        obj.Show();
        this.Hide();
    }

    private void btninventory_Click(object sender, EventArgs e)
    {
        frm6 obj = new frm6();
        obj.Show();
        this.Hide();
    }

    private void btnsalary_Click(object sender, EventArgs e)
    {
        frm9 obj = new frm9();
        obj.Show();
        this.Hide();
    }

    private void btnlogout_Click(object sender, EventArgs e)
    {
        frm1 obj = new frm1();
        obj.Show();
        this.Hide();
        MessageBox.Show("Successfully Logout!!", "Message", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }

    private void btnexit_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void frm2_Load(object sender, EventArgs e)
    {

    }

    private void btnaboutus_Click(object sender, EventArgs e)
    {
        string url =
        "file:///C:/Users/Ravindu%20Haputhanthri/Desktop/COSC%2031112%20Visual%20Programmi
        ng%20Final%20Project%20Group19/car.html";
        System.Diagnostics.Process.Start(url);
    }

    private void btnappointment_MouseEnter(object sender, EventArgs e)

```

```

{
    btnappointment.BackColor = Color.RoyalBlue;
}

private void btnappointment_MouseLeave(object sender, EventArgs e)
{
    btnappointment.BackColor = Color.Black;
}

private void btncustomer_MouseEnter(object sender, EventArgs e)
{
    btncustomer.BackColor = Color.RoyalBlue;
}

private void btncustomer_MouseLeave(object sender, EventArgs e)
{
    btncustomer.BackColor = Color.Black;
}

private void btninventory_MouseEnter(object sender, EventArgs e)
{
    btninventory.BackColor = Color.RoyalBlue;
}

private void btninventory_MouseLeave(object sender, EventArgs e)
{
    btninventory.BackColor = Color.Black;
}

private void btnvehicle_MouseEnter(object sender, EventArgs e)
{
    btnvehicle.BackColor = Color.RoyalBlue;
}

private void btnvehicle_MouseLeave(object sender, EventArgs e)
{
    btnvehicle.BackColor = Color.Black;
}

private void btnemployee_MouseEnter(object sender, EventArgs e)
{
    btnemployee.BackColor = Color.RoyalBlue;
}

private void btnemployee_MouseLeave(object sender, EventArgs e)

```

```

    {
        btnemployee.BackColor = Color.Black;
    }

    private void btnsalary_MouseEnter(object sender, EventArgs e)
    {
        btnsalary.BackColor = Color.RoyalBlue;
    }

    private void btnsalary_MouseLeave(object sender, EventArgs e)
    {
        btnsalary.BackColor = Color.Black;
    }

    private void btnaboutus_MouseEnter(object sender, EventArgs e)
    {
        btnaboutus.BackColor = Color.RoyalBlue;
    }

    private void btnaboutus_MouseLeave(object sender, EventArgs e)
    {
        btnaboutus.BackColor = Color.Black;
    }

    private void btnlogout_MouseEnter(object sender, EventArgs e)
    {
        btnlogout.BackColor = Color.RoyalBlue;
    }

    private void btnlogout_MouseLeave(object sender, EventArgs e)
    {
        btnlogout.BackColor = Color.Black;
    }

    private void btnexit_MouseEnter(object sender, EventArgs e)
    {
        btnexit.BackColor = Color.RoyalBlue;
    }

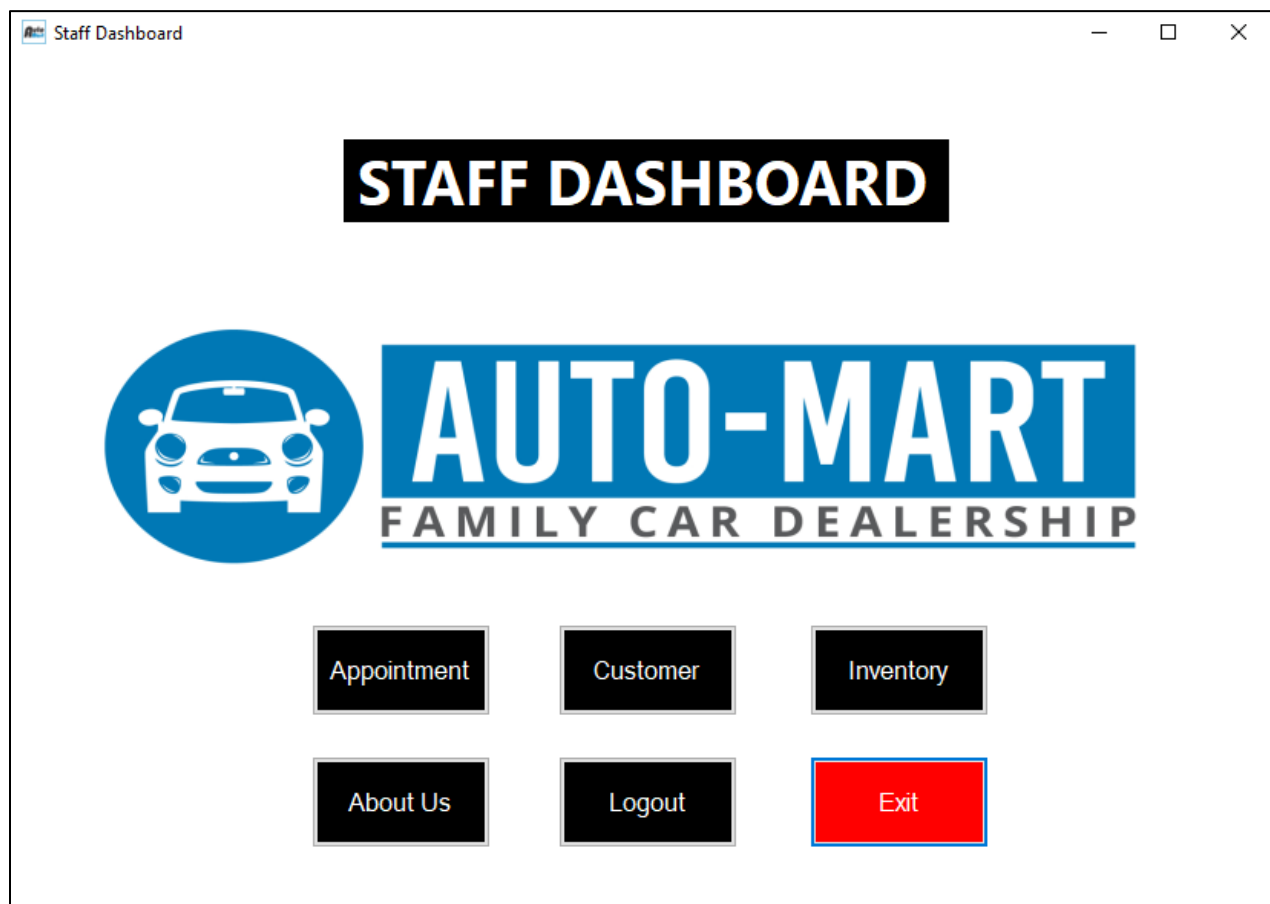
    private void btnexit_MouseLeave(object sender, EventArgs e)
    {
        btnexit.BackColor = Color.Red;
    }
}

```

6.3 Staff Dashboard

Functionality: Role-Based Access

Description: Offers dashboard views tailored to dealership staff members. Depending on their roles (e.g., sales, support), staff can access functionalities relevant to their responsibilities, such as sales management, customer interactions, and appointments.



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm3 : Form
    {
        public frm3()
        {
            InitializeComponent();
        }

        private void btncustomer3_Click(object sender, EventArgs e)
        {
            frm11 obj = new frm11();
            obj.Show();
            this.Hide();
        }

        private void btnappointment3_Click(object sender, EventArgs e)
        {
            frm10 obj = new frm10();
            obj.Show();
            this.Hide();
        }

        private void btninventory3_Click(object sender, EventArgs e)
        {
            frm12 obj = new frm12();
            obj.Show();
            this.Hide();
        }

        private void btnlogout3_Click(object sender, EventArgs e)
        {
            frm1 obj = new frm1();
            obj.Show();
        }
    }
}

```

```

        this.Hide();
        MessageBox.Show("Successfully
Logout!!", "Message", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    private void btnexit3_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void btnaboutus3_Click(object sender, EventArgs e)
    {
        string url =
"file:///C:/Users/Ravindu%20Haputhanthri/Desktop/COSC%2031112%20Visual%20Programmi
ng%20Final%20Project%20Group19/car.html";
        System.Diagnostics.Process.Start(url);
    }

    private void btnappointment3_MouseEnter(object sender, EventArgs e)
    {
        btnappointment3.BackColor = Color.RoyalBlue;
    }

    private void btnappointment3_MouseLeave(object sender, EventArgs e)
    {
        btnappointment3.BackColor = Color.Black;
    }

    private void btncustomer3_MouseEnter(object sender, EventArgs e)
    {
        btncustomer3.BackColor = Color.RoyalBlue;
    }

    private void btncustomer3_MouseLeave(object sender, EventArgs e)
    {
        btncustomer3.BackColor = Color.Black;
    }

    private void btninventory3_MouseEnter(object sender, EventArgs e)
    {
        btninventory3.BackColor = Color.RoyalBlue;
    }

    private void btninventory3_MouseLeave(object sender, EventArgs e)
    {
        btninventory3.BackColor = Color.Black;
    }

```



```

    }

    private void btnaboutus3_MouseEnter(object sender, EventArgs e)
    {
        btnaboutus3.BackColor = Color.RoyalBlue;
    }

    private void btnaboutus3_MouseLeave(object sender, EventArgs e)
    {
        btnaboutus3.BackColor = Color.Black;
    }

    private void btnlogout3_MouseEnter(object sender, EventArgs e)
    {
        btnlogout3.BackColor = Color.RoyalBlue;
    }

    private void btnlogout3_MouseLeave(object sender, EventArgs e)
    {
        btnlogout3.BackColor = Color.Black;
    }

    private void btnexit3_MouseEnter(object sender, EventArgs e)
    {
        btnexit3.BackColor = Color.RoyalBlue;
    }

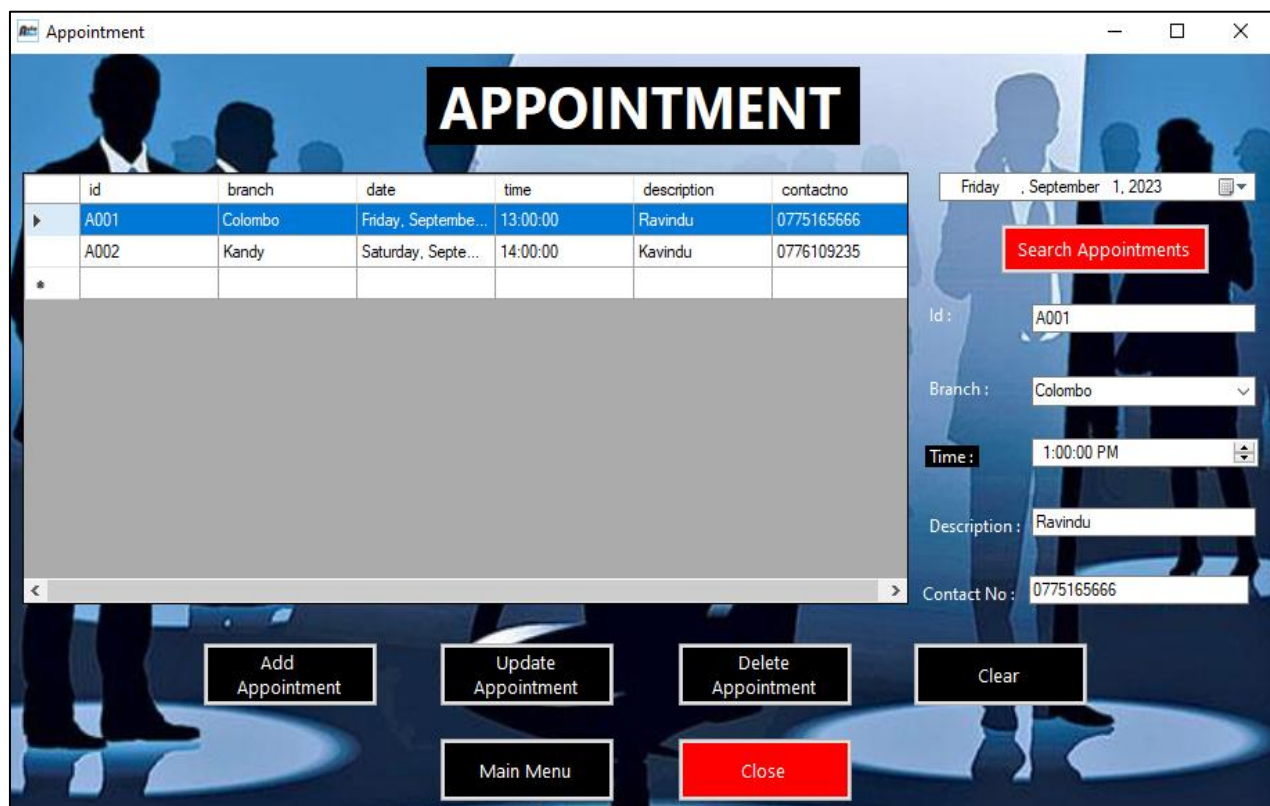
    private void btnexit3_MouseLeave(object sender, EventArgs e)
    {
        btnexit3.BackColor = Color.Black;
    }
}

```

6.4 Appointment

Functionality: Appointment Scheduling and Management

Description: Allows users to schedule and manage appointments for dealership-related activities. Users can create appointments for sales consultations, test drives, service appointments, and more, with options to set reminders and track appointment statuses.



The screenshot displays a software window titled "Appointment". The interface features a table with appointment data, a search bar, and several action buttons. The table contains the following data:

	id	branch	date	time	description	contactno
▶	A001	Colombo	Friday, Septembe...	13:00:00	Ravindu	0775165666
	A002	Kandy	Saturday, Septe...	14:00:00	Kavindu	0776109235
*						

Below the table is a large grey rectangular area. To the right of the table, there is a date selector showing "Friday, September 1, 2023" and a red "Search Appointments" button. Below these are input fields for "Id:" (containing "A001"), "Branch:" (a dropdown menu showing "Colombo"), "Time:" (a dropdown menu showing "1:00:00 PM"), "Description:" (containing "Ravindu"), and "Contact No:" (containing "0775165666"). At the bottom of the window, there are five buttons: "Add Appointment", "Update Appointment", "Delete Appointment", "Clear", and "Main Menu". A red "Close" button is located at the bottom right.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Collections;
using System.Data.Common;
using COSC_31112_Visual_Programming_Final_Project_Group19.AppointmentsTableAdapters;
using System.Xml.Linq;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm4 : Form
    {
        public frm4()
        {
            InitializeComponent();

            private void Form4_Load(object sender, EventArgs e)
            {
                // TODO: This line of code loads data into the 'group19DataSet.Appointment' table. You can move,
                or remove it, as needed.
                this.appointmentTableAdapter.Fill(this.group19DataSet.Appointment);
                cbxbranch.Items.Add("Colombo");
                cbxbranch.Items.Add("Kandy");
                cbxbranch.Items.Add("Galle");
                cbxbranch.Items.Add("Jaffna");
                cbxbranch.Items.Add("Ampara");

                lblbranch.BackColor = Color.Transparent;
                lblid.BackColor = Color.Transparent;
                lbldescription.BackColor = Color.Transparent;
                lblcontactnumber.BackColor = Color.Transparent;

                gridLoad();
            }

```

```

private void btnaddappointment_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);
    SqlCommand com;

    if (txtid.Text.Length < 3)
    {
        MessageBox.Show("Wrong Length", "Length Check", MessageBoxButtons.OKCancel,
        MessageBoxIcon.Information);
        txtid.Clear();
        txtid.Focus();
    }
    else
    {
        try
        {
            con.Open();
            string sql = "insert into Appointment(id,branch,date,time,description,contactno) values('" +
txtid.Text + "','" + cbxbranch.Text + "','" + date4.Text + "','" + time4.Text + "','" + txtdescription.Text + "','" +
txtcontactnumber.Text + "')";
            com = new SqlCommand(sql, con);
            com.ExecuteNonQuery();
            gridLoad();
            MessageBox.Show("Appointment Added Successfully", "Add Appointment",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtid.Text = "";
            date4.Text = "";
            time4.Text = "";
            cbxbranch.Text = "";
            txtdescription.Text = "";
            txtcontactnumber.Text = "";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        finally
        {
            con.Close();
        }
    }
}

```

```

private void btnsearchappointment_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);

    try
    {
        con.Open();
        string sql = "select * from Appointment where date='" + date4.Text + "'";
        using (SqlCommand command = new SqlCommand(sql, con))
        {
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            // Bind the DataTable to the DataGridView
            dataGridView4.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
    }
}

private void dataGridView4_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}

private void btnupdateappointment_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";

    try
    {
        using (SqlConnection con = new SqlConnection(ConnectionString))

```

```

    {
        con.Open();

        if (MessageBox.Show("Do You Want To Update The Appointment Data?", "Confirmation",
            MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) == DialogResult.OK)
        {
            string query = "UPDATE Appointment SET branch = @branch, date = @date, time = @time,
            description = @description, contactno = @contactno WHERE id = @id";

            using (SqlCommand cmd = new SqlCommand(query, con))
            {
                cmd.Parameters.AddWithValue("@id", txtid.Text); // Assuming id is an integer
                cmd.Parameters.AddWithValue("@branch", cbxbranch.Text);
                cmd.Parameters.AddWithValue("@date", date4.Text);
                cmd.Parameters.AddWithValue("@time", time4.Text); // Ensure that the data type
            matches your database
                cmd.Parameters.AddWithValue("@description", txtdescription.Text);
                cmd.Parameters.AddWithValue("@contactno", txtcontactnumber.Text);

                int rowsAffected = cmd.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    gridLoad(); // Reload the data into your DataGridView
                    MessageBox.Show("Appointment successfully updated.", "Information",
            MessageBoxButtons.OK, MessageBoxIcon.Information);

                }
                else
                {
                    MessageBox.Show("Failed to update appointment details.", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show("An error occurred: " + ex.Message, "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

```

```

private void btndeleteappointment_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);

    try
    {
        if (MessageBox.Show("Do You Want To Delete This Appointment Detail, Confirm?", "Successful",
MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) == DialogResult.OK)
        {
            string query = "DELETE FROM Appointment WHERE id = @id";

            using (SqlCommand cmd = new SqlCommand(query, con))
            {
                con.Open();
                cmd.Parameters.AddWithValue("@id", txtid.Text); // Assuming clicked_date is the id
                int rowsAffected = cmd.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    gridLoad();// Refresh your DataGridView or grid
                    MessageBox.Show("Appointment Detail Successfully Deleted.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    con.Close();

                    txtid.Text = "";
                    cbxbranch.Text = "";
                    date4.Text = "";
                    time4.Text = "";
                    txtdescription.Text = "";
                    txtcontactnumber.Text = "";
                }
                else
                {
                    MessageBox.Show("Failed to Delete Appointment Details.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }
    }
    catch (Exception e1)
    {

```

```

        MessageBox.Show("An error occurred: " + e1.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

```

```

private void btnclear4_Click(object sender, EventArgs e)
{
    txtid.Text = "";
    cbxbranch.Text = "";
    date4.Text = "";
    time4.Text = "";
    txtdescription.Text = "";
    txtcontactnumber.Text = "";
    gridLoad();
}

```

```

private void btnmainmenu4_Click(object sender, EventArgs e)
{
    frm2 obj = new frm2();
    obj.Show();
    this.Hide();
}

```

```

private void btnclose4_Click(object sender, EventArgs e)
{
    this.Close();
}

```

```

private void lblbranch_Click(object sender, EventArgs e)
{
}

```

```

private string clicked_id;
private void dataGridView4_CellClick(object sender, DataGridViewCellEventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);

    if (e.RowIndex >= 0)
    {

        clicked_id = dataGridView4.Rows[e.RowIndex].Cells[0].Value.ToString();
    }
}

```



```

string query = "SELECT * FROM Appointment WHERE id = @id";

using (SqlCommand cmd = new SqlCommand(query, con))
{
    con.Open();
    cmd.Parameters.AddWithValue("@id", clicked_id);
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataSet ds = new DataSet();
    da.Fill(ds);

    if (ds.Tables[0].Rows.Count > 0)
    {
        txtid.Text = ds.Tables[0].Rows[0][0].ToString();
        cbxbranch.Text = ds.Tables[0].Rows[0][1].ToString();
        date4.Text = ds.Tables[0].Rows[0][2].ToString();
        time4.Text = ds.Tables[0].Rows[0][3].ToString();
        txtdescription.Text = ds.Tables[0].Rows[0][4].ToString();
        txtcontactnumber.Text = ds.Tables[0].Rows[0][5].ToString();
    }
    con.Close();
}
}

private void gridLoad()
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);
    SqlCommand cmd = new SqlCommand("SELECT * FROM Appointment ", con);
    SqlDataAdapter da1 = new SqlDataAdapter(cmd);
    DataSet ds1 = new DataSet();
    da1.Fill(ds1);

    dataGridView4.DataSource = ds1.Tables[0];

}

}
}

```

6.5 Customer

Functionality: Customer Relationship Management (CRM)

Description: Provides tools for managing customer data. Users can create and update customer profiles, record customer interactions, track purchase history, and send notifications or updates to customers.

CUSTOMER

Personal Details

Customer No : C001

Title : Mr.

First Name : Ravindu

Last Name : Haputhanthri

Date Of Birth : Monday , December 6, 1999

Address : 52/3 Katubedda Moratuwa

NIC/Passport No : 993411205v

Land-Phone No : 0116109235

Mobile No : 0775165666

Email : ravinduheshan99@gmail.com

Vehicle Details

Make : Mercedes-Benz

Model : CLA 200

Chassis No : CHS001

Status : Unregistered

Registration No : R001

Add Customer Update Customer Search Customer Delete Customer

Main Menu Clear Close

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm5 : Form
    {
        public frm5()
        {
            InitializeComponent();

            private void btnaddcustomer_Click(object sender, EventArgs e)
            {
                string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
                SqlConnection con = new SqlConnection(ConnectionString);
                SqlCommand com;

                if (txtcustomerno.Text.Length < 3)
                {
                    MessageBox.Show("Wrong Length", "Length Check", MessageBoxButtons.OKCancel,
                    MessageBoxIcon.Information);
                    txtcustomerno.Clear();
                    txtcustomerno.Focus();
                }
                else
                {
                    try
                    {
                        con.Open();

```

```

        string sql = "insert into
Customer(cusno,title,fname,lname,dob,adrs,nic,lphone,hphone,email,make,model,chassisno,statu
s,regno) values('" + txtcustomerno.Text + "','" + cbxtitle.Text + "','" + txtfn.Text + "','" +
txtln.Text + "','" + dtpdateofbirth.Text + "','" + txtaddress.Text + "','" + txtnic.Text + "','" +
txtlp.Text + "','" + txthp.Text + "','" + txtemail.Text + "','" + cbxmake.Text + "','" + txtmodel.Text
+ "','" + txtchassis.Text + "','" + cbxstatus.Text + "','" + txtregistrationno.Text + "')";
        com = new SqlCommand(sql, con);
        com.ExecuteNonQuery();
        MessageBox.Show("Customer Added Successfully", "Message",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

```

```

        txtcustomerno.Text = "";
        cbxtitle.Text = "";
        txtfn.Text = "";
        txtln.Text = "";
        dtpdateofbirth.Text = "";
        txtaddress.Text = "";
        txtnic.Text = "";
        txtlp.Text = "";
        txthp.Text = "";
        txtemail.Text = "";
        cbxmake.Text = "";
        txtmodel.Text = "";
        txtchassis.Text = "";
        cbxstatus.Text = "";
        txtregistrationno.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
    }
}
}

```

```

private void frm5_Load(object sender, EventArgs e)
{
    cbxtitle.Items.Add("Mr.");
}

```

```
cbxtitle.Items.Add("Mrs.");  
cbxtitle.Items.Add("Miss.");  
cbxtitle.Items.Add("Dr.");  
cbxtitle.Items.Add("Prof.");  
cbxtitle.Items.Add("Rev.");
```

```
cbxmake.Items.Add("Acura");  
cbxmake.Items.Add("Alfa Romeo");  
cbxmake.Items.Add("AM General");  
cbxmake.Items.Add("Audi");  
cbxmake.Items.Add("BMW");  
cbxmake.Items.Add("Chevrolet");  
cbxmake.Items.Add("Chrysler");  
cbxmake.Items.Add("Daihatsu");  
cbxmake.Items.Add("FIAT");  
cbxmake.Items.Add("Ford");  
cbxmake.Items.Add("Honda");  
cbxmake.Items.Add("Hummer");  
cbxmake.Items.Add("Hyundai");  
cbxmake.Items.Add("Isuzu");  
cbxmake.Items.Add("Jaguar");  
cbxmake.Items.Add("Jeep");  
cbxmake.Items.Add("Kia");  
cbxmake.Items.Add("Land Rover");  
cbxmake.Items.Add("Mazda");  
cbxmake.Items.Add("Mercedes-Benz");  
cbxmake.Items.Add("MINI");  
cbxmake.Items.Add("Mitsubishi");  
cbxmake.Items.Add("Nissan");  
cbxmake.Items.Add("Peugeot");  
cbxmake.Items.Add("Porsche");  
cbxmake.Items.Add("Renault");  
cbxmake.Items.Add("Rolls-Royce");  
cbxmake.Items.Add("Subaru");  
cbxmake.Items.Add("Suzuki");  
cbxmake.Items.Add("Toyota");  
cbxmake.Items.Add("Volkswagen");  
cbxmake.Items.Add("Volvo");
```

```
cbxstatus.Items.Add("Registered");  
cbxstatus.Items.Add("Unregistered");
```

```

    }

    private void btnsearchcustomer_Click(object sender, EventArgs e)
    {
        string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
        SqlConnection con = new SqlConnection(ConnectionString);
        SqlCommand com;

        if (txtcustomerno.Text.Length < 3)
        {
            MessageBox.Show("Wrong Length", "Length Check", MessageBoxButtons.OKCancel,
            MessageBoxIcon.Information);
            txtcustomerno.Clear();
            txtcustomerno.Focus();
        }
        else
        {
            try
            {
                con.Open();
                string sql = "select * from Customer where cusno = '" + txtcustomerno.Text + "'";
                com = new SqlCommand(sql, con);
                com.Parameters.AddWithValue("cusno", txtcustomerno.Text);
                SqlDataReader dr;
                dr = com.ExecuteReader();

                if (dr.Read())
                {
                    cbxtitle.Text = dr["title"].ToString();
                    txtfn.Text = dr["fname"].ToString();
                    txtln.Text = dr["lname"].ToString();
                    dtupdateofbirth.Text = dr["dob"].ToString();
                    txtaddress.Text = dr["adrs"].ToString();
                    txtnic.Text = dr["nic"].ToString();
                    txtlp.Text = dr["lphone"].ToString();
                    txtthp.Text = dr["hphone"].ToString();
                    txtemail.Text = dr["email"].ToString();
                    cbxmake.Text = dr["make"].ToString();
                    txtmodel.Text = dr["model"].ToString();
                }
            }
            catch { }
        }
    }
}

```

```

        txtchassis.Text = dr["chassisno"].ToString();
        cbxstatus.Text = dr["status"].ToString();
        txtregistrationno.Text = dr["regno"].ToString() ;
    }
    else
    {
        MessageBox.Show("Customer Not
Found.", "Message", MessageBoxButtons.OKCancel, MessageBoxIcon.Exclamation);
        txtcustomerno.Clear();
        txtcustomerno.Focus();
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    con.Close();
}
}
}

private void btndeletecustomer_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);
    SqlCommand com;

    try
    {
        con.Open();
        string sql = "Delete from Customer where cusno = '"+txtcustomerno.Text+"'";
        com = new SqlCommand(sql, con);
        com.ExecuteNonQuery();
        MessageBox.Show("Customer Deleted Successfully", "Message",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

        txtcustomerno.Text = "";
        cbxtitle.Text = "";
    }
}

```

```

        txtfn.Text = "";
        txtln.Text = "";
        dtpdateofbirth.Text = "";
        txtaddress.Text = "";
        txtnic.Text = "";
        txtlp.Text = "";
        txthp.Text = "";
        txtemail.Text = "";
        cbxmake.Text = "";
        txtmodel.Text = "";
        txtchassis.Text = "";
        cbxstatus.Text = "";
        txtregistrationno.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close ();
    }
}

private void btnupdatecustomer_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);
    SqlCommand com;

    try
    {
        con.Open ();
        string sql = "Update customer set title='" + cbxtitle.Text + "',fname='" + txtfn.Text +
        "',lname='" + txtln.Text + "',dob='" + dtpdateofbirth.Text + "',adrs='" + txtaddress.Text + "',nic='"
        + txtnic.Text + "',lphone='" + txtlp.Text + "',hphone='" + txthp.Text + "',email='" + txtemail.Text
        + "',make='" + cbxmake.Text + "',model='" + txtmodel.Text + "',chassisno='" + txtchassis.Text +
        "',status='" + cbxstatus.Text + "',regno='" + txtregistrationno.Text + "'where cusno='" +
        txtcustomerno.Text + "'";
        com = new SqlCommand(sql, con);
    }
}

```



```
com.ExecuteNonQuery();

MessageBox.Show("Customer Detail Updated Successfully.", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
```

```
txtcustomerno.Text = "";
cbxtitle.Text = "";
txtfn.Text = "";
txtln.Text = "";
dtpdateofbirth.Text = "";
txtaddress.Text = "";
txtnic.Text = "";
txtlp.Text = "";
txthp.Text = "";
txtemail.Text = "";
cbxmake.Text = "";
txtmodel.Text = "";
txtchassis.Text = "";
cbxstatus.Text = "";
txtregistrationno.Text = "";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    con.Close();
}
}

private void btnclear5_Click(object sender, EventArgs e)
{
    txtcustomerno.Text = "";
    cbxtitle.Text = "";
    txtfn.Text = "";
    txtln.Text = "";
    dtpdateofbirth.Text = "";
    txtaddress.Text = "";
    txtnic.Text = "";
    txtlp.Text = "";
```

```

        txthp.Text = "";
        txtemail.Text = "";
        cbxmake.Text = "";
        txtmodel.Text = "";
        txtchassis.Text = "";
        cbxstatus.Text = "";
        txtregistrationno.Text = "";
    }

    private void btnclose5_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void btnmainmenu5_Click(object sender, EventArgs e)
    {
        frm2 obj = new frm2();
        obj.Show();
        this.Hide();
    }
}
}

```

6.6 Inventory

Functionality: Vehicle Inventory Management

Description: Allows users to manage the dealership's vehicle inventory. Features include adding new listings, editing existing listings, tracking vehicle history, and marking vehicles as sold or unavailable.

The screenshot shows a web application window titled "Inventory". The interface features a sidebar with various filters, a main table of vehicle listings, and a bottom navigation bar with action buttons.

Filters:

- Branch :
- Brand :
- Vehicle Type :
- Condition :
- Year Of Manufacture :
- Transmission :
- Fuel Type :
- Body Type :
- Vehicle Register No :
- Chassis No :
- Engine Capacity :
- Price :
- Quantity :

Inventory Table:

brand	vtype	condition	year	transmission	branch	ftype
Mercedes-Benz	Car	Brand New	2023	Auto	Colombo	Petrol
Toyota	Jeep	Brand New	2018	Auto	Kandy	Diesel
Audi	Car	Brand New	2020	Auto	Galle	Petrol
...						

Bottom Bar:

- Search Vehicle
- Main Menu
- Clear
- Close

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm6 : Form
    {
        public frm6()
        {
            InitializeComponent();
        }

        private void lblcustomerno_Click(object sender, EventArgs e)
        {
        }

        private void btnsearchvehicle_Click(object sender, EventArgs e)
        {
            string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
            SqlConnection con = new SqlConnection(ConnectionString);

            try
            {
                con.Open();
                string sql = "select * from Inventory where branch=" + cbxbranch.Text + " or brand="
+ cbxbrand.Text + " or vtype=" + cbxvehicletype.Text + " or condition=" + cbxcondition.Text
+ " or year=" + txtyear.Text + " or transmission=" + cbxtransmission.Text + " or ftype=" +
cbxfueltype.Text + " or btype=" + cbxbodytype.Text + " or vregno=" + txtvehicleregno.Text +
" or ecapacity=" + txtenginecapacity.Text + " or price=" + txtprice.Text + " or qty=" +
txtqty.Text + """;
                using (SqlCommand command = new SqlCommand(sql, con))
                {
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dataTable = new DataTable();

```

```

        adapter.Fill(dataTable);

        // Bind the DataTable to the DataGridView
        dataGridView6.DataSource = dataTable;

        /*cbxbrand.Text = "";
        cbxvehicletype.Text = "";
        cbxcondition.Text = "";
        txtyear.Text = "";
        cbxtransmission.Text = "";
        cbxbranch.Text = "";
        cbxfueltype.Text = "";
        cbxbodytype.Text = "";
        txtvehicleregno.Text = "";
        txtchassisno.Text = "";
        txtenginecapacity.Text = "";
        txtprice.Text = "";
        txtqty.Text = "";*/
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    con.Close();
}
}

private void frm6_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'group19DataSet.Inventory' table. You can
    move, or remove it, as needed.
    this.inventoryTableAdapter.Fill(this.group19DataSet.Inventory);
    cbxbrand.Items.Add("Acura");
    cbxbrand.Items.Add("Alfa Romeo");
    cbxbrand.Items.Add("AM General");
    cbxbrand.Items.Add("Audi");
    cbxbrand.Items.Add("BMW");
    cbxbrand.Items.Add("Chevrolet");
    cbxbrand.Items.Add("Chrysler");
    cbxbrand.Items.Add("Daihatsu");
    cbxbrand.Items.Add("FIAT");
    cbxbrand.Items.Add("Ford");
    cbxbrand.Items.Add("Honda");

```

```
cbxbrand.Items.Add("Hummer");
cbxbrand.Items.Add("Hyundai");
cbxbrand.Items.Add("Isuzu");
cbxbrand.Items.Add("Jaguar");
cbxbrand.Items.Add("Jeep");
cbxbrand.Items.Add("Kia");
cbxbrand.Items.Add("Land Rover");
cbxbrand.Items.Add("Mazda");
cbxbrand.Items.Add("Mercedes-Benz");
cbxbrand.Items.Add("MINI");
cbxbrand.Items.Add("Mitsubishi");
cbxbrand.Items.Add("Nissan");
cbxbrand.Items.Add("Peugeot");
cbxbrand.Items.Add("Porsche");
cbxbrand.Items.Add("Renault");
cbxbrand.Items.Add("Rolls-Royce");
cbxbrand.Items.Add("Subaru");
cbxbrand.Items.Add("Suzuki");
cbxbrand.Items.Add("Toyota");
cbxbrand.Items.Add("Volkswagen");
cbxbrand.Items.Add("Volvo");
```

```
cbxvehicletype.Items.Add("Car");
cbxvehicletype.Items.Add("Van");
cbxvehicletype.Items.Add("Jeep");
cbxvehicletype.Items.Add("Bike");
cbxvehicletype.Items.Add("Threewheeler");
cbxvehicletype.Items.Add("Bus");
cbxvehicletype.Items.Add("Lorry");
cbxvehicletype.Items.Add("Heavy Vehicle");
```

```
cbxcondition.Items.Add("Brand New");
cbxcondition.Items.Add("Used");
cbxcondition.Items.Add("Reconditioned");
```

```
cbxtransmission.Items.Add("Auto");
cbxtransmission.Items.Add("Manual");
cbxtransmission.Items.Add("Tiptronic");
cbxtransmission.Items.Add("Other");
```

```
cbxbranch.Items.Add("Colombo");
cbxbranch.Items.Add("Kandy");
cbxbranch.Items.Add("Galle");
cbxbranch.Items.Add("Jaffna");
cbxbranch.Items.Add("Ampara");
```

```

        cbxfueltype.Items.Add("Petrol");
        cbxfueltype.Items.Add("Diesel");
        cbxfueltype.Items.Add("Hybrid");
        cbxfueltype.Items.Add("Electronic");
        cbxfueltype.Items.Add("Other");

        cbxbodytype.Items.Add("Convertible");
        cbxbodytype.Items.Add("Coup");
        cbxbodytype.Items.Add("Hatchback");
        cbxbodytype.Items.Add("Minivan");
        cbxbodytype.Items.Add("Pickup");
        cbxbodytype.Items.Add("sedan");
        cbxbodytype.Items.Add("SUV");
        cbxbodytype.Items.Add("Wagon");

        gridLoad();
    }

    private void btnclose7_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void btnmainmenu7_Click(object sender, EventArgs e)
    {
        frm2 obj = new frm2();
        obj.Show();
        this.Hide();
    }

    private void btnclear7_Click(object sender, EventArgs e)
    {
        cbxbrand.Text = "";
        cbxvehicletype.Text = "";
        cbxcondition.Text = "";
        txtyear.Text = "";
        cbxtransmission.Text = "";
        cbxbranch.Text = "";
        cbxfueltype.Text = "";
        cbxbodytype.Text = "";
        txtvehicleregno.Text = "";
        txtchassisno.Text = "";
        txtenginecapacity.Text = "";
        txtprice.Text = "";
        txtqty.Text = "";
    }

```

```
        gridLoad();  
    }  
  
    private void gridLoad()  
    {  
        string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial  
Catalog=Group19;Integrated Security=True";  
        SqlConnection con = new SqlConnection(ConnectionString);  
        SqlCommand cmd = new SqlCommand("SELECT * FROM Inventory ", con);  
        SqlDataAdapter da1 = new SqlDataAdapter(cmd);  
        DataSet ds1 = new DataSet();  
        da1.Fill(ds1);  
  
        dataGridView6.DataSource = ds1.Tables[0];  
    }  
}
```


6.7 Add or Edit Vehicle

Functionality: Vehicle Listing Management

Description: This interface is used to manage the dealership's vehicle inventory by adding new vehicle listings or editing existing ones. It provides a user-friendly form where users can input or modify details about vehicles, ensuring that accurate and up-to-date information is available to potential buyers and dealership staff

ADD OR EDIT VEHICLE

Branch : ▾

Brand : ▾

Vehicle Type : ▾

Condition : ▾

Year Of Manufacture :

Transmission : ▾

Fuel Type : ▾

Body Type : ▾

Vehicle Register No :

Chassis No :

Engine Capacity :

Price :

Quantity :

brand	vtype	condition	year	transmission	branch	ftype
Mercedes-Benz	Car	Brand New	2023	Auto	Colombo	Petrol
Toyota	Jeep	Brand New	2018	Auto	Kandy	Diesel
Audi	Car	Brand New	2020	Auto	Galle	Petrol
Isuzu	Van	Used	2009	Manual	Jaffna	Diesel

Search Vehicle Add Vehicle Update Vehicle Delete Vehicle Main Menu Clear Close

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm7 : Form
    {
        public frm7()
        {
            InitializeComponent();
        }

        private void btnmainmenu7_Click(object sender, EventArgs e)
        {
            frm2 obj = new frm2();
            obj.Show();
            this.Hide();
        }

        private void frm7_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'group19DataSet.Inventory' table. You can
            move, or remove it, as needed.
            this.inventoryTableAdapter.Fill(this.group19DataSet.Inventory);

            cbxbrand.Items.Add("Acura");
            cbxbrand.Items.Add("Alfa Romeo");
            cbxbrand.Items.Add("AM General");
            cbxbrand.Items.Add("Audi");
            cbxbrand.Items.Add("BMW");
            cbxbrand.Items.Add("Chevrolet");

```

```
cbxbrand.Items.Add("Chrysler");
cbxbrand.Items.Add("Daihatsu");
cbxbrand.Items.Add("FIAT");
cbxbrand.Items.Add("Ford");
cbxbrand.Items.Add("Honda");
cbxbrand.Items.Add("Hummer");
cbxbrand.Items.Add("Hyundai");
cbxbrand.Items.Add("Isuzu");
cbxbrand.Items.Add("Jaguar");
cbxbrand.Items.Add("Jeep");
cbxbrand.Items.Add("Kia");
cbxbrand.Items.Add("Land Rover");
cbxbrand.Items.Add("Mazda");
cbxbrand.Items.Add("Mercedes-Benz");
cbxbrand.Items.Add("MINI");
cbxbrand.Items.Add("Mitsubishi");
cbxbrand.Items.Add("Nissan");
cbxbrand.Items.Add("Peugeot");
cbxbrand.Items.Add("Porsche");
cbxbrand.Items.Add("Renault");
cbxbrand.Items.Add("Rolls-Royce");
cbxbrand.Items.Add("Subaru");
cbxbrand.Items.Add("Suzuki");
cbxbrand.Items.Add("Toyota");
cbxbrand.Items.Add("Volkswagen");
cbxbrand.Items.Add("Volvo");

cbxvehicletype.Items.Add("Car");
cbxvehicletype.Items.Add("Van");
cbxvehicletype.Items.Add("Jeep");
cbxvehicletype.Items.Add("Bike");
cbxvehicletype.Items.Add("Threewheeler");
cbxvehicletype.Items.Add("Bus");
cbxvehicletype.Items.Add("Lorry");
cbxvehicletype.Items.Add("Heavy Vehicle");

cbxcondition.Items.Add("Brand New");
cbxcondition.Items.Add("Used");
cbxcondition.Items.Add("Reconditioned");

cbxtransmission.Items.Add("Auto");
```

```

        cbxtransmission.Items.Add("Manual");
        cbxtransmission.Items.Add("Tiptronic");
        cbxtransmission.Items.Add("Other");

        cbxbranch.Items.Add("Colombo");
        cbxbranch.Items.Add("Kandy");
        cbxbranch.Items.Add("Galle");
        cbxbranch.Items.Add("Jaffna");
        cbxbranch.Items.Add("Ampara");

        cbxfueltype.Items.Add("Petrol");
        cbxfueltype.Items.Add("Diesel");
        cbxfueltype.Items.Add("Hybrid");
        cbxfueltype.Items.Add("Electronic");
        cbxfueltype.Items.Add("Other");

        cbxbodytype.Items.Add("Convertible");
        cbxbodytype.Items.Add("Coup");
        cbxbodytype.Items.Add("Hatchback");
        cbxbodytype.Items.Add("Minivan");
        cbxbodytype.Items.Add("Pickup");
        cbxbodytype.Items.Add("sedan");
        cbxbodytype.Items.Add("SUV");
        cbxbodytype.Items.Add("Wagon");
    }

    private void btnaddvehicle_Click(object sender, EventArgs e)
    {
        string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
        SqlConnection con = new SqlConnection(ConnectionString);
        SqlCommand com;

        if (txtvehicleregno.Text.Length < 3)
        {
            MessageBox.Show("Wrong Length", "Length Check", MessageBoxButtons.OKCancel,
            MessageBoxIcon.Information);
            txtvehicleregno.Clear();
            txtvehicleregno.Focus();
        }
        else

```

```

    {
        try
        {
            con.Open();
            string sql = "insert into
Inventory(brand,vtype,condition,year,transmission,branch,ftype,btype,vregno,chassisno,ecapacity
,price,qty) values('" + cbxbrand.Text + "','" + cbxvehicletype.Text + "','" + cbxcondition.Text + "','"
+ txtyear.Text + "','" + cbxtransmission.Text + "','" + cbxbranch.Text + "','" + cbxfueltype.Text +
"',"' + cbxbodytype.Text + "','" + txtvehicleregno.Text + "','" + txtchassisno.Text + "','" +
txtenginecapacity.Text + "','" + txtprice.Text + "','" + txtqty.Text + "')";
            com = new SqlCommand(sql, con);
            com.ExecuteNonQuery();
            gridLoad();
            MessageBox.Show("Item Added Successfully", "Message", MessageBoxButtons.OK,
MessageBoxIcon.Information);

            cbxbrand.Text = "";
            cbxvehicletype.Text = "";
            cbxcondition.Text = "";
            txtyear.Text = "";
            cbxtransmission.Text = "";
            cbxbranch.Text = "";
            cbxfueltype.Text = "";
            cbxbodytype.Text = "";
            txtvehicleregno.Text = "";
            txtchassisno.Text = "";
            txtenginecapacity.Text = "";
            txtprice.Text = "";
            txtqty.Text = "";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        finally
        {
            con.Close();
        }
    }
}

```

```

private void btnsearchvehicle_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);

    try
    {
        con.Open();
        string sql = "select * from Inventory where branch='" + cbxbranch.Text + "'";
        using (SqlCommand command = new SqlCommand(sql, con))
        {
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            // Bind the DataTable to the DataGridView
            dataGridView7.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
    }
}

private void btnclose7_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btndeletevehicle_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);

```

```

try
{
    if (MessageBox.Show("Do You Want To Delete This Vehicle Detail, Confirm?",
"Successful", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) == DialogResult.OK)
    {
        string query = "DELETE FROM Inventory WHERE vregno = @vregno";

        using (SqlCommand cmd = new SqlCommand(query, con))
        {
            con.Open();
            cmd.Parameters.AddWithValue("@vregno", txtvehicleregno.Text);
            int rowsAffected = cmd.ExecuteNonQuery();

            if (rowsAffected > 0)
            {
                gridLoad();// Refresh your DataGridView or grid
                MessageBox.Show("Vehicle Detail Successfully Deleted.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                con.Close();

                cbxbrand.Text = "";
                cbxvehicletype.Text = "";
                cbxcondition.Text = "";
                txtyear.Text = "";
                cbxtransmission.Text = "";
                cbxbranch.Text = "";
                cbxfueltype.Text = "";
                cbxbodytype.Text = "";
                txtvehicleregno.Text = "";
                txtchassisno.Text = "";
                txtenginecapacity.Text = "";
                txtprice.Text = "";
                txtqty.Text = "";
            }
            else
            {
                MessageBox.Show("Failed to Delete Vehicle Details.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

```

```

    }
}
catch (Exception e1)
{
    MessageBox.Show("An error occurred: " + e1.Message, "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

private void btnclear7_Click(object sender, EventArgs e)
{
    cbxbrand.Text = "";
    cbxvehicletype.Text = "";
    cbxcondition.Text = "";
    txtyear.Text = "";
    cbxtransmission.Text = "";
    cbxbranch.Text = "";
    cbxfueltype.Text = "";
    cbxbodytype.Text = "";
    txtvehicleregno.Text = "";
    txtchassisno.Text = "";
    txtenginecapacity.Text = "";
    txtprice.Text = "";
    txtqty.Text = "";
    gridLoad();
}

private void btnupdatevehicle_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";

    try
    {
        using (SqlConnection con = new SqlConnection(ConnectionString))
        {
            con.Open();

            if (MessageBox.Show("Do You Want To Update The Vehicle Data?", "Confirmation",
            MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) == DialogResult.OK)
            {

```



```

        string query = "UPDATE Inventory SET brand = @brand, vtype = @vtype,
condition = @condition, year = @year, transmission = @transmission , branch=@branch ,
ftype=@ftype , btype=@btype , chassisno=@chassisno , ecapacity=@ecapacity , price=@price ,
qty=@qty WHERE vregno = @vregno";

```

```

        using (SqlCommand cmd = new SqlCommand(query, con))
        {
            cmd.Parameters.AddWithValue("@brand", cbxbrand.Text); // Assuming id is an
integer
            cmd.Parameters.AddWithValue("@vtype", cbxvehicletype.Text);
            cmd.Parameters.AddWithValue("@condition", cbxcondition.Text);
            cmd.Parameters.AddWithValue("@year", txtyear.Text); // Ensure that the data
type matches your database
            cmd.Parameters.AddWithValue("@transmission", cbxtransmission.Text);
            cmd.Parameters.AddWithValue("@branch", cbxbranch.Text);
            cmd.Parameters.AddWithValue("@ftype", cbxfueltype.Text);
            cmd.Parameters.AddWithValue("@btype", cbxbodytype.Text);
            cmd.Parameters.AddWithValue("@vregno", txtvehicleregno.Text); // Ensure that
the data type matches your database
            cmd.Parameters.AddWithValue("@chassisno", txtchassisno.Text);
            cmd.Parameters.AddWithValue("@ecapacity", txtenginecapacity.Text);
            cmd.Parameters.AddWithValue("@price", txtprice.Text);
            cmd.Parameters.AddWithValue("@qty", txtqty.Text);

```

```

        int rowsAffected = cmd.ExecuteNonQuery();

```

```

        if (rowsAffected > 0)
        {
            gridLoad(); // Reload the data into your DataGridView
            MessageBox.Show("Vehicle Detail Successfully Updated.", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

            cbxbrand.Text = "";
            cbxvehicletype.Text = "";
            cbxcondition.Text = "";
            txtyear.Text = "";
            cbxtransmission.Text = "";
            cbxbranch.Text = "";
            cbxfueltype.Text = "";
            cbxbodytype.Text = "";
            txtvehicleregno.Text = "";

```

```

        txtchassisno.Text = "";
        txtenginecapacity.Text = "";
        txtprice.Text = "";
        txtqty.Text = "";
    }
    else
    {
        MessageBox.Show("Failed to Update Vehicle Details.", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}
}
}
}
catch (Exception ex)
{
    MessageBox.Show("An error occurred: " + ex.Message, "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

```

private void lblvehicleregno_Click(object sender, EventArgs e)
{

}

```

```

private void lblchassisno_Click(object sender, EventArgs e)
{

}

```

```

private void lblenginecapacity_Click(object sender, EventArgs e)
{

}

```

```

private void lblprice_Click(object sender, EventArgs e)

```

```

{

}

private void txtvehicleregno_TextChanged(object sender, EventArgs e)
{

}

private void txtchassisno_TextChanged(object sender, EventArgs e)
{

}

private void txtenginecapacity_TextChanged(object sender, EventArgs e)
{

}

private void txtprice_TextChanged(object sender, EventArgs e)
{

}

private void lbladdoreditvehicle_Click(object sender, EventArgs e)
{

}

private void cbxcondition_SelectedIndexChanged(object sender, EventArgs e)
{

}

private void cbxbrand_SelectedIndexChanged(object sender, EventArgs e)
{

}

private void cbxbranch_SelectedIndexChanged(object sender, EventArgs e)
{

```

```

    }

    private void cbxbodytype_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void cbxtransmission_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void cbxfueltype_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void cbxvehicletype_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void lblfueltype_Click(object sender, EventArgs e)
    {

    }

    private void lblcondition_Click(object sender, EventArgs e)
    {

    }

    private void lblbranch_Click(object sender, EventArgs e)
    {

    }

    private void lblbodytype_Click(object sender, EventArgs e)
    {

```

```

    }

    private void lbltransmission_Click(object sender, EventArgs e)
    {

    }

    private void lblmanufactureyear_Click(object sender, EventArgs e)
    {

    }

    private void lblvehicletype_Click(object sender, EventArgs e)
    {

    }

    private void lblbrand_Click(object sender, EventArgs e)
    {

    }

    private void txtyear_TextChanged(object sender, EventArgs e)
    {

    }

    private void lblqty_Click(object sender, EventArgs e)
    {

    }

    private void txtqty_TextChanged(object sender, EventArgs e)
    {

    }

    private void dataGridView7_CellContentClick(object sender, DataGridViewCellEventArgs
e)
    {

```

```

    }

    private void inventoryBindingSource_CurrentChanged(object sender, EventArgs e)
    {

    }

    private string clicked_brand;
    private void dataGridView7_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
        SqlConnection con = new SqlConnection(ConnectionString);

        if (e.RowIndex >= 0)
        {

            clicked_brand = dataGridView7.Rows[e.RowIndex].Cells[0].Value.ToString();
            string query = "SELECT * FROM Inventory WHERE brand = @brand";

            using (SqlCommand cmd = new SqlCommand(query, con))
            {
                con.Open();
                cmd.Parameters.AddWithValue("@brand", clicked_brand);
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataSet ds = new DataSet();
                da.Fill(ds);

                if (ds.Tables[0].Rows.Count > 0)
                {
                    cbxbrand.Text = ds.Tables[0].Rows[0][0].ToString();
                    cbxvehicletype.Text = ds.Tables[0].Rows[0][1].ToString();
                    cbxcondition.Text = ds.Tables[0].Rows[0][2].ToString();
                    txtyear.Text = ds.Tables[0].Rows[0][3].ToString();
                    cbxtransmission.Text = ds.Tables[0].Rows[0][4].ToString();
                    cbxbranch.Text = ds.Tables[0].Rows[0][5].ToString();
                    cbxfueltype.Text = ds.Tables[0].Rows[0][6].ToString();
                    cbxbodytype.Text = ds.Tables[0].Rows[0][7].ToString();
                    txtvehicleregno.Text = ds.Tables[0].Rows[0][8].ToString();
                    txtchassisno.Text = ds.Tables[0].Rows[0][9].ToString();
                    txtenginecapacity.Text = ds.Tables[0].Rows[0][10].ToString();
                }
            }
        }
    }

```

```

        txtprice.Text = ds.Tables[0].Rows[0][11].ToString();
        txtqty.Text = ds.Tables[0].Rows[0][12].ToString();
    }
    con.Close();
}
}

private void gridLoad()
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);
    SqlCommand cmd = new SqlCommand("SELECT * FROM Inventory ", con);
    SqlDataAdapter da1 = new SqlDataAdapter(cmd);
    DataSet ds1 = new DataSet();
    da1.Fill(ds1);

    dataGridView7.DataSource = ds1.Tables[0];

}
}
}

```

6.8 Add or Edit Employee

Functionality: Employee Management

Description: Supports the management of employee profiles and roles within the dealership. Users can create, edit, and maintain employee details, assign roles, and track historical employee data.

Add or Edit Employee

ADD OR EDIT EMPLOYEE

Personal Details

Employee No :

E001

Title :

Mr.

First Name :

Ravindu

Last Name :

Haputhanthri

Gender :

Male

Date Of Birth :

Monday , December 6, 1999

Address :

52/3 Katubedda Moratuwa

NIC/Passport No :

993411205v

Land-Phone No :


0116109235

Mobile No :

0775165666

Email :

ravinduheshan99@gmail.com



Browse Image

Job Position :

Sales Manager

Working Since :

Tuesday , January 1, 2013

Contract Expiration :

Tuesday , December 31, 2030

Branch :

Colombo

Salary :

450000

Add Employee

Update Employee

Search Employee

Delete Employee

Main Menu

Clear

Close


```
using COSC_31112_Visual_Programming_Final_Project_Group19.Properties;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.ListView;
using System.IO;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
```

```
namespace COSC_31112_Visual_Programming_Final_Project_Group19
```

```
{
    public partial class frm8 : Form
    {
        public frm8()
        {
            InitializeComponent();
        }

        private void Form8_Load(object sender, EventArgs e)
        {
            cbxtitle.Items.Add("Mr.");
            cbxtitle.Items.Add("Mrs.");
            cbxtitle.Items.Add("Miss.");
            cbxtitle.Items.Add("Dr.");
            cbxtitle.Items.Add("Prof.");
            cbxtitle.Items.Add("Rev.");

            cbxgender.Items.Add("Male");
            cbxgender.Items.Add("Female");

            cbxbranch.Items.Add("Colombo");
            cbxbranch.Items.Add("Kandy");
            cbxbranch.Items.Add("Galle");
            cbxbranch.Items.Add("Jaffna");
            cbxbranch.Items.Add("Ampara");
        }
    }
}
```

```

        cbxjobposition.Items.Add("Sales Manager");
        cbxjobposition.Items.Add("Sales Representative");
        cbxjobposition.Items.Add("Fleet Sales Manager");
        cbxjobposition.Items.Add("Finance and Insurance Manager");
        cbxjobposition.Items.Add("Customer Service Representative");
        cbxjobposition.Items.Add("Inventory Manager");
        cbxjobposition.Items.Add("Service Advisor");
        cbxjobposition.Items.Add("Mechanic/Technician");
        cbxjobposition.Items.Add("Detailer");
        cbxjobposition.Items.Add("Lot Attendant");
        cbxjobposition.Items.Add("Marketing Manager");
        cbxjobposition.Items.Add("Human Resources Manager");
        cbxjobposition.Items.Add("Accountant/Finance Controller");
        cbxjobposition.Items.Add("Administrative Assistant");
        cbxjobposition.Items.Add("Security Personnel");
        cbxjobposition.Items.Add("Parts and Accessories Specialist");
        cbxjobposition.Items.Add("Delivery Driver");
        cbxjobposition.Items.Add("Online Sales Specialist");
        cbxjobposition.Items.Add("IT Support Specialist");
        cbxjobposition.Items.Add("Legal Counsel");
    }

    private void btnmainmenu8_Click(object sender, EventArgs e)
    {
        frm2 obj = new frm2();
        obj.Show();
        this.Hide();
    }

    private void btnbrowseimage_Click(object sender, EventArgs e)
    {
        using (OpenFileDialog ofd = new OpenFileDialog() { Filter = "JPEG|*.jpg",
        ValidateNames = true, Multiselect = false })
        {
            if (ofd.ShowDialog() == DialogResult.OK)
            {
                string filename = ofd.FileName;
                pictureBox8.Image = Image.FromFile(filename);
            }
        }
    }

```

```

    }

    private void btnaddemployee_Click(object sender, EventArgs e)
    {
        string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
        SqlConnection con = new SqlConnection(ConnectionString);
        SqlCommand com;

        if (txtemployeeeno.Text.Length < 3)
        {
            MessageBox.Show("Wrong Length", "Length Check", MessageBoxButtons.OKCancel,
            MessageBoxIcon.Information);
            txtemployeeeno.Clear();
            txtemployeeeno.Focus();
        }
        else
        {
            try
            {
                con.Open();
                string sql = "insert into
Employee(enumerator,title,fname,lname,gender,dob,adrs,nic,lphone,hphone,email,image,jposition,
wstart,wend,branch,salary) values('" + txtemployeeeno.Text + "','" + cbxtitle.Text + "','" +
txtfn.Text + "','" + txtln.Text + "','" + cbxgender.Text + "','" + dtpdob.Text + "','" + txtaddress.Text
+ "','" + txtnic.Text + "','" + txtlp.Text + "','" + txthp.Text + "','" + txtemail.Text + "','" +
ConvertImageToBinary(pictureBox8.Image) + "','" + cbxjobposition.Text + "','" +
dtpworkingsince.Text + "','" + dtpcontractexpiration.Text + "','" + cbxbranch.Text + "','"
+ txtsalary.Text + "')";
                com = new SqlCommand(sql, con);
                com.ExecuteNonQuery();
                MessageBox.Show("Employee Added Successfully", "Message",
                MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

                txtemployeeeno.Text = "";
                cbxtitle.Text = "";
                txtfn.Text = "";
                txtln.Text = "";
                cbxgender.Text = "";
                dtpdob.Text = "";
            }
            catch { }
        }
    }
}

```

```

        txtaddress.Text = "";
        txtnic.Text = "";
        txtlp.Text = "";
        txtthp.Text = "";
        txtemail.Text = "";
        pictureBox8.Text = "";
        cbxjobposition.Text = "";
        dtpworkingsince.Text = "";
        dtpcontractexpiration.Text = "";
        cbxbranch.Text = "";
        txtsalary.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
    }
}

private void btnsearchemployee_Click(object sender, EventArgs e)
{
    try
    {
        using (SqlConnection connection = new SqlConnection("Data Source=DESKTOP-
9HG1TI6\\SQLEXPRESS;Initial Catalog=Group19;Integrated Security=True"))
        {
            connection.Open();
            using (SqlCommand command = new SqlCommand("SELECT
title,fname,lname,gender,dob,adrs,nic,lphone,hphone,email,jposition,wstart,wend,branch,salary
FROM Employee where enumber='" + txtemployeeeno.Text + "'", connection))
            {
                using (SqlDataAdapter adapter = new SqlDataAdapter(command))
                {
                    DataTable dataTable = new DataTable();
                    adapter.Fill(dataTable);
                    using (SqlDataReader reader = command.ExecuteReader())
                    {

```

```

        if (reader.Read())
        {
            cbxtitle.Text = reader["title"].ToString();
            txtfn.Text = reader["fname"].ToString();
            txtln.Text = reader["lname"].ToString();
            cbxgender.Text = reader["gender"].ToString();
            dtpdob.Text = reader["dob"].ToString();
            txtaddress.Text = reader["adrs"].ToString();
            txtnic.Text = reader["nic"].ToString();
            txtlp.Text = reader["lphone"].ToString();
            txtthp.Text = reader["hphone"].ToString();
            txtemail.Text = reader["email"].ToString();
            cbxjobposition.Text = reader["jposition"].ToString();
            dtpworkingsince.Text = reader["wstart"].ToString();
            dtpcontractexpiration.Text = reader["wend"].ToString();
            cbxbranch.Text = reader["branch"].ToString();
            txtsalary.Text = reader["salary"].ToString();
        }
        else
        {
            MessageBox.Show("Employee Does Not Exsist");
        }
    }
}

}

using (SqlCommand command1 = new SqlCommand("SELECT image FROM
Employee where enumber='" + txtemployeeno.Text + "'", connection))
{
    using (SqlDataAdapter adapter = new SqlDataAdapter(command1))
    {
        DataTable dataTable = new DataTable();
        adapter.Fill(dataTable);
        using (SqlDataReader reader = command1.ExecuteReader())
        {
            if (reader.Read())
            {
                Byte[] photo = (byte[])reader["image"];
                pictureBox8.Image = ConvertBinaryToImage(photo);
            }
        }
    }
}

```

```

        }
    }
    connection.Close();
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

```

Image ConvertBinaryToImage(byte[] data)

```

{
    try
    {
        using (MemoryStream ms = new MemoryStream(data))
        {
            return Image.FromStream(ms);
        }
    }
    catch (ArgumentException ex)
    {
        Console.WriteLine($"Error converting binary to image: {ex.Message}");
        // Handle the ArgumentException appropriately in your application
        return null;
    }
    catch (OutOfMemoryException ex)
    {
        Console.WriteLine($"Error converting binary to image: {ex.Message}");
        // Handle the OutOfMemoryException appropriately in your application
        return null;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error converting binary to image: {ex.Message}");
        // Handle other exceptions appropriately in your application
        return null;
    }
}

```

```

byte[] ConvertImageToBinary(Image img)
{
    try
    {
        using (MemoryStream ms = new MemoryStream())
        {
            // Save the image to the MemoryStream
            img.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);

            // Ensure all data is written to the MemoryStream and then get the byte array
            ms.Flush();

            return ms.ToArray();
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error converting image to binary: {ex.Message}");
        return null; // Handle the error appropriately in your application
    }
}

```

```

private void btnupdateemployee_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);
    SqlCommand com;

    try
    {
        con.Open();
        string sql = "Update Employee set title='" + cbxtitle.Text + "',fname='" + txtfn.Text +
        "',lname='" + txtln.Text + "',gender='" + cbxgender.Text + "',dob='" + dtpdob.Text + "',adrs='" +
        txtaddress.Text + "',nic='" + txtnic.Text + "',lphone='" + txtlp.Text + "',hphone='" + txthp.Text +
        "',email='" + txtemail.Text + "',image='" + pictureBox8.Text + "',jposition='" +
        cbxjobposition.Text + "',wstart='" + dtpworkingsince.Text + "',wend='" +
        dtpcontractexpiration.Text + "',branch='" + cbxbranch.Text + "',salary='" + txtsalary.Text + "' where
        enumber='" + txtemployeeno.Text + "'";
    }
}

```

```

        com = new SqlCommand(sql, con);
        com.ExecuteNonQuery();

        MessageBox.Show("Employee Detail Updated Successfully.", "Message",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

        txtemployeeno.Text = "";
        cbxtitle.Text = "";
        txtfn.Text = "";
        txtln.Text = "";
        dtpdob.Text = "";
        txtaddress.Text = "";
        txtnic.Text = "";
        txtlp.Text = "";
        txthp.Text = "";
        txtemail.Text = "";
        pictureBox8.Text = "";
        cbxjobposition.Text = "";
        dtpworkingsince.Text = "";
        dtpcontractexpiration.Text = "";
        cbxbranch.Text = "";
        txtsalary.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
    }
}

private void btndeleteemployee_Click(object sender, EventArgs e)
{
    string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
    SqlConnection con = new SqlConnection(ConnectionString);
    SqlCommand com;

    try

```



```

{
    con.Open();
    string sql = "Delete from Employee where enumber = " + txtemployeeno.Text + "";
    com = new SqlCommand(sql, con);
    com.ExecuteNonQuery();
    MessageBox.Show("Employee Deleted Successfully", "Message",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

```

```

    txtemployeeno.Text = "";
    cbxtitle.Text = "";
    txtfn.Text = "";
    txtln.Text = "";
    dtpdob.Text = "";
    txtaddress.Text = "";
    txtnic.Text = "";
    txtlp.Text = "";
    txthp.Text = "";
    txtemail.Text = "";
    pictureBox8.Text = "";
    cbxjobposition.Text = "";
    dtpworkingsince.Text = "";
    dtpcontractexpiration.Text = "";
    cbxbranch.Text = "";
    txtsalary.Text = "";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    con.Close();
}
}

private void btnclear8_Click(object sender, EventArgs e)
{
    txtemployeeno.Text = "";
    cbxtitle.Text = "";
    txtfn.Text = "";
    txtln.Text = "";

```

```
        dtpdob.Text = "";
        txtaddress.Text = "";
        txtnic.Text = "";
        txtlp.Text = "";
        txtthp.Text = "";
        txtemail.Text = "";
        pictureBox8.Text = "";
        cbxjobposition.Text = "";
        dtpworkingsince.Text = "";
        dtpcontractexpiration.Text = "";
        cbxbranch.Text = "";
        txtsalary.Text = "";
    }

    private void btnclose8_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
```

6.9 Employee's Salary Calculator

Functionality: Salary Calculation and Payroll

Description: Offers tools for calculating employee salaries based on various components, such as base salary, bonuses, deductions, and benefits. It generates detailed payroll reports for record-keeping and payment processing.

EMPLOYEE SALARY CALCULATOR

Employee No : E001
First Name : Ravindu
Last Name : Haputhanthri
Job Position : Sales Manager
Branch : Colombo
Basic Salary(Rs) : 450000

Extra Hours Worked : 45
Extra Earnings : 5625
Gross Salary(Rs) : 455625
ETF : 18225
Net Salary(Rs) : 437400

Special Note :
125 Rs per Extra Hour
ETF Percentage : 4%

Net Salary

Main Menu Clear Close

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace COSC_31112_Visual_Programming_Final_Project_Group19
{
    public partial class frm9 : Form
    {
        public frm9()
        {
            InitializeComponent();
        }

        private void btnmainmenu9_Click(object sender, EventArgs e)
        {
            frm2 obj = new frm2();
            obj.Show();
            this.Hide();
        }

        private void btnload_Click(object sender, EventArgs e)
        {
            string ConnectionString = "Data Source=DESKTOP-9HG1TI6\\SQLEXPRESS;Initial
Catalog=Group19;Integrated Security=True";
            SqlConnection con = new SqlConnection(ConnectionString);
            SqlCommand com;

            if (txtemployeeno.Text.Length < 3)
            {
                MessageBox.Show("Wrong Length", "Length Check", MessageBoxButtons.OKCancel,
                MessageBoxIcon.Information);
                txtemployeeno.Clear();
                txtemployeeno.Focus();
            }
            else
            {
                try
                {
                    con.Open();

```

```

        string sql = "select fname,lname,jposition,branch,salary,image from Employee where
enumber = '" + txtemployeeeno.Text + "'";
        com = new SqlCommand(sql, con);
        com.Parameters.AddWithValue("enumber", txtemployeeeno.Text);
        SqlDataReader dr;
        dr = com.ExecuteReader();

        if (dr.Read())
        {

            txtfn.Text = dr["fname"].ToString();
            txtln.Text = dr["lname"].ToString();
            txtjposition.Text = dr["jposition"].ToString();
            txtbranch.Text = dr["branch"].ToString();
            txtbasicsalary.Text = dr["salary"].ToString();

        }
        else
        {
            MessageBox.Show("Employee Not Found.", "Message",
MessageBoxButtons.OKCancel, MessageBoxIcon.Exclamation);
            txtemployeeeno.Clear();
            txtemployeeeno.Focus();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
    }
}

private void btnnetsalary_Click(object sender, EventArgs e)
{
    double sal, extra, tot, etf, netsal;
    int hours;

    hours = int.Parse(txtextrahoursworked.Text);
    extra = (hours * 125);
    txtextraearnings.Text = extra.ToString();
    sal = double.Parse(txtbasicsalary.Text);
    tot = extra + sal;

```

```

        txtgrosssalary.Text = tot.ToString();
        etf = tot * 4 / 100;
        txtetf.Text = etf.ToString();
        netsal = tot - etf;
        txtnetsalary.Text = netsal.ToString();
    }

    private void btnclear9_Click(object sender, EventArgs e)
    {
        txtemployeeno.Text = "";
        txtfn.Text = "";
        txtln.Text = "";
        txtjobposition.Text = "";
        txtbranch.Text = "";
        txtbasicsalary.Text = "";
        txtextrahoursworked.Text = "";
        txtextraearnings.Text = "";
        txtgrosssalary.Text = "";
        txtetf.Text = "";
        txtnetsalary.Text = "";
    }

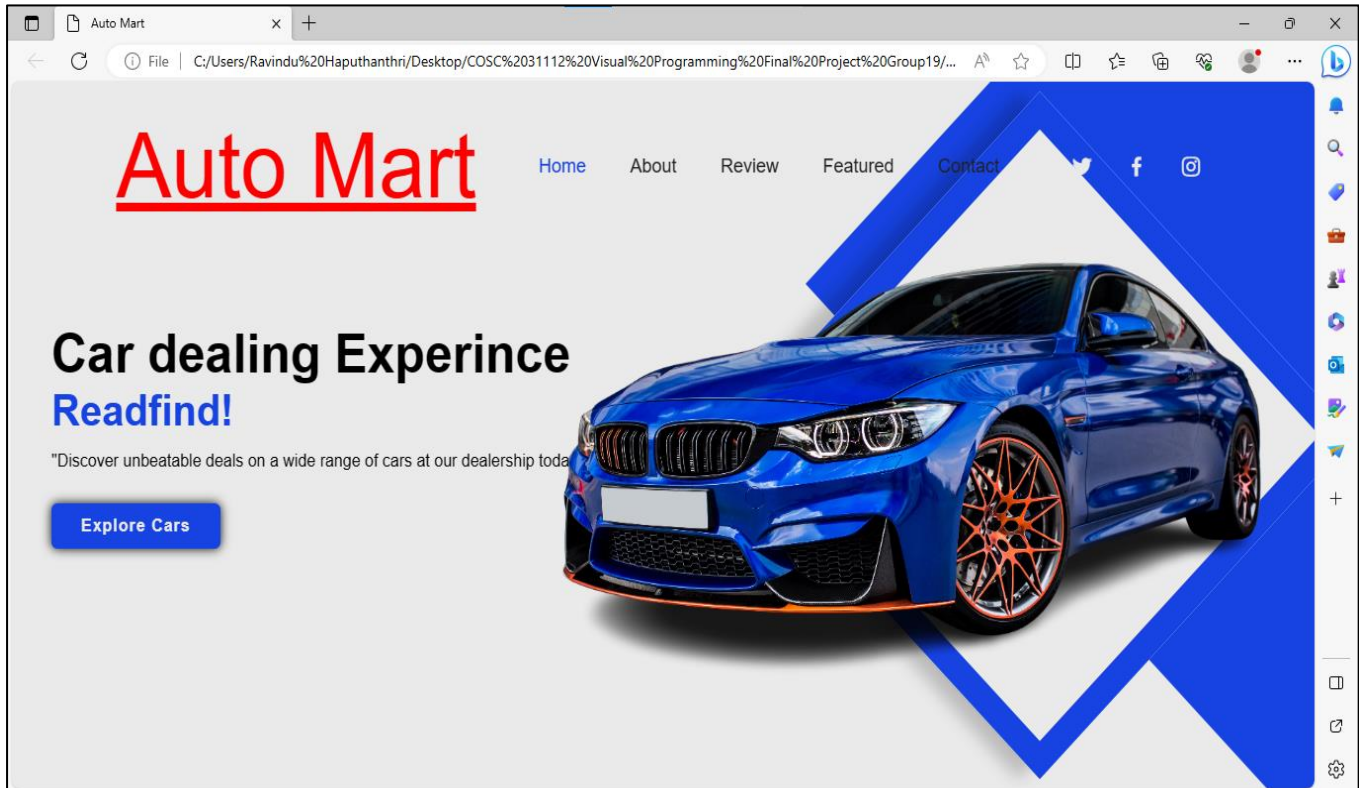
    private void btnclose9_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

6.10 About Us Web Page

Functionality: Informational

Description: Displays information about your dealership, including its history, mission, values, and other relevant details. It serves as an informational resource for customers and visitors to learn more about your dealership.



6.10.1 car.html file

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Auto Mart</title>
  <link rel="stylesheet" href="stylesheet2.css">
  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
</head>
<body>
  <header class="header">
    <a href="#" class="logo"><big><u>Auto Mart</u></big></a>
    <nav class="navbar">
      <a href="#" style="--i:1" class="active">Home</a>
      <a href="#" style="--i:2">About</a>
      <a href="#" style="--i:3">Review</a>
      <a href="#" style="--i:4">Featured</a>
      <a href="#" style="--i:5">Contact</a>
    </nav>

    <div class="social-media">
      <a href="#" style="--i:1"><i class='bx bxl-twitter'></i></a>
      <a href="#" style="--i:2"><i class='bx bxl-facebook'></i></a>
      <a href="#" style="--i:3"><i class='bx bxl-instagram' ></i></a>
    </div>
  </header>
  <section class="home">
    <div class="home-content">
      <h1>Car dealing Experince</h1>
      <h3>Readfind!</h3>
      <p>"Discover unbeatable deals on a wide range of cars at our
dealership today!"</p>
      <a href="#" class="btn">Explore Cars</a>
    </div>
    <div class="home-img">
      <div class="rhombus">
        
      </div>
    </div>
    <div class="rhombus2"></div>
  </section>
</body>
</html>
```


6.10.2 stylesheet2.css file

```
@import url('https://fonts.googleapis.com/css2? family=Poppins: wght@300;400;
500; 600; 700; 800; 900& display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins',sans-serif;
}
body{
    background: #eaeaea;
}
.header{
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    padding: 30px 8%;
    background: transparent;
    display: flex;
    justify-content: space-between;
    align-items: center;
    z-index: 100;
}

.logo{
    font-size: 70px;
    color: red;
    text-decoration: none;
    text-align:left
    font-weight: 100;
    opacity: 0;
    animation: slideRight 1s ease forwards;
}

.navbar a{
    display: inline-block;
    font-size: 18px;
    color: #222;
    text-decoration: none;
    font-weight: 500;
    margin: 0 20px;
    transition: .3s;
    opacity: 0;
    animation: slideTop .5s ease forwards;
    animation-delay: calc(.2s * var(--i));
```

```

}
.navbar a:hover,
.navbar a.active{
    color: #1743e3;
}
.social-media{
    display: flex;
    justify-content: space-between;
    width: 150px;
    height: 40px;
}
.social-media a {
    display: inline-flex;
    justify-content: center;
    align-items: center;
    width: 40px;
    height: 40px;
    background : transparent;
    border: 2px solid transparent;
    text-decoration: none;
    transform: rotate(45deg);
    transition: .5s;
    opacity: 0;
    animation: slideSci .5s ease forwards;
    animation-delay: calc(.2s * var(--i));
}

.social-media a:hover{
    border-color: #eaeaea;
}

.social-media a i {
    font-size: 24px;
    color: #eaeaea;
    transform: rotate(-45deg);
}
.home{
    position: relative;
    width: 100%;
    height: 100vh;
    justify-content: space-between;
    display: flex;
    align-items: center;

```

```

        overflow: hidden;
    }
    .home-content h1{
        font-size: 50px;
        line-height: 1.2;
        opacity: 0;
        animation: slideBottem 1s ease forwards ;
        animation-delay: 1s;
        margin-left: 40px;
    }
    .home-content h3{
        font-size: 40px;
        color: #1743e3;
        opacity: 0;
        animation: slideRight 1.3s ease forwards ;
        animation-delay: 1.6s;
        margin-left: 40px;
    }
    .home-content p{
        font-size: 16px;
        margin: 15px 0 30px;
        opacity: 0;
        animation: slideLeft 1s ease forwards ;
        animation-delay: 1s;
        margin-left: 40px;
    }
    .btn{
        display: inline-block;
        padding: 10px 28px;
        background : #1743e3;
        border: 2px solid #1743e3;
        border-radius: 6px;
        box-shadow: 0 0 10px rgb(0, 0, 0.1);
        font-size: 16px;
        color: #eaeaea;
        letter-spacing: 1px;
        text-decoration: none;
        font-weight: 600;
        transition: .5s;
        opacity: 0;
        animation: slideTop 1s ease forwards;
        transition-delay: 2s;
        margin-left: 40px;
    }

```

```

.btn:hover{
    background: transparent;
    color: #1743e3;
}
.home-img{
    position: relative;
    right: 4%;
    width: 450px;
    height: 450px;
    transform: rotate(45deg);
}

.home-img .rhombus{
    position: absolute;
    width: 100%;
    height: 100%;
    background: #eaeaea;
    border: 25px solid #1743e3;
    box-shadow: -15px 15px 15px rgba(0, 0, 0, .2);
    opacity: 0;
    animation: zoomOut 1s ease forwards;
    animation-delay: 1.6s;
}
.home-img .rhombus img{
    position: absolute;
    top: 110px;
    left: -250px;
    max-width: 750px;
    transform: rotate(-45deg);
}
.home .rhombus2 {
    position: absolute;
    top: -25%;
    right: -25%;
    width: 700px;
    height: 700px;
    background: #1743e3;
    transform: rotate(-45deg);
    z-index: -1;
}
@keyframes slideRight {
    0%{
        transform: translateX(-100px);
        opacity: 0;
    }
}

```

```

        100%{
            transform: translateX(0);
            opacity: 1;
        }
    }

    @keyframes slideLeft {
        0%{
            transform: translateX(100px);
            opacity: 0;
        }
        100%{
            transform: translateX(0);
            opacity: 1;
        }
    }

    @keyframes slideTop {
        0%{
            transform: translateY(100px);
            opacity: 0;
        }
        100%{
            transform: translateY(0);
            opacity: 1;
        }
    }

    @keyframes slideBottem {
        0%{
            transform: translateY(-100px);
            opacity: 0;
        }
        100%{
            transform: translateY(0);
            opacity: 1;
        }
    }

    @keyframes slideSci {
        0%{
            transform: translateX(100px) rotate(45deg);
            opacity: 0;
        }
        100%{

```

```
        transform: translateY(0) rotate(45deg);
        opacity: 1;
    }
}

@keyframes zoomOut {
    0%{
        transform: scale(1.1);
        opacity: 0;
    }
    100%{
        transform: scale(1);
        opacity: 1;
    }
}
```

7. DATA TABLE STRUCTURE

7.1 Inventory

DESKTOP-9HG1TI6\S...19 - dbo.Inventory - X DESKTOP-9HG1TI6\S...19 - d			
	Column Name	Data Type	Allow Nulls
►	brand	varchar(50)	<input type="checkbox"/>
	vtype	varchar(50)	<input type="checkbox"/>
	condition	varchar(50)	<input type="checkbox"/>
	year	int	<input type="checkbox"/>
	transmission	varchar(50)	<input type="checkbox"/>
	branch	varchar(50)	<input type="checkbox"/>
	ftype	varchar(50)	<input type="checkbox"/>
	btype	varchar(50)	<input type="checkbox"/>
	vregno	nvarchar(50)	<input type="checkbox"/>
	chassisno	nvarchar(50)	<input type="checkbox"/>
	ecapacity	nvarchar(50)	<input type="checkbox"/>
	price	nvarchar(50)	<input type="checkbox"/>
	qty	int	<input type="checkbox"/>
			<input type="checkbox"/>

7.2 Employee

DESKTOP-9HG1TI6\...19 - dbo.Employee			
	Column Name	Data Type	Allow Nulls
▶	enumber	varchar(50)	<input type="checkbox"/>
	title	varchar(50)	<input type="checkbox"/>
	fname	varchar(50)	<input type="checkbox"/>
	lname	varchar(50)	<input type="checkbox"/>
	gender	varchar(50)	<input type="checkbox"/>
	dob	nvarchar(50)	<input type="checkbox"/>
	adrs	nvarchar(50)	<input type="checkbox"/>
	nic	nvarchar(50)	<input type="checkbox"/>
	lphone	nvarchar(50)	<input type="checkbox"/>
	hphone	nvarchar(50)	<input type="checkbox"/>
	email	nvarchar(50)	<input type="checkbox"/>
	image	image	<input type="checkbox"/>
	jposition	varchar(50)	<input type="checkbox"/>
	wstart	nvarchar(50)	<input type="checkbox"/>
	wend	nvarchar(50)	<input type="checkbox"/>
	branch	varchar(50)	<input type="checkbox"/>
	salary	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

7.3 Customer

DESKTOP-9HG1TI6\...19 - dbo.Customer			
	Column Name	Data Type	Allow Nulls
▶	cusno	nvarchar(50)	<input type="checkbox"/>
	title	varchar(50)	<input type="checkbox"/>
	fname	varchar(50)	<input type="checkbox"/>
	lname	varchar(50)	<input type="checkbox"/>
	dob	nvarchar(50)	<input type="checkbox"/>
	adrs	nvarchar(MAX)	<input type="checkbox"/>
	nic	nvarchar(50)	<input type="checkbox"/>
	lphone	nvarchar(50)	<input type="checkbox"/>
	hphone	nvarchar(50)	<input type="checkbox"/>
	email	nvarchar(50)	<input type="checkbox"/>
	make	varchar(50)	<input type="checkbox"/>
	model	nvarchar(50)	<input type="checkbox"/>
	chassisno	nvarchar(50)	<input type="checkbox"/>
	status	varchar(50)	<input type="checkbox"/>
	regno	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

7.4 Appointment

DESKTOP-9HG1TI6\...- dbo.Appointment			
	Column Name	Data Type	Allow Nulls
▶	id	nvarchar(50)	<input type="checkbox"/>
	branch	varchar(50)	<input type="checkbox"/>
	date	nvarchar(50)	<input type="checkbox"/>
	time	time(7)	<input type="checkbox"/>
	description	varchar(50)	<input type="checkbox"/>
	contactno	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

8. TOOLS

- ❖ **Programming Language:** The primary programming language used for this project is C#, which is commonly used for Windows Forms applications and integrates with various libraries and frameworks.
- ❖ **Development Environment:** The project is likely developed using Visual Studio, a popular integrated development environment (IDE) for C# and .NET development. Visual Studio provides a user-friendly interface for building Windows Forms applications and offers debugging and design tools.
- ❖ **SQL Server Database:** The project connects to a SQL Server database using System.Data.SqlClient. This is evident in the code where SQL Server connections and commands are used for data storage and retrieval.
- ❖ **Windows Forms:** The user interface (UI) for the project is built using Windows Forms. Windows Forms is a graphical user interface framework provided by Microsoft for developing desktop applications in C#.
- ❖ **Database:** The project appears to interact with a database to store information related to employees, customers, items, and sales. The exact database management system (DBMS) is SQL Server.

9. WORK CONTRIBUTION

9.1 Individual Work Contribution

Student No.	Name	Work Contribution
PS/2019/171	N.D.R.H.Haputhanthri	Vehicle Management, Employee Management, Project Documentation
PS/2019/164	J.A.K.N.JAYAKODY	About Us Web Page, View Inventory
PS/2019/075	S.N.Y.A.GUNASEKARA	Login, Manage Appointments
PS/2019/072	E.S.N.DILKI	Admin Dashboard, Customer Management
PS/2017/012	G.B.M.W.G.S.R.BANDARA	Staff Dashboard, Employee Salary Calculator

9.2 DETAILS OF GROUP MEMBERS

Student No.	Name	Contact Info
PS/2019/171	N.D.R.H.HAPUTHANTHRI	0775165666
PS/2019/164	J.A.K.N.JAYAKODY	0768016139
PS/2019/075	S.N.Y.A.GUNASEKARA	0768120305
PS/2019/072	E.S.N.DILKI	0716449484
PS/2017/012	G.B.M.W.G.S.R.BANDARA	0773649686

9.3 Challenges

- ❖ **Data Complexity and Volume:** Vehicle management systems deal with extensive data, including vehicle specifications, sales records, customer information, and maintenance history. Managing and processing this data efficiently can be a challenge.
- ❖ **Integration with External Systems:** Integration with external systems like DMVs, financial institutions for payment processing, and third-party data providers for vehicle information can be complex and require well-defined APIs and data formats.
- ❖ **User Authentication and Security:** Ensuring the security of user data, including customer and employee information, is critical. Implementing robust authentication and authorization mechanisms is essential.
- ❖ **User Interface Design:** Creating a user-friendly and intuitive interface for various user roles (administrators, sales staff, customers) can be challenging. Ensuring that the system is accessible and easy to use is important for user adoption.
- ❖ **Scalability:** As the system grows and more data is added, scalability becomes crucial. The system should be able to handle an increasing number of users and records without significant performance degradation.
- ❖ **Real-Time Updates:** Keeping inventory, sales, and customer information up to date in real-time can be challenging, especially when multiple users are accessing and modifying data simultaneously.
- ❖ **Complex Business Logic:** Vehicle sales involve various complex business rules, such as pricing calculations, tax calculations, and financing options. Implementing and maintaining these rules accurately is vital.
- ❖ **Data Quality and Accuracy:** Maintaining accurate and high-quality data is essential for vehicle listings and customer records. Implementing data validation and verification processes is necessary.
- ❖ **Testing and Quality Assurance:** Thorough testing, including unit testing, integration testing, and user acceptance testing, is essential to identify and fix bugs and ensure the system functions as expected.
- ❖ **Change Management:** Implementing the VSMS may require changes in dealership processes and workflows. Managing change within the organization and ensuring that users adapt to the new system can be challenging.
- ❖ **Data Migration:** If migrating from an existing system, data migration can be complex. Ensuring that historical data is accurately transferred to the new system is crucial.
- ❖ **User Training and Support:** Providing adequate training and ongoing support for users, especially for new features and updates, is essential for user satisfaction and system adoption.
- ❖ **Performance Optimization:** Continuously monitoring and optimizing system performance to handle increased loads and maintain responsiveness is an ongoing challenge.

10. FUTURE ENHANCEMENT OF THE PROJECT

- ❖ **AI-Driven Customer Engagement:** Description: Implement advanced AI chatbots and virtual assistants to engage with customers in real-time. These AI-powered agents can answer inquiries, assist in scheduling appointments, and provide personalized vehicle recommendations based on customer preferences and browsing history.
- ❖ **Enhanced CRM Integration:** Description: Further integrate customer relationship management (CRM) functionality to track and manage customer interactions comprehensively. Utilize AI algorithms to identify sales opportunities and optimize customer engagement strategies.
- ❖ **IoT Integration for Vehicle Telemetry:** Description: Connect vehicles to the Internet of Things (IoT) to gather real-time telemetry data. Monitor vehicle health, fuel efficiency, and usage patterns. Provide customers with detailed vehicle performance reports and predictive maintenance alerts.
- ❖ **Advanced Inventory Management:** Description: Enhance the inventory management module with automated price adjustments based on market trends, demand, and vehicle condition. Implement predictive analytics to optimize inventory stocking levels and reduce overstock or shortages.
- ❖ **AR/VR Showroom Experience:** Description: Create an immersive augmented reality (AR) or virtual reality (VR) showroom experience. Allow customers to virtually explore vehicles, customize features, and take simulated test drives, providing a unique and interactive sales experience.

- ❖ **Blockchain-Based Vehicle History:** Description: Implement blockchain technology to create an immutable and transparent ledger of each vehicle's history. This includes service records, accident reports, ownership transfers, and maintenance details, boosting transparency and trust for buyers.
- ❖ **Enhanced Mobile Accessibility:** Description: Optimize the VSMS for mobile devices with responsive design and a dedicated mobile app. Allow users to perform all critical functions, from browsing inventory to scheduling appointments, seamlessly on smartphones and tablets.
- ❖ **Advanced Reporting and Predictive Analytics:** Description: Strengthen the reporting and analytics capabilities with machine learning algorithms for predictive analytics. Provide actionable insights into sales trends, customer behavior, and inventory management, enabling data-driven decisions.
- ❖ **Voice-Activated Features:** Description: Enable voice-activated commands within the VSMS. Customers and staff can interact with the system using voice assistants, simplifying tasks like searching for vehicles, scheduling appointments, and obtaining information.

11. REFERENCES

Web Sites:

- ❖ <http://stackoverflow.com/>
- ❖ <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>
- ❖ <http://www.codeproject.com/>
- ❖ <https://www.youtube.com/user/ProgrammingKnowledge>
- ❖ <https://www.youtube.com/user/Nemboolisoft/featured>
- ❖ <https://www.youtube.com/channel/UCEwi4t2RiptCROIt dg9W1vA>