



**UNIVERSITY OF KELANIYA**

**DEPARTMENT OF STATISTICS & COMPUTER  
SCIENCE**

**ACADEMIC YEAR 2021/2022**

**FINAL PROJECT REPORT**

**POST OFFICE MANAGEMENT SYSTEM**

**GROUP 15**

## Table of Contents

### Contents

Table of Contents .....	2
1. INTRODUCTION .....	3
2. PROBLEM IDENTIFICATION .....	4
3. OBJECTIVES.....	5
4. FUNCTIONAL AND NON- FUNCTIONAL REQUIREMENTS .....	6
4.1 Functional Requirements.....	6
4.2 Non-Functional Requirements.....	7
5. DESIGN.....	8
6. IMPLEMENTATION .....	23
7. TOOLS .....	76
8. INTIAL PLAN VS ACTUAL PLAN .....	77
9. WORK CONTRIBUTION.....	78
10. FUTURE ENHANCEMENT OF THE PROJECT .....	79
11. DETAILS OF GROUP MEMBERS .....	79

## 1. INTRODUCTION

In an increasingly digital world, postal services remain an essential component of global communication and commerce. A Postal Management System (PMS) is a sophisticated software solution designed to streamline and enhance the operations of postal organizations. This system plays a pivotal role in modernizing and optimizing the traditional postal services, ensuring efficiency, accuracy, and customer satisfaction.

The primary goal of a Postal Management System is to facilitate the efficient handling, sorting, and delivery of mail and packages while also providing a range of services to meet the evolving needs of both individual and business customers. This comprehensive system integrates various functionalities, such as tracking and tracing, address validation, postage calculation, and customer relationship management.

Key features and benefits of a Postal Management System is are Mail Processing, Tracking and Tracing, Postage Calculation, Inventory Management.

In an era of digital transformation, a well-implemented Postal Management System is essential for postal organizations to remain competitive, responsive, and capable of meeting the evolving needs of customers and businesses. It serves as the backbone of modern postal services, combining technology, automation, and data management to deliver efficient, reliable, and customer-centric postal solutions.

## 2. PROBLEM IDENTIFICATION

The purpose of designing a Windows Application for a post office is to enhance and optimize the operations and services provided by the post office with modern technology. The application aims to address the specific needs and challenges faced by post offices, offering a digital solution that improves efficiency, user experience, customer experience, and overall effectiveness.

- Manage letters and packages details received by mail
  - Ordinary(normal) letters received by daily mail are not recorded by the office. The introduced Windows Application provides a feature to record the details of all letters and packages.
  - Details regarding the delivery of letters and packages are also allowed to be entered into the database. Thus, when a customer comes to the office and inquires about a letter from a customer, he can get the details of whether such a letter has been received by the office and if the letter has been delivered to the customer, then by whom and on what date, and if the letter is in the office, he can get the letter.
- Salary creation
  - In the current process, the creation of employee salaries is done manually. This application provides a salary calculator to make it more convenient and efficient.
- Manage letters and packages details received by customers
  - Currently, the information regarding the letters and packages that have been received from the customs office is noted manually on the sheets. This application made a way to enter the data into the system without noticing manually and sorted them according to the cities.
- Report generating
  - The report about all the cash transactions made at the end of the day is currently being mentioned manually in the cash sheet and with this a report about all the cash transactions made at the end of the day can be generated.

### 3. OBJECTIVES

- Automate the traditional manual processes carried out by postal services in order to reduce paperwork and errors.
- By providing digital solutions for tasks such as user registration, login, data entry, searching options and report generating.
- Enhance overall operational productivity by optimizing workflows and reducing processing time.
- Provide a user-friendly interface for the users.
- Ensure accurate and timely information about packages and letters status.
- Maintain accurate letter and package records, details, and transaction information within a secure database.
- At the end of the day, generating reports for regular money transactions.
- Providing customers with high-quality services in an efficient way.

## 4. FUNCTIONAL AND NON- FUNCTIONAL REQUIREMENTS

### 4.1 Functional Requirements

Functional requirements of this project outline the specific features and capabilities the system must possess to meet the needs of the postal organization and its customers. Here are some key functional requirements along with their descriptions:

#### **1. Tracking and Tracing**

- Customers and postal employees should be able to track the status of parcels and letters. Detailed information about delivered letters and parcels.

#### **2. Postage Calculation**

- The PMS should calculate accurate postage rates based on factors such as weight, size, destination, and service type. It should support various postage options, including express and international shipping.

#### **3. Reporting and Analytics**

- The PMS should generate reports on such as daily payments , letters to be mailed, stamp management and delivered letters. It should also provide data analytics capabilities for performance monitoring and decision-making.

#### **4. Financial Management**

- The PMS should record financial transactions related to postage sales, shipping fees, and other postal services. It should generate financial reports and support budgeting and auditing processes.

#### **5. User Authentication and Authorization**

- The system should have robust user authentication to ensure secure access. It should also implement role-based authorization to control user privileges and data access.

## 4.2 Non-Functional Requirements

---

In addition to the functional requirements, the development of the postal management system must also consider various non-functional requirements that define the system's performance, security, and usability aspects:

### **1. Security and Privacy:**

- Data Security: Different levels of permissions and privileges granted to users or entities to access and interact with the system's resources.

### **2. Performance:**

- Response Time: Ensure that the system responds quickly to user interactions data retrieval, and transaction processing.

### **3. Reliability:**

- Availability: Ensure high system availability, minimizing downtime and disruptions.

### **4. Usability:**

- User-Friendly Interfaces: Design intuitive and user-friendly interfaces for all types of users, including clear navigation and helpful error messages.

---

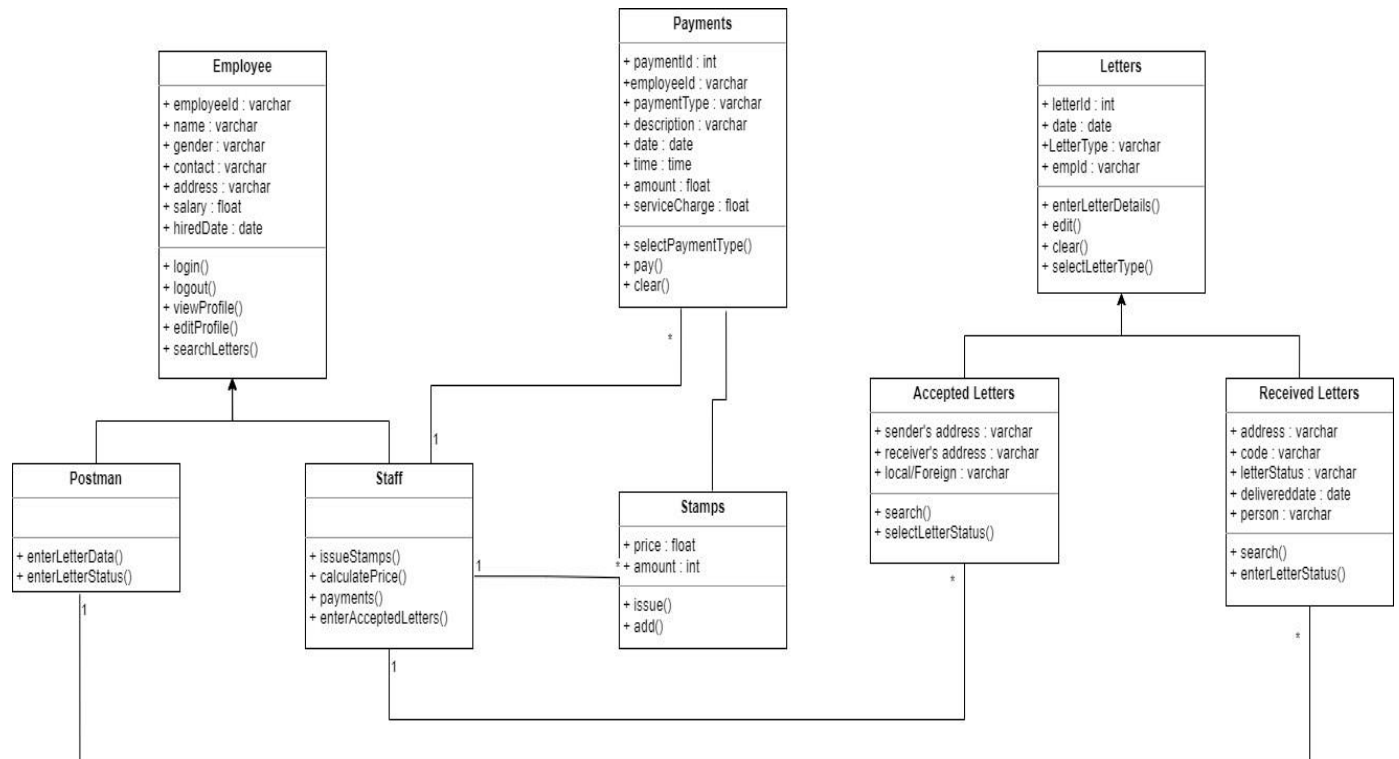
### **5. Maintainability:**

- Code Maintainability: Write readable and maintainable code to facilitate future updates and enhancements.

---

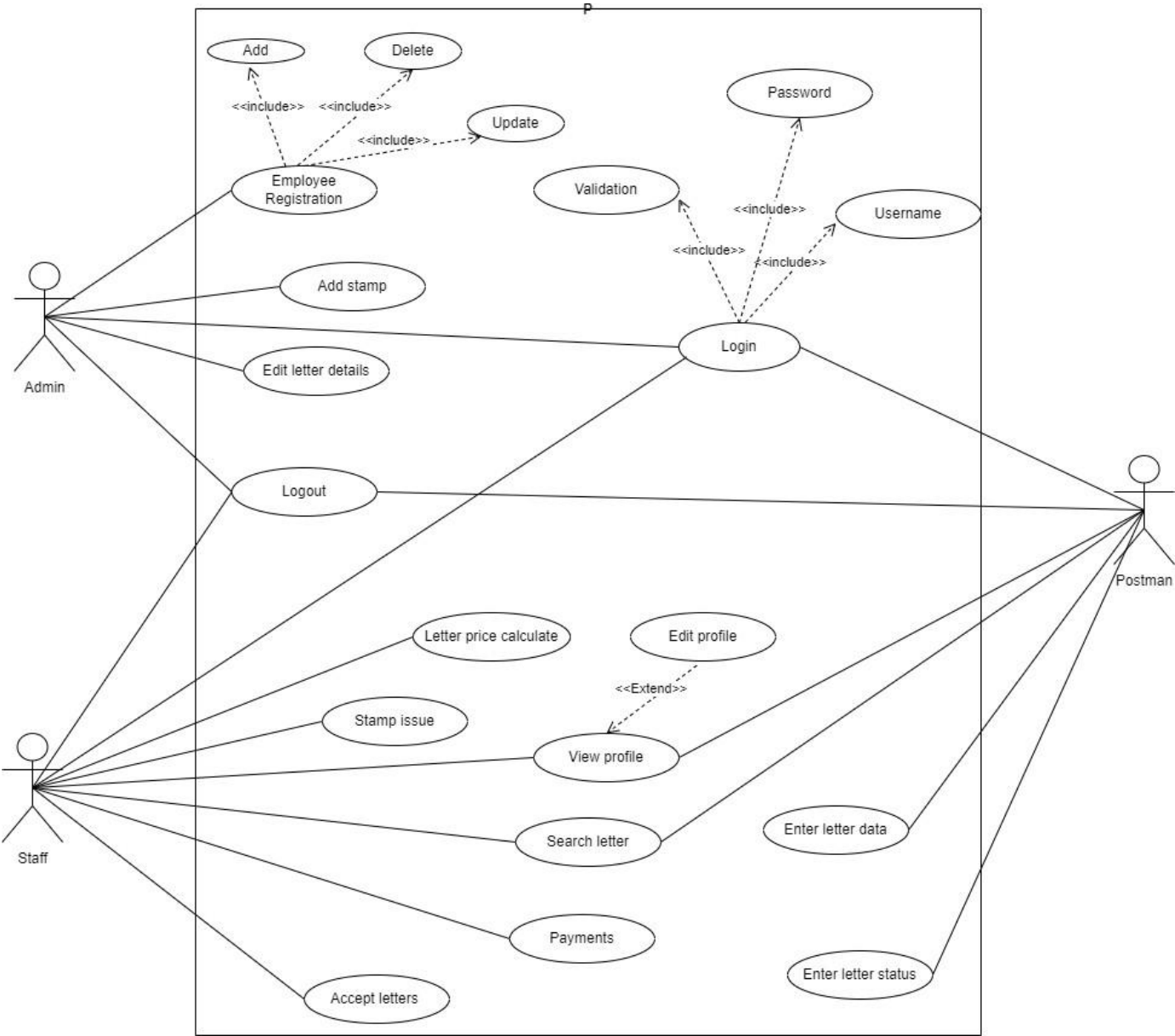
## 5. DESIGN

### Class Diagram

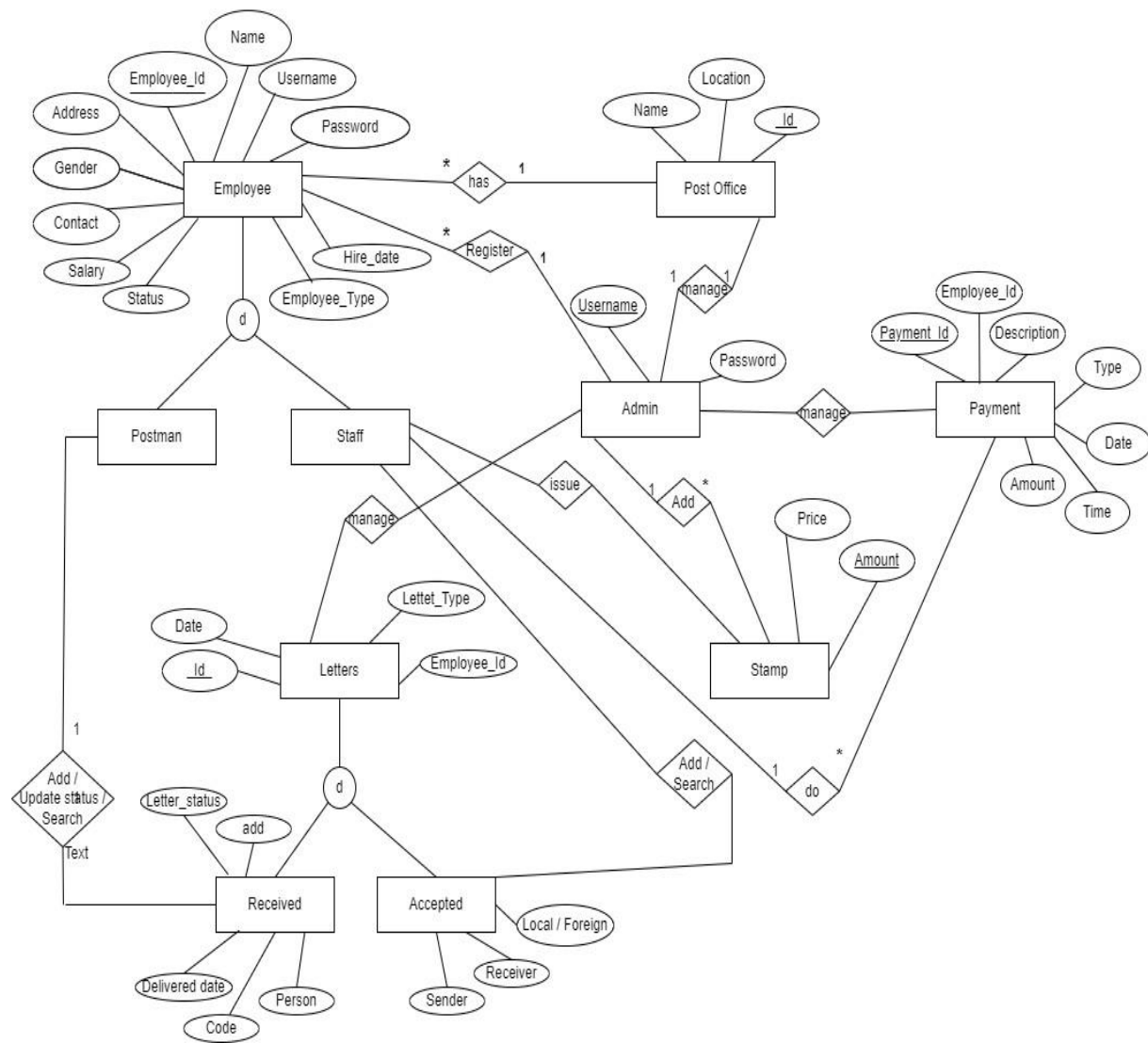




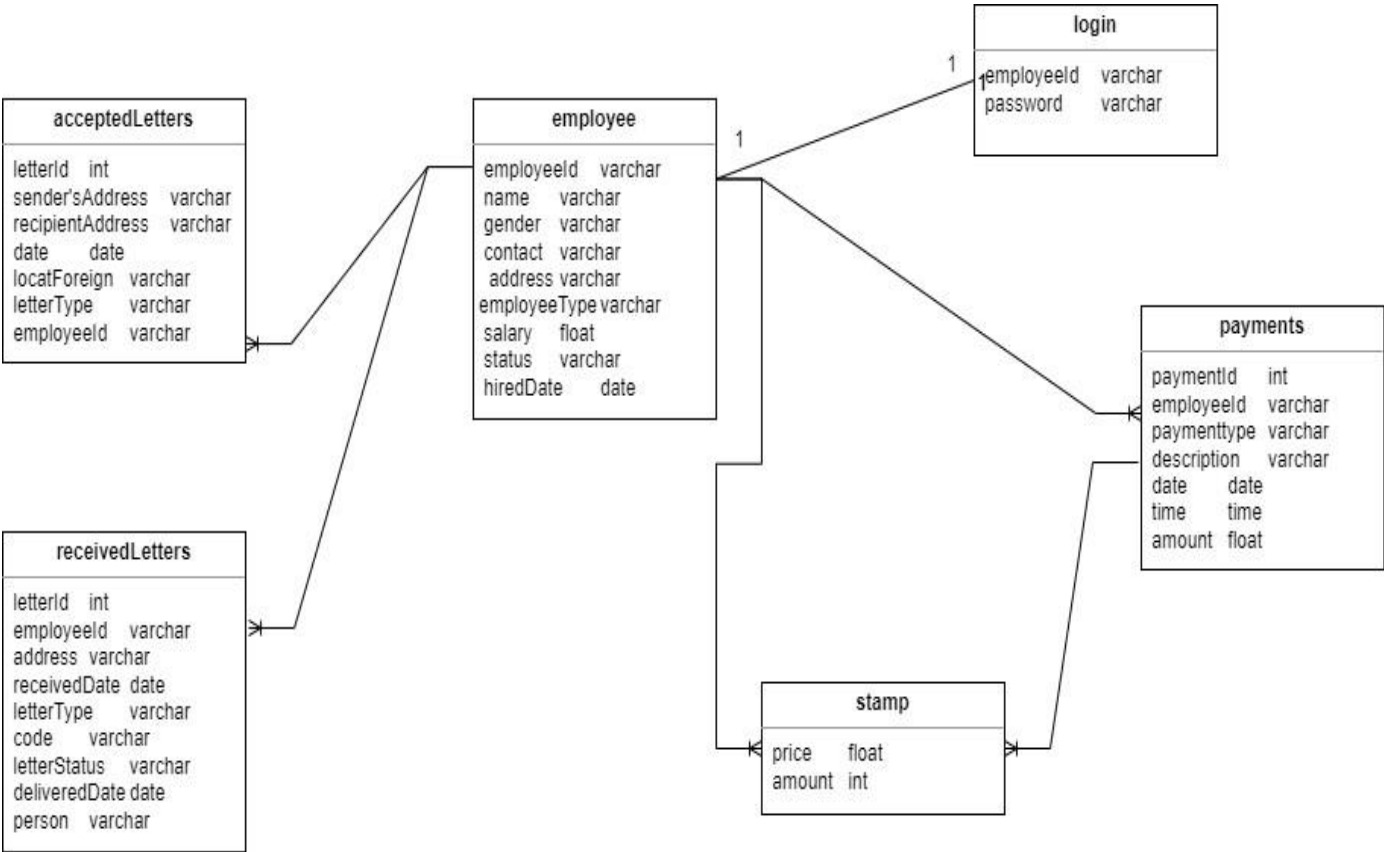
Use Case Diagram



## ER Diagram

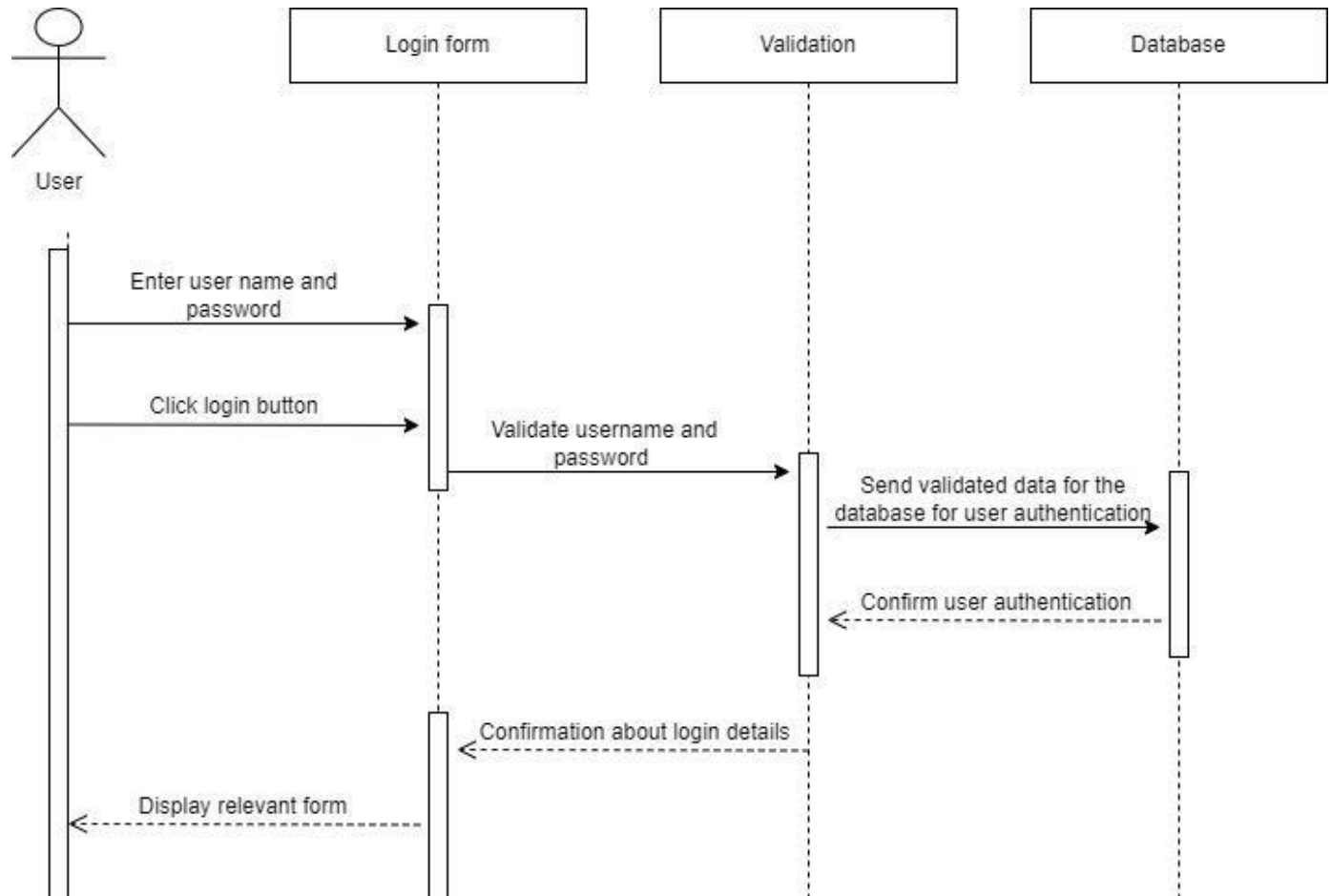


# Database Structure

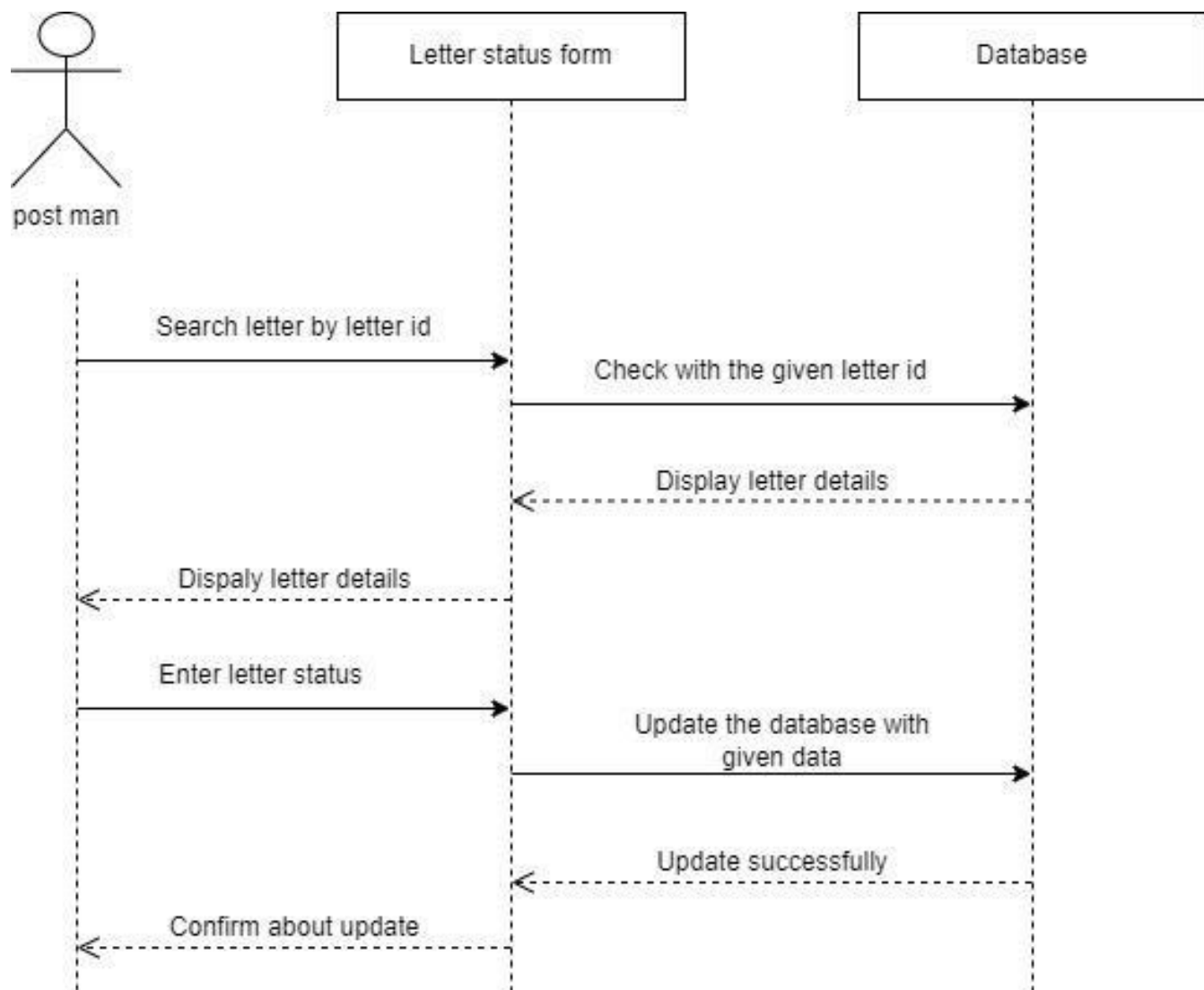


## Sequence Diagrams

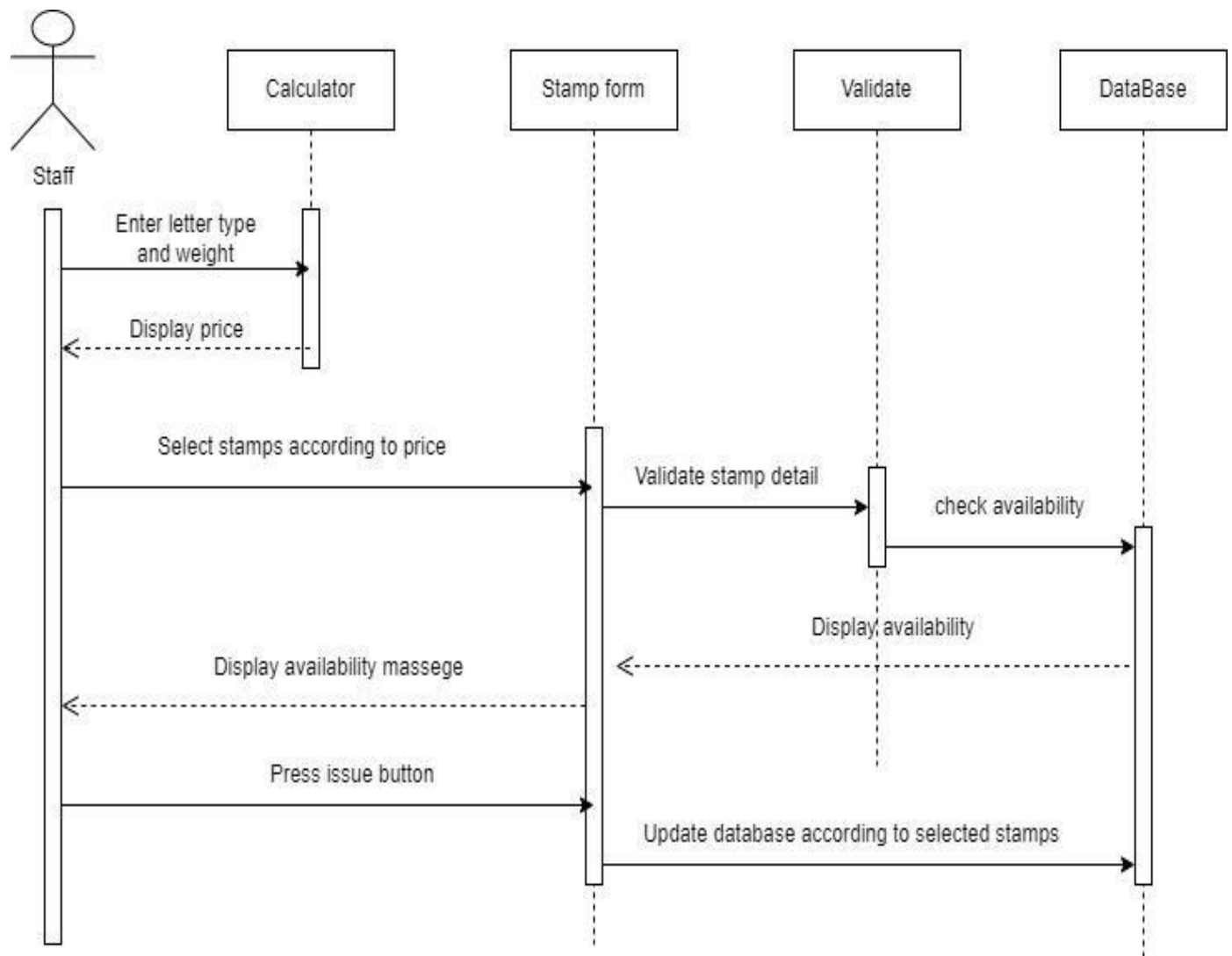
1.



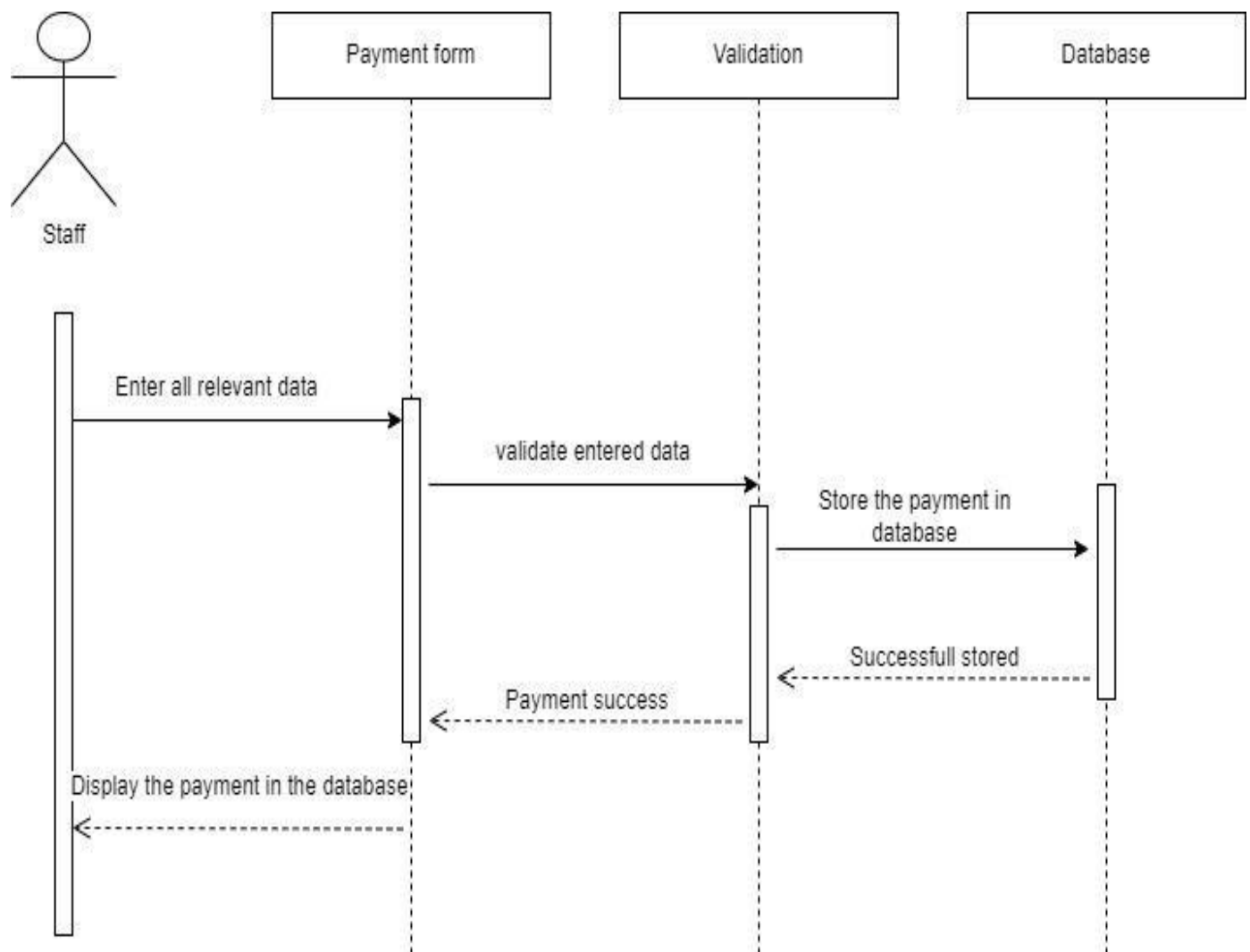
2.



3.



4.



## Login Form



The screenshot shows a web browser window titled "login". On the left is the Sri Lanka Post logo, which consists of a stylized "S" made of colorful segments (blue, green, yellow, red, purple) and a red envelope icon. Below the logo, the text "Sri Lanka POST" is displayed in a large, bold font, with "Sri Lanka" in blue and "POST" in red. Underneath, in Tamil, it says "ශ්‍රී ලංකා තැපෑලේ-இலங்கை அஞ்சல்". At the bottom left of the logo area, it says "Hot Line : 011 4597781". On the right side of the window, there is a login form. At the top right of the form area, there are labels for "Date" and "Time". Below these, the text "Welcome" and "Post Office Yakkala" is displayed. The form includes fields for "User Name" and "Password". Below the password field, there is a checkbox labeled "Show Password" and a link "Forget Password...". A "Login" button with a green arrow icon is present, along with "Clear" and "Exit" buttons.

login

Date Time

Welcome

Post Office Yakkala

User Name

Password

☐ Show Password

[Forget Password...](#)


Login

Clear Exit

Hot Line : 011 4597781

ශ්‍රී ලංකා තැපෑලේ-இலங்கை அஞ்சல்

## Postman Dashboard Form



The screenshot shows a web browser window titled "postmanDashboard". On the left is a blue sidebar with a user profile icon at the top. Below the icon, the text "empId" and "empName" is displayed. The sidebar contains several buttons: "Enter Letter Data", "Enter Letter Status", "Search Letter", "View Profile", and "Logout". The main content area on the right features the Sri Lanka Post logo, which is the same as in the login form. Below the logo, the text "Sri Lanka POST" is displayed in a large, bold font, with "Sri Lanka" in blue and "POST" in red. Underneath, in Tamil, it says "ශ්‍රී ලංකා තැපෑලේ-இலங்கை அஞ்சல்". At the bottom right of the main content area, it says "Hot Line : 011 4597781". At the top right of the main content area, there are labels for "Time" and "Date".

postmanDashboard

Time Date

empId

empName

Enter Letter Data

Enter Letter Status

Search Letter

View Profile

Logout

Hot Line : 011 4597781

ශ්‍රී ලංකා තැපෑලේ-இலங்கை அஞ்சல்

Sri Lanka POST




## Staff Dashboard Form


Time

Date


empId

empName







Stamp Issue




Payments




Letters and  
Parcels



Search Letter



View Profile



Employee  
Registration

Logout

Hot Line : 011 4597781

## Accepted Letters Form

Sender's Address

Recipient's Address

Local / Foreign

Letter Type

Insert

Clear

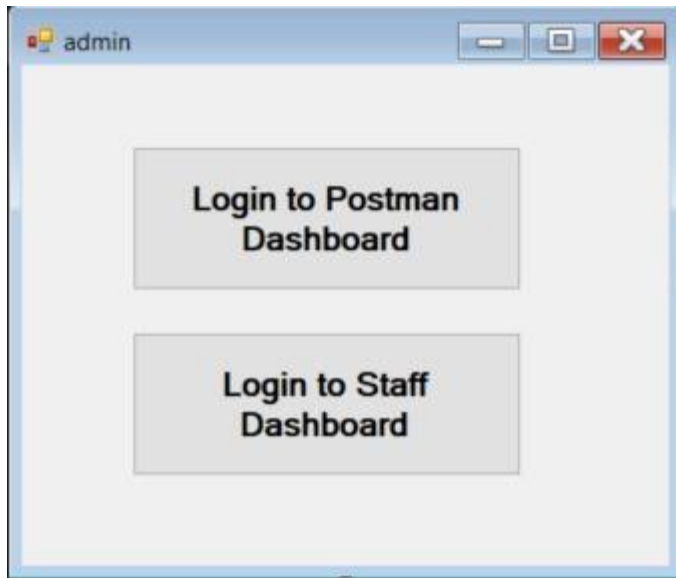
Search by Receiver's Address...

Enter Letter ID to Edit or Delete

Edit

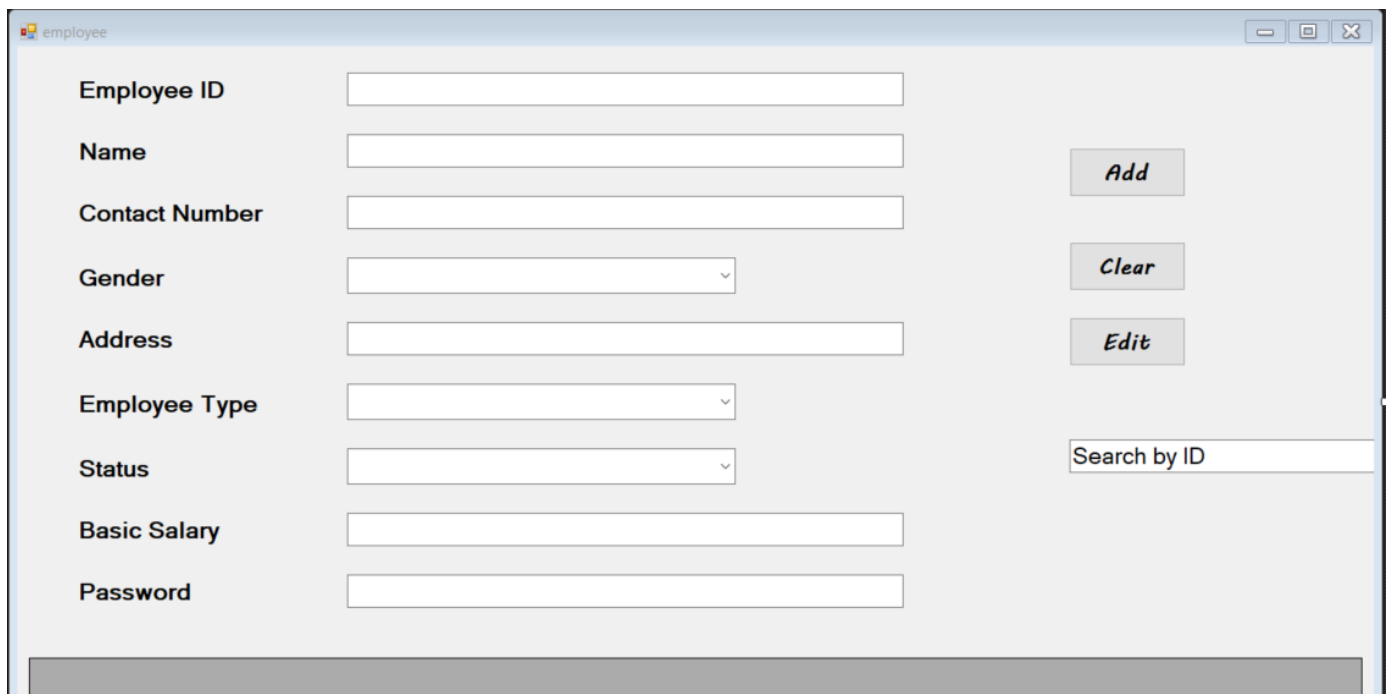
Delete

## Admin Form



A screenshot of a web application window titled "admin". The window has a light gray background and a blue border. It contains two large, light gray rectangular buttons stacked vertically. The top button is labeled "Login to Postman Dashboard" and the bottom button is labeled "Login to Staff Dashboard". Both buttons have a subtle drop shadow.

## Employee Form



A screenshot of a web application window titled "employee". The window has a light gray background and a blue border. It contains a form with several input fields and buttons. The form fields are arranged in a vertical list on the left, and the buttons are on the right. The fields are: "Employee ID", "Name", "Contact Number", "Gender" (a dropdown menu), "Address", "Employee Type" (a dropdown menu), "Status" (a dropdown menu), "Basic Salary", and "Password". The buttons are "Add", "Clear", and "Edit". There is also a search bar labeled "Search by ID" at the bottom right.

Field Label	Field Type
Employee ID	Text Input
Name	Text Input
Contact Number	Text Input
Gender	Dropdown Menu
Address	Text Input
Employee Type	Dropdown Menu
Status	Dropdown Menu
Basic Salary	Text Input
Password	Text Input

Buttons: Add, Clear, Edit

Search by ID

Letter Data Form

letterData

Letter ID

Address

Letter Type

Code

Clear

Insert

Edit

Letter Status Form

letterStatus

Letter ID

Search

Letter Status

▼

Name of the Person

Insert

Clear

# Payment Form

payment

Payment Type

Name

ID

Amount

Service Charge

Amount

Pay

Clear

mm/dd/yy

Print

printDocument1

printPreviewDialog1

# Profile Form

profile

Employee ID

label10

Name

label5

Gender

label2

Contact Number

label13

Address

label7

Employee Type

label8

Basic Salary

label14

Status

label4

Edit My Profile

Edit Name

Edit Contact Number

Edit Address

## Search Letters Form

The screenshot shows a Java Swing application window titled "searchLetter". The window's title bar includes standard minimize, maximize, and close buttons. The main content area is light gray and contains three primary elements: a text input field with the placeholder text "Enter Employee ID", a button labeled "Print", and another text input field with the placeholder text "Search By Address". Below these elements is a large, solid gray rectangular area that takes up the bottom half of the window. At the bottom of the screen, the Windows taskbar is visible, showing two active application icons: "printDocument1" and "printPreviewDialog1".

## Stamp Issue Form

stamplssue

0.50

1

2

3

5

10

15

20

25

30

35

40

50

60

75

100

110

500

1000

Add Stamps

Price

Issue

Enter Letter Type

Enter Weight (g)

Price

Clear

Calculate

## 6. IMPLEMENTATION

### Login



login

10/1/2023 4:13 PM

Welcome  
Post Office Yakkala

User Name

Password

☐ Show Password  
[Forget Password...](#)

Login

Clear Exit

Hot Line : 011 4597781

ශ්‍රී ලංකා තැපෑල

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

namespace PostOfficeManagement
{
    public partial class login : Form
    {
        public static string user;
```

```

public login()
{
    InitializeComponent();
}

SqlConnection conn = new SqlConnection(@"Data Source=LafitOff-
NLDBQLGG\SQL EXPRESS;Initial Catalog=flostOffice;Integrated Security=True");

private void login_Load(object sender, EventArgs e)
{
    txtUsername.Focus();

    timer1.Start();
    lblDate.Text = DateTime.Now.ToShortDateString();
    lblTime.Text = DateTime.Now.ToShortTimeString();

}

private void btnLogin_Click(object sender, EventArgs e)
{
    if (txtUsername.Text == "")
    {
        MessageBox.Show("Please enter username", "Log in Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtUsername.Focus();
        return;
    }

    if (txtPassword.Text == "")
    {
        MessageBox.Show("Please enter password", "Log in Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtPassword.Focus();
        return;
    }

    string username, password;

    try
    {
        string query = "SELECT * FROM login WHERE employeeId = '" +
        txtUsername.Text + "' AND password = '" + txtPassword.Text + "'";
        SqlDataAdapter adapter = new SqlDataAdapter(query, conn);

        DataTable dt = new DataTable();
        adapter.Fill(dt);

        if (dt.Rows.Count > 0)
        {

```



```

        username = txtUsername.Text;
        password = txtflassword.Text;
        user = username;

        if (username.Contains("emp"))
        {
            postmanDashboard postmanDashboard = new postmanDashboard();
            postmanDashboard.Show();
            this.Hide();
        }
        else if (username.Contains("stf"))
        {
            staffDashboard staffDashboard = new staffDashboard();
            staffDashboard.Show();
            this.Hide();
        }
        else if (username == "admin")
        {
            admin admin = new admin();
            admin.Show();
            this.Hide();
        }
    }
    else
    {
        MessageBox.Show("Invalid Username or flassword", "Log in Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}
catch
{
    MessageBox.Show("Error");
}
finally
{
    conn.Close();
}

}

private void label5_MouseHover(object sender, EventArgs e)
{
    label5.ForeColor = Color.Blue;
}

private void label5_MouseLeave(object sender, EventArgs e)
{
    label5.ForeColor = Color.Black;
}

```

```

private void btnClear_Click(object sender, EventArgs e)
{
    txtUsername.Clear();
    txtUsername.Focus();
    txtflassword.Clear();
}

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

private void pictureBox1_Click(object sender, EventArgs e)
{
}

private void label7_Click(object sender, EventArgs e)
{
}

private void timer1_Tick(object sender, EventArgs e)
{
    lblTime.Text = DateTime.Now.ToShortTimeString();
    timer1.Start();
}

private void label5_Click(object sender, EventArgs e)
{
    MessageBox.Show("Contact the head of the institution", "Login Information",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}

private void checkBoxShow_CheckedChanged(object sender, EventArgs e)
{
    if (checkBoxShow.Checked == true)
    {
        txtflassword.UseSystemflasswordChar = false;
    }
    else
    {
        txtflassword.UseSystemflasswordChar = true;
    }
}
}
}
}

```

## Postman Dashboard



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

```
namespace flostOfficeManagement
```

```
{
    public partial class postmanDashboard : Form
    {
```

```
        public postmanDashboard()
        {
            InitializeComponent();
        }
```

```
        SqlConnection conn = new SqlConnection(@"Data Source=LAfitOfi-
NLDBQLGG\SQLExpress;Initial Catalog=flostOffice;Integrated Security=True");
```

```
        private void label2_Click(object sender, EventArgs e)
```

```

{
}

private void postmanDashboard_Load(object sender, EventArgs e)
{
    if (login.user == "admin")
    {
        lblEmpName.Hide();
        btnViewprofile.Hide();
    }

    btnLetterData.Focus();
    lblEmpId.Text = login.user;

    SqlCommand cmd = new SqlCommand("SELECT [name] FROM [dbo].[employee] WHERE
[employeeid] = ''' + login.user + '''", conn);
    conn.Open();

    SqlDataReader myR = cmd.ExecuteReader();
    if (myR.HasRows)
    {
        while (myR.Read())
        {
            lblEmpName.Text = myR[0].ToString();
        }
    }
    conn.Close();

    timer1.Start();
    lblDate.Text = DateTime.Now.ToShortDateString();
    lblTime.Text = DateTime.Now.ToShortTimeString();
}

private void timer1_Tick(object sender, EventArgs e)
{
    lblTime.Text = DateTime.Now.ToShortTimeString();
    timer1.Start();
}

private void btnViewprofile_Click(object sender, EventArgs e)
{
    profile profile = new profile();

    profile.TopLevel = false;
    pnlContent.Controls.Add(profile);
    profile.BringToFront();
    profile.Show();
}

private void btnSearch_Click(object sender, EventArgs e)
{

```

```

        searchLetter letterStatus = new searchLetter();

        letterStatus.TopLevel = false;
        pnlContent.Controls.Add(letterStatus);
        letterStatus.BringToFront();
        letterStatus.Show();
    }

    private void btnLetterStatus_Click(object sender, EventArgs e)
    {
        letterStatus letterStatus = new letterStatus();

        letterStatus.TopLevel = false;
        pnlContent.Controls.Add(letterStatus);
        letterStatus.BringToFront();
        letterStatus.Show();
    }

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

    private void btnLetterData_Click(object sender, EventArgs e)
    {
        letterData letterData = new letterData();

        letterData.TopLevel = false;
        pnlContent.Controls.Add(letterData);
        letterData.BringToFront();
        letterData.Show();
    }

    private void panel2_fla int(object sender, fl a intEventArgs e)
    {
    }

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

    private void btnLetterData_MouseHover(object sender, EventArgs e)
    {
        btnLetterData.BackColor = Color.AliceBlue;
    }

    private void pictureBox1_Click(object sender, EventArgs e)
    {
    }

```

```

private void btnLetterData_MouseHover_1(object sender, EventArgs e)
{
    btnLetterData.BackColor = Color.AliceBlue;
}

private void btnLetterStatus_MouseHover(object sender, EventArgs e)
{
    btnLetterStatus.BackColor = Color.AliceBlue;
}

private void btnSearch_MouseHover(object sender, EventArgs e)
{
    btnSearch.BackColor = Color.AliceBlue;
}

private void btnViewflrofile_MouseHover(object sender, EventArgs e)
{
    btnViewflrofile.BackColor = Color.AliceBlue;
}

private void btnLogout_MouseHover(object sender, EventArgs e)
{
    btnLogout.BackColor = Color.AliceBlue;
}

private void btnLetterData_MouseLeave(object sender, EventArgs e)
{
    btnLetterData.BackColor = Color.Beige;
}

private void btnLetterStatus_MouseLeave(object sender, EventArgs e)
{
    btnLetterStatus.BackColor = Color.Beige;
}

private void btnSearch_MouseLeave(object sender, EventArgs e)
{
    btnSearch.BackColor = Color.Beige;
}

private void btnViewflrofile_MouseLeave(object sender, EventArgs e)
{
    btnViewflrofile.BackColor = Color.Beige;
}

private void btnLogout_MouseLeave(object sender, EventArgs e)
{
    btnLogout.BackColor = Color.Beige;
}

private void btnLogout_Click_1(object sender, EventArgs e)
{
    login login = new login();

```

```
this.Hide();
login.Show();
```

```
}
}
}
```

## Enter Letter Data

postmanDashboard

4:16 PM 10/1/2023

letterData

emp001  
sanjeewa

Enter Letter Data

Enter Letter Status

Search Letter

View Profile

Logout

Address

Letter Type  
Normal

Clear Insert

letterId	employeeId	address	receivedDate	letterType	code
1	1	kaluthara	fgfg	Normal	yuty
2	56	rtyr	uujh.	ghmhrfhy	ytuykhj
3	label1	sad	9/22/2023	Normal	sds
4	label1	sad	9/22/2023	Normal	sds
5	label1	sad	9/22/2023	Normal	sds
8		idigolla gampaha			
9		105/A	9/30/2023	Normal	None
10	emp001	105	9/30/2023	Normal	None
11	emp001	kandy	9/30/2023	Normal	None
12	emp001	colombo	9/30/2023	Register	sdf
14	emp001	105/A, Ambagaspiya, ...	9/30/2023	Parcel	1550
6	1	ambagaspiya gampaha			
7		yakkala gampaha			
13	emp001	fff	9/30/2023	Newspaper	None

Hot Line : 011 4597781

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

```
namespace flostOfficeManagement
{
    public partial class letterData : Form
    {
```

```

public LetterData()
{
    InitializeComponent();

    SqlConnection conn = new SqlConnection(@"Data Source=LafitOff-
NLDBQLGG\SQLSERVER;Initial Catalog=flostOffice;Integrated Security=True");

    private void LetterData_Load(object sender, EventArgs e)
    {
        if (login.user.Contains("emp"))
        {
            lblLetterId.Hide();
            btnEdit.Hide();
        }

        cmbLetterType.SelectedIndex = 0;
        String date = DateTime.Now.ToShortDateString();

        getLetterDetails();
    }

    private void getLetterDetails()
    {
        SqlCommand cmd = new SqlCommand("SELECT letterId, employeeId, address,
receivedDate, letterType, code FROM receivedLetters", conn);
        DataTable dt = new DataTable();

        conn.Open();

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewLetterData.DataSource = dt;
    }

    private void btnInsert_Click(object sender, EventArgs e)
    {
        if (txtAddress.Text == "")
        {
            MessageBox.Show("Enter letter address", "Letter Date Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtAddress.Focus();
            return;
        }

        if (cmbLetterType.SelectedIndex != 0 || cmbLetterType.SelectedIndex != 4 ||
cmbLetterType.SelectedIndex != 5)

```



```

    {
        if (txtCode.Text == "")
        {
            MessageBox.Show("Enter letter code", "Letter Date Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtCode.Focus();
            return;
        }
    }

    SqlCommand cmd = new SqlCommand(@"INSERT INTO
[dbo].[receivedLetters]([employeeId], [address], [receivedDate], [letterType], [code],
[letterStatus],[deliveredDate], [person]) VALUES ('" + logIn.user + "', '" +
txtAddress.Text + "', '" + DateTime.Now.ToShortDateString() + "', '" +
cmbLetterType.SelectedItem.ToString() + "', '" + txtCode.Text + "', '" + "On fireman's" +
"', '" + "Not Delivered Yet" + "', '" + "Not Delivered Yet" + "')", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    getLetterDetails();

    txtCode.Clear();
    txtAddress.Clear();
    cmbLetterType.SelectedIndex = 0;
    txtAddress.Focus();

}

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

private void cmbLetterType_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (cmbLetterType.SelectedItem.ToString())
    {
        case "Normal" :
        case "Open float":
        case "Newspaper":
            txtCode.Text = "None";
            txtCode.Hide();
            lblCode.Hide();
            break;

        case "Register":
        case "parcel":
        case "Speed float":
            txtCode.Text = "";
            txtCode.Show();
    }
}

```

```

        lblCode.Show();
        break;
    }
}

private void lbl1_Click(object sender, EventArgs e)
{
}

private void lbl2_Click(object sender, EventArgs e)
{
}

private void btnClear_Click(object sender, EventArgs e)
{
    txtCode.Clear();
    txtAddress.Clear();
    cmbLetterType.SelectedIndex = 0;
    txtAddress.Focus();
}

private void dataGridViewLetterData_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    if (e.RowIndex >= 0)
    {
        DataGridViewRow row = dataGridViewLetterData.Rows[e.RowIndex];

        txtAddress.Text = row.Cells["address"].Value.ToString();
        cmbLetterType.SelectedItem = row.Cells["letterType"].Value.ToString();
        txtCode.Text = row.Cells["code"].Value.ToString();
        lblLetterId.Text = row.Cells["letterId"].Value.ToString();
    }
}

private void btnEdit_Click(object sender, EventArgs e)
{
    SqlCommand cmd = new SqlCommand(@"UPDATE [dbo].[receivedLetters] SET
[address] = '"+txtAddress.Text+"', [letterType] =
 '"+cmbLetterType.SelectedItem.ToString()+"', [code] = '"+txtCode.Text+"' WHERE
[letterId] = '"+lblLetterId.Text+"'", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
    getLetterDetails();
}
}
}

```

---

## Enter Letter Status

postmanDashboard

4:18 PM 10/1/2023

letterStatus

emp001  
sanjeewa

Enter Letter Data

Enter Letter Status

Search Letter

View Profile

Logout

Letter ID

10/1/2023

Delivered

Name of the Person

Insert Clear

letterId	employeeId	address	receivedDate	letterType	code	letterStatus	deliveredDate	person
1	1	kaluthara	fgfg	Normal	yuty	Delivered	9/24/2023	dinethra
2	56	rtjr	uujh.	ghmhrfny	ytuykhj	hmjhj	yukuk	hm,hjm,
3	label1	sad	9/22/2023	Normal	sds	On Premise	Not Delivered ...	Not Delivered ...
4	label1	sad	9/22/2023	Normal	sds	On Premise	Not Delivered ...	Not Delivered ...
5	label1	sad	9/22/2023	Normal	sds	On Premise	Not Delivered ...	Not Delivered ...
8		idigolla gampa...						
9		105/A	9/30/2023	Normal	None	On Premise	Not Delivered ...	Not Delivered ...
10	emp001	105	9/30/2023	Normal	None	Delivered	9/30/2023	ganepola
11	emp001	kandy	9/30/2023	Normal	None	On Premise	Not Delivered ...	Not Delivered ...
12	emp001	colombo	9/30/2023	Register	sdf	On Premise	Not Delivered ...	Not Delivered ...
14	emp001	105/A, Ambag...	9/30/2023	Parcel	1550	On Premise	Not Delivered ...	Not Delivered ...
6	1	ambagasptiya...						
7		yakkala gamp...						
13	emp001	fff	9/30/2023	Newspaper	None	On Premise	Not Delivered ...	Not Delivered ...
*								

Hot Line : 011 4597781

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Runtime.CompilerServices.RuntimeHelpers;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace flostopOfficeManagement
{
    public partial class letterStatus : Form
    {
        public letterStatus()
        {
            InitializeComponent();

            SqlConnection conn = new SqlConnection(@"Data Source=LafitOf-
NLDBQLGG\SQLSERVER;Initial Catalog=flostopOffice;Integrated Security=True");

            private void letterStatus_Load(object sender, EventArgs e)
            {

```

```

        cmbStatus.SelectedIndex = 0;
        lblDate.Text = DateTime.Now.ToShortDateString();

        getLetterDetails();
    }

    private void getLetterDetails()
    {
        SqlCommand cmd = new SqlCommand("SELECT letterId, employeeId, address,
receivedDate, letterType, code, letterStatus, deliveredDate, person FROM
receivedLetters", conn);
        DataTable dt = new DataTable();

        conn.Open();

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewLetterStatus.DataSource = dt;
    }

    private void btnInsert_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(txtLetterId.Text))
        {
            MessageBox.Show("Please enter a value in the Letter ID field.",
"Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtLetterId.Focus();
            return;
        }

        if (string.IsNullOrEmpty(txtflerson.Text))
        {
            MessageBox.Show("Please enter a value in the Name of the flerson field.",
"Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtflerson.Focus();
            return;
        }

        SqlCommand cmd = new SqlCommand(@"UPDATE [dbo].[receivedLetters] SET
[letterStatus] = ''+cmbStatus.SelectedItem.ToString()+'', [deliveredDate] = ''+
DateTime.Now.ToShortDateString() + '', [person] = ''+txtflerson.Text+'' WHERE letterId =
''+txtLetterId.Text+''", conn);
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        getLetterDetails();

        txtLetterId.Clear();
        txtflerson.Clear();
        cmbStatus.SelectedIndex = 0;
    }

```

```

        txtLetterId.Focus();
    }

    private void button1_Click(object sender, EventArgs e)
    {

    }

    private void btnSearch_Click(object sender, EventArgs e)
    {
        SqlCommand cmd = new SqlCommand("SELECT letterId, employeeId, address,
receivedDate, letterType, code, letterStatus, deliveredDate, person FROM receivedLetters
WHERE letterId = '" + txtLetterId.Text + "'", conn);
        DataTable dt = new DataTable();

        conn.Open();

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewLetterStatus.DataSource = dt;
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        txtLetterId.Clear();
        txtPerson.Clear();
        cmbStatus.SelectedIndex = 0;
        txtLetterId.Focus();
    }
}
}
}

```

## Search Letter

postmanDashboard

4:18 PM 10/1/2023

searchLetter

Enter Employee ID

Search By Address

emp001  
sanjeewa

Enter Letter Data

Enter Letter Status

Search Letter

View Profile

Logout

letterId	employeeId	address	receivedDate	letterType	code	letterStatus	deliveredDate	person
1	1	kaluthara	fgfg	Normal	yuty	Delivered	9/24/2023	dinethra
2	56	rtyr	uujh.	ghmhfthy	ytuykhj	hmghj	yukuk	hm.hjm.
3	label1	sad	9/22/2023	Normal	sds	On Premise	Not Delivered Yet	Not Delivered Yet
4	label1	sad	9/22/2023	Normal	sds	On Premise	Not Delivered Yet	Not Delivered Yet
5	label1	sad	9/22/2023	Normal	sds	On Premise	Not Delivered Yet	Not Delivered Yet
8		idigolla gampaha						
9		105/A	9/30/2023	Normal	None	On Premise	Not Delivered Yet	Not Delivered Yet
10	emp001	105	9/30/2023	Normal	None	Delivered	9/30/2023	ganepola
11	emp001	kandy	9/30/2023	Normal	None	On Premise	Not Delivered Yet	Not Delivered Yet
12	emp001	colombo	9/30/2023	Register	sdf	On Premise	Not Delivered Yet	Not Delivered Yet
14	emp001	105/A, Ambagaspitiya...	9/30/2023	Parcel	1550	On Premise	Not Delivered Yet	Not Delivered Yet
6	1	ambagaspitiya gampa...						
7		yakkala gampaha						
13	emp001	fff	9/30/2023	Newspaper	None	On Premise	Not Delivered Yet	Not Delivered Yet
*								

Hot Line : 011 4597781

```

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

namespace flostOfficeManagement
{
    public partial class searchLetter : Form
    {
        public searchLetter()
        {
            InitializeComponent();
        }

        SqlConnection conn = new SqlConnection(@"Data Source=LafitOf-
NLDBQLGG\SQLSERVER;Initial Catalog=flostOffice;Integrated Security=True");

        private void btnSearch_Click(object sender, EventArgs e)
        {

```

```

    }

    private void searchLetter_Load(object sender, EventArgs e)
    {
        if (login.user.Contains("stf"))
        {
            textBox1.Hide();
            btnflrint.Hide();
        }

        getLetterDetails();
    }

    private void getLetterDetails()
    {
        SqlCommand cmd = new SqlCommand("SELECT letterId, employeeId, address,
receivedDate, letterType, code, letterStatus, deliveredDate, person FROM
receivedLetters", conn);
        DataTable dt = new DataTable();

        conn.Open();

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewSearchLetter.DataSource = dt;
    }

    private void txtSearch_Keyfress(object sender, KeyfressEventArgs e)
    {
        SqlCommand cmd = new SqlCommand("SELECT letterId, employeeId, address,
receivedDate, letterType, code, letterStatus, deliveredDate, person FROM rece ivedLetters
WHERE address LIKE '%" + txtSearch.Text + "%';", conn);
        DataTable dt = new DataTable();
        DataView dv = dt.DefaultView;

        conn.Open();

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewSearchLetter.DataSource = dt;
    }

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

    private void txtSearch_Click(object sender, EventArgs e)
    {

```

```

        txtSearch.Text = "";
    }

    private void searchLetter_Click(object sender, EventArgs e)
    {
        txtSearch.Text = "Search By Address";
        textBox1.Text = "Enter Employee ID";
    }

    private void textBox1_Keyfress(object sender, KeyfressEventArgs e)
    {
        SqlCommand cmd = new SqlCommand("SELECT letterId, employeeId, address,
receivedDate, letterType, code, letterStatus, deliveredDate, person FROM receivedLetters
WHERE employeeId LIKE '%" + textBox1.Text + "%' AND letterStatus = '" + "On firemise" + "'",
conn);

        DataTable dt = new DataTable();
        DataView dv = dt.DefaultView;

        conn.Open();

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewSearchLetter.DataSource = dt;
    }

    private void textBox1_Click(object sender, EventArgs e)
    {
        textBox1.Text = "";
    }

    private void printDocument1_flr intflage(object sender,
System.Drawing.flr inting.flr intflageEventArgs e)
    {
        Bitmap bitmap = new Bitmap(dataGridViewSearchLetter.Width,
dataGridViewSearchLetter.Height);
        dataGridViewSearchLetter.DrawToBitmap(bitmap, new Rectangle(0, 0,
dataGridViewSearchLetter.Width, dataGridViewSearchLetter.Height));
        e.Graphics.DrawImage(bitmap, 1, 1);
    }

    private void btnflrint_Click(object sender, EventArgs e)
    {
        printflreviewDialog1.Document = printDocument1;
        printflreviewDialog1.flr intflreviewControl.Zoom = 1;
        printflreviewDialog1.ShowDialog();
    }
}
}
}

```



## View Profile

Employee ID	emp001
Name	sanjeewa
Gender	Male
Contact Number	0758009265
Address	105/ A, Ambagasipitiya, Yakkala
Employee Type	Post Master
Basic Salary	5000
Status	Permanent
Hire Date	9/28/2023

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

```
namespace flostOfficeManagement
```

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

```
SqlConnection conn = new SqlConnection(@"Data Source=LafitOfi-
NLDBQLGG\SQLXFIRESS;Initial Catalog=flostOffice;Integrated Security=True");
```

```
private void profile_Load(object sender, EventArgs e)
{
    txtAddress.Hide();
    txtContact.Hide();
}
```

```

txtName.Hide();
txtflassword.Hide();
btnEdit.Hide();
lblA.Hide();
lblC.Hide();
lblN.Hide();
lblp.Hide();

```

```

SqlCommand cmd = new SqlCommand("SELECT [employeeId], [name], [gender],
[contact], [address], [employeeType], [BasicSalary], [status], [hireDate] FROM
[dbo].[employee] WHERE [employeeId] = '" + login.user + "'", conn);
conn.Open();

```

```

SqlDataReader myR = cmd.ExecuteReader();
if (myR.HasRows)
{
    while (myR.Read())
    {
        lblEmpId.Text = myR[0].ToString();
        lblname.Text = myR[1].ToString();
        lblGender.Text = myR[2].ToString();
        lblContact.Text = myR[3].ToString();
        lblAddress.Text = myR[4].ToString();
        lblEmpType.Text = myR[5].ToString();
        lblSalary.Text = myR[6].ToString();
        lblStatus.Text = myR[7].ToString();
        lblHireDate.Text = myR[8].ToString();
    }
}
conn.Close();

```

```

SqlCommand cmd1 = new SqlCommand("SELECT [password] FROM [dbo].[login] WHERE
[employeeId] = '" + login.user + "'", conn);
conn.Open();

```

```

SqlDataReader myR1 = cmd1.ExecuteReader();
if (myR1.HasRows)
{
    while (myR1.Read())
    {
        lblflassword.Text = myR1[0].ToString();
    }
}
conn.Close();

```

```

private void txtName_Click(object sender, EventArgs e)
{
    txtName.Text = lblname.Text;
}

```

```

private void txtContact_Click(object sender, EventArgs e)
{
    txtContact.Text = lblContact.Text;
}

```

```

private void txtAddress_Click(object sender, EventArgs e)
{
    txtAddress.Text = lblAddress.Text;
}

private void txtflassword_Click(object sender, EventArgs e)
{
    txtflassword.Text = lblflassword.Text;
}

private void profile_Click(object sender, EventArgs e)
{
    txtAddress.Text = lblAddress.Text;
    txtContact.Text = lblContact.Text;
    txtName.Text = lblname.Text;
    txtflassword.Text = lblflassword.Text;
}

private void btnflrofile_Click(object sender, EventArgs e)
{
    txtAddress.Show();
    txtContact.Show();
    txtName.Show();
    txtflassword.Show();
    btnEdit.Show();
    lblA.Show();
    lblC.Show();
    lblN.Show();
    lblp.Show();

    txtAddress.Text = lblAddress.Text;
    txtContact.Text = lblContact.Text;
    txtName.Text = lblname.Text;
    txtflassword.Text = lblflassword.Text;
}

private void btnEdit_Click(object sender, EventArgs e)
{
    if (txtName.Text == "")
    {
        MessageBox.Show("Enter Name", "flrofile Edit Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtName.Focus();
        return;
    }
    if (txtContact.Text == "")
    {
        MessageBox.Show("Enter contact number", "flrofile Edit Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtContact.Focus();
        return;
    }
    if (txtAddress.Text == "")
    {

```

```

        MessageBox.Show("Enter address", "Profile Edit Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtAddress.Focus();
        return;
    }
    if (txtPassword.Text == "")
    {
        MessageBox.Show("Enter password", "Profile Edit Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtPassword.Focus();
        return;
    }
    else
    {
        if (txtPassword.Text.Length > 10)
        {
            MessageBox.Show("Maximum 10 characters only", "Profile Edit Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtPassword.Focus();
            return;
        }
    }

    var result = MessageBox.Show("Are you sure you want to edit details", "Edit Profile Information", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

    if (result == DialogResult.Yes)
    {
        SqlCommand cmd1 = new SqlCommand(@"UPDATE [dbo].[employee] SET [name] = 
        '' + txtName.Text + '', [contact] = '' + txtContact.Text + '', [address] = '' + 
        txtAddress.Text + '' WHERE [employeeId] = '' + lblEmpId.Text + '', conn);
        conn.Open();
        cmd1.ExecuteNonQuery();
        conn.Close();

        SqlCommand cmd2 = new SqlCommand(@"UPDATE [dbo].[login] SET [password] = 
        '' + txtPassword.Text + '' WHERE [employeeId] = '' + lblEmpId.Text + '', conn);
        conn.Open();
        cmd2.ExecuteNonQuery();
        conn.Close();

        SqlCommand cmd = new SqlCommand("SELECT [employeeId], [name], [gender], 
        [contact], [address], [employeeType], [BasicSalary], [status], [hireDate] FROM 
        [dbo].[employee] WHERE [employeeId] = '' + login.user + '', conn);
        conn.Open();

        SqlDataReader myR = cmd.ExecuteReader();
        if (myR.HasRows)
        {
            while (myR.Read())
            {
                lblEmpId.Text = myR[0].ToString();
                lblName.Text = myR[1].ToString();
                lblGender.Text = myR[2].ToString();
                lblContact.Text = myR[3].ToString();
            }
        }
    }
}

```

```

        lblAddress.Text = myR[4].ToString();
        lblEmpType.Text = myR[5].ToString();
        lblSalary.Text = myR[6].ToString();
        lblStatus.Text = myR[7].ToString();
        lblHireDate.Text = myR[8].ToString();
    }
}
conn.Close();

SqlCommand cmd3 = new SqlCommand("SELECT [password] FROM [dbo].[logIn]
WHERE [employeeId] = '" + logIn.user + "'", conn);
conn.Open();

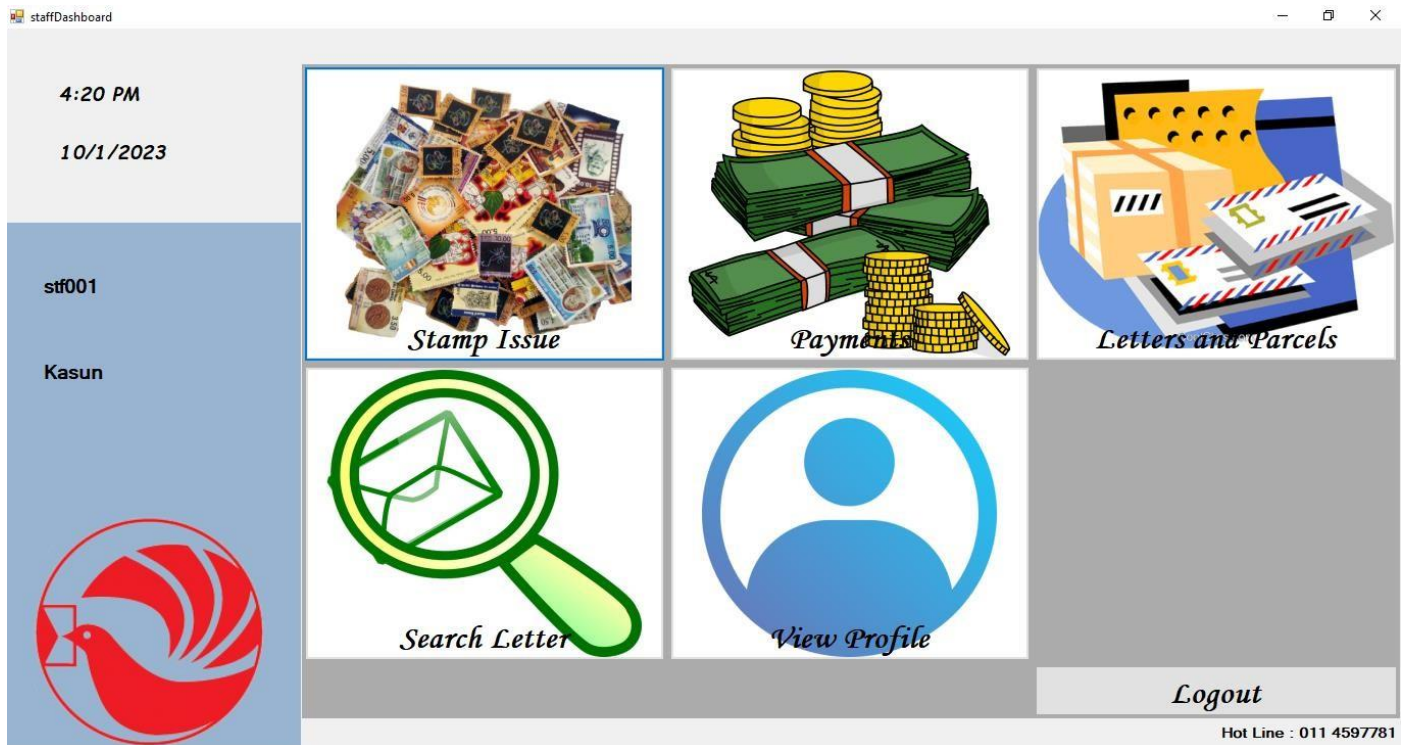
SqlDataReader myR1 = cmd3.ExecuteReader();
if (myR1.HasRows)
{
    while (myR1.Read())
    {
        lblflassword.Text = myR1[0].ToString();

    }
}
conn.Close();

txtAddress.Hide();
txtContact.Hide();
txtName.Hide();
txtflassword.Hide();
btnEdit.Hide();
lblA.Hide();
lblC.Hide();
lblN.Hide();
lblp.Hide();
}
if (result == DialogResult.No)
{
    txtAddress.Hide();
    txtContact.Hide();
    txtName.Hide();
    txtflassword.Hide();
    btnEdit.Hide();
    lblA.Hide();
    lblC.Hide();
    lblN.Hide();
    lblp.Hide();
}
}
}
}

```

## Staff Dashboard



```
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 flostOfficeManagement
{
    public partial class staffDashboard : Form
    {
        public staffDashboard()
        {
            InitializeComponent();
        }
        SqlConnection conn = new SqlConnection(@"Data Source=LafitOfi-
NLDBQLGG\SQLSERVER;Initial Catalog=flostOffice;Integrated Security=True");

        private void panel1_flaint(object sender, flaintEventArgs e)
        {
        }

        private void staffDashboard_Load(object sender, EventArgs e)
        {
        }
    }
}
```

```

{
    if (login.user.Contains("stf"))
    {
        button6.Hide();
    }
    if (login.user == "admin")
    {
        button4.Hide();
        lblEmpName.Hide();
    }

    lblEmpId.Text = login.user;

    SqlCommand cmd = new SqlCommand("SELECT [name] FROM [dbo].[employee] WHERE
[employeeid] = '' + login.user + ''", conn);
    conn.Open();

    SqlDataReader myR = cmd.ExecuteReader();
    if (myR.HasRows)
    {
        while (myR.Read())
        {
            lblEmpName.Text = myR[0].ToString();
        }
    }
    conn.Close();

    timer1.Start();
    lblDate.Text = DateTime.Now.ToShortDateString();
    lblTime.Text = DateTime.Now.ToShortTimeString();
}

private void button4_Click(object sender, EventArgs e)
{
    searchLetter letterStatus = new searchLetter();

    letterStatus.TopLevel = false;
    pnlContent.Controls.Add(letterStatus);
    letterStatus.BringToFront();
    letterStatus.Show();
}

private void button1_Click(object sender, EventArgs e)
{
    stampIssue stampIssue = new stampIssue();
    stampIssue.TopLevel = false;
    pnlContent.Controls.Add(stampIssue);
    stampIssue.BringToFront();
    stampIssue.Show();
}

private void panel5_flaint(object sender, flaintEventArgs e)
{

```

```

    }

    private void pnlContent_fla int(object sender, fla intEventArgs e)
    {

    }

    private void lbl1_Click(object sender, EventArgs e)
    {

    }

    private void lbl1_Click_1(object sender, EventArgs e)
    {

    }

    private void button3_Click(object sender, EventArgs e)
    {
        acceptedLetters acceptedLetters = new acceptedLetters();
        acceptedLetters.TopLevel = false;
        pnlContent.Controls.Add(acceptedLetters);
        acceptedLetters.BringToFront();
        acceptedLetters.Show();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        payment payment = new payment();
        payment.TopLevel = false;
        pnlContent.Controls.Add(payment);
        payment.BringToFront();
        payment.Show();
    }

    private void tableLayoutflanel1_fla int(object sender, fla intEventArgs e)
    {

    }

    private void tableLayoutflanel1_fla int_1(object sender, fla intEventArgs e)
    {

    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        lblTime.Text = DateTime.Now.ToShortTimeString();
        timer1.Start();
    }

    private void button4_Click_1(object sender, EventArgs e)
    {
        profile profile = new profile();
    }

```



```

        profile.TopLevel = false;
        pnlContent.Controls.Add(profile);
        profile.BringToFront();
        profile.Show();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        login login = new login();

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

    private void button6_Click(object sender, EventArgs e)
    {
        employee employee = new employee();

        employee.TopLevel = false;
        pnlContent.Controls.Add(employee);
        employee.BringToFront();
        employee.Show();
    }
}

```

## Stamp Issue

4:22 PM  
10/1/2023

stf001  
Kasun

stampPrice	quantity
0.5	263
1	176
2	90
3	10
5	100
10	99
15	100
20	100
25	200
30	100
35	100
40	110
50	200
60	100

Price

Issue

Enter Letter Type

Enter Weight (g)

Price

Clear Calculate

Hot Line : 011 4597781

```
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.Security.Principal;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml.Linq;
```

```
namespace flostOfficeManagement
{
```

```
    public partial class stampIssue : Form
    {
```

```
        public stampIssue()
        {
            InitializeComponent();
        }
```

```
        SqlConnection conn = new SqlConnection(@"Data Source=LafitOffice-
NLDBQLG\SQLSERVER;Initial Catalog=flostOffice;Integrated Security=True");
```

```
        private void textBox1_TextChanged(object sender, EventArgs e)
        {
        }
    }
```

```

private void btnIssue_Click(object sender, EventArgs e)
{
    MessageBox.Show("Please double check the availability of stamps before issue", "Stamp Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
    DialogResult result = MessageBox.Show("Confirmation availability", "Stamp Information", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

    if (result == DialogResult.Yes)
    {
        double price = 0;

        if (cb0.Checked == true)
        {
            stamp(0.50, txt0.Text);
            price = price + (0.50 * int.Parse(txt0.Text));
        }
        if (cb1.Checked == true)
        {
            stamp(1, txt1.Text);
            price = price + (1 * int.Parse(txt1.Text));
        }
        if (cb2.Checked == true)
        {
            stamp(2, txt2.Text);
            price = price + (2 * int.Parse(txt2.Text));
        }
        if (cb3.Checked == true)
        {
            stamp(3, txt3.Text);
            price = price + (3 * int.Parse(txt3.Text));
        }
        if (cb5.Checked == true)
        {
            stamp(5, txt5.Text);
            price = price + (5 * int.Parse(txt5.Text));
        }
        if (cb10.Checked == true)
        {
            stamp(10, txt10.Text);
            price = price + (10 * int.Parse(txt10.Text));
        }
        if (cb15.Checked == true)
        {
            stamp(15, txt15.Text);
            price = price + (15 * int.Parse(txt15.Text));
        }
        if (cb20.Checked == true)
    }
}

```

```
{
    stamp(20, txt20.Text);
    price = price + (20 * int.Parse(txt20.Text));
}
if (cb25.Checked == true)
{
    stamp(25, txt25.Text);
    price = price + (25 * int.Parse(txt25.Text));
}
if (cb30.Checked == true)
{
    stamp(30, txt30.Text);
    price = price + (30 * int.Parse(txt30.Text));
}
if (cb35.Checked == true)
{
    stamp(35, txt35.Text);
    price = price + (35 * int.Parse(txt35.Text));
}
if (cb40.Checked == true)
{
    stamp(40, txt40.Text);
    price = price + (40 * int.Parse(txt40.Text));
}
if (cb50.Checked == true)
{
    stamp(50, txt50.Text);
    price = price + (50 * int.Parse(txt50.Text));
}
if (cb60.Checked == true)
{
    stamp(60, txt60.Text);
    price = price + (60 * int.Parse(txt60.Text));
}
if (cb75.Checked == true)
{
    stamp(75, txt75.Text);
    price = price + (75 * int.Parse(txt75.Text));
}
if (cb100.Checked == true)
{
    stamp(100, txt100.Text);
    price = price + (100 * int.Parse(txt100.Text));
}
if (cb110.Checked == true)
{
    stamp(110, txt110.Text);
```

```

        price = price + (110 * int.Parse(txt110.Text));
    }
    if (cb500.Checked == true)
    {
        stamp(500, txt500.Text);
        price = price + (500 * int.Parse(txt500.Text));
    }
    if (cb1000.Checked == true)
    {
        stamp(1000, txt1000.Text);
        price = price + (1000 * int.Parse(txt1000.Text));
    }

    lblStampPrice.Text = price.ToString();

    float stampAmount = float.Parse(lblStampPrice.Text);
    SqlCommand cmd = new SqlCommand(@"INSERT INTO [dbo].[payments]
([employeeid], [paymentType], [description], [date], [time], [amount]) VALUES ('" +
login.user + "'", "'" + "Stamp Issue" + "'", "'" + "Stamp Issue" + "'", "'" +
DateTime.Now.ToShortDateString() + "'", "'" + DateTime.Now.ToShortTimeString() + "'", "'" +
stampAmount + "')", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    clear();
}
else if (result == DialogResult.No)
{
    clear();
}
}

private void clear()
{
    initialize();
    cb0.Checked = false;
    cb1.Checked = false;
    cb2.Checked = false;
    cb3.Checked = false;
    cb5.Checked = false;
    cb10.Checked = false;
    cb15.Checked = false;
    cb20.Checked = false;
    cb25.Checked = false;
    cb30.Checked = false;
    cb35.Checked = false;
    cb40.Checked = false;
    cb50.Checked = false;
    cb60.Checked = false;
    cb75.Checked = false;
    cb100.Checked = false;
    cb110.Checked = false;

```

```

        cb500.Checked = false;
        cb1000.Checked = false;
        //lblStampPrice.Text = "";
    }

    private void stamp(double price, string amount)
    {
        int num = 0;
        int stampCount;
        int updateVal;

        stampCount = int.Parse(amount);
        SqlCommand cmd = new SqlCommand("SELECT quantity FROM [dbo].[stamp] WHERE
stapmflrice = '"+price+"'", conn);
        conn.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            num = int.Parse(reader["quantity"].ToString());

        }
        conn.Close();
        if (stampCount > num)
        {
            MessageBox.Show("");

        }
        else
        {
            updateVal = num - stampCount;
            SqlCommand cmd1 = new SqlCommand(@"UPDATE [dbo].[stamp] SET [quantity] =
'' + updateVal + '' WHERE stapmflrice = '"+price+"'", conn);
            conn.Open();
            cmd1.ExecuteNonQuery();
            conn.Close();
            getStampDetails();
        }
    }

    private void lblPrice_Click(object sender, EventArgs e)
    {

    }

    private bool IsNumeric(string input)
    {
        Regex regex = new Regex("^[0-9]+$");
        return regex.IsMatch(input);
    }

    private void btnCalc_Click(object sender, EventArgs e)
    {
        if (cmbLetterType.Text == "")
        {

```

```

        MessageBox.Show("Please enter letter type", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        cmbLetterType.Focus();
        return;
    }

    if (txtWeight.Text == "")
    {
        MessageBox.Show("Please enter weight", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        txtWeight.Focus();
        return;
    }
    else if (!IsNumeric(txtWeight.Text))
    {
        MessageBox.Show("Invalid weight", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        txtWeight.Focus();
        return;
    }

    if (cmbLetterType.SelectedIndex == 0)
    {
        weightAmount(50, 20, 10);
    }
    if (cmbLetterType.SelectedIndex == 1)
    {
        weightAmount(110, 20, 10);
    }
    if (cmbLetterType.SelectedIndex == 2)
    {
        weightAmount(150, 250, 50);
    }
    if (cmbLetterType.SelectedIndex == 3)
    {
        weightAmount(200, 250, 50);
    }
    if (cmbLetterType.SelectedIndex == 4)
    {
        lblPrice.Text = "30";
    }
}

private void weightAmount(double minPrice, double minWeight, double addedAmount)
{
    double price = minPrice;
    for (double i = minWeight; i <= double.Parse(txtWeight.Text); i = i +
minWeight)
    {
        price = price + addedAmount;
    }
    lblPrice.Text = price.ToString();
}

```

```

    }

    private void stampIssue_Load(object sender, EventArgs e)
    {
        if (login.user.Contains("stf"))
        {
            btnAddStamps.Hide();
        }

        initialize();
        getStampDetails();
    }

    private void getStampDetails()
    {
        SqlCommand cmd = new SqlCommand("SELECT * FROM [dbo].[stamp]", conn);
        DataTable dt = new DataTable();

        conn.Open();

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewStampDetails.DataSource = dt;
    }

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

    private void addStamp(double price, string amount)
    {
        int num = 0;
        int stampCount;
        int updateVal;

        stampCount = int.Parse(amount);
        SqlCommand cmd = new SqlCommand("SELECT quantity FROM [dbo].[stamp] WHERE stampprice = '" + price + "'", conn);
        conn.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            num = int.Parse(reader["quantity"].ToString());
            conn.Close();
            updateVal = num + stampCount;
            SqlCommand cmd1 = new SqlCommand(@"UPDATE [dbo].[stamp] SET [quantity] = '" + updateVal + "' WHERE stampprice = '" + price + "'", conn);
            conn.Open();
            cmd1.ExecuteNonQuery();
            conn.Close();
        }
    }

```



```
        getStampDetails();

    }

}

private void btnAddStamps_Click(object sender, EventArgs e)
{
    if (cb0.Checked == true)
    {
        addStamp(0.50, txt0.Text);
    }
    if (cb1.Checked == true)
    {
        addStamp(1, txt1.Text);
    }
    if (cb2.Checked == true)
    {
        addStamp(2, txt2.Text);
    }
    if (cb3.Checked == true)
    {
        addStamp(3, txt3.Text);
    }
    if (cb5.Checked == true)
    {
        addStamp(5, txt5.Text);
    }
    if (cb10.Checked == true)
    {
        addStamp(10, txt10.Text);
    }
    if (cb15.Checked == true)
    {
        addStamp(15, txt15.Text);
    }
    if (cb20.Checked == true)
    {
        addStamp(20, txt20.Text);
    }
}
```

```
}  
if (cb25.Checked == true)  
{  
    addStamp(25, txt25.Text);  
  
}  
if (cb30.Checked == true)  
{  
    addStamp(30, txt30.Text);  
  
}  
if (cb35.Checked == true)  
{  
    addStamp(35, txt35.Text);  
  
}  
if (cb40.Checked == true)  
{  
    addStamp(40, txt40.Text);  
  
}  
if (cb50.Checked == true)  
{  
    addStamp(50, txt50.Text);  
  
}  
if (cb60.Checked == true)  
{  
    addStamp(60, txt60.Text);  
  
}  
if (cb75.Checked == true)  
{  
    addStamp(75, txt75.Text);  
  
}  
if (cb100.Checked == true)  
{  
    addStamp(100, txt100.Text);  
  
}  
if (cb110.Checked == true)  
{  
    addStamp(110, txt110.Text);  
  
}  
if (cb500.Checked == true)
```

```

        {
            addStamp(500, txt500.Text);

        }
        if (cb1000.Checked == true)
        {
            addStamp(1000, txt1000.Text);

        }

        clear();
    }

    private void initialize()
    {
        txt0.Text = "0";
        txt1.Text = "0";
        txt2.Text = "0";
        txt3.Text = "0";
        txt5.Text = "0";
        txt10.Text = "0";
        txt15.Text = "0";
        txt20.Text = "0";
        txt25.Text = "0";
        txt30.Text = "0";
        txt35.Text = "0";
        txt40.Text = "0";
        txt50.Text = "0";
        txt60.Text = "0";
        txt75.Text = "0";
        txt1000.Text = "0";
        txt110.Text = "0";
        txt500.Text = "0";
        txt1000.Text = "0";

    }

    private void txt0_Click(object sender, EventArgs e)
    {
        txt0.Clear();
    }

    private void txt1_Click(object sender, EventArgs e)
    {
        txt1.Clear();
    }

    private void txt2_Click(object sender, EventArgs e)
    {
        txt2.Clear();
    }

    private void txt3_Click(object sender, EventArgs e)

```

```
{
    txt3.Clear();
}

private void txt5_Click(object sender, EventArgs e)
{
    txt5.Clear();
}

private void txt10_Click(object sender, EventArgs e)
{
    txt10.Clear();
}

private void txt15_Click(object sender, EventArgs e)
{
    txt15.Clear();
}

private void txt20_Click(object sender, EventArgs e)
{
    txt20.Clear();
}

private void txt25_Click(object sender, EventArgs e)
{
    txt25.Clear();
}

private void txt30_Click(object sender, EventArgs e)
{
    txt30.Clear();
}

private void txt35_Click(object sender, EventArgs e)
{
    txt35.Clear();
}

private void txt40_Click(object sender, EventArgs e)
{
    txt40.Clear();
}

private void txt50_Click(object sender, EventArgs e)
{
    txt50.Clear();
}

private void txt60_Click(object sender, EventArgs e)
{
    txt60.Clear();
}

private void txt75_Click(object sender, EventArgs e)
{

```

```

        txt75.Clear();
    }

    private void txt100_Click(object sender, EventArgs e)
    {
        txt100.Clear();
    }

    private void txt110_Click(object sender, EventArgs e)
    {
        txt110.Clear();
    }

    private void txt500_Click(object sender, EventArgs e)
    {
        txt500.Clear();
    }

    private void txt1000_Click(object sender, EventArgs e)
    {
        txt1000.Clear();
    }

    private void cb0_CheckedChanged(object sender, EventArgs e)
    {
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        txtWeight.Clear();
        cmbLetterType.Text = "";
    }
}

```

## Payment


staffDashboard

4:22 PM

10/1/2023

stf001

Kasun



Payment Type

Amount  Service Charge Amount

Pay Clear  mm/dd/yy Print

paymentId	employeeId	paymentType	description	date	time	amount
5		Ceylon Electricity Board	123dgdg	9/22/2023	11:06 PM	100
7		Sri Lanka Telecom	Account No.4534535 45...	9/24/2023	6:51 PM	20
8		Sri Lanka Telecom	Account No.4534535 45...	9/24/2023	6:51 PM	20
9		Sri Lanka Telecom	Account No.7897+99+5+...	9/24/2023	6:51 PM	20
12		Ceylon Electricity Board	qwerty	9/24/2023	7:16 PM	120
13		National Water Supply an...	12	9/24/2023	7:17 PM	1020
1007		Stamp Issue	Stamp Issue	9/30/2023	10:08 PM	0
1008	stf001	Stamp Issue	Stamp Issue	9/30/2023	10:09 PM	1000

Hot Line : 011 4597781

```

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.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace flostOfficeManagement
{
    public partial class payment : Form
    {
        public payment()
        {
            InitializeComponent();

            SqlConnection conn = new SqlConnection(@"Data Source=LafitOfi-
NLDBQLGG\SQLSERVER;Initial Catalog=flostOffice;Integrated Security=True");

            private void cmbPaymentType_SelectedIndexChanged(object sender, EventArgs e)
            {
                pnlAccount.Visible = false;
                pnlLife.Visible = false;
            }
        }
    }
}

```

```

        pnlMobile.Visible = false;
        pnlVehicle.Visible = false;
        pnlOther.Visible = false;
        pnlExam.Visible = false;

        switch (cmbflaymentType.SelectedIndex)
        {
            case 0:
            case 1:
                lblServiceCharge.Text = "20";
                pnlAccount.Show();
                break;
            case 2:
            case 3:
                lblServiceCharge.Text = "20";
                pnlMobile.Show();
                break;
            case 4:
                lblServiceCharge.Text = "50";
                pnlLife.Show();
                break;
            case 5:
                lblServiceCharge.Text = "50";
                pnlAccount.Visible = false;
                pnlVehicle.Show();
                break;
            case 6:
                lblServiceCharge.Text = "10";
                pnlExam.Show();
                break;
            case 7:
            case 8:
            case 9:
                lblServiceCharge.Text = "50";
                pnlOther.Show();
                break;
        }
    }

    private void payment_Load(object sender, EventArgs e)
    {
        pnlAccount.Visible = false;
        pnlLife.Visible = false;
        pnlMobile.Visible = false;
        pnlVehicle.Visible = false;
        pnlOther.Visible = false;
        pnlExam.Visible = false;

        getflaymentDetails();
    }

    private void getflaymentDetails()
    {

```

```

SqlCommand cmd = new SqlCommand("SELECT [paymentId], [employeeId],
[paymentType], [description], [date], [time], [amount] FROM [dbo].[payments]", conn);
DataTable dt = new DataTable();

conn.Open();

SqlDataReader reader = cmd.ExecuteReader();
dt.Load(reader);
conn.Close();

dataGridViewflayment.DataSource = dt;
}

private bool IsNumeric(string input)
{
    Regex regex = new Regex("[0-9]+$");
    return regex.IsMatch(input);
}

private void btnflay_Click(object sender, EventArgs e)
{
    if (cmbflaymentType.Text == "")
    {
        MessageBox.Show("Please enter payment type", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        cmbflaymentType.Focus();
        return;
    }
    if (txtAmount.Text == "")
    {
        MessageBox.Show("Please enter amount", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        txtAmount.Focus();
        return;
    }
    else if (!IsNumeric(txtAmount.Text))
    {
        MessageBox.Show("Invalid amount", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        txtAmount.Focus();
        return;
    }

    var result = MessageBox.Show("Confirm payment", "flayment Informat ion",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information);

    if (result == DialogResult.Yes)
    {
        float totalAmount = float.Parse(txtAmount.Text) +
        float.Parse(lblServiceCharge.Text);

        SqlCommand cmd = new SqlCommand(@"INSERT INTO [dbo].[payments]
([employeeId], [paymentType], [description], [date], [time], [amount]) VALUES ('" + log
n.user + "', '" + cmbflaymentType.SelectedItem.ToString() + "', '" + txtAccount.Text
+ "' + ' " + " " + "' + ' " + txtTelephoneNumber.Text + "' + ' " + "\n" + ' " + ' " +

```



```

txtVehicalNumber.Text + "','" + " " + "','" + txtChassisNumber.Text + "','" + "\n"
+ "','" + txtPolicyNumber.Text + "','" + "\n" + "','" + txtExamCode.Text + "','" +
+ "\n" + "','" + txtName.Text + "','" + " " + "','" + txtID.Text + "','" +
DateTime.Now.ToShortDateString() + "','" + DateTime.Now.ToShortTimeString() + "','" +
totalAmount + "','"", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    getflaymentDetails();
}
else if (result == DialogResult.No)
{
    return;
}

txtAccount.Clear();
txtTelephoneNumber.Clear();
txtVehicalNumber.Clear();
txtChassisNumber.Clear();
txtPolicyNumber.Clear();
txtExamCode.Clear();
txtName.Clear();
txtID.Clear();
txtAmount.Clear();
cmbflaymentType.Text = "";
cmbflaymentType.Focus();

}

private void btnClear_Click(object sender, EventArgs e)
{
    txtAccount.Clear();
    txtTelephoneNumber.Clear();
    txtVehicalNumber.Clear();
    txtChassisNumber.Clear();
    txtPolicyNumber.Clear();
    txtExamCode.Clear();
    txtName.Clear();
    txtID.Clear();
    txtAmount.Clear();
    cmbflaymentType.Text = "";
    cmbflaymentType.Focus();
}

private void textBox2_Keyfress(object sender, KeyfressEventArgs e)
{
    SqlCommand cmd = new SqlCommand("SELECT * FROM payments WHERE date LIKE '%"
+ textBox2.Text + "%';", conn);
    DataTable dt = new DataTable();
    DataView dv = dt.DefaultView;

    conn.Open();

```

```

        SqlDataReader reader = cmd.ExecuteReader();
        dt.Load(reader);
        conn.Close();

        dataGridViewflayment.DataSource = dt;
    }

    private void textBox2_Click(object sender, EventArgs e)
    {
        textBox2.Text = "";
    }

    private void payment_Click(object sender, EventArgs e)
    {
        textBox2.Text = "mm/dd/yy";
    }

    private void printDocument1_flrintflage(object sender,
System.Drawing.flrinting.flrintflageEventArgs e)
    {
        Bitmap bitmap = new Bitmap(dataGridViewflayment.Width,
dataGridViewflayment.Height);
        dataGridViewflayment.DrawToBitmap(bitmap, new Rectangle(0, 0,
dataGridViewflayment.Width, dataGridViewflayment.Height));
        e.Graphics.DrawImage(bitmap, 1, 1);
    }

    private void btnflrint_Click(object sender, EventArgs e)
    {
        printflreviewDialog1.Document = printDocument1;
        printflreviewDialog1.flrintflreviewControl.Zoom = 1;
        printflreviewDialog1.ShowDialog();
    }
}

```

## Accepted Letters

letterId	senderAddress	recipientAddress	date	localForeign	letterType
1	asd	fhgf	9/23/2023	Local	Normal
2	asd	fhgf	9/23/2023	Local	Normal
3	dddd	fhgf	9/23/2023	Local	Normal
4	105	sdsad	9/24/2023	Local	Normal
1004	gampaha	negambo	9/30/2023	Local	Normal
6	105/A,Ambagaspitiya,Yakkala	58/3,Welweniya West, Welwe...	9/24/2023	Local	Speed Post

Hot Line : 011 4597781

```

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

namespace flostOfficeManagement
{
    public partial class acceptedLetters : Form
    {
        public acceptedLetters()
        {
            InitializeComponent();
        }

        SqlConnection conn = new SqlConnection(@"Data Source=LAfitOfi-
NLDBQLGG\SQLSERVER;Initial Catalog=flostOffice;Integrated Security=True");

        private void acceptedLetters_Load(object sender, EventArgs e)
        {
            if (login.user.Contains("stf"))
            {
                txtLetterId.Hide();
                btnEdit.Hide();
                btnDelete.Hide();
            }
        }
    }
}

```

```

    }

    cmbLetterType.SelectedIndex = 0;
    cmbStatus.SelectedIndex = 0;

    getAcceptedLettersDetails();
}

private void getAcceptedLettersDetails()
{
    SqlCommand cmd = new SqlCommand("SELECT * FROM acceptedLetters", conn);
    DataTable dt = new DataTable();

    conn.Open();

    SqlDataReader reader = cmd.ExecuteReader();
    dt.Load(reader);
    conn.Close();

    dataGridViewAcceptedLetters.DataSource = dt;
}

private void btnInsert_Click(object sender, EventArgs e)
{
    if (txtSender.Text == "")
    {
        MessageBox.Show("please enter sender's address", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtSender.Focus();
        return;
    }
    if (txtReceiver.Text == "")
    {
        MessageBox.Show("please enter receiver's address", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtReceiver.Focus();
        return;
    }

    SqlCommand cmd = new SqlCommand(@"INSERT INTO [dbo].[acceptedLetters]
([senderAddress], [recipientAddress], [date], [location], [letterType]) VALUES
('"+txtSender.Text+"', '"+txtReceiver.Text+"', '"+ DateTime.Now.ToShortDateString() +
"', '"+cmbStatus.SelectedItem.ToString()+"',
 '"+cmbLetterType.SelectedItem.ToString()+"')", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    getAcceptedLettersDetails();

    txtReceiver.Clear();
    txtSender.Clear();
    cmbLetterType.SelectedIndex = 0;
    cmbStatus.SelectedIndex = 0;
}

```

```

        txtSender.Focus();
    }

    private void dataGridViewAcceptedLetters_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

    private void dataGridViewAcceptedLetters_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        if (e.RowIndex >= 0)
        {
            DataGridViewRow row = dataGridViewAcceptedLetters.Rows[e.RowIndex];

            txtSender.Text = row.Cells["senderAddress"].Value.ToString();
            cmbLetterType.SelectedItem = row.Cells["letterType"].Value.ToString();
            cmbStatus.SelectedItem = row.Cells["locatForeign"].Value.ToString();
            txtReceiver.Text = row.Cells["recipientAddress"].Value.ToString();
        }
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void btnEdit_Click(object sender, EventArgs e)
    {
        SqlCommand cmd = new SqlCommand(@"UPDATE [dbo].[acceptedLetters] SET
[senderAddress] = ''+txtSender.Text+'', [recipientAddress] = ''+txtReceiver.Text+'',
[locatForeign] = ''+cmbStatus.SelectedItem.ToString()+'', [letterType] =
''+cmbLetterType.SelectedItem.ToString()+'' WHERE letterId = ''+txtLetterId.Text+''",
conn);

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
        getAcceptedLettersDetails();
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        txtReceiver.Clear();
        txtSender.Clear();
        cmbLetterType.SelectedIndex = 0;
        cmbStatus.SelectedIndex = 0;
        txtSender.Focus();
    }

    private void txtSearch_TextChanged(object sender, EventArgs e)
    {

    }

```

```

private void btnDelete_Click(object sender, EventArgs e)
{
    SqlCommand cmd = new SqlCommand(@"DELETE FROM [dbo].[acceptedLetters] WHERE
letterId = '" + txtLetterId.Text + "'", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
    getAcceptedLettersDetails();
}

private void txtSearch_Click(object sender, EventArgs e)
{
    txtSearch.Clear();
}

private void txtLetterId_Click(object sender, EventArgs e)
{
    txtLetterId.Clear();
}

private void txtSearch_Keyfress(object sender, KeyfressEventArgs e)
{
    SqlCommand cmd = new SqlCommand("SELECT [letterId], [senderAddress],
[recipientAddress], [date], [location], [letterType] FROM [dbo].[acceptedLetters]
WHERE recipientAddress LIKE '%" + txtSearch.Text + "%'", conn);
    DataTable dt = new DataTable();
    DataView dv = dt.DefaultView;

    conn.Open();

    SqlDataReader reader = cmd.ExecuteReader();
    dt.Load(reader);
    conn.Close();

    dataGridViewAcceptedLetters.DataSource = dt;
}

private void acceptedLetters_Click(object sender, EventArgs e)
{
    txtSearch.Text = "Search by Receiver's Address...";
    txtLetterId.Text = "Enter Letter ID to Edit or Delete";
}
}
}


```

## Employee

staffDashboard

4:24 PM  
10/1/2023

admin



Employee ID

Name

Contact Number

Gender

Address

Employee Type

Status

Basic Salary

Password

Add

Clear

Edit

Search by ID

employeeid	name	gender	contact	address	employeeType	BasicSalary	status
emp001	sanjeewa	Male	0758009265	105/A, Ambagaspiya, ...	Post Master	5000	Permanent
stf001	Kasun	Male	078	gampaha	Staff	50000	Permanent

Hot Line : 011 4597781

```

using flostOfficeManagement;
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;
using System.Xml.Linq;
using static System.Runtime.CompilerServices.RuntimeHelpers;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.StartPanel;

namespace flostOfficeManagement
{
    public partial class employee : Form
    {
        public employee()
        {
            InitializeComponent();

            SqlConnection conn = new SqlConnection(@"Data Source=LafitOfi-
NLDBQLGG\SQLSERVER;Initial Catalog=flostOffice;Integrated Security=True");

```

```

private void employee_Load(object sender, EventArgs e)
{
    txtEmployeeId.Focus();

    getEmployeeDetails();
}

private void getEmployeeDetails()
{
    SqlCommand cmd = new SqlCommand("SELECT * FROM employee", conn);
    DataTable dt = new DataTable();

    conn.Open();

    SqlDataReader reader = cmd.ExecuteReader();
    dt.Load(reader);
    conn.Close();

    dataGridViewEmployeeDetails.DataSource = dt;
}

private void btnClear_Click(object sender, EventArgs e)
{
    clear();
}

private void clear()
{
    txtEmployeeId.Clear();
    txtName.Clear();
    txtContact.Clear();
    cmbGender.Text = "";
    cmbEmployeeType.Text = "";
    cmbStatus.Text = "";
    txtAddress.Clear();
    txtPassword.Clear();
    txtSalary.Clear();
}

private void btnAdd_Click(object sender, EventArgs e)
{
    if (txtPassword.Text.Length > 10 || txtPassword.Text.Length <= 0)
    {
        MessageBox.Show("Max password size is 10 characters", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtPassword.Focus();
        return;
    }

    SqlCommand cmd = new SqlCommand(@"INSERT INTO [dbo].[employee]
([employeeId], [name], [gender], [contact], [address], [employeeType], [BasicSalary],
[status], [hireDate]) VALUES ('" + txtEmployeeId.Text + "', '" + txtName.Text + "', '"

```



```

+ cmbGender.SelectedItem.ToString() + "',''" + txtContact.Text + "',''" +
txtAddress.Text + "',''" + cmbEmployeeType.SelectedItem.ToString() + "',''" +
txtSalary.Text + "',''" + cmbStatus.SelectedItem.ToString() + "',''" +
DateTime.Now.ToShortDateString() + "')", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    SqlCommand cmd1 = new SqlCommand(@"INSERT INTO [dbo].[logIn] ([employeeId],
[password]) VALUES ('"+txtEmployeeId.Text+"', '"+txtflassword.Text+"')", conn);
    conn.Open();
    cmd1.ExecuteNonQuery();
    conn.Close();

    getEmployeeDetails();
}

private void dataGridViewEmployeeDetails_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    if (e.RowIndex >= 0)
    {
        DataGridViewRow row = dataGridViewEmployeeDetails.Rows[e.RowIndex];

        txtEmployeeId.Text = row.Cells["employeeId"].Value.ToString();
        txtName.Text = row.Cells["name"].Value.ToString();
        txtContact.Text = row.Cells["contact"].Value.ToString();
        txtSalary.Text = row.Cells["BasicSalary"].Value.ToString();
        txtAddress.Text = row.Cells["address"].Value.ToString();

        cmbEmployeeType.SelectedItem =
row.Cells["employeeType"].Value.ToString();
        cmbGender.SelectedItem = row.Cells["gender"].Value.ToString();
        cmbStatus.SelectedItem = row.Cells["status"].Value.ToString();
    }

    SqlCommand cmd = new SqlCommand("SELECT [password] FROM [dbo].[logIn] where
[employeeId] = '"+txtEmployeeId.Text+"'", conn);
    conn.Open();

    SqlDataReader myR = cmd.ExecuteReader();
    if (myR.HasRows)
    {
        while (myR.Read())
        {
            txtflassword.Text = myR[0].ToString();
        }
    }
    conn.Close();
}

private void btnEdit_Click(object sender, EventArgs e)
{
    SqlCommand cmd = new SqlCommand(@"UPDATE [dbo].[employee] SET [employeeId] =
 '"+txtEmployeeId.Text+"', [name] = '"+txtName.Text+"', [gender] =
 '"+cmbGender.SelectedItem.ToString()+"', [contact] = '"+txtContact.Text+"', [address] =

```

```

""+txtAddress.Text+""", [employeeType] = ""+cmbEmployeeType.SelectedItem.ToString()+""",
[BasicSalary] = ""+txtSalary.Text+""", [status] = ""+cmbStatus.SelectedItem.ToString()+
"" WHERE [employeeId] = ""+txtEmployeeId.Text+""", conn);
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    SqlCommand cmd1 = new SqlCommand(@"UPDATE [dbo].[login] SET [password] =
""+txtfpassword.Text+"" WHERE [employeeId] = ""+txtEmployeeId.Text+""", conn);
    conn.Open();
    cmd1.ExecuteNonQuery();
    conn.Close();
    getEmployeeDetails();
}

private void txtSearch_Click(object sender, EventArgs e)
{
    txtSearch.Clear();
}

private void employee_Click(object sender, EventArgs e)
{
    txtSearch.Text = "Search by ID";
}

private void txtSearch_Keyfress(object sender, KeyfressEventArgs e)
{
    SqlCommand cmd = new SqlCommand("SELECT [employeeId], [name], [gender],
[contact], [address], [employeeType], [BasicSalary], [status], [hireDate] FROM
[dbo].[employee] WHERE [employeeId] LIKE '%" + txtSearch.Text + "%'", conn);
    DataTable dt = new DataTable();
    DataView dv = dt.DefaultView;

    conn.Open();

    SqlDataReader reader = cmd.ExecuteReader();
    dt.Load(reader);
    conn.Close();

    dataGridViewEmployeeDetails.DataSource = dt;
}
}
}

```

## Admin

```
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 flostOfficeManagement
{
    public partial class admin : Form
    {
        public admin()
        {
            InitializeComponent();

            private void btnflozman_Click(object sender, EventArgs e)
            {
                postmanDashboard postmanDashboard = new postmanDashboard();
                postmanDashboard.Show();
                this.Hide();
            }

            private void button2_Click(object sender, EventArgs e)
            {
                staffDashboard staffDashboard = new staffDashboard();
                staffDashboard.Show();
                this.Hide();
            }

            private void admin_Load(object sender, EventArgs e)
            {
            }
        }
    }
}
```

## 7. TOOLS

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

## 8. INITIAL PLAN VS ACTUAL PLAN

### **Initial Plan:**

Initial plan is for a Postal Management System (PMS) envisions the development of a comprehensive software solution to modernize and optimize postal services within a national postal organization. The scope of the project includes streamlining mail processing, tracking, and customer interactions. Key objectives involve enhancing operational efficiency, improving customer satisfaction, and reducing errors in mail handling. The system will incorporate features such as automated mail sorting using advanced technologies like Optical Character Recognition (OCR), real-time tracking and tracing for customers, address validation to minimize undeliverable mail, and a customer relationship management (CRM) module for personalized services. The technology stack will consist of a cloud-based infrastructure to ensure scalability and flexibility, with a user-friendly interface accessible via web and mobile applications. The plan outlines a phased development approach with rigorous testing, security measures, and training for postal employees. Continuous evaluation and improvement are integral to the plan, ensuring that the PMS aligns with evolving postal regulations and the changing needs of customers and the organization.

### **Actual Plan:**

In the actual implementation of the Postal Management System (PMS), the main focus was on letter handling, successfully achieving the goals outlined in the initial plan. Additionally, other features such as report generation and payment management were implemented according to the initial plan.

However, due to the complexity and challenges associated with accurately inputting and managing salary data within the system, the decision was made to exclude the salary calculation module from the actual project. This adjustment was made to ensure the feasibility and efficiency of the system implementation.

## 9. WORK CONTRIBUTION

### 9.1 Individual Work Contribution

Work Contribution	Group Member
Login Form	S. A. S. LAKSHAN
Dashboard Form	G.A.D.D. GANEPOLA
Profile Form	T.C. HATHIRINGE
Data Entry Form	H.K.J. PUNSADINIE
Database	H.B.G.HANDUWALA

### 9.2 Challenges

- **Scheduling Conflicts:** Coordinating the schedules of four group members is challenging, making it difficult to find suitable meeting times.
  - **Task Management:** Handling multiple tasks requires clear responsibilities and deadlines to prevent overlaps or omissions.
  - **Online Collaboration:** Adapting to and effectively using online collaboration tools can be a learning curve for some group members.
  - **Communication Gaps:** Misunderstandings and lack of clarity can occur due to differing schedules, leading to communication gaps.
  - **Motivation and Accountability:** Keeping all group members motivated and accountable, especially when working remotely, poses a challenge.
  - **Conflict Resolution:** Addressing disagreements or conflicts within the group requires effective resolution strategies.
  - **Workload Management:** Balancing project work with individual commitments and responsibilities can lead to stress and time management challenges.
  - **Progress Tracking:** Tracking project progress, especially when group members work at different times, can be complex.
  - **Peer Coordination:** Ensuring alignment between different members' contributions can be tricky.
  - **Flexibility:** Adapting to changing schedules and unforeseen circumstances requires flexibility and adaptability.
-

## 10. FUTURE ENHANCEMENT OF THE PROJECT

In an era marked by rapid technological advancement and evolving consumer expectations, the postal service industry is not exempt from the winds of change. To remain relevant and competitive, postal services worldwide are exploring innovative ways to enhance their operations. A crucial aspect of this evolution is the continuous improvement of their Enterprise Resource Planning (ERP) systems.

Looking ahead, the future of postal service ERP systems promises transformative changes. Real-time tracking and predictive analytics are set to revolutionize parcel delivery, offering customers unparalleled visibility into their shipments while optimizing route planning and resource allocation. The customer experience is poised for a significant upgrade, with self-service portals and mobile apps simplifying the process of sending and receiving parcels.

Automation will play a pivotal role, with advanced sorting machines and robotics streamlining operations and reducing manual errors. Furthermore, electronic document management will usher in a paperless era, benefiting the environment and simplifying record-keeping.

Global expansion and green initiatives will be at the forefront of postal service enhancements, catering to the growing international e-commerce market while promoting sustainability. The future postal service ERP will also leverage machine learning for fraud detection and employ chatbots for efficient customer support.

In summary, the future of postal service ERP systems is bright, promising increased efficiency, customer satisfaction, and environmental responsibility. Embracing these enhancements will undoubtedly help postal services navigate the complex landscape of modern logistics and meet the demands of tomorrow's world.

## 11. DETAILS OF GROUP MEMBERS

Student Number	Student Name	Contact Number	Mail
PS/2019/013	S. A. S. LAKSHAN	071-7897294	<a href="mailto:lakshan-ps19013@stu.kln.ac.lk">lakshan-ps19013@stu.kln.ac.lk</a>
PS/2019/036	G.A.D.D. GANEPOLA	075-8009265	<a href="mailto:ganepol-ps19036@stu.kln.ac.lk">ganepol-ps19036@stu.kln.ac.lk</a>
PS/2019/041	T.C. HATHIRINGE	077-3636316	<a href="mailto:Hathiri-ps19041@stu.kln.ac.lk">Hathiri-ps19041@stu.kln.ac.lk</a>
PS/2019/054	H.K.J. PUNSADINIE	077-4683480	<a href="mailto:punsadi-ps19054@stu.kln.ac.lk">punsadi-ps19054@stu.kln.ac.lk</a>
PS/2019/289	H.B.G.HANDUWALA	071-5916316	<a href="mailto:handuwa-ps19289@stu.kln.ac.lk">handuwa-ps19289@stu.kln.ac.lk</a>