Project Title : Inboxify

Submitted by:

Muhammad Omer (2023-CS-68)

Supervised by: Sir Leeaq

Department of Computer Science Uet Lahore

Short Description of your Project:

Description:

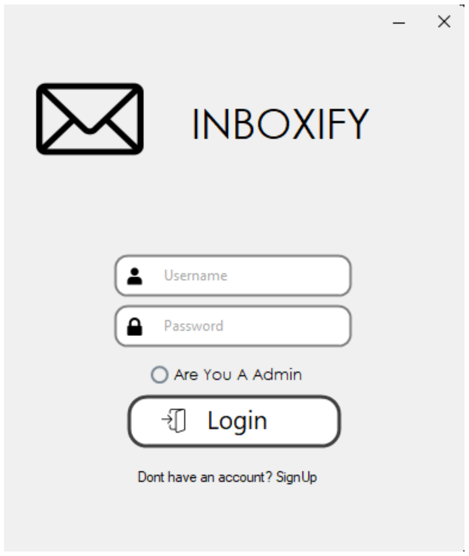
Inboxify is a versatile email client application designed to streamline email management for both personal and business users. The application offers a user-friendly interface for sending, receiving, organizing, and managing emails efficiently. Users can create accounts, categorize emails, attach files, and perform various actions such as archiving, deleting, and marking emails as spam. Inboxify supports multiple email accounts and allows users to switch between them seamlessly. Additionally, the application provides administrative features for managing user accounts and permissions.

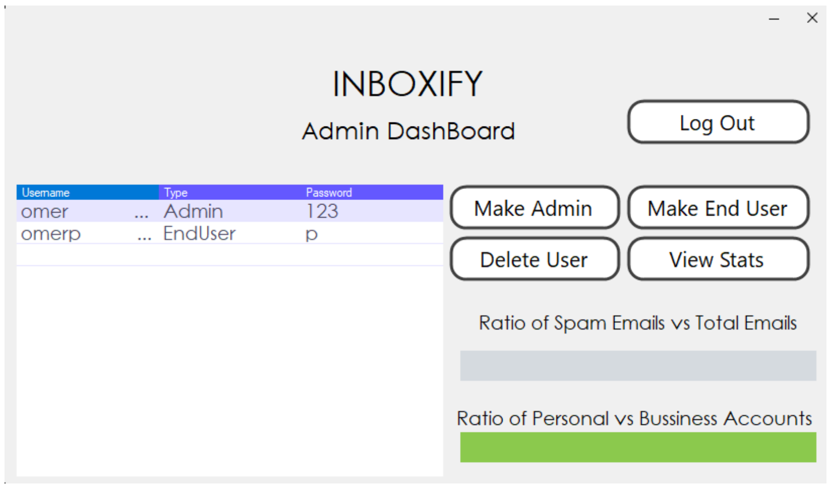
Key Features:

- User-friendly interface for email management.
- Support for personal and business email accounts.
- Categorization of emails with tags like Inbox, Archived, Deleted, and Spam.
- Administrative features for managing user accounts and permissions.
- Secure authentication and password management.
- Integration with SQL Server database for data storage and retrieval.

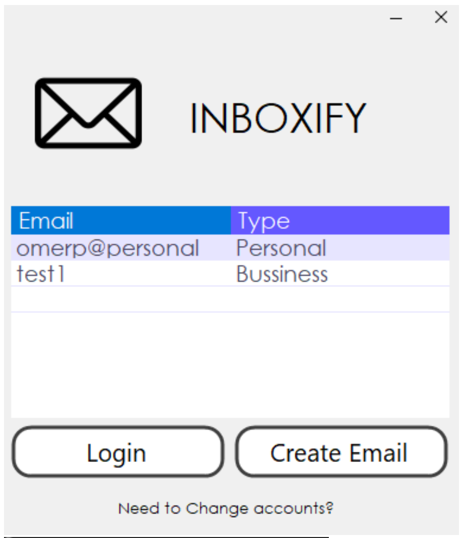
Inboxify aims to enhance productivity and organization in email communication, offering a comprehensive solution for users to effectively manage their email correspondence.

WireFrames:

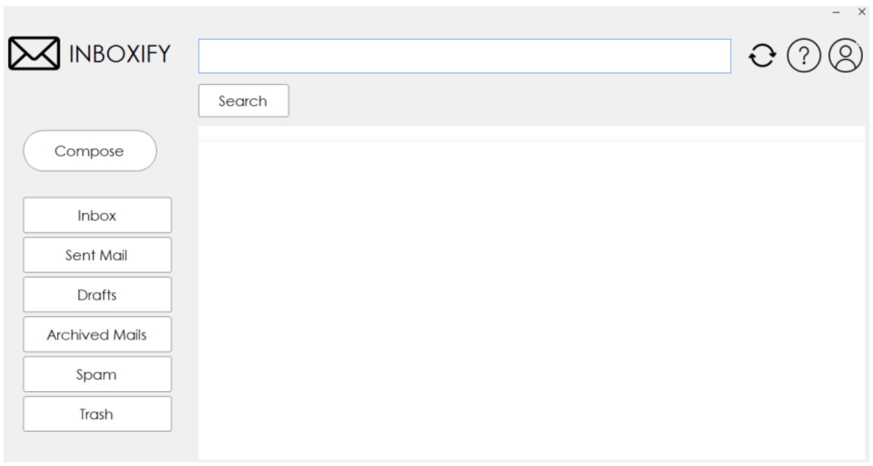


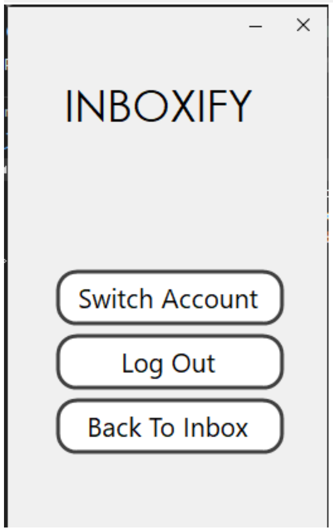

INBOXIFY
Admin DashBoard
Log Out
Make Admin
Make End User
Delete User
View Stats
Ratio of Spam Emails vs Total Emails
Ratio of Personal vs Bussiness Accounts

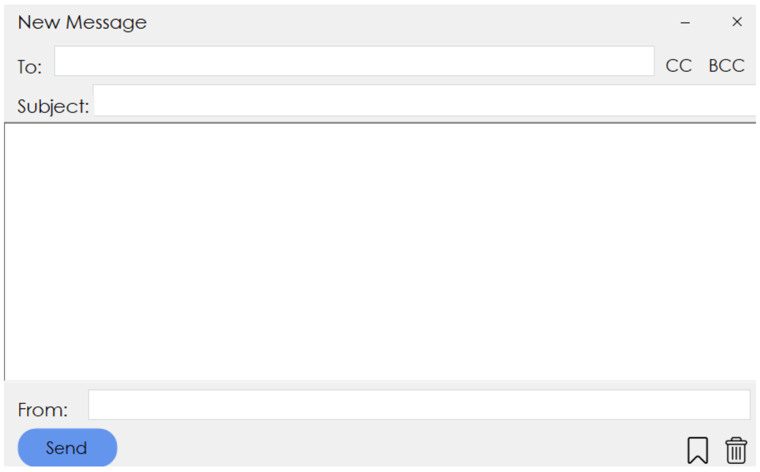
Username		Type	Password
omer	...	Admin	123
omerp	...	EndUser	p



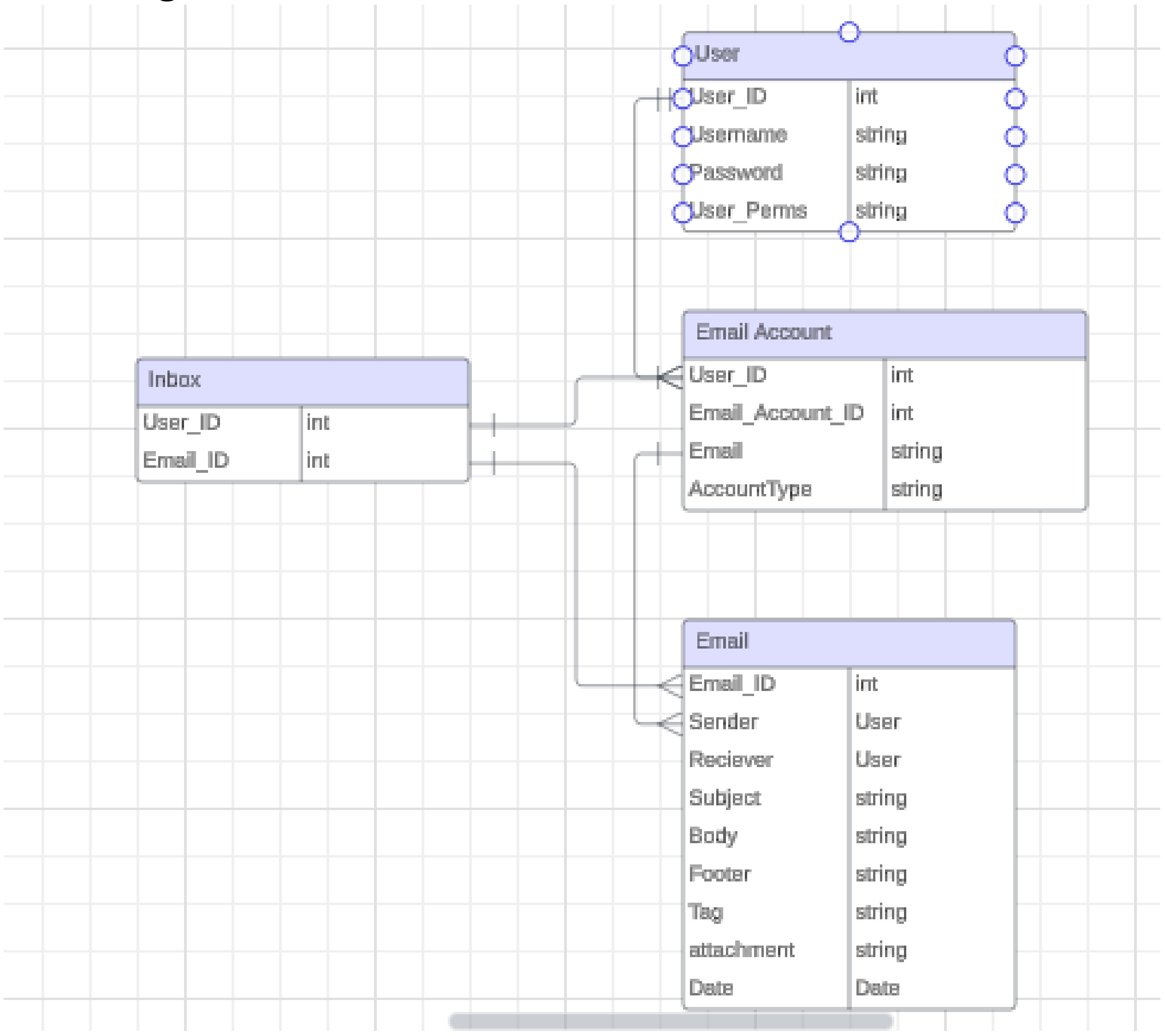
Email	Type
omerp@personal	Personal
test1	Bussiness







CRC Diagram:



Code:

```
using Email_client.BL;
using Inboxify_Backend.Interfaces;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Linq;

namespace Inboxify_Backend.DL
{
    public class EmailDL: IEmailDL
    {
        public void AddEmail(Email email)
        {
            SqlConnection connection = new
SqlConnection(Connection.connectionString);
            connection.Open();
            string query = "INSERT INTO Emails(Sender,
Receiver,Subject,Body,Footer,Tag,Attachment,Date) VALUES
(@Sender,@Receiver,@Subject,@Body,@Footer,@Tag,@Attachment,@Date)";
            SqlCommand command = new SqlCommand(query, connection);

            command.Parameters.Add(new SqlParameter("@Sender", email.Get_Sender()));
            command.Parameters.Add(new SqlParameter("@Receiver",
email.Get_receiver()));
            command.Parameters.Add(new SqlParameter("@Subject",
email.Get_Email_Subject()));
            command.Parameters.Add(new SqlParameter("@Body",
email.Get_Email_Body()));
            command.Parameters.Add(new SqlParameter("@Footer",
email.Get_Email_Footer()));
            command.Parameters.Add(new SqlParameter("@Tag", email.Get_Tag()));
            command.Parameters.Add(new SqlParameter("@Attachment", ""));
            command.Parameters.Add(new SqlParameter("@Date", DateTime.Now));
            command.ExecuteNonQuery();
            connection.Close();
        }
        public Email GetEmail(string subject,string sender)
        {
            SqlConnection connection = new
SqlConnection(Connection.connectionString);
            connection.Open();
            string query = "SELECT * FROM Emails WHERE Sender = @Sender AND Subject
= @Subject";

            SqlCommand command = new SqlCommand(query, connection);

            command.Parameters.AddWithValue("@Sender", sender);
            command.Parameters.AddWithValue("@Subject", subject);

            using (SqlDataReader reader = command.ExecuteReader())
            {

                if (reader.Read())
                {
                    // Populate the user object with data from the database
                    int Email_ID = reader.GetInt32(reader.GetOrdinal("Email_ID"));
                    string Sender = reader.GetString(reader.GetOrdinal("Sender"));
                    string Receiver =
reader.GetString(reader.GetOrdinal("Receiver"));
                    string Subject = reader.GetString(reader.GetOrdinal("Subject"));
                    string Body = reader.GetString(reader.GetOrdinal("Body"));
                    string Footer = reader.GetString(reader.GetOrdinal("Footer"));
                    string Tag = reader.GetString(reader.GetOrdinal("Tag"));



                    Email email = new Email(Email_ID, Sender, Receiver, Subject,
Body, Footer, Tag);
                    return email;
                }
                else
                {
                    // If no matching user is found, return null
                    Email email = new Email(0, "0", "0", "0", "0", "0","0");
                    return null;
                }
            }
        }
        public void UpdateTag(string tag)
        {
            SqlConnection connection = new
SqlConnection(Connection.connectionString);
            connection.Open();
            string query = "UPDATE Emails SET Tag = @Tag WHERE Email_ID =
@Email_ID";
            SqlCommand command = new SqlCommand(query, connection);
            command.Parameters.Add(new SqlParameter("@Tag", tag));
            command.Parameters.Add(new SqlParameter("@Email_ID",
Connection.currentEmail.Get_ID()));
            command.ExecuteNonQuery();
            connection.Close();
        }
    }
}
using Email_client.BL;
using Inboxify_Backend.Interfaces;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inboxify_Backend.DL.FH
{
    internal class FileEmailDL : IEmailDL
    {
        private readonly string filePath = "emails.txt";
        public void AddEmail(Email email)
        {

            string emailData =
$"{email.Get_ID()},{email.Get_Sender()},{email.Get_receiver()},{email.Get_Email_Subj
ect()},{email.Get_Email_Body()},{email.Get_Email_Footer()},{email.Get_Tag()}";

            using (StreamWriter writer = File.AppendText(filePath))
            {
                writer.WriteLine(emailData);
            }
        }
        public Email GetEmail(string subject, string sender)
        {
            string[] lines = File.ReadAllLines(filePath);

            foreach (string line in lines)
            {
                string[] data = line.Split(',');

                if (data[3] == subject && data[1] == sender)
                {
                    Email email = new Email(int.Parse(data[0]), data[1], data[2],
data[3], data[4], data[5], data[6]);
                    return email;
                }
            }
            return null;
        }
        public void UpdateTag(string tag)
        {

            string[] lines = File.ReadAllLines(filePath);

            List<string> updatedLines = new List<string>();

            foreach (string line in lines)
            {
                string[] data = line.Split(',');

                if (int.Parse(data[0]) == Connection.currentEmail.Get_ID())
                {

                    data[6] = tag;
                }

                string updatedLine = string.Join(",", data);

                updatedLines.Add(updatedLine);
            }

            File.WriteAllLines(filePath, updatedLines);
        }
    }
}
using Email_client.BL;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inboxify_Backend.Interfaces
{
    internal interface IEmailDL
    {
        void AddEmail(Email email);
        Email GetEmail(string subject, string sender);

    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inboxify_console.UIConsole
```

```
{
    public class AdminMenuUI
    {
        public static int PrintMenu()
        {
            Console.WriteLine("======================== Admin Authentication 
=======================");
            Console.WriteLine("======================== 1.     Log In 
=======================");
            Console.WriteLine("======================== 2.     Exit 
=======================\n\n");
            Console.WriteLine("Enter your choice: ");
            int choice = int.Parse(Console.ReadLine());
            return choice;
        }
        public static int PrintMainMenu()
        {
            Console.WriteLine("\n====================          Admin Menu 
====================");
            Console.WriteLine("==================== 1. View Users 
==================");
            Console.WriteLine("==================== 2. Make Admin 
==================");
            Console.WriteLine("==================== 3. Make User 
==================");
            Console.WriteLine("==================== 4. View Stats 
==================");
            Console.WriteLine("==================== 5. Exit 
==================");

            Console.WriteLine("Enter your choice: ");
            int choice = int.Parse(Console.ReadLine());
            return choice;
        }
        public static void FailCred()
        {
            Console.WriteLine("Authentication failed. Please try again.");
            Console.ReadKey();
        }

    }
}
using Email_client.DL;
using Inboxify_Backend.DL;
using Inboxify_Backend.DL.User;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inboxify_Backend.Interfaces
{
    public static class ObjectHandler
    {
        public static AdminDL AdminDL = new AdminDL();
        public static EndUserDL EndUserDL = new EndUserDL();
        public static EmailAccountDL EmailAccountDL = new EmailAccountDL();
        public static EmailDL EmailDL = new EmailDL();

    }
}
using Email_client.BL;
using Inboxify_Backend.BL.Emails;
using Inboxify_Backend.Interfaces;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Security.Principal;
using System.Text;
using System.Threading.Tasks;

namespace Inboxify_Backend.DL.User
{
    public class EmailAccountDL : IEmailAccount
    {
        public List<Account> GetAllAccounts(Users user)
        {
            List<Account> accounts = new List<Account>();

            SqlConnection con = new SqlConnection(Connection.connectionString);
            con.Open();
            string query = "SELECT * FROM Accounts WHERE User_Id = @UserId";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                Account Account = new Account(rdr.GetInt32(1), rdr.GetString(2),
rdr.GetString(3));
                accounts.Add(Account);
            }
            con.Close();

            return accounts;
        }
        public List<Account> GetAllAccounts()
        {
            List<Account> accounts = new List<Account>();

            SqlConnection con = new SqlConnection(Connection.connectionString);
            con.Open();
            string query = "SELECT * From Accounts";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                Account Account = new Account(rdr.GetInt32(1), rdr.GetString(2),
rdr.GetString(3));
                accounts.Add(Account);
            }
            con.Close();

            return accounts;
        }
        public void Add_Account(Account account)
        {
            SqlConnection con = new SqlConnection(Connection.connectionString);
            con.Open();
            string query = "INSERT INTO Accounts (User_Id, Email, AccountType)
VALUES (@UserId, @Email, @AccountType)";
            SqlCommand command = new SqlCommand(query, con);
            command.Parameters.AddWithValue("@UserId", account.UserId);
            command.Parameters.AddWithValue("@Email", account.EmailAccount);
            command.Parameters.AddWithValue("@AccountType", account.EmailType);
            command.ExecuteNonQuery();
            con.Close();
        }
        public List<Account> GetAllBussinessAccounts()
        {
            List<Account> accounts = new List<Account>();

            SqlConnection con = new SqlConnection(Connection.connectionString);
            con.Open();
            string query = "SELECT * From Accounts WHERE AccountType = 'Bussiness'";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                Account Account = new Account(rdr.GetInt32(1), rdr.GetString(2),
rdr.GetString(3));
                accounts.Add(Account);
            }
            con.Close();

            return accounts;
        }
        public List<Account> GetAllPersonalAccounts()
        {
            List<Account> accounts = new List<Account>();

            SqlConnection con = new SqlConnection(Connection.connectionString);
            con.Open();
            string query = "SELECT * From Accounts WHERE AccountType = 'Personal'";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                Account Account = new Account(rdr.GetInt32(1), rdr.GetString(2),
rdr.GetString(3));
                accounts.Add(Account);
            }
            con.Close();

            return accounts;
        }
        public Account GetAccount(string Email_Account)
        {
            SqlConnection con = new SqlConnection(Connection.connectionString);
            con.Open();
            string query = "SELECT * FROM Accounts WHERE Email = @Email";
            SqlCommand command = new SqlCommand(query, con);
            command.Parameters.AddWithValue("@Email", Email_Account);
            SqlDataReader reader = command.ExecuteReader();
            Account account = new Account();
            if (reader.Read())
            {
                // Populate the account object with data from the database
                account.UserId = reader.GetInt32(reader.GetOrdinal("User_Id"));
                account.EmailAccountId = 
reader.GetInt32(reader.GetOrdinal("Email_Account_Id"));
                account.EmailAccount = reader.GetString(reader.GetOrdinal("Email"));
                account.EmailType = 
reader.GetString(reader.GetOrdinal("AccountType"));
            }
            con.Close();
            return account;

        }

    }
}
using Email_client.BL;
using Inboxify_Backend.Interfaces;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Inboxify_Backend.DL
{
    public class EndUserDL : IUserDL
    {
        public void Add_User(Users user)
        {
            SqlConnection connection = new 
SqlConnection(Connection.connectionString);
            connection.Open();

            string query = "INSERT INTO Users(username, password, perms) VALUES 
(@username, @password, @perms)";
            SqlCommand command = new SqlCommand(query, connection);

            command.Parameters.Add(new SqlParameter("@username", user.Get_Email()));
            command.Parameters.Add(new SqlParameter("@password", 
user.Get_Password()));
            command.Parameters.Add(new SqlParameter("@perms", 
user.Get_User_Perm()));
            command.ExecuteNonQuery();
            connection.Close();
        }
        public Users Check_Credentials(string email, string password)
        {
            using (SqlConnection connection = new 
SqlConnection(Connection.connectionString))
            {
                connection.Open();
                string query = "SELECT * FROM Users WHERE username = @username AND 
Password = @Password";
                using (SqlCommand cmd = new SqlCommand(query, connection))
                {
                    cmd.Parameters.AddWithValue("@username", email);
                    cmd.Parameters.AddWithValue("@Password", password);

                    using (SqlDataReader rdr = cmd.ExecuteReader())
                    {
                        if (rdr.Read())
                        {
                            return Connection.currentUser = new
Users(rdr.GetString(1), rdr.GetString(2), rdr.GetString(3));
                        }
                    }
                }
            }
            return null;
        }
        public List<Users> GetUserList()
        {
            List<Users> UserList = new List<Users>();

            SqlConnection con = new SqlConnection(Connection.connectionString);
            con.Open();
            string query = "SELECT * From Users Where perms = 'EndUser'";
            SqlCommand cmd = new SqlCommand(query, con);
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                Users user = new Users(rdr.GetString(1), rdr.GetString(2),
rdr.GetString(3));
                UserList.Add(user);
            }
            con.Close();

            return UserList;
        }
        public Users FillInfo(string email, string password)
        {
            SqlConnection connection = new
SqlConnection(Connection.connectionString);
            connection.Open();
            string query = "SELECT * FROM Users WHERE username = @username AND
password = @password";

            SqlCommand command = new SqlCommand(query, connection);

            command.Parameters.AddWithValue("@username", email);
            command.Parameters.AddWithValue("@password", password);

            using (SqlDataReader reader = command.ExecuteReader())
                        {

                            if (reader.Read())
                            {
                                // Populate the user object with data from the
database
                                int id =
reader.GetInt32(reader.GetOrdinal("UserID"));
```

```
                                string email1 = 
reader.GetString(reader.GetOrdinal("username"));
                                string password1 = 
reader.GetString(reader.GetOrdinal("password"));
                                string perms = 
reader.GetString(reader.GetOrdinal("perms"));
                    Users user = new Users(id,email1,password1,perms);
                    return user;
                }
                            else
                            {
                                // If no matching user is found, return null
                                return null;
                            }
                        }




        }
    }
}
```