Bank Management System

Session: 2022 – 2026

Submitted by:

Mohammad Zaid 2022R\2021-CS-214

Supervised by:

Prof. Dr. Muhammad Awais Hassan

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Table of Contents

Youtube Video Link

Complete Journey: https://youtu.be/PndvBJ_Ufa8

Introduction

Overview

Bank Management System is a backend management application developed in C#, catering to two types of users: Admin and Customer. It utilizes object-oriented programming concepts to organize and manage various tasks within a Bank environment. The system provides efficient management and operational support, enabling seamless workflow and enhanced customer service.

Objectives

The objectives of this system includes

1. **Efficient Backend Management**: The system aims to streamline backend operations by providing tools and functionalities to effectively manage users their amounts and accounts.
2. **Improved Customer Service**: By maintaining customer records, processing deposites and transfers accurately, and offering support, the system aims to enhance customer satisfaction and provide personalized services.

Functionality

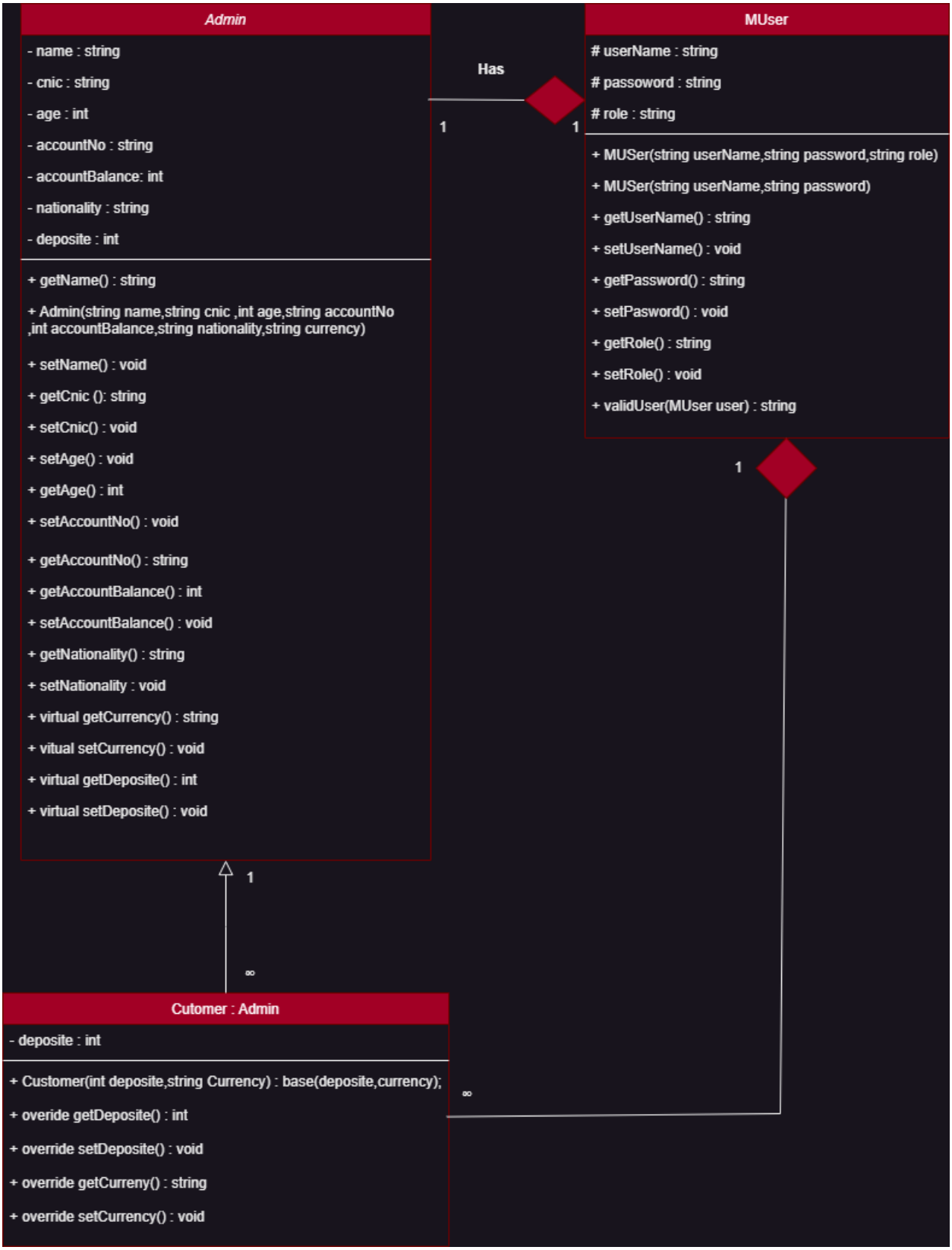
Admin:

1. **Admin Management:** Add, edit, or delete user and beverage accounts, search uaser, manage accounts, and update attributes.

Customer:

1. **Customer Management:** Deposite,transfer their amounts check their account balance,check their account number and user name,change their user password;
2. **Deposite Processing:** Give customer his deposite amount then Admin check his amount and his amount deposited successfully.

Class Responsibility Collaboration Diagrams



Object Oriented Programming

Object-oriented programming (OOP) is a programming paradigm based on the concepts of the “objects”, which contains data and code called as attributes and behavior of the class respectively. The main concepts of OOP includes association, inheritance and polymorphism. I have used this programming paradigm in my project.

○ Association

In Object-oriented programming, Association is a relation between two separate classes which establishes through their objects. Association can be one-to-one, one-to-many, many-to-one, many-to-many. There are two types of Association, **Aggregation** and **Composition**. I have used both Composition & Aggregation in my project. It is used in two places.

1. Admin and Customer class have Credential object. It is a relation of one-to-one because an admin class and customer class contains only one user object. Credentials cannot exist without admin and customer. So, I have used composition in this context.
2. Admin class have lists of Customers. It is a relation of one-to-many because a Admin class contains many Customer. Because Customer can exist without the object of Admin, so I used aggregation for this particular purpose.

Advantage

If I compare this with my procedural programming concepts, I can observe that there is a clear advantage of OOP. There was disjoint data of Customers in procedural programming which is rectified in OOP. Now the lists of Customer are within the classes of Admin respectively.

○ Inheritance

Inheritance is one of the core concepts of Object-oriented programming approach. It is a feature that allows a new class to derive from an existing class. The new class inherits all the public or protected attributes and the member functions of the base class. I have used this OOP concept in one place in this management system.

1. Admin class is a parent class. Customer class are derived from this class. Customer is only user. So, that is the reason I have applied the concept of inheritance here as they are inheriting user object which contains name, password, role,the MUser class.

Advantage

Inheritance gives various advantages over procedural programming. It promotes code-

reusability and reduces redundancy. It helps in organizing the program's structure. It allows flexibility in the code as you will adjust in one place and the rest of the code will work smoothly.

○ Polymorphism

Polymorphism is also one of the core concepts of Object-oriented programming approach. This concept refers to the ability of a function to perform multiple operation under different circumstances. There are two types of Polymorphism. The type of polymorphism used to extend the functionality of common functions in parent and child classes is called Dynamic Polymorphism. I have used Dynamic Polymorphism in few places.

1. ToString() functions in inheritance (Admin – Customer).This function in parent class of Admin will return the name as chosen by the user. But when it is overridden in child classes, for instance in "Customer" class, it will return same name, it will also return Currency and Deposite.
2. I have also used this type of Polymorphism in startInterface() function. This function would load the interface of the user according to their role when logged in correctly.

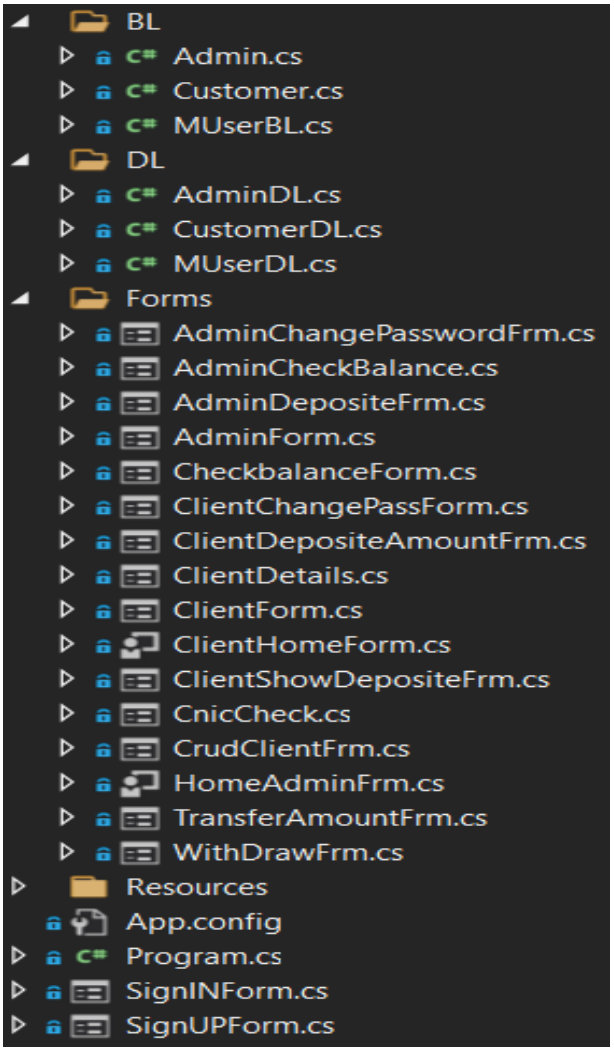
The other type of Polymorphism is Static Polymorphism. I have used this polymorphism only for the constructors.

Advantage

Polymorphism allowed us to extend the functionalities from the base class to use them for the child classes. Due to polymorphism, our code has become shorter because it didn't required us to copy paste the whole code and then make changes to it. Dynamic Polymorphism has enabled the programmer to use the same function in different manner. We lacked this when we were making our projects in procedural programming.

• Design Pattern Implementation

The directory structure for the project is given below



- **Business Logic (BL)**

  The Business Logic folder contains the main classes of the project. It includes the classes of Credentials,Admin,Customer.

- **Data Layer (DL)**

  The Data Layer folder contains the static Lists and functions of the project. It contains the list of users which contains the objects of Customers. There are static functions of each class as well such as storing and loading of data to and from files. It also includes the other static functions.

- **User Interface (UI Forms)**

  This folder contains all the code for Windows Form (.Net Framework) and input of the data. This folder deals with the interaction of user with the application. It includes the menus and other functions which are used for Showing Windows form and taking inputs

from the user.

Class Details

- ## Admin

  An Employee class is used for generalizing the only class of users. It is a parent class of these two Customers. It contains the common attributes of the class. It contains the user object of MUser class since every Admin and Customer object created will have their own credentials.

- ## Customer

  This class inherits the attributes and behavior of the parent class Admin. Its attributes includes the list of customers and function to add the objects of Customers to their respective list. All the attributes of the Customer class are private and are accessed by the getter and setter functions.

- ## MUser

  Credentials class contains strings of username, password and role of the user. Its attributes are also private. This class is created to facilitate in the sign-in and sign-up procedure. There are two constructors in this class. One constructor with three arguments are used when sign-up functionality is used. The other constructor with two arguments is used when we use sign-in function. So, that Customer and Admin can only login through his/her id once signing up.

Conclusion

In conclusion, the Bank Management System is built using the object-oriented programming approach. Its key functionalities includes the CRUD operations. Important concepts object-oriented concepts such as association, inheritance and polymorphism are used in this system. I faced several challenges during this phase. I faced difficulty in designing an effective class diagram collaboration model for the system and managing the key concepts of OOP paradigm. Throughout the period of designing, production and development of this project, I have learned how to create an effective system using object-oriented theory. The object-oriented approach can be really helpful in scaling of the project. It also helps the programmers in future to maintain the software.

CODE

MUserBL Class:

```
namespace GUI_BMS_Project.BL
{
    class MUserBL
    {
        private string userName;
        private string password;
        private string role;

        public string UserName { get => userName; set => userName = value; }
        public string Password { get => password; set => password = value; }
        public string Role { get => role; set => role = value; }

        public MUserBL(string userName, string password, string role)
        {
            this.UserName = userName;
            this.Password = password;
            this.Role = role;
        }

        public MUserBL(string userName, string password)
        {
            this.UserName = userName;
            this.Password = password;
            this.Role = "NA";
        }

        /*public bool isAdmin()
        {
            if(role == "Admin")
            {
                return true;
            }
            else if(role == "Customer")
            {
                return true;
            }
            return false;
        }*/


        public static string isvalidUser(MUserBL existingUser)
        {
            foreach (MUserBL user in MUserDL.userList)
            {
                if (user.UserName == existingUser.UserName && user.Password ==
existingUser.Password)
                {
```

```
                    return user.Role;
                }
            }
            return "User NOT Found!!!";
        }
    }
}
```

MUserDL Class

```
namespace GUI_BMS_Project.DL
{
    class MUserDL
    {
        public static List<MUserBL> userList = new List<MUserBL>();
        public static string path = "Credentials.txt";
        public static void addUserIntoList(MUserBL user)
        {
            bool flag = true;
            foreach (MUserBL us in userList)
            {
                if (us.UserName == user.UserName && us.Password == user.UserName)
                {
                    flag = false;
                }
            }
            if (flag == true)
            {
                userList.Add(user);
                storeUserIntoFile();
            }
        }

        public static MUserBL SignIn(MUserBL user)
        {
            foreach (MUserBL storedUser in userList)
            {
                if (storedUser.UserName == user.UserName && storedUser.Password ==
user.Password)
                {
                    return storedUser;
                }
            }
            return null;
        }

        public static string parseData(string record, int field)
        {
            int comma = 1;
```

```
            string item = "";

            for (int x = 0; x < record.Length; x++)
            {
                if (record[x] == ',')
                {
                    comma++;
                }

                else if (comma == field)
                {
                    item = item + record[x];
                }
            }
            return item;
        }

        public static bool readDataFromFile()
        {
            if (File.Exists(path))
            {
                StreamReader fileVariable = new StreamReader(path);
                string record;
                while ((record = fileVariable.ReadLine()) != null)
                {
                    string userName = parseData(record, 1);
                    string password = parseData(record, 2);
                    string role = parseData(record, 3);
                    MUserBL user = new MUserBL(userName, password, role);
                    userList.Add(user);
                }
                fileVariable.Close();
                return true;
            }
            else
                return false;
        }

        public static void storeUserIntoFile()
        {
            StreamWriter file = new StreamWriter(path);
            foreach (MUserBL user in userList)
            {
                file.WriteLine(user.UserName + "," + user.Password + "," + user.Role);
            }
            file.Flush();
            file.Close();
        }
    }

}
```

Admin Class:

```
namespace GUI_BMS_Project.BL
{
    class Admin
    {
        private string name;
        private string cnic;
        private int age;
        private string accountNo;
        private int accBalance;
        private string nationality;
        private string currency;
        private int deposite;

        //private variables-----------------------------------

        public Admin(int deposite, string currency)
        {
            this.Deposite = deposite;
            this.Currency = currency;
        }

        public Admin(string name, string cnic, int age, string acountNo, int accBalance,
int deposite, string nationality, string currency)
        {
            this.Name = name;
            this.Cnic = cnic;
            this.Age = age;
            this.AccountNo = acountNo;
            this.AccBalance = accBalance;
            this.Deposite = deposite;
            this.Nationality = nationality;
            this.Currency = currency;
        }

        //Constructors=---------------------------------------

        public virtual string Name { get => name; set => name = value; }
        public virtual string Cnic { get => cnic; set => cnic = value; }
        public virtual int Age { get => age; set => age = value; }
        public virtual string AccountNo { get => accountNo; set => accountNo = value; }
        public virtual int AccBalance { get => accBalance; set => accBalance = value; }
        public virtual string Nationality { get => nationality; set => nationality =
value; }
        public virtual string Currency { get => currency; set => currency = value; }
        public virtual int Deposite { get => deposite; set => deposite = value; }
    }
}
```

AdminDL Class:

```
namespace GUI_BMS_Project.DL
{
    class AdminDL
    {
        public static List<Admin> adminUserList = new List<Admin>();
        public static string path = "Client.txt";
        public static void addAdminUserIntoList(Admin User)
        {
            if (User.Name != null && User.Cnic != null)
            {
                adminUserList.Add(User);
            }
        }

        public static string parseData(string record, int field)
        {
            int comma = 1;
            string item = "";

            for (int x = 0; x < record.Length; x++)
            {
                if (record[x] == ',')
                {
                    comma++;
                }

                else if (comma == field)
                {
                    item = item + record[x];
                }
            }
            return item;
        }

        public static bool deleteuserRecords(string name, string accnum, string path)
        {
            for (int x = 0; x < AdminDL.adminUserList.Count; x++)
            {
                if (name == AdminDL.adminUserList[x].Name && accnum ==
AdminDL.adminUserList[x].AccountNo)
                {
                    AdminDL.adminUserList[x].Name = null;
                    AdminDL.adminUserList[x].Cnic = null;
                    AdminDL.adminUserList[x].Age = 0;
                    AdminDL.adminUserList[x].AccountNo = null;
                    AdminDL.adminUserList[x].Deposite = 0;
                    AdminDL.adminUserList[x].AccBalance = 0;
                    AdminDL.adminUserList[x].Nationality = null;
                    AdminDL.adminUserList[x].Currency = null;
                    storeClientIntoFile();
                    return true;
                }
```

```
            }
            return false;
        }

        public static bool UpdateUserRecord(string name, string accnum, string path)
        {
            foreach (Admin adm in adminUserList)
            {
                if (name == adm.Name && accnum == adm.AccountNo)
                {
                    Console.WriteLine("\nEnter the New Name: ");
                    string newname = Console.ReadLine();
                    adm.Name = newname;

                    Console.WriteLine("Enter the New CNIC: ");
                    string cnic = Console.ReadLine();
                    adm.Cnic = cnic;

                    Console.WriteLine("Enter the New Age: ");
                    int age = int.Parse(Console.ReadLine());
                    adm.Age = age;

                    Console.WriteLine("Enter the New Nationality: ");
                    string nationality = Console.ReadLine();
                    adm.Nationality = nationality;

                    Console.WriteLine("Enter the New Currency you Want in your Account:
");
                    string currency = Console.ReadLine();
                    adm.Currency = currency;

                    AdminDL.storeClientIntoFile();
                    return true;
                }

            }
            return false;
        }

        public static bool EditPassword(string name, string password, string path)
        {
            for (int x = 0; x < MUserDL.userList.Count; x++)
            {
                if (name == MUserDL.userList[x].Role && password ==
MUserDL.userList[x].Password)
                {
                    Console.WriteLine("Enter a new Password: ");
                    string pass = Console.ReadLine();
                    MUserDL.userList[x].Password = pass;
                    Console.WriteLine("\nYour New Password is: ");
                    Console.WriteLine(MUserDL.userList[x].Password);
                    string nam = MUserDL.userList[x].UserName;
                    string role = MUserDL.userList[x].Role;
                    MUserDL.storeUserIntoFile();
                    return true;
                }
            }
            return false;
```

```
        }

        public static void ReadClientDataFromFile()
        {
            if (File.Exists(path))
            {
                StreamReader fileVariable = new StreamReader(path);
                string record;
                while ((record = fileVariable.ReadLine()) != null)
                {
                    string[] fields = record.Split(',');
                    if (fields.Length >= 8)
                    {
                        string name = fields[0].Trim();
                        string cnic = fields[1].Trim();
                        int age = int.Parse(fields[2].Trim());
                        string account = fields[3].Trim();
                        int balance = int.Parse(fields[4].Trim());
                        int deposit = int.Parse(fields[5].Trim());
                        string nation = fields[6].Trim();
                        string currency = fields[7].Trim();
                        Admin adm = new Admin(name, cnic, age, account, balance, deposit,
nation, currency);
                        adminUserList.Add(adm);
                    }
                }
                fileVariable.Close();
            }
        }


        public static void storeClientIntoFile()
        {
            StreamWriter file = new StreamWriter(path, false);
            foreach (Admin adm in adminUserList)
            {
                if (adm.Name != "" && adm.Cnic != null && adm.Nationality != null)
                {
                    file.WriteLine(adm.Name + "," + adm.Cnic + "," + adm.Age + "," +
adm.AccountNo + "," + adm.AccBalance + "," + adm.Deposite + "," + adm.Nationality + "," +
adm.Currency);
                }
            }
            file.Flush();
            file.Close();
        }
    }
}
```

Customer Class:

```
class Customer:Admin
```

```
    {
        public Customer(int deposite, string currency) : base(deposite, currency)
        {
        }

        /*public Customer(string currency, double trans) : base(currency)
        {
            this.transfer = trans;
        }*/


        public override string AccountNo { get => base.AccountNo; set => base.AccountNo =
value; }
        public override int AccBalance { get => base.AccBalance; set => base.AccBalance =
value; }
        public override int Deposite { get => base.Deposite; set => base.Deposite =
value; }

    }
```

CustomerDL Class:

```
namespace GUI_BMS_Project.DL
{
    class CustomerDL
    {
        public static bool ChangeClientPassword(string rol, string usname, string
password, string path)
        {
            foreach (MUserBL us in MUserDL.userList)
            {
                if (rol.ToLower() == "customer" && usname == us.UserName && password ==
us.Password)
                {
                    Console.WriteLine("Enter the UserName: ");
                    string uName = Console.ReadLine();

                    Console.WriteLine("Enter a new Password: ");
                    string pass = Console.ReadLine();

                    Console.WriteLine("Your Old UserName Was: " + us.UserName);
                    Console.WriteLine("Your Old Password Was: " + us.Password);

                    us.UserName = uName;
                    us.Password = pass;

                    Console.WriteLine("\nYour New User Name is: " + us.UserName);
                    Console.WriteLine("\nYour New Password is: " + us.Password);
                    string role = us.Role;
                    MUserDL.storeUserIntoFile();
                    return true;
```

```
                    //break;
                }
            }
            return false;
        }
    }

}
```

SignUpFrm Form:

```
namespace GUI_BMS_Project
{
    public partial class SignUPForm : Form
    {
        public SignUPForm()
        {
            InitializeComponent();
        }

        private void BtnSignUp_Click(object sender, EventArgs e)
        {
            if (textName.Text != "" && textPassword.Text != "" && textRole.Text != "")
            {
                string name = textName.Text;
                string Passw = textPassword.Text;
                string roll = textRole.Text;
                MUserBL U3 = new MUserBL(name, Passw, roll);
                MUserBL user = MUserDL.userList.Find(u => u.UserName == name &&
u.Password == Passw);
                if (user != null)
                {
                    lblShow_Click(sender, e);
                    Thread.Sleep(600);
                    lblShow_Click(sender, e);
                }
                else
                {
                    MUserBL U1 = new MUserBL(name, Passw, roll);
                    MUserDL.addUserIntoList(U1);
                    MUserDL.storeUserIntoFile();
                    MessageBox.Show("Sign Up Successfully");
                    textName.Text = string.Empty;
                    textPassword.Text = string.Empty;
                    textRole.Text = string.Empty;
                }
            }
            else
            {
                lblShow.Text = "Please Provide Information !!!";
```

```
                lblShow.Visible = true;
            }
        }

        private void lblShow_Click(object sender, EventArgs e)
        {
            lblShow.Visible = true;
            textName.Text = string.Empty;
            textPassword.Text = string.Empty;
            textRole.Text = string.Empty;

        }

        private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
        {
            this.Hide();
            SignINForm SI1 = new SignINForm();
            SI1.Show();
        }

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

SignInFrm Form:

```
namespace GUI_BMS_Project.DL
{
    public partial class SignINForm : Form
    {
        public SignINForm()
        {
            InitializeComponent();
        }

        private void BtnSignIn_Click(object sender, EventArgs e)
        {
            if (TxtName.Text != "" && TxtPassword.Text != "")
            {
                string name = TxtName.Text;
                string pas = TxtPassword.Text;
```

```
                MUserBL user1 = new MUserBL(name, pas);
                MUserBL user2 = MUserDL.userList.Find(u => u.UserName == name &&
u.Password == pas && (u.Role == "Admin" || u.Role == "admin") || (u.Role == "Client" ||
u.Role == "client"));
                if (user2 != null)
                {
                    lblWarning.Hide();
                    if ((user2.Role == "Admin" || user2.Role == "admin") &&
TxtPassword.Text == user2.Password && TxtName.Text == user2.UserName)
                    {
                        MessageBox.Show("SignIn Successfully !!!");
                        this.Hide();
                        AdminForm f1 = new AdminForm();
                        f1.Show();
                    }

                    if ((user2.Role == "Client" || user2.Role == "client") &&
TxtPassword.Text == user2.Password && TxtName.Text == user2.UserName)
                    {
                        MessageBox.Show("Sign In Successfully !!!");
                        this.Hide();
                        CnicCheck cc = new CnicCheck();
                        cc.Show();
                    }
                    else
                    {
                        lblWarning_Click(sender, e);
                    }
                }
                else
                {
                    lblWarning_Click(sender, e);
                }
            }
            else
            {
                lblWarning.Text = "Provide UserNAme and Password";
                lblWarning.Visible = true;
            }
        }

        private void lblWarning_Click(object sender, EventArgs e)
        {
            lblWarning.Text = "⚠ Invalid Username and Password ⚠";
            lblWarning.Visible = true;
            TxtName.Text = string.Empty;
            TxtPassword.Text = string.Empty;
        }

        private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
        {
            this.Hide();
            SignUPForm SF1 = new SignUPForm();
            SF1.Show();
        }

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

```
        {
            Application.Exit();
        }
    }
}
```

CheckCnicForClient Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class CnicCheck : Form
    {
        public CnicCheck()
        {
            InitializeComponent();
        }

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

        private void label3_Click(object sender, EventArgs e)
        {
            this.Hide();
            SignINForm si = new SignINForm();
            si.Show();
        }

        private void btnCheck_Click(object sender, EventArgs e)
        {
            if (txtCnicCheck.Text != "")
            {
                int c = 0;
                for (int i = 0; i < AdminDL.adminUserList.Count; i++)
                {
                    if (AdminDL.adminUserList[i].Cnic == txtCnicCheck.Text)
                    {
                        this.Hide();
                        ClientForm cf = new ClientForm(i);
                        cf.Show();
                    }
                    else
                    {
                        c++;
                    }
                }
                if(c == AdminDL.adminUserList.Count)
                {
```

```
                    MessageBox.Show("Please Enter Valid Cnic !!!");
                    txtCnicCheck.Text = string.Empty;
                }
            }
            else
            {
                MessageBox.Show("Please Provide Information !!!");
            }
        }
    }
}
```

AdminForm Form:

```
namespace GUI_BMS_Project
{
    public partial class AdminForm : Form
    {
        public AdminForm()
        {
            InitializeComponent();
        }

        private void btnAddClient_Click(object sender, EventArgs e)
        {
            this.Hide();
            CrudClientFrm ccf = new CrudClientFrm();
            ccf.Show();
        }

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

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            this.Hide();
            SignINForm SI = new SignINForm();
            SI.Show();
        }

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

```
            CheckBalance ccf = new CheckBalance();
            ccf.Show();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Hide();
            AdminDepositeFrm adf = new AdminDepositeFrm();
            adf.Show();
        }

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

        private void lblmenu_Click(object sender, EventArgs e)
        {
            if (panel5.Visible == false)
            {
                panel4.Visible = true;
                panel5.Visible = true;
                panel6.Visible = true;
                panel7.Visible = true;
                panel8.Visible = true;
            }
            else
            {
                panel4.Visible = false;
                panel5.Visible = false;
                panel6.Visible = false;
                panel7.Visible = false;
                panel8.Visible = false;
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
            AdminChangePasswordFrm acpf = new AdminChangePasswordFrm();
            acpf.Show();
        }
    }
}
```

CrudClientFrm Form:

```
namespace GUI_BMS_Project
{
    public partial class CrudClientFrm : Form
    {
        public CrudClientFrm()
        {
            InitializeComponent();
            GridViewClient.SelectionChanged += dataGridView1_SelectionChanged;
        }

        private void BtnAdd_Click(object sender, EventArgs e)
        {
            if (TxtAge.Text != "" && TxtName.Text != "")
            {
                string n = TxtName.Text;
                string cnic = TxtCNIC.Text;
                int age = int.Parse(TxtAge.Text);
                string Account_no = TxtAccount.Text;
                int initial = int.Parse(TxtBalance.Text);
                int dep = 0;
                string Nat = TxtNationality.Text;
                string Curr = TxtCurrency.Text;
                if (age > 18 && age < 60 && initial > 0)
                {
                    Admin AddClientObj = new Admin(n, cnic, age, Account_no, initial,
dep, Nat, Curr);
                    AdminDL.addAdminUserIntoList(AddClientObj);
                    AdminDL.storeClientIntoFile();
                    MessageBox.Show("Added Successfully!!!");
                    GridViewClient.DataSource = null;
                    GridViewClient.DataSource = AdminDL.adminUserList;
                    GridViewClient.Refresh();
                }
                else
                {
                    MessageBox.Show("Age greater than 18 / Balance greater than 0 !!!");
                }
            }
            else
            {
                MessageBox.Show("Please Provide Information !!!");
            }
        }

        private void CrudClientFrm_Load(object sender, EventArgs e)
        {
            GridViewClient.DataSource = AdminDL.adminUserList;
        }

        private void dataGridView1_SelectionChanged(object sender, EventArgs e)
        {
            if (GridViewClient.SelectedRows.Count > 0)
            {
                DataGridViewRow selectedRow = GridViewClient.SelectedRows[0];
```

```
            // Retrieve the values from the selected row
            string Name = selectedRow.Cells["Name"].Value.ToString();
            string cnic = selectedRow.Cells["CNIC"].Value.ToString();
            int age = Convert.ToInt32(selectedRow.Cells["Age"].Value);
            string Acc_num = selectedRow.Cells["Accountno"].Value.ToString();
            int Acc_balance = Convert.ToInt32(selectedRow.Cells["AccBalance"].Value);
            string national = selectedRow.Cells["Nationality"].Value.ToString();
            string currency = selectedRow.Cells["Currency"].Value.ToString();

            // Update the text boxes with the retrieved values
            TxtName.Text = Name;
            TxtCNIC.Text = cnic;
            TxtAge.Text = age.ToString();
            TxtAccount.Text = Acc_num;
            TxtBalance.Text = Acc_balance.ToString();
            TxtNationality.Text = national;
            TxtCurrency.Text = currency;
        }
    }

    private void btnDelete_Click(object sender, EventArgs e)
    {
        Admin M = (Admin)GridViewClient.CurrentRow.DataBoundItem;
        AdminDL.adminUserList.Remove(M);
        AdminDL.storeClientIntoFile();
        GridViewClient.DataSource = null;
        GridViewClient.DataSource = AdminDL.adminUserList;
        GridViewClient.Refresh();
        MessageBox.Show("Deleted Successfully!!!");
    }

    private void btnEdit_Click(object sender, EventArgs e)
    {
        if (GridViewClient.SelectedRows.Count > 0)
        {
            DataGridViewRow selectedRow = GridViewClient.SelectedRows[0];

            string n = TxtName.Text;
            string cnic = TxtCNIC.Text;
            int age = int.Parse(TxtAge.Text);
            string Account_no = TxtAccount.Text;
            int initial = int.Parse(TxtBalance.Text);
            int dep = 0;
            string Nat = TxtNationality.Text;
            string Curr = TxtCurrency.Text;

            if (age > 18 && age < 60 && initial > -1)
            {

                selectedRow.Cells["Name"].Value = n;
                selectedRow.Cells["CNIC"].Value = cnic;
                selectedRow.Cells["Age"].Value = age;
                selectedRow.Cells["Accountno"].Value = Account_no;
                selectedRow.Cells["AccBalance"].Value = initial;
                selectedRow.Cells["Nationality"].Value = Nat;
                selectedRow.Cells["Currency"].Value = Curr;
```

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

                    Admin adm = new Admin(n, cnic, age, Account_no, initial, dep, Nat,
Curr);
                    //AdminDL.addAdminUserIntoList(adm);
                    AdminDL.storeClientIntoFile();
                }
                else
                {
                    MessageBox.Show("Age greater than 18 / Balance greater than 0 !!!");
                }

            }
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            TxtName.Text = string.Empty;
            TxtCNIC.Text = string.Empty;
            TxtAge.Text = string.Empty;
            TxtAccount.Text = string.Empty;
            TxtBalance.Text = string.Empty;
            TxtNationality.Text = string.Empty;
            TxtCurrency.Text = string.Empty;
        }

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            AdminForm af = new AdminForm();
            af.Show();
        }

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

AdminCheckBalance Form:

```
namespace GUI_BMS_Project
{
    public partial class CheckBalance : Form
```

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

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

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            AdminForm af = new AdminForm();
            af.Show();
        }

        private void btnCheck_Click(object sender, EventArgs e)
        {
            string n = TxtName.Text;
            string a = txtAccountNo.Text;
            SignINForm frm = new SignINForm();
            int c = 0;
            label4.Visible = false;

            for (int i = 0; i < AdminDL.adminUserList.Count; i++)
            {
                if (AdminDL.adminUserList[i].Name == n &&
AdminDL.adminUserList[i].AccountNo == a)
                {
                    txtBalance.Text = AdminDL.adminUserList[i].AccBalance.ToString();
                    label4.Text = AdminDL.adminUserList[i].Name;
                    label4.Visible = true;
                    label5.Visible = true;
                }
                else
                {
                    c++;
                }
            }
            if (c == AdminDL.adminUserList.Count)
            {
                MessageBox.Show("Enter Valid Name/Account !!!");
            }
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            txtAccountNo.Text = string.Empty;
            txtBalance.Text = string.Empty;
            TxtName.Text = string.Empty;

            label5.Visible = false;
            label4.Visible = false;
        }
    }
```

```
}
```

AdminDepositeFrm Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class AdminDepositeFrm : Form
    {
        public AdminDepositeFrm()
        {
            InitializeComponent();
        }

        private void btnCheck_Click(object sender, EventArgs e)
        {
            int c = 0;
            if(TxtName.Text != "" && txtAccountNo.Text != "")
            {
                foreach(Admin ad in AdminDL.adminUserList)
                {
                    if(TxtName.Text == ad.Name && txtAccountNo.Text == ad.AccountNo &&
ad.Deposite > 0)
                    {
                        label5.Visible = true;
                        label6.Visible = true;
                        label4.Visible = true;
                        label3.Visible = true;
                        label4.Text = ad.Name;
                        label6.Text = ad.Deposite.ToString();
                        int depBal = ad.AccBalance + ad.Deposite;
                        ad.AccBalance = depBal;
                        string curr = ad.Currency;
                        ad.Currency = curr;
                        ad.Deposite = 0;
                        AdminDL.storeClientIntoFile();
                        MessageBox.Show("Amount Deposited SuccessFully !!!");

                        label5.Visible = false;
                        label6.Visible = false;
                        label4.Visible = false;
                        label3.Visible = false;
                        txtAccountNo.Text = string.Empty;
                        TxtName.Text = string.Empty;
                    }
                    else
                    {
                        c++;
                    }
```

```
                }
                if(c == AdminDL.adminUserList.Count)
                {
                    txtAccountNo.Text = string.Empty;
                    TxtName.Text = string.Empty;
                    MessageBox.Show("Enter Valid Strings or Client may not be deposite
any Amount yet.....");
                    c = 0;
                }
            }
            else
            {
                MessageBox.Show("Please Provide Information !!!");
            }
        }
        private void lblclose_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            AdminForm af = new AdminForm();
            af.Show();
        }
    }

}
```

AdminChangePasswordFrm Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class AdminChangePasswordFrm : Form
    {
        public AdminChangePasswordFrm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int c = 0;
            if (textoldName.Text != "" && textoldPassword.Text != "" && textoldRole.Text
!= "")
            {
                foreach (MUserBL us in MUserDL.userList)
                {
```

```
                    if ((textoldRole.Text.ToLower() == "admin" ||
textoldRole.Text.ToLower() == "client") && textoldName.Text == us.UserName &&
textoldPassword.Text == us.Password)
                    {
                        NewUN.Visible = true;
                        NewP.Visible = true;
                        txtNewName.Visible = true;
                        txtNewPassword.Visible = true;

                        if (txtNewName.Text != "" && txtNewPassword.Text != "")
                        {
                            label2.Visible = true;
                            label7.Visible = true;
                            label10.Visible = true;
                            label11.Visible = true;
                            label12.Visible = true;
                            label13.Visible = true;
                            label14.Visible = true;
                            label15.Visible = true;

                            label7.Text = us.UserName;
                            label11.Text = us.Password;

                            us.UserName = txtNewName.Text;
                            us.Password = txtNewPassword.Text;

                            label12.Text = us.UserName;
                            label15.Text = us.Password;

                            MUserDL.storeUserIntoFile();
                            break;
                        }
                        else
                        {
                            MessageBox.Show("Please Enter New Password And UserName
!!!");
                        }
                    }
                    else
                    {
                        c++;
                    }
                }

                if (c == MUserDL.userList.Count)
                {
                    MessageBox.Show("No Such Account Found... Enter Valid Information
!!!");
                    textoldName.Text = string.Empty;
                    textoldPassword.Text = string.Empty;
                    textoldRole.Text = string.Empty;
                }
            }
            else
            {
                MessageBox.Show("Please Provide InFormation !!!");
            }
        }
```

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

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            AdminForm af = new AdminForm();
            af.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            NewUN.Visible = false;
            NewP.Visible = false;
            txtNewName.Visible = false;
            txtNewPassword.Visible = false;

            label2.Visible = false;
            label7.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
            label12.Visible = false;
            label13.Visible = false;
            label14.Visible = false;
            label15.Visible = false;

            textoldName.Text = string.Empty;
            textoldPassword.Text = string.Empty;
            textoldRole.Text = string.Empty;

            txtNewName.Text = string.Empty;
            txtNewPassword.Text = string.Empty;
        }
    }

}
```

ClientFrm Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class ClientForm : Form
    {
        int index = 0;
        public ClientForm(int x)
        {
```

```
        InitializeComponent();
        index = x;
    }

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

    private void pictureBox1_Click(object sender, EventArgs e)
    {
        this.Hide();
        SignINForm si = new SignINForm();
        si.Show();
    }

    private void btnAddClient_Click(object sender, EventArgs e)
    {
        this.Hide();
        ClientDetails cd = new ClientDetails(index);
        cd.Show();
    }

    private void btnCheckBalance_Click(object sender, EventArgs e)
    {
        this.Hide();
        CheckbalanceForm cbf = new CheckbalanceForm(index);
        cbf.Show();
    }

    private void btnWithDraw_Click(object sender, EventArgs e)
    {
        this.Hide();
        WithDrawFrm wF = new WithDrawFrm(index);
        wF.Show();
    }

    private void btnTransfer_Click(object sender, EventArgs e)
    {
        this.Hide();
        TransferAmountFrm taf = new TransferAmountFrm(index);
        taf.Show();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Hide();
        ClientDepositeAmountFrm cdaf = new ClientDepositeAmountFrm(index);
        cdaf.Show();
    }

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

```
        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientShowDepositeFrm csdf = new ClientShowDepositeFrm(index);
            csdf.Show();
        }

        private void lblmenu_Click(object sender, EventArgs e)
        {
            if(panel4.Visible == false)
            {
                panel4.Visible = true;
                panel5.Visible = true;
                panel6.Visible = true;
                panel7.Visible = true;
                panel8.Visible = true;
                panel9.Visible = true;
                panel10.Visible = true;
                panel11.Visible = true;
            }
            else
            {
                panel4.Visible = false;
                panel5.Visible = false;
                panel6.Visible = false;
                panel7.Visible = false;
                panel8.Visible = false;
                panel9.Visible = false;
                panel10.Visible = false;
                panel11.Visible = false;
            }
        }

        private void button4_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientChangePassForm ccpf = new ClientChangePassForm(index);
            ccpf.Show();
        }
    }

}
```

ClientDetail Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class ClientDetails : Form
    {
```

```
        int idx = 0;
        public ClientDetails(int x)
        {
            InitializeComponent();
            idx = x;

            label3.Visible = true;
            label3.Text = AdminDL.adminUserList[x].Name;

            label4.Visible = true;
            label4.Text = AdminDL.adminUserList[x].AccountNo;

            label2.Visible = true;
            label2.Text = AdminDL.adminUserList[x].Currency;
        }

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

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientForm cf = new ClientForm(idx);
            cf.Show();
        }
    }
}
```

ClientCheckBalance Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class CheckbalanceForm : Form
    {
        int idx = 0;
        public CheckbalanceForm(int x)
        {
            InitializeComponent();
            idx = x;

            label3.Visible = true;
            label3.Text = (AdminDL.adminUserList[x].AccBalance).ToString();
        }
```

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

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientForm cf = new ClientForm(idx);
            cf.Show();
        }
    }

}
```

ClientWithDrawal Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class WithDrawFrm : Form
    {
        int idx = 0;
        int balance;
        public WithDrawFrm(int x)
        {
            InitializeComponent();
            idx = x;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (txtwithDraw.Text != "")
            {
                balance = AdminDL.adminUserList[idx].AccBalance;
                if (int.Parse(txtwithDraw.Text) < balance && int.Parse(txtwithDraw.Text)
> 0)
                {
                    balance = balance - int.Parse(txtwithDraw.Text);
                    AdminDL.adminUserList[idx].AccBalance = balance;
                    AdminDL.storeClientIntoFile();
                    label6.Visible = true;
                    lblwithDraw.Visible = true;
                    lblwithDraw.Text = balance.ToString();
                    MessageBox.Show("Amount WithDraw Successfully !!!");
                    lblwithDraw.Visible = false;
                    label6.Visible = false;
                    txtwithDraw.Text = string.Empty;
                }
```

```
                else
                {
                    MessageBox.Show("Your Amount is not Enough for this Transaction
!!!");
                    txtwithDraw.Text = string.Empty;
                }
            }
            else
            {
                MessageBox.Show("Please Enter Amount !!!");
            }
        }

        private void lblclose_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientForm cf = new ClientForm(idx);
            cf.Show();
        }
    }

}
```

ClientTransfer Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class TransferAmountFrm : Form
    {
        int idx = 0;
        int rm = 0;
        int c = 0;
        public TransferAmountFrm(int x)
        {
            InitializeComponent();
            idx = x;
        }
        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientForm cf = new ClientForm(idx);
            cf.Show();
        }

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

```
        {
            Application.Exit();
        }

        private void btnTransfer_Click(object sender, EventArgs e)
        {
            if (txtTrans.Text != "")
            {
                string accNo = txtTrans.Text;
                foreach (Admin ad in AdminDL.adminUserList)
                {
                    if (accNo == ad.AccountNo && accNo !=
AdminDL.adminUserList[idx].AccountNo)
                    {
                        lbltitle.Visible = true;
                        lblaccno.Visible = true;
                        label8.Visible = true;
                        label4.Visible = true;
                        label8.Text = ad.Name;
                        label4.Text = ad.AccountNo;
                        label2.Visible = true;
                        txttransAmount.Visible = true;
                        if (txttransAmount.Text != "")
                        {
                            int amount = int.Parse(txttransAmount.Text);
                            if (amount > 0 && AdminDL.adminUserList[idx].AccBalance >
amount)
                            {
                                rm = AdminDL.adminUserList[idx].AccBalance - amount;
                                AdminDL.adminUserList[idx].AccBalance = rm;
                                int tr = ad.AccBalance + amount;
                                ad.AccBalance = tr;
                                AdminDL.storeClientIntoFile();
                                label6.Visible = true;
                                lbltrans.Visible = true;
                                lbltrans.Text = rm.ToString();
                                MessageBox.Show("Transaction complete Successfully !!!");

                                label6.Visible = false;
                                lbltrans.Visible = false;
                                lbltitle.Visible = false;
                                lblaccno.Visible = false;
                                label8.Visible = false;
                                label4.Visible = false;
                                label2.Visible = false;
                                txttransAmount.Visible = false;
                                txtTrans.Text = string.Empty;
                                txttransAmount.Text = string.Empty;
                            }
                            else
                            {
                                MessageBox.Show("Your amount is not Enough For this
transaction !!!");
                            }
                        }
                        else
                        {
                            MessageBox.Show("Please Enter Amount !!!");
```

```
                    }
                }
                else
                {
                    c++;
                }
            }

            if (c == AdminDL.adminUserList.Count)
            {
                MessageBox.Show("Please Enter Valid Account Number !!!");
                c = 0;
            }
        }
        else
        {
            MessageBox.Show("Please Provide Information !!!");
        }
    }
}
```

ClientDeposite Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class ClientDepositeAmountFrm : Form
    {
        int idx = 0;
        public ClientDepositeAmountFrm(int x)
        {
            InitializeComponent();
            idx = x;
        }

        private void btnCheck_Click(object sender, EventArgs e)
        {

            if (AdminDL.adminUserList[idx].Deposite == 0)
            {
                if (txtAccountNo.Text != "" && TxtAmount.Text != "" && txtCurrency.Text
!= "")
                {
                    if (int.Parse(TxtAmount.Text) > 0)
                    {
```

```
                        int dep = int.Parse(TxtAmount.Text);

                        string curr = txtCurrency.Text;
                        if (txtAccountNo.Text == AdminDL.adminUserList[idx].AccountNo &&
txtCurrency.Text == AdminDL.adminUserList[idx].Currency)
                        {
                            Customer cust = new Customer(dep, curr);
                            AdminDL.adminUserList[idx].Deposite = dep;
                            AdminDL.adminUserList[idx].Currency = curr;
                            AdminDL.addAdminUserIntoList(cust);
                            AdminDL.storeClientIntoFile();

                            MessageBox.Show("SuccessFully Proceed !!! Your Amount will Be
Deposite When Admin Check it......");

                            txtAccountNo.Text = string.Empty;
                            txtCurrency.Text = string.Empty;
                            TxtAmount.Text = string.Empty;
                        }
                        else
                        {
                            MessageBox.Show("Please Enter Valid Account Number or
Currency Name Please Check Your Details First... !!!");
                            txtAccountNo.Text = string.Empty;
                            txtCurrency.Text = string.Empty;
                        }
                    }
                    else
                    {
                        MessageBox.Show("Deposite must be greater or Equal to 0 !!! Enter
Again.....");
                        TxtAmount.Text = string.Empty;
                    }
                }
                else
                {
                    MessageBox.Show("Please Provide Information !!!");
                }
            }
            else
            {
                MessageBox.Show("Your last amount is not deposited by Admin yet...Please
visit to Admin and Deposite Your Last Amount First !!!");
                this.Hide();
                ClientForm cf = new ClientForm(idx);
                cf.Show();
            }
        }

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

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientForm cf = new ClientForm(idx);
```

```
            cf.Show();
        }
    }
}
```

ClientShowDeposite Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class ClientShowDepositeFrm : Form
    {
        int idx = 0;
        public ClientShowDepositeFrm(int x)
        {
            InitializeComponent();
            idx = x;

            if(AdminDL.adminUserList[idx].Deposite <= 0)
            {
                lbltitle.Visible = true;
                label3.Visible = true;
                label3.Text = AdminDL.adminUserList[idx].Deposite.ToString();
                label2.Visible = true;
                label2.Text = "You can Deposite Your Amount....";
            }
            else
            {
                lbltitle.Visible = true;
                label3.Visible = true;
                label3.Text = AdminDL.adminUserList[idx].Deposite.ToString();
                label2.Visible = true;
                label2.Text = "Please Go To Admin and Deposite Your Last amount....";
            }
        }

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

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientForm cf = new ClientForm(idx);
            cf.Show();
```

```
        }
    }
}
```

ClientChangePassword Form:

```
namespace GUI_BMS_Project.Forms
{
    public partial class ClientChangePassForm : Form
    {
        int idx = 0;
        public ClientChangePassForm(int x)
        {
            InitializeComponent();
            idx = x;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int c = 0;
            if(textoldName.Text != "" && textoldPassword.Text != "" && textoldRole.Text
!= "")
            {
                foreach(MUserBL us in MUserDL.userList)
                {
                    if(textoldRole.Text.ToLower() == "client" && textoldName.Text ==
us.UserName && textoldPassword.Text == us.Password)
                    {
                        NewUN.Visible = true;
                        NewP.Visible = true;
                        txtNewName.Visible = true;
                        txtNewPassword.Visible = true;

                        if(txtNewName.Text != "" && txtNewPassword.Text != "")
                        {
                            label2.Visible = true;
                            label7.Visible = true;
                            label10.Visible = true;
                            label11.Visible = true;
                            label12.Visible = true;
                            label13.Visible = true;
                            label14.Visible = true;
                            label15.Visible = true;

                            label7.Text = us.UserName;
                            label11.Text = us.Password;

                            us.UserName = txtNewName.Text;
```

```
                            us.Password = txtNewPassword.Text;

                            label12.Text = us.UserName;
                            label15.Text = us.Password;

                            MUserDL.storeUserIntoFile();
                            break;
                        }
                        else
                        {
                            MessageBox.Show("Please Enter New Password And UserName
!!!");
                        }
                    }
                    else
                    {
                        c++;
                    }
                }

                if(c == MUserDL.userList.Count)
                {
                    MessageBox.Show("No Such Account Found... Enter Valid Information
!!!");
                    textoldName.Text = string.Empty;
                    textoldPassword.Text = string.Empty;
                    textoldRole.Text = string.Empty;
                }
            }
            else
            {
                MessageBox.Show("Please Provide InFormation !!!");
            }
        }

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

        private void label9_Click(object sender, EventArgs e)
        {
            this.Hide();
            ClientForm cf = new ClientForm(idx);
            cf.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            NewUN.Visible = false;
            NewP.Visible = false;
            txtNewName.Visible = false;
            txtNewPassword.Visible = false;

            label2.Visible = false;
            label7.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
```

```
            label12.Visible = false;
            label13.Visible = false;
            label14.Visible = false;
            label15.Visible = false;

            textoldName.Text = string.Empty;
            textoldPassword.Text = string.Empty;
            textoldRole.Text = string.Empty;

            txtNewName.Text = string.Empty;
            txtNewPassword.Text = string.Empty;
        }
    }

}
```

THE END ☺