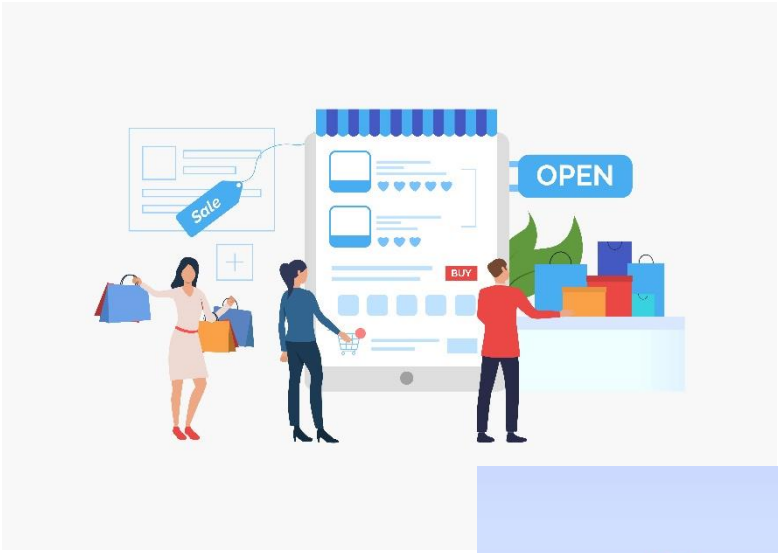
9/30/2023

Super Market Management System



Group 08
UNIVERSITY OF KELANIYA
DEPARTMENT OF STATISCTICS AND COMPUTER SCIENCE

Table of Contents

Introduction

We have developed a Supermarket Management System for Mini Good Super. By automating various processes in the supermarket and integrating essential features to streamline the operations, optimize inventory management, Report Generation and provide a seamless shopping experience for Employees and Customers.

In our management system there are 5 employee types as Cashiers, Manager, Storekeeper, Sales Team, HR Team. And each of them has access to different processes in the system.

Cashier - The cashier can handle sales of the customer through the POS system and manage customers in the database.

Manager - The Manager Has the access to the generate reports of the Total Sales and view Employee Details, Customer Details, Product details, Stock Details, Discount Details.

Storekeeper - The storekeeper is the person that manages the inventory on the supermarket. Add new stock to the supermarket and remove stock from the supermarket.

Sales Team - The sales team Manages Discounts in the supermarket.

HR Team - The HR team manages the employees in the supermarket, they add new employees to the system and remove employees from the system and they manage the employee salary.

Problem Identification

- Users, especially cashiers and storekeepers, need to quickly learn and adapt to the software.
- Data Security and Privacy: User authentication to check reduce the unauthorized logins.
- User Roles and Permissions: Only several roles can access several modules.
- Error Handling: Error Messages pop up.
- Inventory Management: Check if the current system efficiently manages product listings, stock levels, and updates in real-time.

Objectives

The primary objectives of implementing the Supermarket Management System are as follows:

1. Streamline Operations: Automate various supermarket processes, such as inventory management, sales transactions, and employee management, Report Generation, Customer Registration to reduce manual effort and increase efficiency.
2. Enhance Customer Experience: Improve customer satisfaction through quick and accurate service and loyalty programs.
3. Enhance Employee Experience: usability of the system.
4. Optimize Inventory Management: Ensure optimal stock levels, reduce the shortage of stock.
5. Improve Decision-Making: Generate insightful reports and analytics to facilitate informed decision-making and strategic planning.
6. Increase Sales and Revenue: Implement promotional offers, discounts, and loyalty programs to attract and retain customers, ultimately boosting sales.

Functional & non-functional requirements of the application

Functional requirements

1. **User Authentication and Authorization**:
    - Users must be able to log in to the system with valid credentials.
    - Different user roles (e.g., Cashier, HR, Storekeeper) must have different levels of access to the system.
2. **Employee Management**:
    - HR users must be able to add, edit, and delete employee details.
    - HR users must be able to search for employee details.
3. **Point of Sale (POS)**:
    - Cashier users must be able to add products to a shopping cart.
    - Cashier users must be able to clear the cart and finalize the bill.
    - The system should calculate the total bill, apply discounts if applicable, and update customer transaction totals.
4. **Product Management**:
    - Storekeeper users must be able to add, edit, and delete product details.
    - Storekeeper users must be able to search for product details.
5. **Database Interaction**:
    - The system must interact with a database to store and retrieve employee, product, and transaction data.

Non-functional requirements

1. **Security**:
    - User authentication and data access must be secure to prevent unauthorized access.
    - Personal and sensitive information (e.g., employee details, customer transactions) must be protected.
2. **Performance**:
    - The system should provide reasonable response times for user interactions (e.g., searching for products, processing sales transactions).
3. **Reliability**:
    - The system should be reliable and available during business hours.
    - Data integrity must be maintained to prevent data corruption or loss.
4. **Usability**:
    - The user interface should be intuitive and user-friendly to minimize user training requirements.
5. **Scalability**:
    - The system should be able to handle an increasing number of users, products, and transactions without a significant drop in performance.
6. **Data Backup and Recovery**:
    - Regular data backups should be performed to ensure data can be recovered in case of system failure or data loss.
7. **Compliance**:
    - The system should comply with relevant legal and regulatory requirements related to data privacy and security.
8. **Error Handling**:
    - The system should provide clear error messages and gracefully handle exceptions to prevent data corruption or loss.

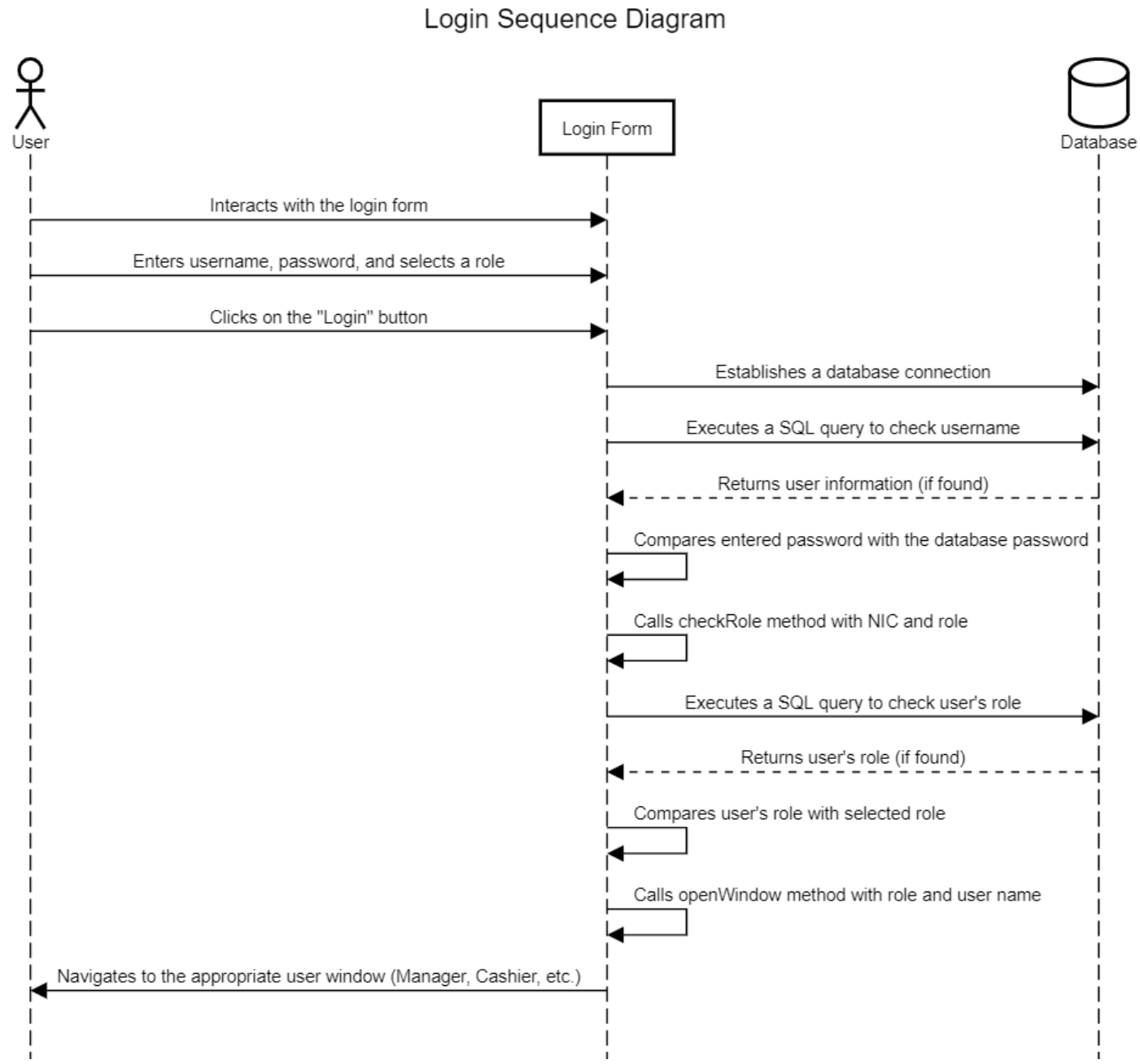
Design of the application

Use case diagram

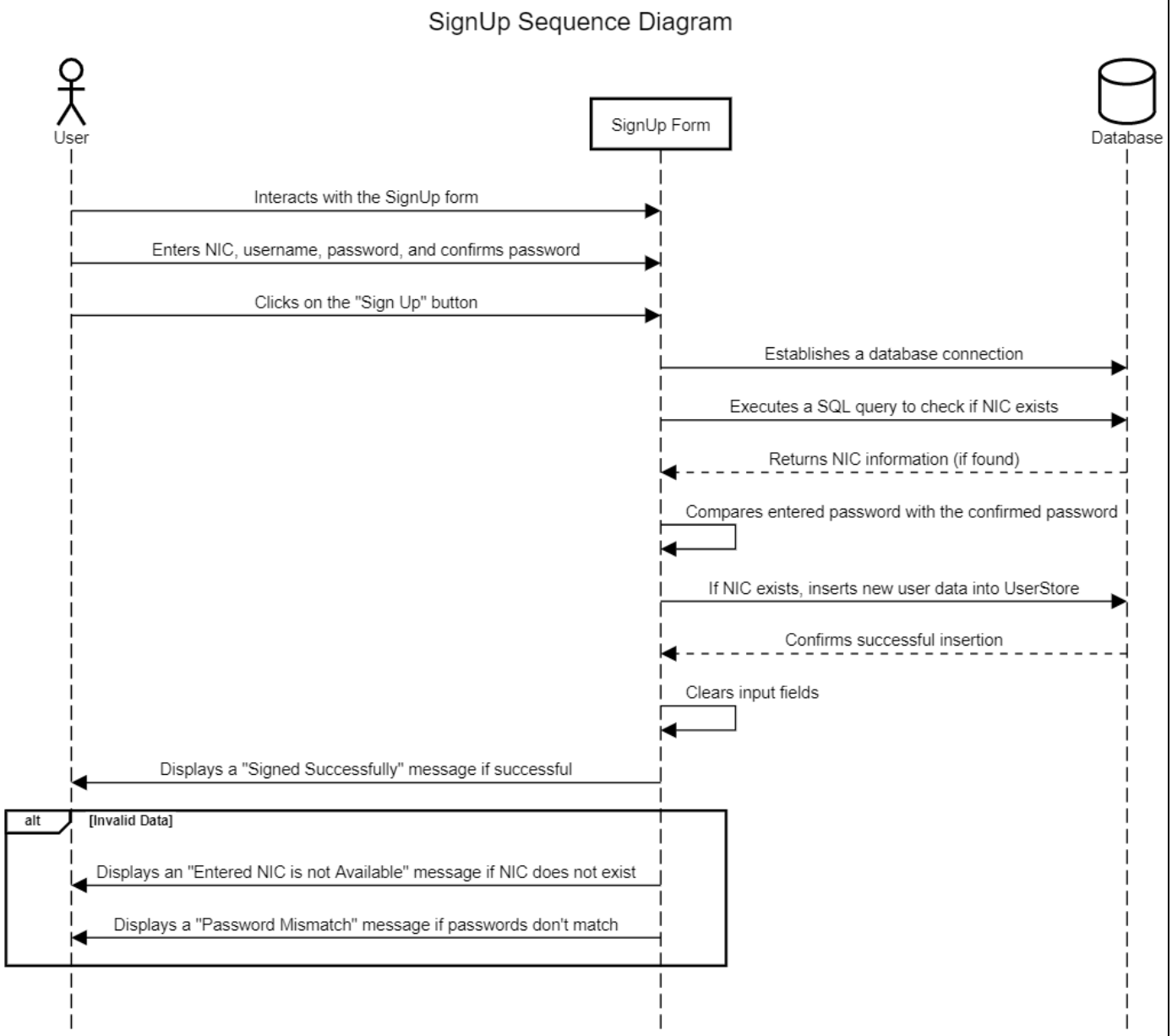
Class diagram

Sequence diagram

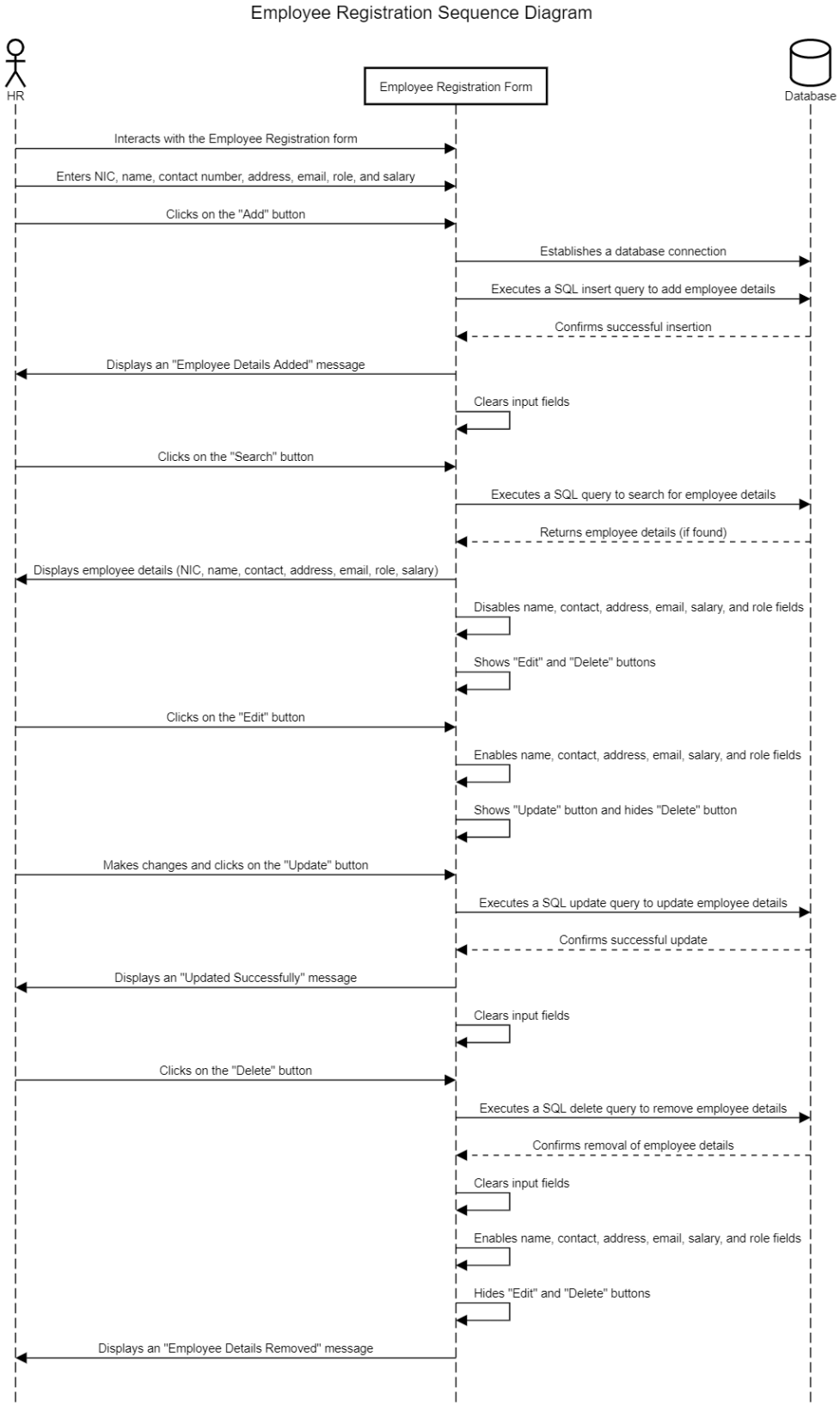
Login Sequence Diagram:



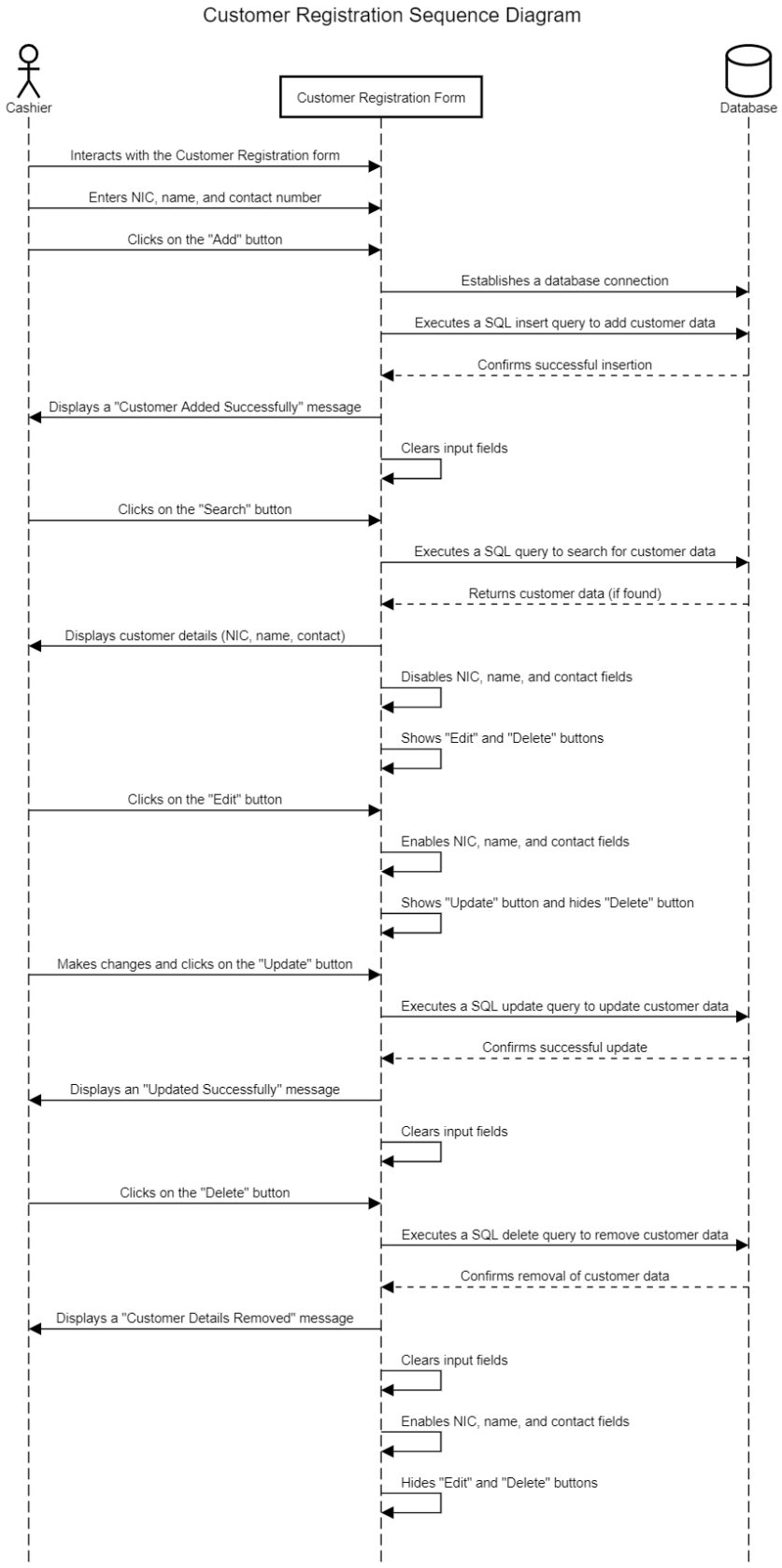
SignUp Sequence Diagram:



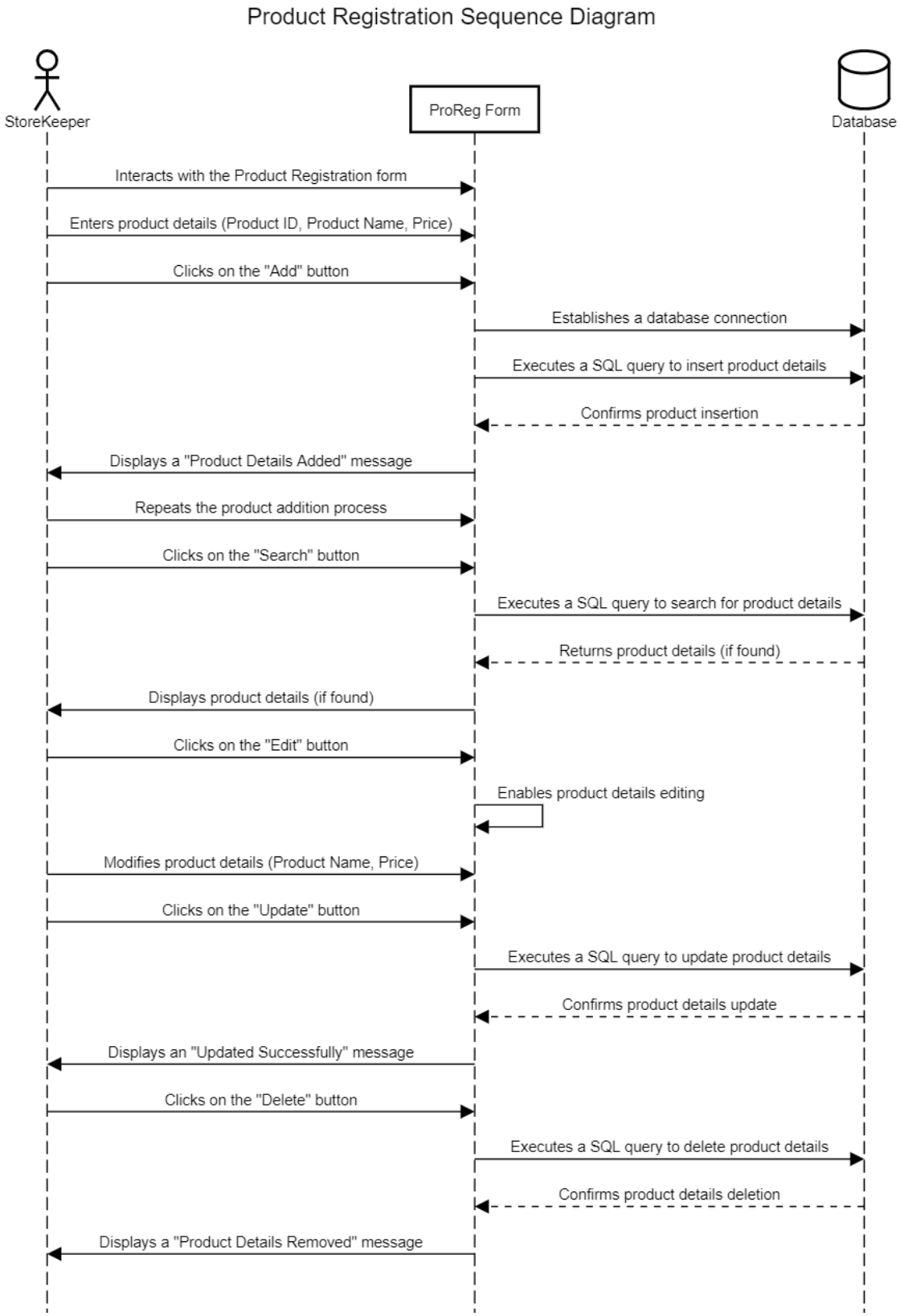
Employee Registration Sequence Diagram:



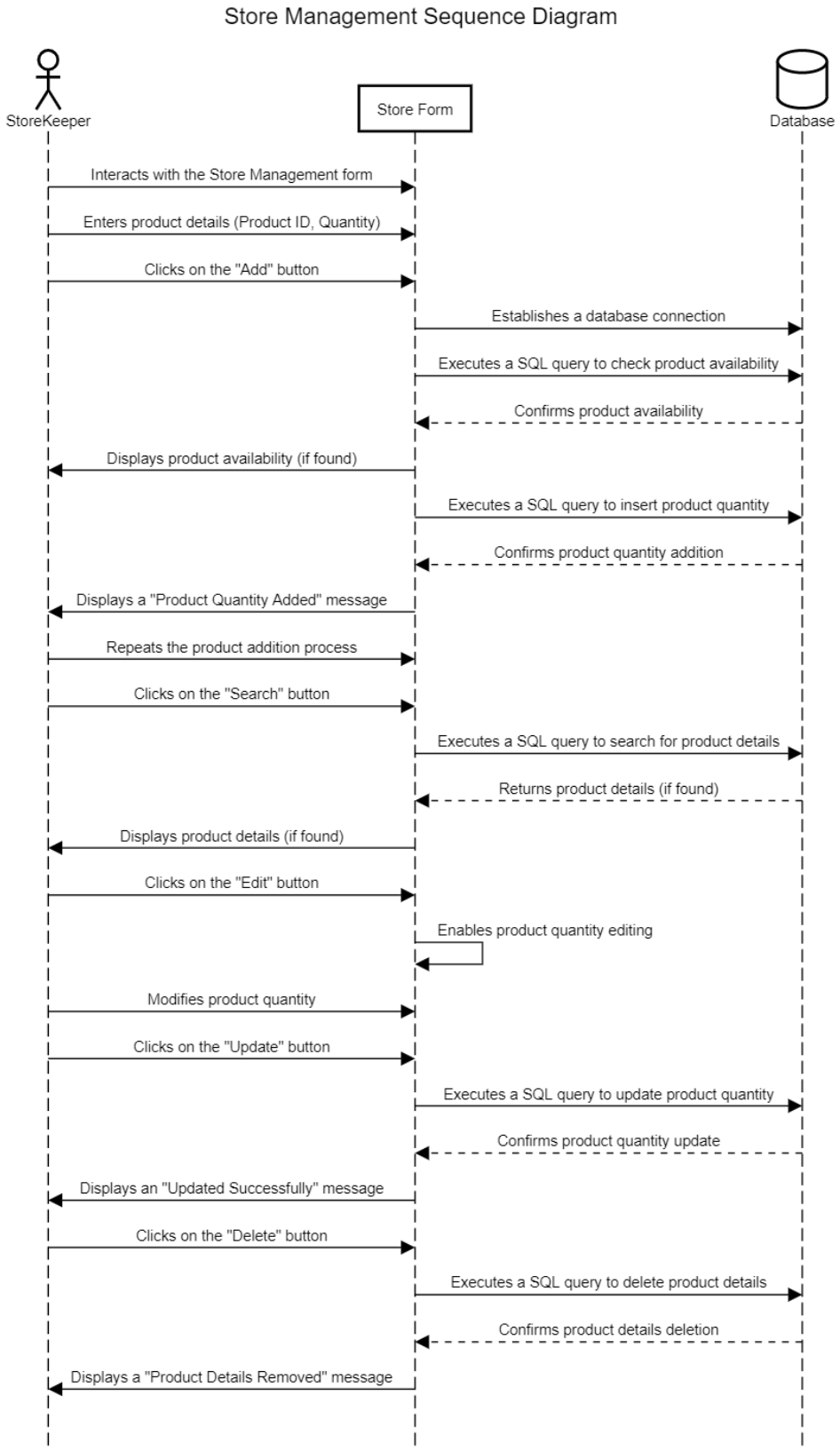
Customer Registration Sequence Diagram:



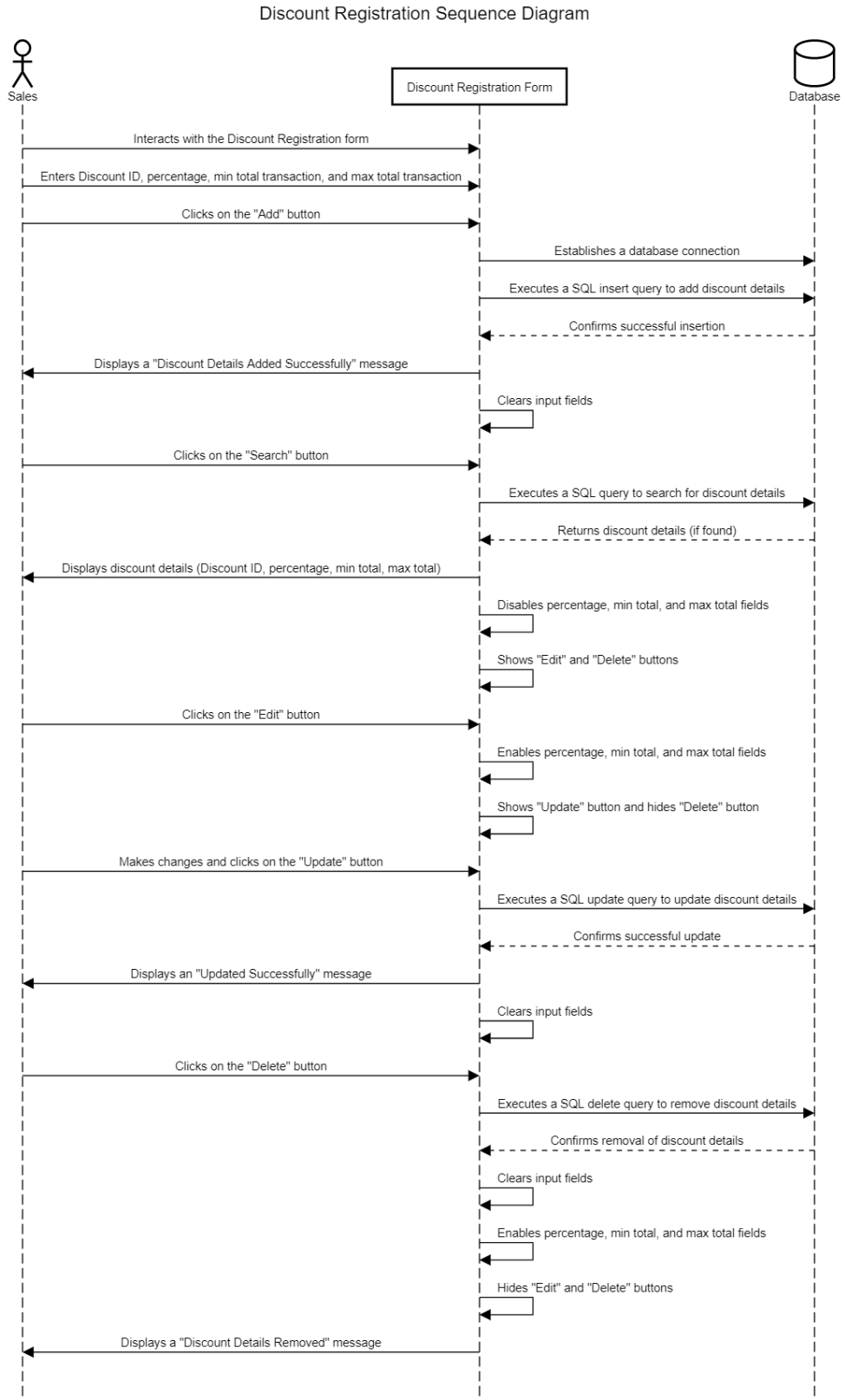
Product Registration Sequence Diagram:



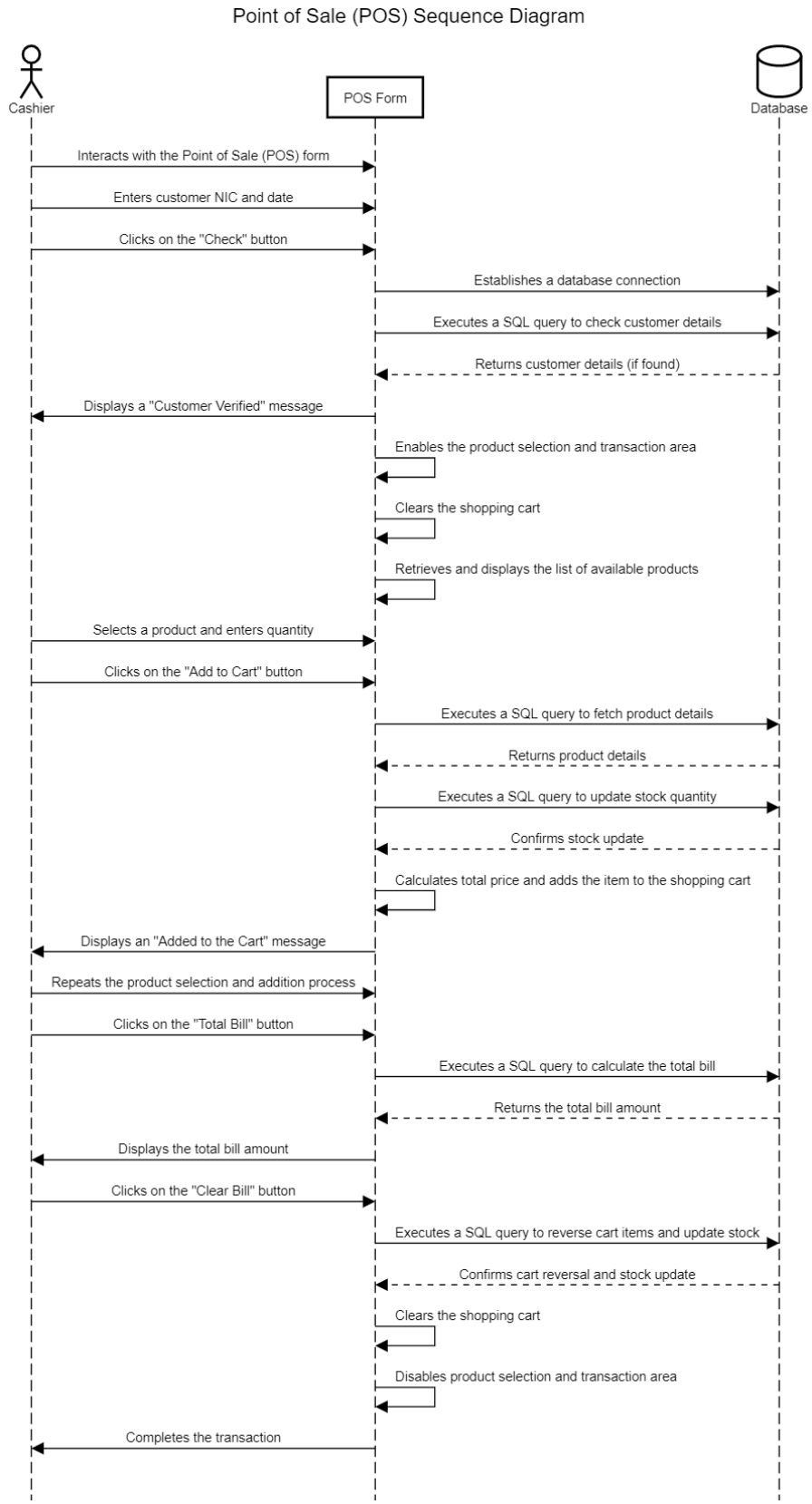
Stock Management Sequence Diagram:



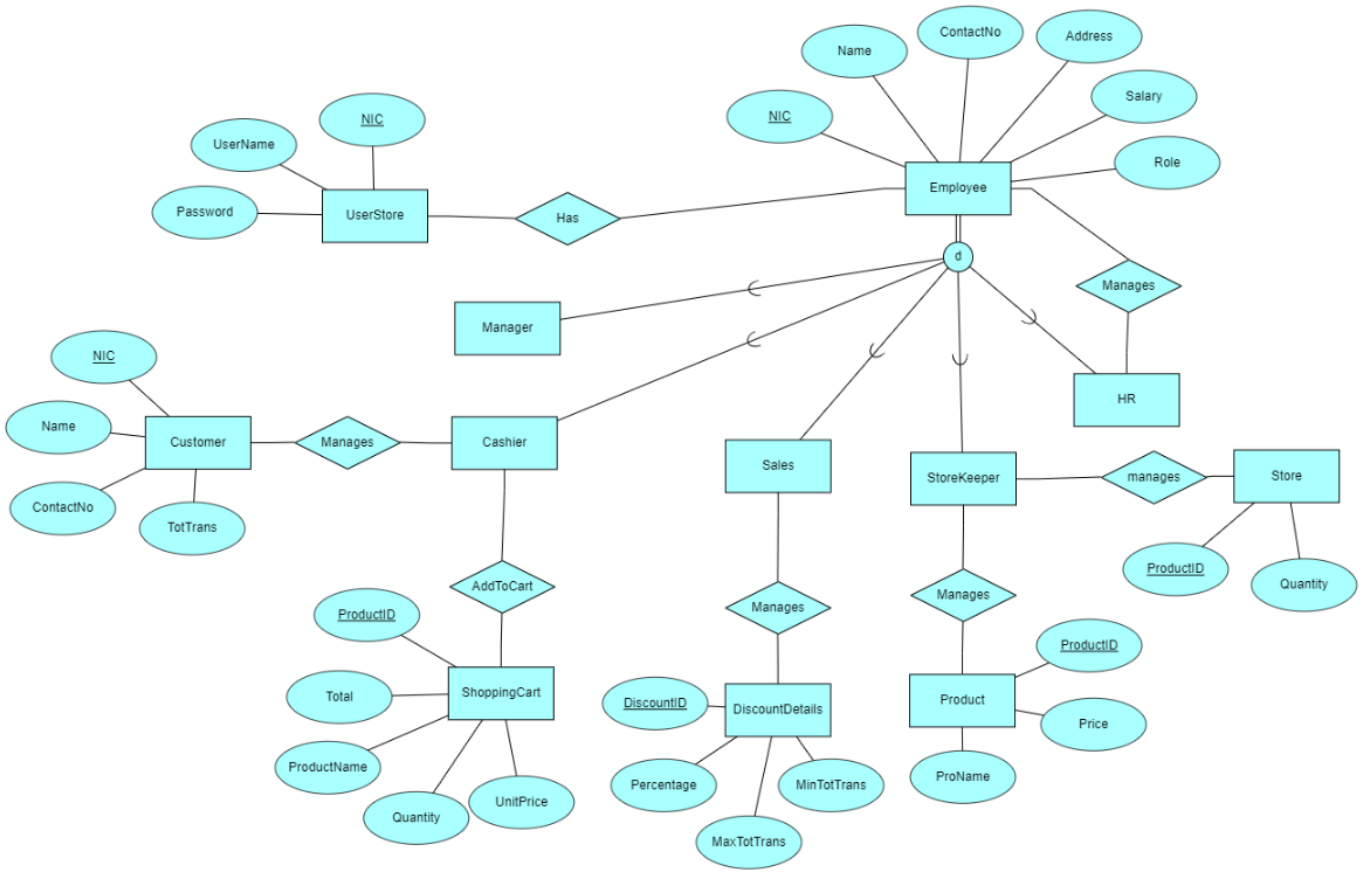
Discount Details Registration Sequence Diagram:



Point of Sale (POS) Sequence Diagram:



ER diagram



Database structure

Server Name: HARINDU
Database Name: SuperMarket

The Database has 7 tables as follows.

- Employee
- Product
- Customer
- Store
- UserStore
- DiscountDetails
- ShoppingCart

Harindu.SuperMarket - dbo.Employee

Column Name	Data Type	Allow Nulls
NIC	varchar(15)	☐
Name	varchar(50)	☑
ContactNo	nchar(10)	☑
Address	varchar(100)	☑
Email	varchar(50)	☑
Salary	int	☑
Role	varchar(50)	☑
		☐

Harindu.SuperMarket - dbo.Product

Column Name	Data Type	Allow Nulls
ProductID	varchar(10)	☐
ProName	varchar(50)	☑
Price	float	☑
		☐

Harindu.SuperMark...ket - dbo.Customer

Column Name	Data Type	Allow Nulls
Nic	varchar(15)	☐
Name	varchar(50)	☑
ContactNo	nchar(10)	☑
TotTrans	float	☑
		☐

Harindu.SuperMarket - dbo.Store

Column Name	Data Type	Allow Nulls
ProductID	varchar(10)	☐
Quantity	int	☑
		☐

Harindu.SuperMarket - dbo.UserStore

Column Name	Data Type	Allow Nulls
NIC	varchar(15)	☐
UserName	varchar(50)	☑
Password	varchar(15)	☑
		☐

Harindu.SuperMark...bo.DiscountDetails

Column Name	Data Type	Allow Nulls
DiscountID	varchar(10)	☐
Percentage	float	☑
MinTotTrans	int	☑
MaxTotTrans	int	☑
		☐

Harindu.SuperMark...dbo.ShoppingCart

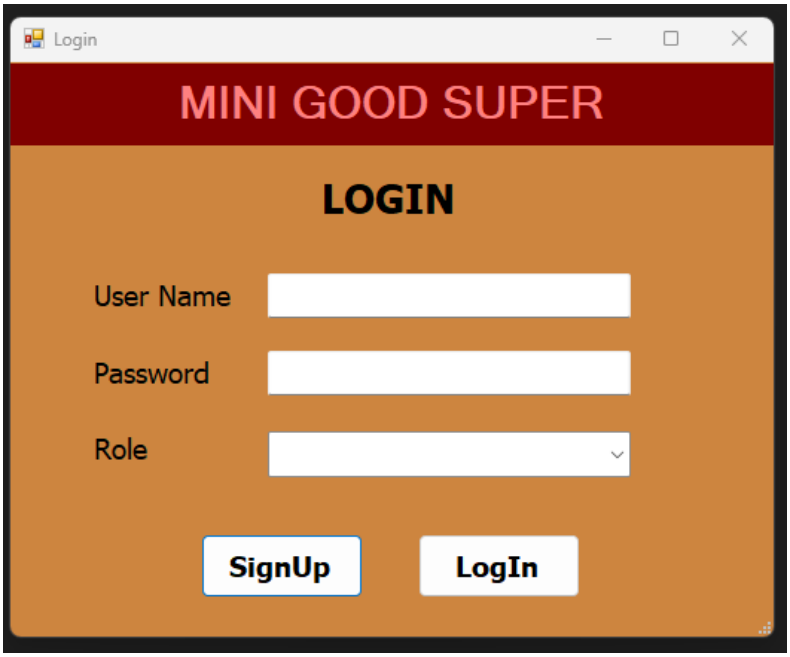
Column Name	Data Type	Allow Nulls
ProductID	varchar(10)	☐
ProductName	varchar(50)	☑
UnitPrice	float	☑
Quantity	int	☑
Total	float	☑
		☐

Design of forms and their Implementation

Main Loading Form --- >

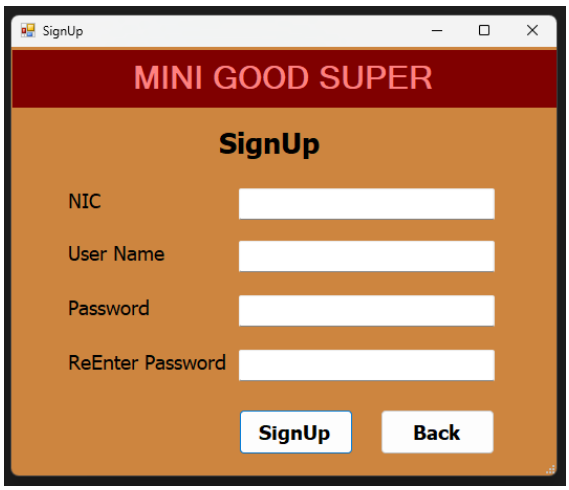
Login Form --- >

In this form users can enter their username password and the role to login to the system if the user hasn't sign-In, they can go to the signup page. The system will check the Database for authentication.



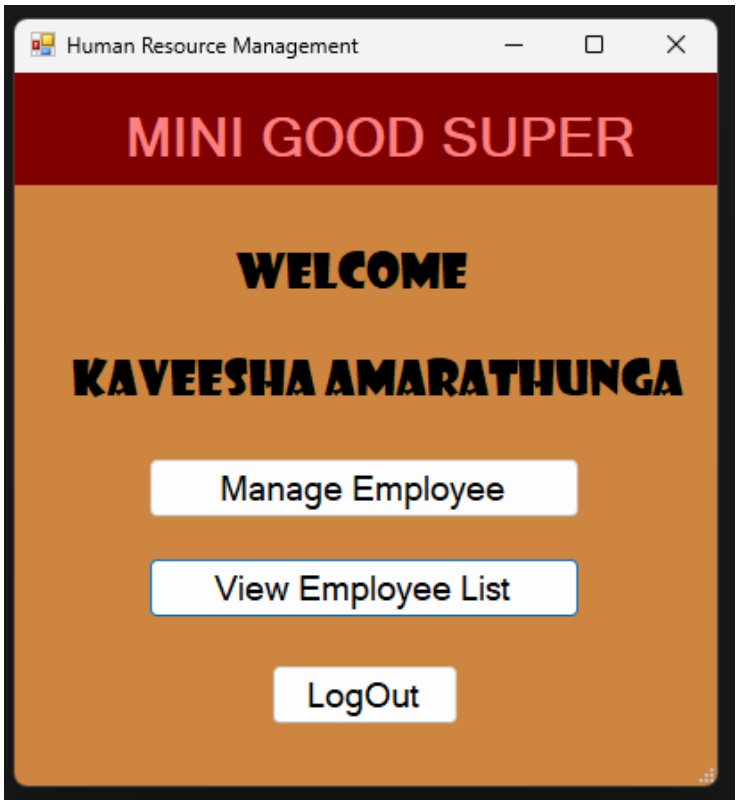
Sign-up Form --- >

The User Can Sign-In to the system by entering their NIC, Username and Password. The user needs to be in the System Database for registration.



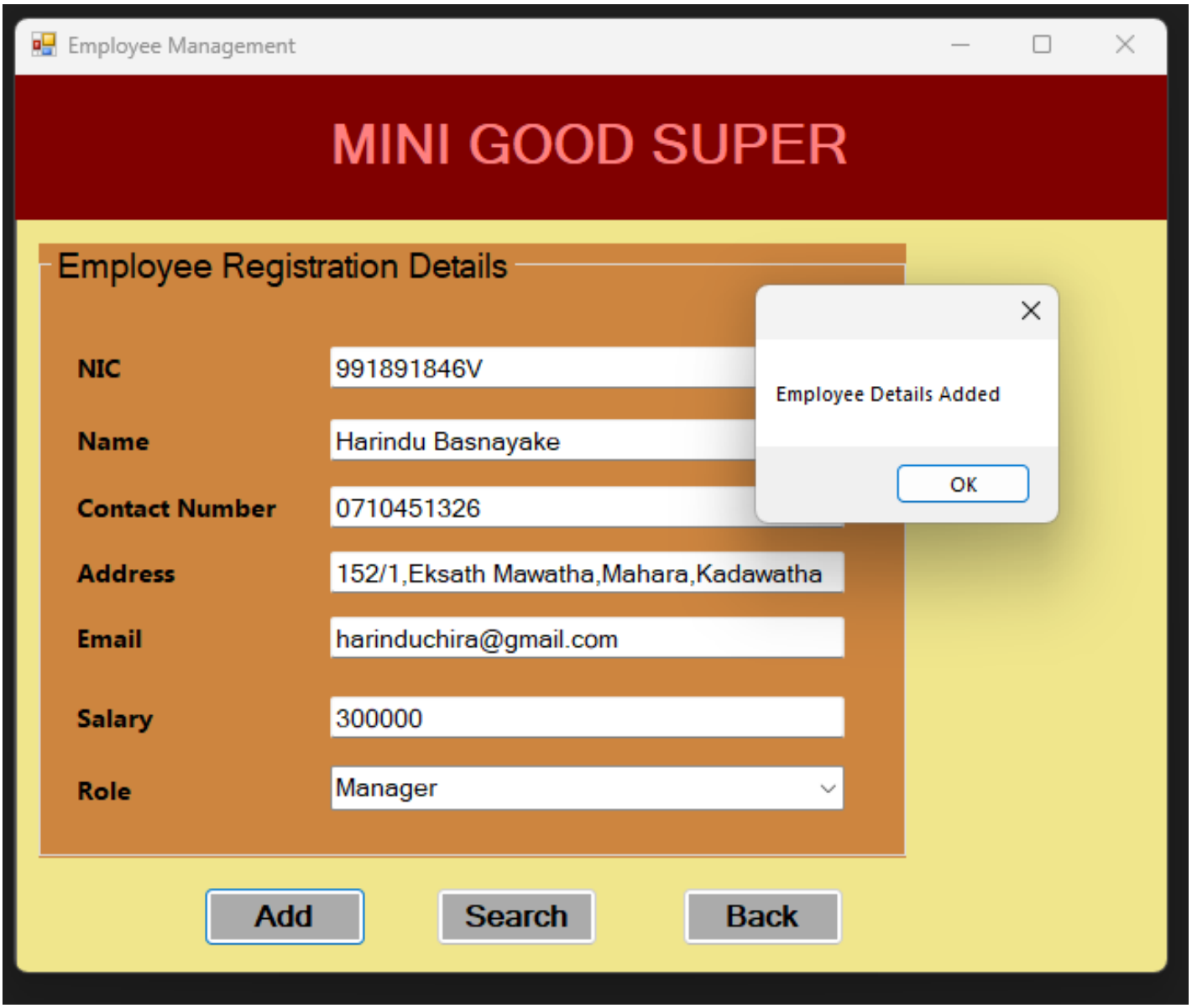
HR Form --- >

The Employee Kaveesha Amarathunga Has logged in to the system as a HR Employee. The HR can Manage Employee Details and View Employee Details.



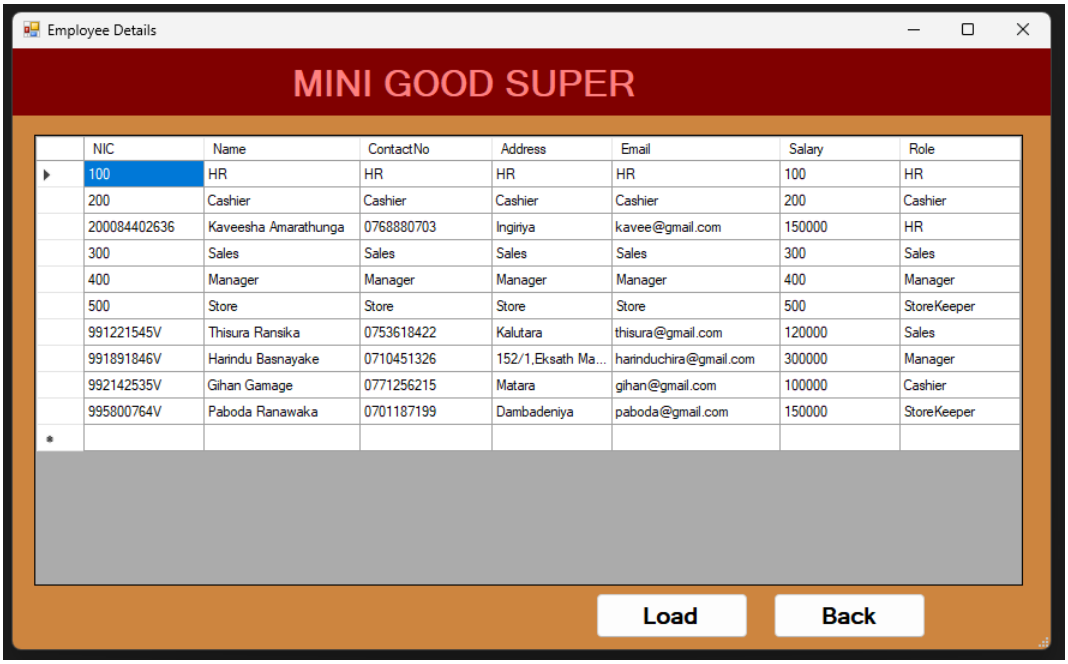
Employee Management Form --- >

Employee Details Can Be added to the system database in this form. And can search for any employee details by entering their NIC and Click Search. IF the searched data is available the form will display edit and delete buttons to edit or delete the employee details. Once the edit button is clicked the Employee details will be able to edit and update.



Employee Details List Form --- >

The HR and The Manager Can view the Employee details by clicking the load button on this form.



Storekeeper Form --- >

The Employee Paboda Ranawaka has logged into the system as a Storekeeper. Storekeeper can Manage Product Details, Manage Stock Details and view Product and stock Details.

Product Management Form --- >

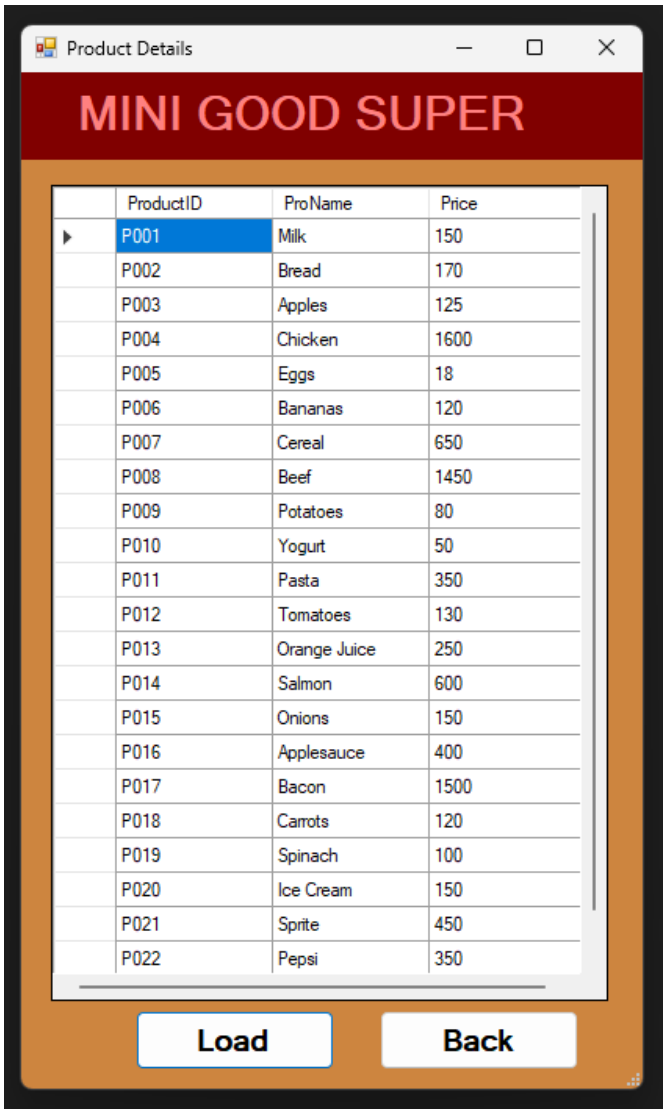
Product Details Can Be added to the system database in this form. And can search for any Product details by entering their ProductID and Click Search. IF the searched data is available the form will display edit and delete buttons to edit or delete the Product details. Once the edit button is clicked the Product details will be able to edit and update.

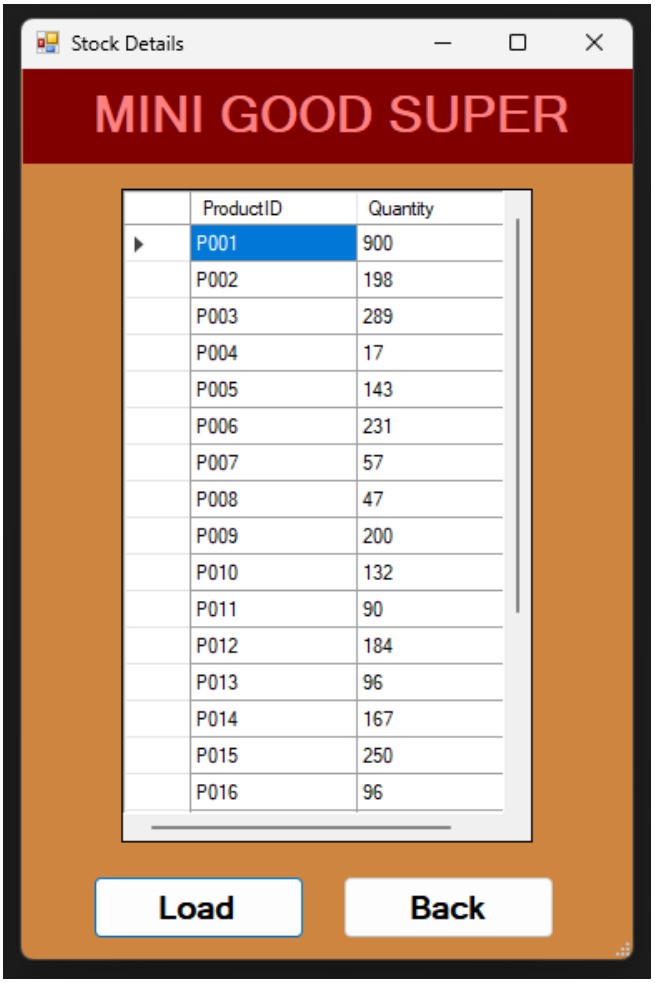
Stock Management Form --- >

Product Stock Can Be added to the system database in this form. And can search for any Product Stock details by entering their ProductID and Click Search. IF the searched data is available the form will display edit and delete buttons to edit or delete the Product Stock details. Once the edit button is clicked the Product Stock details will be able to edit and update.

Product List and Product Stock List Form --- >

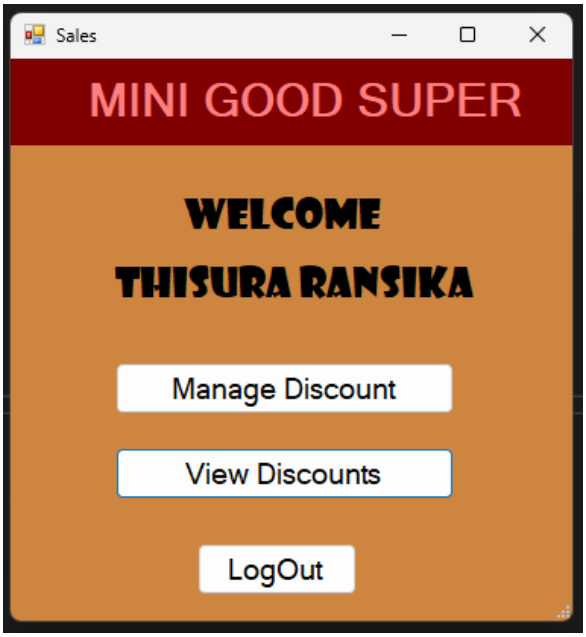
Product List and Product Stock list can only be Viewed by the Manager and the Storekeeper. Once the Load button is clicked it will display the lists. The product Details need be added to the database to add product stock details.





Sales Form --- >

The Employee Thisuara Ransika has logged into the system as Sales. Sales can Manage Discount Details and view Discount Details.

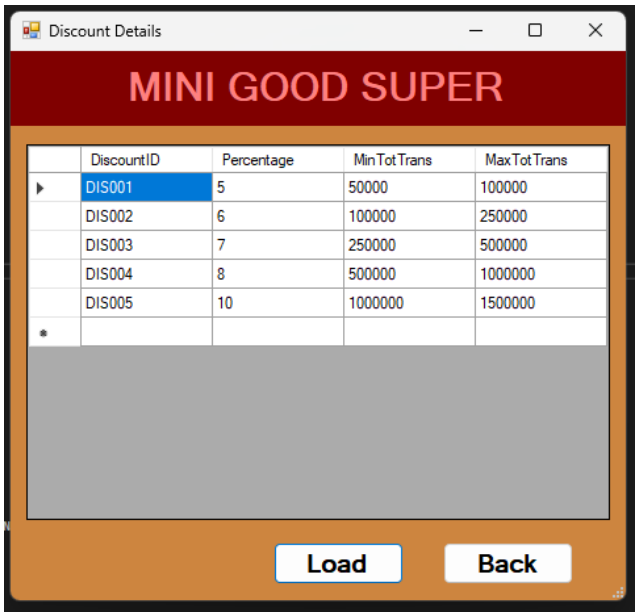


Discount Management Form --- >

Discount Details Can Be added to the system database in this form. And can search for any Discount details by entering their Discount ID and Click Search. IF the searched data is available the form will display edit and delete buttons to edit or delete the Discount details. Once the edit button is clicked the Discount details will be able to edit and update.

Discount List Form --- >

Discount List can only be Viewed by the Manager and the Sales. Once the Load button is clicked it will display the lists.



Cashier Form --- >

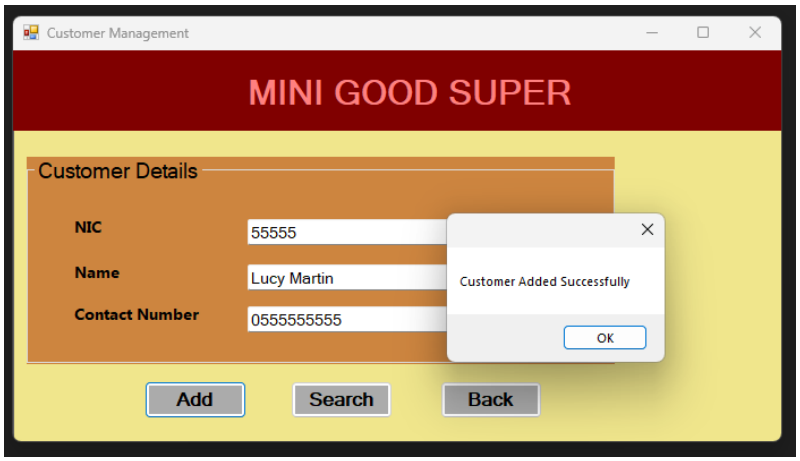
The Employee Gihan Gamage has logged into the system as Cashier. Cashier can Manage Customer Details and Do Sales via POS.

Cashier Form --- >

Cashier Can do sales by entering and verify the customer details first. The customer needs to be registered in the system to do transactions. Once the customer details are verified the Add Product and Shopping Cart fields will be accessible. The Cashier can Search the product Name from the dropdown menu and enter the quantity and add it into the cart. Once the Add to cart button click it will check the product stock whether the quantity is available or not if quantity Is available it will add to the shopping cart. Once the Cashier clicked Clear Bill button It will clear the shopping cart and reverse the added product quantity to the store database. Once the Cashier clicked Total bill the Total of the shopping cart will display and the clear the current shopping cart and the total sale amount will be added to the customer details for promotions. If the Customer is eligible for discount, it will display once the total bill is clicked. The Cashier can add the discount to the customer bill.

Customer Management Form --- >

Customer Details Can Be added to the system database in this form. And can search for any Customer details by entering their NIC and Click Search. If the searched data is available, the form will display edit and delete buttons to edit or delete the Customer details. Once the edit button is clicked the Customer details will be able to edit and update.



Manager Form --- >

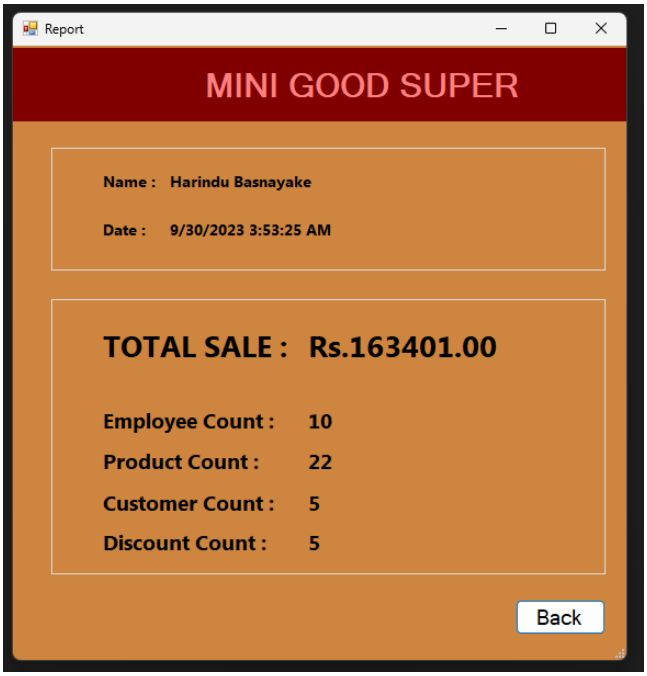
The Employee Harindu Basnayake has logged into the system as Manager. Manager can View Employee Details, Customer Details, Product Details, Product Stock Details, Discount Details and View Reports.

Customer List Form --- >

Customer List can only be Viewed by the Manager. Once the Load button is clicked it will display the list of customer details.

Report Form --- >

The manager can view the report form and it will display the total sales of the Super Market and the Count of each Employees, Customers, Products and Discounts available in the system.



Implementation

Program.cs -->

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SuperMarket
{
    internal static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Main());
        }
    }
}
```

Program.cs -->

```
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 SuperMarket
{
    public partial class Main : Form
    {
        int startPoint = 0;

        public Main()
        {
            InitializeComponent();
        }

        private void Main_Load(object sender, EventArgs e)
        {
            timer1.Start();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            startPoint += 1;

            progressBar1.Value = startPoint;

            if (progressBar1.Value == 100)
            {
                progressBar1.Value = 0;
```

```
                timer1.Stop();

                this.Hide();
                Login login = new Login();
                login.ShowDialog();
                this.Close();
            }
        }
    }
}
```

Login.cs -->

```
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 SuperMarket
{
    public partial class Login : Form
    {

        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        string userName, pass, role, dbpass, dbrole, dbNic, dbName;

        public Login()
        {
            InitializeComponent();
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            userName = txtUserName.Text;
            pass = txtPass.Text;
            role = cmbRole.SelectedItem.ToString();

            string checkUserPass = "SELECT * FROM UserStore WHERE UserName =
@SearchValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using(SqlCommand command = new
SqlCommand(checkUserPass,connection))
                {
                    command.Parameters.AddWithValue("@SearchValue", userName);

                    using(SqlDataReader reader = command.ExecuteReader() )
                    {
                        if(reader.HasRows)
                        {
```

```
                    while(reader.Read())
                    {
                        dbNic = reader["NIC"].ToString();
                        dbpass = reader["Password"].ToString();

                        if(dbpass == pass)
                        {
                            if(checkRole(dbNic,role))
                            {
                                openWindow(role,dbName);
                            }
                        }
                        else
                        {
                            MessageBox.Show("Password is Invalid");
                        }
                    }
                }
                else
                {
                    MessageBox.Show("User Name is Invalid");
                }
            }
        }
    }

    private void btnSignUp_Click(object sender, EventArgs e)
    {
        this.Hide();
        SignUp signUp = new SignUp();
        signUp.ShowDialog();
        this.Close();
    }

    private bool checkRole(string dbNic,string role)
    {
        string checkRole = "SELECT * FROM Employee WHERE NIC = @SearchValue";

        using(SqlConnection connection = new SqlConnection(conString))
        {
            connection.Open();

            using(SqlCommand command = new SqlCommand(checkRole,connection))
            {
                command.Parameters.AddWithValue("@SearchValue", dbNic);

                using(SqlDataReader reader = command.ExecuteReader())
                {
                    if(reader.HasRows)
                    {
                        while(reader.Read())
                        {
                            dbrole = reader["Role"].ToString();
                            dbName = reader["Name"].ToString();
                        }
                    }
                    else
                    {
                        MessageBox.Show("NIC is Invalid");
                    }
                }
            }
```

```
                connection.Close();
            }

            if (dbrole == role)
            {
                return true;
            }
            else
            {
                MessageBox.Show("Role is Invalid");
                return false;
            }
        }

        public void openWindow(string role,string user)
        {
            switch (role)
            {
                case "Manager":
                    this.Hide();
                    Manager manager = new Manager(user);
                    manager.ShowDialog();
                    this.Close();
                    break;

                case "Cashier":
                    this.Hide();
                    Cashier cashier = new Cashier(user);
                    cashier.ShowDialog();
                    this.Close();
                    break;

                case "StoreKeeper":
                    this.Hide();
                    StoreKeeper storeKeeper = new StoreKeeper(user);
                    storeKeeper.ShowDialog();
                    this.Close();
                    break;

                case "HR":
                    this.Hide();
                    HR hR = new HR(user);
                    hR.ShowDialog();
                    this.Close();
                    break;

                case "Sales":
                    this.Hide();
                    Sales sales = new Sales(user);
                    sales.ShowDialog();
                    this.Close();
                    break;
            }
        }
    }
}
```

SignUp.cs -->

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

namespace SuperMarket
{
    public partial class SignUp : Form
    {
        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        string nic, userName, pass, passCon;

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.ShowDialog();
            this.Close();
        }

        public SignUp()
        {
            InitializeComponent();
        }

        private void btnSignIN_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;
            userName = txtUserName.Text;
            pass = txtPass.Text;
            passCon = txtPassCon.Text;

            if (pass == passCon)
            {
                string insertSql = "INSERT INTO UserStore (NIC,UserName,Password)
VALUES (@Value1, @Value2,@value3)";

                string selectSql = "SELECT * FROM Employee WHERE NIC =
@SearchValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand(selectSql,
connection))
                    {
                        command.Parameters.AddWithValue("@SearchValue", nic);

                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            if (reader.HasRows)
```

```
                            {
                                reader.Close();

                                using (SqlCommand cmd = new SqlCommand(insertSql,
connection))
                                {
                                    cmd.Parameters.AddWithValue("@Value1", nic);
                                    cmd.Parameters.AddWithValue("@Value2",
userName);
                                    cmd.Parameters.AddWithValue("@Value3", pass);

                                    cmd.ExecuteNonQuery();
                                }

                                MessageBox.Show("Signed Successfully");

                                txtNIC.Clear();
                                txtUserName.Clear();
                                txtPass.Clear();
                                txtPassCon.Clear();
                            }
                            else
                            {
                                MessageBox.Show("Entered NIC IS not Available");
                            }
                        }
                    }
                    connection.Close();
                }
            }
            else
            {
                MessageBox.Show("Password Mismatch");
            }
        }
    }
}
```

HR.cs -->

```
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 SuperMarket
{
    public partial class HR : Form
    {
        string userName;
        string role = "HR";
        public HR(string user)
        {
            InitializeComponent();

            userName = user;
```

```
            lblDisplay.Text =  user;
        }

        private void btnEmpReg_Click(object sender, EventArgs e)
        {
            this.Hide();
            EmpReg empReg = new EmpReg(userName);
            empReg.ShowDialog();
            this.Close();
        }

        private void btnLogOut_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.ShowDialog();
            this.Close();
        }

        private void btnEmpList_Click(object sender, EventArgs e)
        {
            this.Hide();
            EmpList empList = new EmpList(userName,role);
            empList.ShowDialog();
            this.Close();
        }
    }
}
```

EmpReg.cs -->

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

namespace SuperMarket
{
    public partial class EmpReg : Form
    {
        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        string nic, name, contactNO, address, email, role, oldnic;

        string userName;

        int salary;
```

```
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            HR hr = new HR(userName);
            hr.ShowDialog();
            this.Close();
        }

        public EmpReg(string userName)
        {
            InitializeComponent();

            btnEdit.Visible = false;
            btnUpdate.Visible = false;
            btnDelete.Visible = false;

            this.userName = userName;
        }

        private void btnAdd_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;
            name = txtName.Text;
            contactNO = txtContact.Text;
            address = txtAddress.Text;
            email = txtEmail.Text;
            role = cmbRole.SelectedItem.ToString();
            salary = int.Parse(txtSalary.Text);


            string insertSql = "INSERT INTO Employee
(NIC,Name,ContactNo,Address,Email,Salary,Role) VALUES (@Value1,
@Value2,@Value3,@Value4,@Value5,@Value6,@Value7)";


            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(insertSql,
connection))
                {
                    command.Parameters.AddWithValue("@Value1", nic);
                    command.Parameters.AddWithValue("@Value2", name);
                    command.Parameters.AddWithValue("@Value3", contactNO);
                    command.Parameters.AddWithValue("@Value4", address);
                    command.Parameters.AddWithValue("@Value5", email);
                    command.Parameters.AddWithValue("@Value6", salary);
                    command.Parameters.AddWithValue("@Value7", role);

                    command.ExecuteNonQuery();
                }

                connection.Close();
            }

            MessageBox.Show("Employee Details Added");
```

```
            txtNIC.Clear();
            txtName.Clear();
            txtContact.Clear();
            txtAddress.Clear();
            txtEmail.Clear();
            txtSalary.Clear();
        }

        private void btnSearch_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;

            oldnic = nic;

            if (nic == null)
            {
                MessageBox.Show("Please enter the NIC to Search");
            }
            else
            {
                string selectSql = "SELECT * FROM Employee WHERE NIC =
@SearchValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand(selectSql,
connection))
                    {
                        command.Parameters.AddWithValue("@SearchValue", nic);

                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            if (reader.HasRows)
                            {
                                 while (reader.Read())
                                 {
                                    txtNIC.Text = reader["NIC"].ToString();
                                    txtName.Text = reader["Name"].ToString();
                                    txtContact.Text =
reader["ContactNo"].ToString();
                                    txtAddress.Text =
reader["Address"].ToString();
                                    txtEmail.Text = reader["Email"].ToString();
                                    txtSalary.Text = reader["Salary"].ToString();
                                    cmbRole.Text = reader["Role"].ToString();

                                    MessageBox.Show("Employee Details
Available");

                                    btnEdit.Visible = true;
                                    btnDelete.Visible = true;

                                    txtName.Enabled = false;
                                    txtContact.Enabled = false;
                                    txtAddress.Enabled = false;
                                    txtEmail.Enabled = false;
```

```
                                txtSalary.Enabled = false;
                                cmbRole.Enabled = false;
                            }
                        }
                        else {
                            MessageBox.Show("Entered NIC is not Available");
                        }
                    }
                }
                connection.Close();
            }
        }
    }

    private void btnEdit_Click(object sender, EventArgs e)
    {
        btnUpdate.Visible = true;
        btnDelete.Visible = false;

        txtName.Enabled = true;
        txtContact.Enabled = true;
        txtAddress.Enabled = true;
        txtEmail.Enabled =  true;
        txtSalary.Enabled = true;
        cmbRole.Enabled = true;
    }

    private void btnUpdate_Click(object sender, EventArgs e)
    {
        nic = txtNIC.Text;
        name = txtName.Text;
        contactNO = txtContact.Text;
        address = txtAddress.Text;
        email = txtEmail.Text;
        role = cmbRole.SelectedItem.ToString();
        salary = int.Parse(txtSalary.Text);

        string updateSql = "UPDATE Employee SET NIC = @NewValue1, Name =
@NewValue2 , ContactNo = @NewValue3 , Address = @NewValue4 , Email = @NewValue5 ,
Salary = @NewValue6 , Role = @NewValue7 WHERE NIC = @PrimaryKeyValue";

        using (SqlConnection connection = new SqlConnection(conString))
        {
            connection.Open();

            using (SqlCommand command = new SqlCommand(updateSql,
connection))
            {
                command.Parameters.AddWithValue("@NewValue1", nic);
                command.Parameters.AddWithValue("@NewValue2", name);
                command.Parameters.AddWithValue("@NewValue3", contactNO);
                command.Parameters.AddWithValue("@NewValue4", address);
                command.Parameters.AddWithValue("@NewValue5", email);
                command.Parameters.AddWithValue("@NewValue6", salary);
                command.Parameters.AddWithValue("@NewValue7", role);
                command.Parameters.AddWithValue("@PrimaryKeyValue", oldnic);

                // Execute the SQL command
```

```
                    command.ExecuteNonQuery();

                    updateUserStore();

                    MessageBox.Show("Updated Successfully");

                    txtNIC.Clear();
                    txtName.Clear();
                    txtContact.Clear();
                    txtAddress.Clear();
                    txtEmail.Clear();
                    txtSalary.Clear();

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;

                    oldnic = "";
                }
                connection.Close();
            }

            void updateUserStore()
            {
                string updateSqlUserStore = "UPDATE UserStore SET NIC =
@NewValue1 WHERE NIC = @PrimaryKeyValue";

                using(SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using(SqlCommand command = new SqlCommand(updateSqlUserStore,
connection))
                    {
                        command.Parameters.AddWithValue("@NewValue1", nic);
                        command.Parameters.AddWithValue("@PrimaryKeyValue",
oldnic);

                        command.ExecuteNonQuery();
                    }
                }
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;

            string deleteSql = "DELETE FROM Employee WHERE NIC =
@PrimaryKeyValue";

            using(SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(deleteSql,
connection))
                {
```

```
                    command.Parameters.AddWithValue("@PrimaryKeyValue", nic);

                    command.ExecuteNonQuery();

                    deleteUserStoreDetails(nic);

                    MessageBox.Show("Employee Details Removed");

                    txtNIC.Clear();
                    txtName.Clear();
                    txtContact.Clear();
                    txtAddress.Clear();
                    txtEmail.Clear();
                    txtSalary.Clear();

                    txtName.Enabled = true;
                    txtContact.Enabled = true;
                    txtAddress.Enabled = true;
                    txtEmail.Enabled = true;
                    txtSalary.Enabled = true;
                    cmbRole.Enabled = true;

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;

                }
                connection.Close() ;
            }

            void deleteUserStoreDetails(string Nic)
            {
                string deleteSqlUserStore = "DELETE FROM UserStore WHERE NIC =
@PrimaryKeyValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using (SqlCommand command = new
SqlCommand(deleteSqlUserStore, connection))
                    {
                        command.Parameters.AddWithValue("@PrimaryKeyValue", Nic);

                        command.ExecuteNonQuery();
                    }

                    connection.Close();
                }
            }
        }
    }
```

EmpList.cs -->

```
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 SuperMarket
{
    public partial class EmpList : Form
    {
        string userName;
        string role;
        public EmpList(string userName,string role)
        {
            InitializeComponent();
            this.userName = userName;
            this.role = role;
        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
            empList11.Clear();
            sqlDataAdapter1.Fill(empList11);
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role, userName);
            this.Close();
        }
    }
}
```

StoreKeeper.cs -->

```
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 SuperMarket
{
    public partial class StoreKeeper : Form
    {
        string userName;
        string role = "StoreKeeper";
        public StoreKeeper(string user)
        {
            InitializeComponent();
            userName = user;
            lblDisplay.Text =  user;
        }

        private void btnProReg_Click(object sender, EventArgs e)
        {
            this.Hide();
            ProReg proReg = new ProReg(userName);
            proReg.ShowDialog();
            this.Close();
        }

        private void btnLogOut_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.ShowDialog();
            this.Close();
        }

        private void btnStore_Click(object sender, EventArgs e)
        {
            this.Hide();
            Store store = new Store(userName);
            store.ShowDialog();
            this.Close();
        }

        private void btnViewProList_Click(object sender, EventArgs e)
        {
            this.Hide();
            ProList proList = new ProList(userName,role);
            proList.ShowDialog();
            this.Close();
        }

        private void btnViewStore_Click(object sender, EventArgs e)
        {
            this.Hide();
            StockList stckList = new StockList(userName,role);
            stckList.ShowDialog();
            this.Close();
        }
    }
}
```

ProReg.cs -->

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

namespace SuperMarket
{
    public partial class ProReg : Form
    {
        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        string proID, proName, oldProID;

        string userName;

        float price;

        private void btnSearch_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;

            oldProID = proID;

            if (proID == null)
            {
                MessageBox.Show("Please enter the Product ID to Search");
            }
            else
            {
                string selectSql = "SELECT * FROM Product WHERE ProductID =
@SearchValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand(selectSql,
connection))
                    {
                        command.Parameters.AddWithValue("@SearchValue", proID);

                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            if (reader.HasRows)
                            {
                                while (reader.Read())
                                {
                                    txtPID.Text = reader["ProductID"].ToString();
                                    txtProName.Text =
reader["ProName"].ToString();
```

```
                                txtPrice.Text = reader["Price"].ToString();

                                MessageBox.Show("Product Details Available");

                                btnEdit.Visible = true;
                                btnDelete.Visible = true;

                                txtProName.Enabled = false;
                                txtPrice.Enabled = false;
                            }
                        }
                        else
                        {
                            MessageBox.Show("Entered Product ID is not
Available");
                        }
                    }
                }
                connection.Close();
            }
        }

        private void btnEdit_Click(object sender, EventArgs e)
        {
            btnUpdate.Visible = true;
            btnDelete.Visible = false;

            txtProName.Enabled = true;
            txtPrice.Enabled = true;
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;
            proName = txtProName.Text;
            price = float.Parse(txtPrice.Text);

            string updateSql = "UPDATE Product SET ProductID = @NewValue1,
ProName = @NewValue2 , Price = @NewValue3 WHERE ProductID = @PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(updateSql,
connection))
                {
                    command.Parameters.AddWithValue("@NewValue1", proID);
                    command.Parameters.AddWithValue("@NewValue2", proName);
                    command.Parameters.AddWithValue("@NewValue3", price);
                    command.Parameters.AddWithValue("@PrimaryKeyValue",
oldProID);

                    // Execute the SQL command
                    command.ExecuteNonQuery();

                    MessageBox.Show("Updated Successfully");

                    txtPID.Clear();
                    txtProName.Clear();
                    txtPrice.Clear();
```

```
                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;

                    oldProID = "";
                }
                connection.Close();
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;

            string deleteSql = "DELETE FROM Product WHERE ProductID =
@PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(deleteSql,
connection))
                {
                    command.Parameters.AddWithValue("@PrimaryKeyValue", proID);

                    command.ExecuteNonQuery();

                    MessageBox.Show("Product Details Removed");

                    txtPID.Clear();
                    txtProName.Clear();
                    txtPrice.Clear();

                    txtPID.Enabled = true;
                    txtProName.Enabled = true;
                    txtPrice.Enabled = true;

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;

                }
                connection.Close();
            }
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            StoreKeeper storeKeeper = new StoreKeeper(userName);
            storeKeeper.ShowDialog();
            this.Close();
        }

        public ProReg(string userName)
        {
            InitializeComponent();

            this.userName = userName;

            btnEdit.Visible = false;
            btnUpdate.Visible = false;
```

```
            btnDelete.Visible = false;
        }

        private void btnAdd_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;
            proName = txtProName.Text;
            price = float.Parse(txtPrice.Text);

            string insertSql = "INSERT INTO Product (ProductID,ProName,Price)
VALUES (@Value1, @Value2,@Value3)";


            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(insertSql,
connection))
                {
                    command.Parameters.AddWithValue("@Value1", proID);
                    command.Parameters.AddWithValue("@Value2", proName);
                    command.Parameters.AddWithValue("@Value3", price);

                    command.ExecuteNonQuery();
                }

                connection.Close();
            }

            MessageBox.Show("Product Details Added");

            txtPID.Clear();
            txtProName.Clear();
            txtPrice.Clear();
        }


    }
}
```

ProList.cs -->

```
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 SuperMarket
{
    public partial class ProList : Form
    {
        string userName;
        string role;
        public ProList(string userName, string role)
```

```
        {
            InitializeComponent();

            this.userName = userName;
            this.role = role;
        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
            proList11.Clear();
            sqlDataAdapter1.Fill(proList11);
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role, userName);
            this.Close();
        }
    }
}
```

Store.cs -->

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

namespace SuperMarket
{
    public partial class Store : Form
    {
        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        string proID, oldProID;

        string userName;

        int qty;


        public Store(string userName)
        {
            InitializeComponent();

            this.userName = userName;

            btnEdit.Visible = false;
            btnUpdate.Visible = false;
            btnDelete.Visible = false;
        }
```

```
        private void btnSearch_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;

            oldProID = proID;

            if (proID == null)
            {
                MessageBox.Show("Please enter the Product ID to Search");
            }
            else
            {
                string selectSql = "SELECT * FROM Store WHERE ProductID =
@SearchValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand(selectSql,
connection))
                    {
                        command.Parameters.AddWithValue("@SearchValue", proID);

                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            if (reader.HasRows)
                            {
                                while (reader.Read())
                                {
                                    txtPID.Text = reader["ProductID"].ToString();
                                    txtQuantity.Text =
reader["Quantity"].ToString();

                                    MessageBox.Show("Product Details Available");

                                    btnEdit.Visible = true;
                                    btnDelete.Visible = true;

                                    txtQuantity.Enabled = false;
                                }
                            }
                            else
                            {
                                MessageBox.Show("Entered Product ID is not
Available");
                            }
                        }
                    }
                    connection.Close();
                }
            }
        }

        private void btnEdit_Click(object sender, EventArgs e)
        {
            btnUpdate.Visible = true;
            btnDelete.Visible = false;

            txtQuantity.Enabled = true;
        }
```

```
        private void btnUpdate_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;
            qty = int.Parse(txtQuantity.Text);

            string updateSql = "UPDATE Store SET ProductID = @NewValue1, Quantity
= @NewValue2 WHERE ProductID = @PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(updateSql,
connection))
                {
                    command.Parameters.AddWithValue("@NewValue1", proID);
                    command.Parameters.AddWithValue("@NewValue2", qty);
                    command.Parameters.AddWithValue("@PrimaryKeyValue",
oldProID);

                    command.ExecuteNonQuery();

                    MessageBox.Show("Updated Successfully");

                    txtPID.Clear();
                    txtQuantity.Clear();

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;

                    oldProID = "";
                }
                connection.Close();
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;

            string deleteSql = "DELETE FROM Store WHERE ProductID =
@PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(deleteSql,
connection))
                {
                    command.Parameters.AddWithValue("@PrimaryKeyValue", proID);

                    command.ExecuteNonQuery();

                    MessageBox.Show("Product Details Removed");

                    txtPID.Clear();
                    txtQuantity.Clear();

                    txtPID.Enabled = true;
                    txtQuantity.Enabled = true;
```

```
                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;
                }
                connection.Close();
            }
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            StoreKeeper storeKeeper = new StoreKeeper(userName);
            storeKeeper.ShowDialog();
            this.Close();
        }

        private void btnAdd_Click(object sender, EventArgs e)
        {
            proID = txtPID.Text;

            qty = int.Parse(txtQuantity.Text);

            string insertSql = "INSERT INTO Store (ProductID,Quantity) VALUES
(@Value1, @Value2)";

            string selectSql = "SELECT * FROM Product WHERE ProductID =
@SearchValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(selectSql,
connection))
                {
                    command.Parameters.AddWithValue("@SearchValue", proID);

                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        if (reader.HasRows)
                        {
                            reader.Close();

                            using(SqlCommand cmd = new SqlCommand(insertSql,
connection))
                            {
                                cmd.Parameters.AddWithValue("@Value1", proID);
                                cmd.Parameters.AddWithValue("@Value2", qty);

                                cmd.ExecuteNonQuery();
                            }

                            MessageBox.Show("Product Quantity Added");

                            txtPID.Clear();
                            txtQuantity.Clear();
                        }
                        else
                        {
                            MessageBox.Show("Entered Product ID is not
Available");
                        }
                    }
```

```
                }
                connection.Close();
            }
        }
    }
}
```

StockList.cs -->

```
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 SuperMarket
{
    public partial class StockList : Form
    {
        string userName;
        string role;
        public StockList(string userName,string role)
        {
            InitializeComponent();

            this.userName = userName;
            this.role = role;
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role,userName);
            this.Close();
        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
            stockList11.Clear();
            sqlDataAdapter1.Fill(stockList11);
        }
    }
}
```

Sales.cs -->

```
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 SuperMarket
{
    public partial class Sales : Form
    {
        string userName;
        string role = "Sales";

        public Sales(string user)
        {
            InitializeComponent();

            userName = user;

            lblUser.Text = userName;
        }

        private void btnLogOut_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.ShowDialog();
            this.Close();
        }

        private void btnDisReg_Click(object sender, EventArgs e)
        {
            this.Hide();
            DisReg disReg = new DisReg(userName,role);
            disReg.ShowDialog();
            this.Close();
        }

        private void btnDisView_Click(object sender, EventArgs e)
        {
            this.Hide();
            DisList disList = new DisList(userName, role);
            disList.ShowDialog();
            this.Close();
        }
    }
}
```

DisReg.cs -->

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

namespace SuperMarket
{
    public partial class DisReg : Form
    {
        string role, userName;

        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        string disID,oldDisID;

        float disPercentage;

        int minTotTrans, maxTotTrans;

        public DisReg(string userName,string role)
        {
            InitializeComponent();

            this.role = role;
            this.userName = userName;

            btnEdit.Visible = false;
            btnDelete.Visible = false;
            btnUpdate.Visible = false;
        }

        private void btnEdit_Click(object sender, EventArgs e)
        {
            btnUpdate.Visible = true;
            btnDelete.Visible = false;

            txtDisPer.Enabled = true;
            txtMaxTot.Enabled = true;
            txtMinTot.Enabled = true;

        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {
            disID = txtDisID.Text;
            disPercentage = float.Parse(txtDisPer.Text);
            minTotTrans = int.Parse(txtMinTot.Text);
            maxTotTrans = int.Parse(txtMaxTot.Text);
```

```
            string updateSql = "UPDATE DiscountDetails SET DiscountID =
@NewValue1, Percentage = @NewValue2 , MinTotTrans = @NewValue3 , MaxTotTrans =
@NewValue4  WHERE DiscountID = @PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(updateSql,
connection))
                {
                    command.Parameters.AddWithValue("@NewValue1", disID);
                    command.Parameters.AddWithValue("@NewValue2", disPercentage);
                    command.Parameters.AddWithValue("@NewValue3", minTotTrans);
                    command.Parameters.AddWithValue("@NewValue4", maxTotTrans);

                    command.Parameters.AddWithValue("@PrimaryKeyValue",
oldDisID);

                    command.ExecuteNonQuery();

                    MessageBox.Show("Updated Successfully");

                    txtDisID.Clear();
                    txtDisPer.Clear();
                    txtMinTot.Clear();
                    txtMaxTot.Clear();

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;

                    oldDisID = "";
                }
                connection.Close();
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            disID = txtDisID.Text;

            string deleteSql = "DELETE FROM DiscountDetails WHERE DiscountID =
@PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(deleteSql,
connection))
                {
                    command.Parameters.AddWithValue("@PrimaryKeyValue", disID);

                    command.ExecuteNonQuery();

                    MessageBox.Show("Discount Details Removed");

                    txtDisID.Clear();
                    txtDisPer.Clear();
                    txtMaxTot.Clear();
                    txtMinTot.Clear();
```

```
                    txtDisPer.Enabled = true;
                    txtMaxTot.Enabled = true;
                    txtMinTot.Enabled = true;

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;
                }
                connection.Close();
            }
        }

        private void btnSearch_Click(object sender, EventArgs e)
        {
            disID = txtDisID.Text;

            oldDisID = disID;

            if (disID == null)
            {
                MessageBox.Show("Please enter the Discount ID to Search");
            }
            else
            {
                string selectSql = "SELECT * FROM DiscountDetails WHERE
DiscountID = @SearchValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand(selectSql,
connection))
                    {
                        command.Parameters.AddWithValue("@SearchValue", disID);

                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            if (reader.HasRows)
                            {
                                while (reader.Read())
                                {
                                    txtDisID.Text =
reader["DiscountID"].ToString();
                                    txtDisPer.Text =
reader["Percentage"].ToString();
                                    txtMinTot.Text =
reader["MinTotTrans"].ToString();
                                    txtMaxTot.Text =
reader["MaxTotTrans"].ToString();

                                    MessageBox.Show("Discount Details
Available");

                                    btnEdit.Visible = true;
                                    btnDelete.Visible = true;

                                    txtDisPer.Enabled = false;
                                    txtMinTot.Enabled = false;
                                    txtMaxTot.Enabled = false;
                                }
                            }
                            else
```

```
                            {
                                MessageBox.Show("Entered Discount ID is not
Available");
                            }
                        }
                    }
                    connection.Close();
                }
            }
        }

        private void btnAdd_Click(object sender, EventArgs e)
        {
            disID = txtDisID.Text;
            disPercentage = float.Parse(txtDisPer.Text);
            minTotTrans = int.Parse(txtMinTot.Text);
            maxTotTrans = int.Parse(txtMaxTot.Text);

            string insertSql = "INSERT INTO DiscountDetails
(DiscountID,Percentage,MinTotTrans,MaxTotTrans) VALUES (@Value1,
@Value2,@Value3,@Value4)";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(insertSql,
connection))
                {
                    command.Parameters.AddWithValue("@Value1", disID);
                    command.Parameters.AddWithValue("@Value2", disPercentage);
                    command.Parameters.AddWithValue("@Value3", minTotTrans);
                    command.Parameters.AddWithValue("@Value4", maxTotTrans);


                    command.ExecuteNonQuery();
                }
                connection.Close();
            }

            MessageBox.Show("Discount Details Added Successfully");

            txtDisID.Clear();
            txtDisPer.Clear();
            txtMinTot.Clear();
            txtMaxTot.Clear();
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role, userName);
            this.Close();
        }
    }
}
```

DisList.cs -->

```
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 SuperMarket
{
    public partial class DisList : Form
    {
        string userName, role;
        public DisList(string userName, string role)
        {
            InitializeComponent();

            this.userName = userName;
            this.role = role;
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role, userName);
            this.Close();
        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
            discountList1.Clear();
            sqlDataAdapter1.Fill(discountList1);
        }
    }
}
```

Cashier.cs -->

```
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 SuperMarket
{
    public partial class Cashier : Form
    {
        string userName;

        public Cashier(string user)
        {
            InitializeComponent();

            userName = user;

            lblDisplay.Text = user;
        }

        private void btnLogOut_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.ShowDialog();
            this.Close();
        }

        private void btnCusReg_Click(object sender, EventArgs e)
        {
            this.Hide();
            CusReg cusReg = new CusReg(userName);
            cusReg.ShowDialog();
            this.Close();
        }

        private void btnSale_Click(object sender, EventArgs e)
        {
            this.Hide();
            POS pOS = new POS(userName);
            pOS.ShowDialog();
            this.Close();
        }
    }
}
```

CusReg.cs -->

```
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 static System.Windows.Forms.VisualStyles.VisualStyleElement.ListView;
using System.Net;

namespace SuperMarket
{

    public partial class CusReg : Form
    {
        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        string nic, name, contactNO, oldnic;

        string userName;

        float totTrans;

        private void btnEdit_Click(object sender, EventArgs e)
        {
            btnUpdate.Visible = true;
            btnDelete.Visible = false;

            txtName.Enabled = true;
            txtContact.Enabled = true;
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;
            name = txtName.Text;
            contactNO = txtContact.Text;

            string updateSql = "UPDATE Customer SET NIC = @NewValue1, Name =
@NewValue2 , ContactNo = @NewValue3  WHERE NIC = @PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(updateSql,
connection))
                {
                    command.Parameters.AddWithValue("@NewValue1", nic);
                    command.Parameters.AddWithValue("@NewValue2", name);
                    command.Parameters.AddWithValue("@NewValue3", contactNO);
                    command.Parameters.AddWithValue("@PrimaryKeyValue", oldnic);

                    command.ExecuteNonQuery();

                    MessageBox.Show("Updated Successfully");
```

```
                    txtNIC.Clear();
                    txtName.Clear();
                    txtContact.Clear();

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;

                    oldnic = "";
                }
                connection.Close();
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;

            string deleteSql = "DELETE FROM Customer WHERE NIC =
@PrimaryKeyValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(deleteSql,
connection))
                {
                    command.Parameters.AddWithValue("@PrimaryKeyValue", nic);

                    command.ExecuteNonQuery();

                    MessageBox.Show("Customer Details Removed");

                    txtNIC.Clear();
                    txtName.Clear();
                    txtContact.Clear();

                    txtName.Enabled = true;
                    txtContact.Enabled = true;

                    btnEdit.Visible = false;
                    btnUpdate.Visible = false;
                    btnDelete.Visible = false;
                }
                connection.Close();
            }
        }



        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Cashier cashier = new Cashier(userName);
            cashier.ShowDialog();
            this.Close();
        }

        private void btnSearch_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;
```

```
            oldnic = nic;

            if (nic == null)
            {
                MessageBox.Show("Please enter the NIC to Search");
            }
            else
            {
                string selectSql = "SELECT * FROM Customer WHERE NIC =
@SearchValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand(selectSql,
connection))
                    {
                        command.Parameters.AddWithValue("@SearchValue", nic);

                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            if (reader.HasRows)
                            {
                                while (reader.Read())
                                {
                                    txtNIC.Text = reader["NIC"].ToString();
                                    txtName.Text = reader["Name"].ToString();
                                    txtContact.Text =
reader["ContactNo"].ToString();

                                    MessageBox.Show("Customer Details
Available");

                                    btnEdit.Visible = true;
                                    btnDelete.Visible = true;

                                    txtName.Enabled = false;
                                    txtContact.Enabled = false;
                                }
                            }
                            else
                            {
                                MessageBox.Show("Entered NIC is not Available");
                            }
                        }
                    }
                    connection.Close();
                }
            }
        }

        public CusReg(string user)
        {
            InitializeComponent();

            this.userName = user;

            btnEdit.Visible = false;
            btnUpdate.Visible = false;
            btnDelete.Visible = false;
        }
```

```
        private void btnAdd_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;
            name = txtName.Text;
            contactNO = txtContact.Text;

            totTrans = 0;



            string insertSql = "INSERT INTO Customer
(NIC,Name,ContactNo,TotTrans) VALUES (@Value1, @Value2,@Value3,@Value4)";


            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(insertSql,
connection))
                {
                    command.Parameters.AddWithValue("@Value1", nic);
                    command.Parameters.AddWithValue("@Value2", name);
                    command.Parameters.AddWithValue("@Value3", contactNO);
                    command.Parameters.AddWithValue("@Value4", totTrans);

                    command.ExecuteNonQuery();
                }

                connection.Close();
            }

            MessageBox.Show("Customer Added Successfully");
            txtNIC.Clear();
            txtName.Clear();
            txtContact.Clear();
        }
    }
}
```

POS.cs -->

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Net.Http.Headers;
using System.Windows.Forms;

namespace SuperMarket
{

    public partial class POS : Form
    {
        string userName;
        string role = "Cashier";

        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";
```

```
        string nic, proId, proName, date, unitPrice;

        int qty;
        float newtot;
        float tempTran;

        private void btnAddtoCart_Click(object sender, EventArgs e)
        {
            proName = cmbProduct.SelectedItem.ToString();
            qty = int.Parse(txtQty.Text);

            string selectSql = "SELECT * FROM Product WHERE ProName =
@SearchValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(selectSql,
connection))
                {
                    command.Parameters.AddWithValue("@SearchValue", proName);

                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            proId = reader["ProductID"].ToString();
                            unitPrice = reader["Price"].ToString();

                            float uPrice = float.Parse(unitPrice);

                            float totPrice = qty * uPrice;

                            if (updateStock(proId, qty))
                            {
                                addToCart(proId, proName, uPrice, qty, totPrice);
                            }
                            else
                            {
                                MessageBox.Show("Low Stock");
                            }
                        }
                    }
                }
                connection.Close();
            }
        }

        private void addToCart(string ProductID, string ProductName, float
UnitPrice, int Quantity, float Total)
        {
            string insertSql = "INSERT INTO ShoppingCart
(ProductID,ProductName,UnitPrice,Quantity,Total) VALUES (@Value1,
@Value2,@Value3,@Value4,@Value5)";

            using (SqlConnection connection = new SqlConnection(conString))
            {
```

```
                connection.Open();

                using (SqlCommand command = new SqlCommand(insertSql,
connection))
                {
                    command.Parameters.AddWithValue("@Value1", ProductID);
                    command.Parameters.AddWithValue("@Value2", ProductName);
                    command.Parameters.AddWithValue("@Value3", UnitPrice);
                    command.Parameters.AddWithValue("@Value4", Quantity);
                    command.Parameters.AddWithValue("@Value5", Total);

                    command.ExecuteNonQuery();

                    txtQty.Clear();

                    MessageBox.Show("Added to the Cart");

                    sqlDataAdapter1.Fill(shopingDataList1);
                }
                connection.Close();
            }
        }

        private bool updateStock(string ProductID, int quantity)
        {
            int newqty;
            int oldqty;
            int dbqty;
            bool flag = true;

            string searchSql = "SELECT Quantity FROM Store WHERE ProductID =
@value";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(searchSql,
connection))
                {
                    command.Parameters.AddWithValue("@value", ProductID);

                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            dbqty = int.Parse(reader["Quantity"].ToString());

                            oldqty = dbqty;

                            newqty = dbqty - quantity;

                            if (newqty < 0)
                            {
                                newqty = oldqty;
                                flag = false;
                            }
                            else
```

```
                            {
                                update(newqty);
                                flag = true;
                            }
                        }
                    }
                }
                connection.Close();
            }

            if (flag)
            {
                return true;
            }
            else
            {
                return false;
            }

            void update(int qty)
            {
                using (SqlConnection connection = new SqlConnection(conString))
                {
                    string updateSql = "UPDATE Store SET Quantity = @NewValue1
WHERE ProductID = @PrimaryKeyValue";

                    connection.Open();
                    using (SqlCommand cmd = new SqlCommand(updateSql,
connection))
                    {
                        cmd.Parameters.AddWithValue("@NewValue1", qty);
                        cmd.Parameters.AddWithValue("@PrimaryKeyValue",
ProductID);

                        cmd.ExecuteNonQuery();
                    }
                    connection.Close();
                }
            }
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role, userName);
            this.Close();
        }

        private void btnClearBill_Click(object sender, EventArgs e)
        {

            reverseCart();

            clearCart();
            shopingDataList1.Clear();
            sqlDataAdapter1.Fill(shopingDataList1);
```

```
            MessageBox.Show("Bill Cleared");
        }


        private void reverseCart()
        {
            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                string selectSql = "SELECT ProductID,Quantity FROM ShoppingCart";

                using (SqlCommand command = new SqlCommand(selectSql,
connection))
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        if (reader.HasRows)
                        {
                            string proId;
                            int qty;

                            while (reader.Read())
                            {
                                proId = reader["ProductID"].ToString();
                                qty = int.Parse(reader["Quantity"].ToString());

                                reverseQty(proId, qty);
                            }
                        }
                        else
                        {
                            MessageBox.Show("No Transactions");
                        }
                    }
                }
                connection.Close();
            }

            void reverseQty(string proid, int qty)
            {
                string selectStockSql = "SELECT Quantity FROM Store WHERE
ProductID = @value";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using(SqlCommand command = new SqlCommand(selectStockSql,
connection))
                    {
                        command.Parameters.AddWithValue("@value", proid);

                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            int oldQty;
                            int newQty;
```

```
                            while(reader.Read())
                            {
                                oldQty =
int.Parse(reader["Quantity"].ToString());

                                newQty = oldQty + qty;

                                updateReverseStock(proid, newQty);
                            }
                        }
                    }
                    connection.Close();
                }
            }

            void updateReverseStock(string proid, int newqty)
            {
                string updateStockSql = "UPDATE Store SET Quantity = @NewValue1
WHERE ProductID = @PrimaryKeyValue";

                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open ();

                    using(SqlCommand command = new SqlCommand(updateStockSql,
connection))
                    {
                        command.Parameters.AddWithValue("@NewValue1", newqty);
                        command.Parameters.AddWithValue("@PrimaryKeyValue",
proid);

                        command.ExecuteNonQuery();
                    }
                    connection.Close ();
                }
            }
        }


        private void btnTotalBill_Click(object sender, EventArgs e)
        {
            float total;

            string totalSql = "SELECT SUM(Total) as Tot FROM ShoppingCart";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(totalSql, connection))
                {
                    using(SqlDataReader reader = command.ExecuteReader())
                    {
                        while(reader.Read())
                        {
                            string tot = reader["Tot"].ToString();

                            total = float.Parse(tot);
```

```
                            float newTot = checkDiscount(nic, total);

                            lblBillTot.Text = newTot.ToString();

                            updateCusTot(nic, newTot);

                            MessageBox.Show("Bill Finalized" +
Environment.NewLine + "The Total Bill is : " + newTot );

                            this.Hide();
                            POS pos = new POS(userName);
                            pos.ShowDialog();
                            this.Close();
                        }
                    }
                }
                connection.Close();
            }

        }

        private float checkDiscount(string nic, float tot)
        {
            this.newtot = tot;
            this.tempTran = tot;

            string selectCustrans = "SELECT TotTrans FROM CUSTOMER WHERE NIC =
@SearchValue";


            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(selectCustrans,
connection))
                {
                    command.Parameters.AddWithValue("@SearchValue", nic);

                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            float cusTrans =
float.Parse(reader["TotTrans"].ToString());

                            checkRange(cusTrans);
                        }
                    }
                }
                connection.Close();
            }
            return newtot; // Return the updated total after applying discounts
        }

        private void checkRange(float custrans)
        {
```

```
            string selectDis = "SELECT
DiscountID,Percentage,MinTotTrans,MaxTotTrans FROM DiscountDetails";
            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(selectDis,
connection))
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            string disID = reader["DiscountID"].ToString();
                            float percentage =
float.Parse(reader["Percentage"].ToString());
                            int minTran =
int.Parse(reader["MinTotTrans"].ToString());
                            int maxTran =
int.Parse(reader["MaxTotTrans"].ToString());

                            if (minTran <= custrans && custrans < maxTran)
                            {
                                DialogResult result = MessageBox.Show("Customer
is Eligible for Discount Code : " + disID + " of " + percentage + "%" +
Environment.NewLine + "Apply Discount ?", "Discount Available",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question);

                                if (result == DialogResult.OK)
                                {
                                    this.newtot = tempTran * ((100 - percentage)
/ 100);

                                    break;
                                }
                            }
                        }
                    }
                }
                connection.Close();
            }
        }

        private void updateCusTot(String NIC , float totalTrans)
        {
            float oldVal;
            float newVal;

            string updateSql = "UPDATE Customer SET TotTrans = @NewValue1 WHERE
NIC = @PrimaryKeyValue";

            string selectSql = "SELECT TotTrans FROM CUSTOMER WHERE NIC =
@SearchValue";

            using(SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open() ;

                using(SqlCommand command = new SqlCommand(selectSql, connection))
```

```
                {
                    command.Parameters.AddWithValue("@SearchValue", NIC);

                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            string dbVal = reader["TotTrans"].ToString();
                            oldVal = float.Parse(dbVal);

                            newVal = oldVal + totalTrans;

                            Update();
                        }
                    }
                }
                connection.Close() ;
            }

            void Update()
            {
                using (SqlConnection connection = new SqlConnection(conString))
                {
                    connection.Open();

                    using(SqlCommand command = new SqlCommand(updateSql,
connection))
                    {

                        command.Parameters.AddWithValue("@NewValue1", newVal);
                        command.Parameters.AddWithValue("@PrimaryKeyValue", NIC);

                        command.ExecuteNonQuery();
                    }
                    connection.Close();
                }
            }
        }

        public POS(string userName)
        {
            InitializeComponent();
            this.userName = userName;

            groupBox2.Enabled = false;
            groupBox3.Enabled = false;

        }

        private void addPro()
        {
            cmbProduct.Items.Clear();

            string selectSql = "SELECT ProName FROM Product";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();
```

```
                using (SqlCommand command = new SqlCommand(selectSql,
connection))
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            string productName = reader["ProName"].ToString();
                            cmbProduct.Items.Add(productName);
                        }
                    }
                }
                connection.Close();
            }
        }

        private void clearCart()
        {
            string deleteSql = "DELETE FROM ShoppingCart";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(deleteSql,
connection))
                {
                    command.ExecuteNonQuery();
                }
                connection.Close();
            }
        }

        private void btnCheck_Click(object sender, EventArgs e)
        {
            nic = txtNIC.Text;
            date = dateTimePicker.Text;


            string selectSql = "SELECT * FROM Customer WHERE NIC = @SearchValue";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(selectSql,
connection))
                {
                    command.Parameters.AddWithValue("@SearchValue", nic);

                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        if (reader.HasRows)
                        {
                            MessageBox.Show("Customer Verified");

                            lblCusId.Text = nic;
```

```
                            lblDate.Text = date;

                            groupBox2.Enabled = true;
                            groupBox3.Enabled = true;

                            addPro();

                            clearCart();
                        }
                        else
                        {
                            MessageBox.Show("Entered Customer Details is
unavailable " + Environment.NewLine + "Please add the Customer details to the
database");
                        }
                    }
                }
                connection.Close();
            }
        }
    }
}
```

Manager.cs -->

```
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 SuperMarket
{
    public partial class Manager : Form
    {
        string userName;
        string role = "Manager";
        public Manager(string user)
        {
            InitializeComponent();

            userName = user;

            lblDisplay.Text =  user;
        }

        private void btnLogOut_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.ShowDialog();
            this.Close();
        }

        private void btnEmpList_Click(object sender, EventArgs e)
        {
            this.Hide();
```

```
            EmpList empList = new EmpList(userName,role);
            empList.ShowDialog();
            this.Close();
        }

        private void btnInvList_Click(object sender, EventArgs e)
        {
            this.Hide();
            StockList stkList = new StockList(userName,role);
            stkList.ShowDialog();
            this.Close();
        }

        private void btnProList_Click(object sender, EventArgs e)
        {
            this.Hide();
            ProList proList = new ProList(userName, role);
            proList.ShowDialog();
            this.Close();
        }

        private void btnCusList_Click(object sender, EventArgs e)
        {
            this.Hide();
            CusList cusList = new CusList(userName,role);
            cusList.ShowDialog();
            this.Close();
        }

        private void btnReport_Click(object sender, EventArgs e)
        {
            this.Hide();
            Report report = new Report(userName, role);
            report.ShowDialog();
            this.Close();
        }
    }
}
```

CusList.cs -->

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

namespace SuperMarket
{
    public partial class CusList : Form
    {
        string userName;
        string role;
        public CusList(string userName,string role)
        {
```

```
            InitializeComponent();

            this.userName = userName;
            this.role = role;
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role,userName);
            this.Close();
        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
            cusDataList1.Clear();
            sqlDataAdapter1.Fill(cusDataList1);
        }
    }
}
```

Report.cs -->

```
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 SuperMarket
{
    public partial class Report : Form
    {
        string userName, role;

        string conString = "Data Source=HARINDU;Initial
Catalog=SuperMarket;Integrated Security=True";

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Hide();
            Login login = new Login();
            login.openWindow(role, userName);
            this.Close();
        }

        public Report(string userName,string role)
        {
            InitializeComponent();

            this.userName = userName;
            this.role = role;

            loadDetails();
        }
```

```
private void loadDetails()
{
    lblName.Text = userName;
    lblDate.Text = DateTime.Now.ToString();

    loadTotalSale();

    loadEmpCount();

    loadCusCount();

    loadProCount();

    loadDisCount();
}

private void loadTotalSale()
{
    string totalSql = "SELECT SUM(TotTrans) as TotalSale FROM Customer";

    using(SqlConnection connection = new SqlConnection(conString))
    {
        connection.Open();

        using (SqlCommand command = new SqlCommand(totalSql, connection))
        {
            using (SqlDataReader reader = command.ExecuteReader())
            {
                reader.Read();

                string totalSale = reader["TotalSale"].ToString();

                lblTotal.Text = "Rs." + totalSale ;

                reader.Close();
            }
        }
        connection.Close();
    }
}

private void loadEmpCount()
{
    string countSql = "SELECT Count(NIC) as EmpCount FROM Employee";

    using (SqlConnection connection = new SqlConnection(conString))
    {
        connection.Open();

        using (SqlCommand command = new SqlCommand(countSql, connection))
        {
            using (SqlDataReader reader = command.ExecuteReader())
            {
                reader.Read();

                string EmpCount = reader["EmpCount"].ToString();

                lblEmpCount.Text = EmpCount ;

                reader.Close();
            }
        }
        connection.Close();
```

```
            }
        }

        private void loadProCount()
        {
            string countSql = "SELECT Count(ProductID) as ProCount FROM Product";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(countSql, connection))
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        reader.Read();

                        string ProCount = reader["ProCount"].ToString();

                        lblProCount.Text = ProCount;

                        reader.Close();
                    }
                }
                connection.Close();
            }
        }

        private void loadCusCount()
        {
            string countSql = "SELECT Count(NIC) as CusCount FROM Customer";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(countSql, connection))
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        reader.Read();

                        string CusCount = reader["CusCount"].ToString();

                        lblCusCount.Text = CusCount;

                        reader.Close();
                    }
                }
                connection.Close();
            }
        }

        private void loadDisCount()
        {
            string countSql = "SELECT Count(DiscountID) as DisCount FROM
DiscountDetails";

            using (SqlConnection connection = new SqlConnection(conString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(countSql, connection))
```

```
                {
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        reader.Read();

                        string DisCount = reader["DisCount"].ToString();

                        lblDisCount.Text = DisCount;

                        reader.Close();
                    }
                }
                connection.Close();
            }
        }
    }
}
```

Tools used in design & development of the application.

- Visual Studio 2022
- C# programming language
- .NET framework
- MS SQL Server Management Studio
- Draw.io
- SequenceDiagram.org

Initial plan Vs. Actual plan

In the Initial plan we intend to develop a customer loyalty program and some promotions for products, but we didn't include it in the system.

In the initial plan we intend to develop a method to inform the Storekeeper when the stock is low but in the actual plan the storekeeper must check the product stock manually by looking at the stock list view.

In the Initial plan we intend to design a forget password portal but in the actual plan if the Employee forget his password the employee details need to be completely remove form the database by HR and re add the employee details again, after that employee need to sign up again.

Work contribution table

TASK	MEMBER
Login Page, HR team Pages	H C B Basnayake
Draw UML diagrams to the system	A A K N Amarathunga
Creating Cashier, Manager Pages	M R P S Ranawaka
Creating Sales, Storekeeper Pages	W T R De Silva

Future enhancements of the project

Inventory Forecasting: Implement advanced algorithms and machine learning models to predict future inventory needs more accurately, reducing overstocking and understocking issues.

Supplier Portal: Create a portal where suppliers can access real-time inventory data, place orders, and track deliveries. Streamline the supplier management process.

Integration with E-commerce Platforms: Integrate the system with popular e-commerce platforms to enable online shopping and synchronize online and in-store inventory.

Enhanced Loyalty Program: Expand the loyalty program with tiered rewards, personalized discounts, and a mobile app for customers to track their loyalty points and rewards.

Forget Password Portal: Develop a forget password reset method.

Customer Loyalty programs: Include customer loyalty programs and Product promotions.

Stock updates: Include a low stock update method for storekeeper.

Details of group members

- PS/2019/152 - H C B Basnayake - 0710451326
- PS/2019/089 - A A K N Amarathunga - 0768880703
- PS/2019/180 - M R P S Ranawaka - 0701187199
- PS/2019/240 - W T R De Silva - 0753618422