**https://www.youtube.com/channel/UC0ZpJKLuJL4U03c7m23rnGw**

**Bank Management System**

****

Session: 2022 – 2026

**Submitted by:**

Harmain Iftikhar 2022-CS-07

**Supervised by:**

Prof. Dr. Laeeq Khan Niazi

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**Table of Contents**

**Introduction………………………………………………………………….…………...……3**

**Class Responsibility Collaboration Card……………………………...………..……………4**

**Object Oriented Programming……………………………………………..….…………...…5**

**Design Pattern Implementation..…………………………………………………….……...7,8**

**Class Details……........……………………………………………………………..….......….8,9**

**Conclusion……........……………………………………………………………..……........…..9**

**Introduction**

* **Overview**

Bank Management System is a backend management application developed in C#, catering to Three types of users: Admin, Employee and Customer. It utilizes object-oriented programming concepts to organize and manage various tasks like ATM, Transactions and account manage within a Bank. The system provides efficient management and operational support, enabling seamless workflow and outclass 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 customers, Atm, Transactions according to account type (savings and current) and Personal Information of almost every customer and Employee and Admin.
2. **Improved Customer Service**: By maintaining users records, processing transactions accurately, and offering loan, ATM, Transactions, the system aims to enhance customer satisfaction and provide personalized services.

* **Functionality**

**Admin:**

1. **Bank Management:** Add, see and edit bank details like Bank Name, Bank Email, Bank No, Bank Id and Bank Address also allow every user to see these details.
2. **Employee Management:** Add, search, edit, or remove Employee, assign different salary of Employee, update personal information of Employee like Salary, employee ID and date of joining.
3. **Personal Details Management:** Admin can update his own information like Name, Phone No, CNIC, Email, Address and Email Password.
4. **Customer Management:** Admin can search, remove any customer he wants and can see all details of customer.

**Employee:**

1. **Customer Management:** Add, see and edit customer, creation of ATM for customer and set account type of customer (Saving and current).
2. **Personal Details Management:** Employee can set his personal information except Date of Joining, Salary and Id.
3. **Bank Details:** Employee can see bank details like phone no, email, address, bank branch id, bank name.

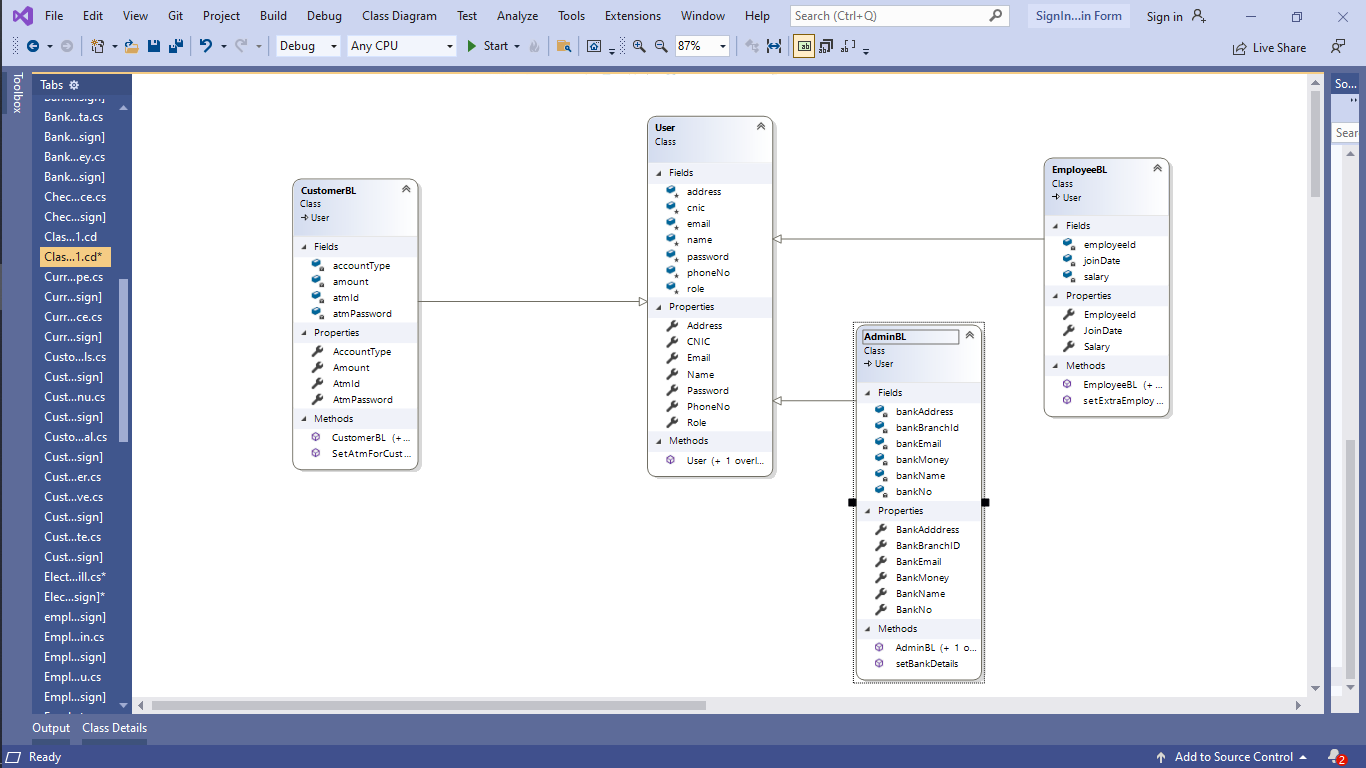
**Customer:**

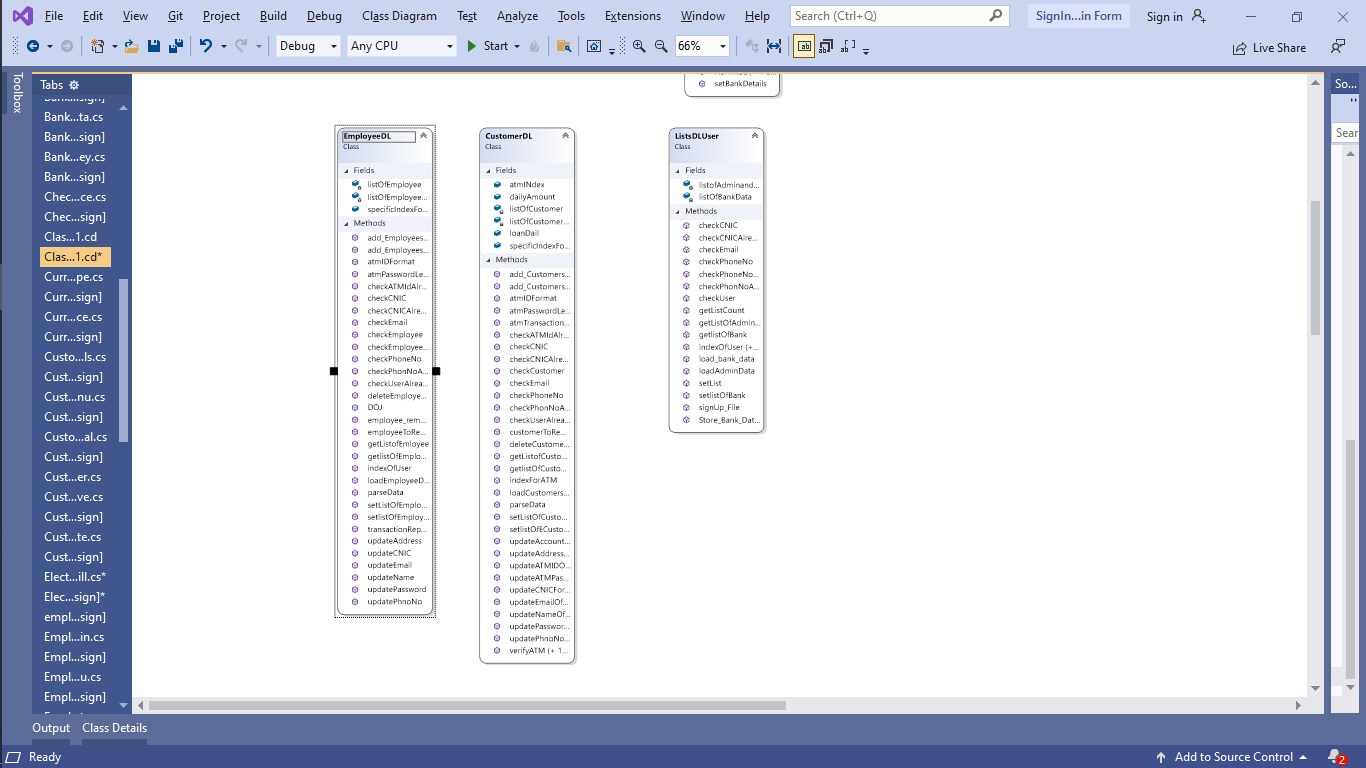
1. **Transaction Process:** Customer can perform every transaction based on his account type if customer account is a “current” then he can withdraw money, send money, can add money, take loans from bank, pay online bills (gas, electricity and many more). Customer can also take loan for specific period of time.
2. **Personal Details Management:** Customer can update his own information like Name, Phone No, CNIC, Email, Address, Atm Id, Atm Password and Email Password.
3. **Bank Details:** Customer can see bank details like phone no, email, address, bank branch id, bank name.

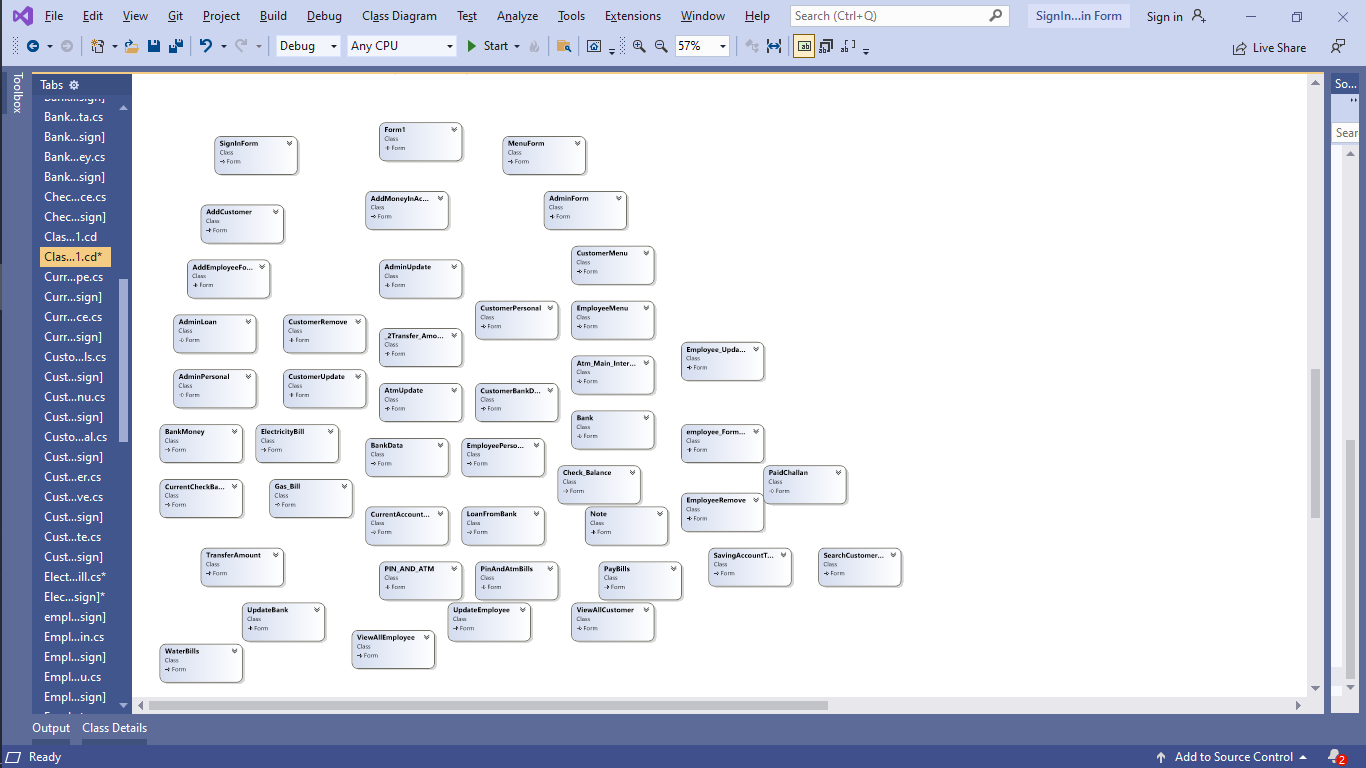
**Validations in System ----------------------------->**

There are so many validations in the system like user can never enter the blank information. User can never enter wrong Email format email must contains “@gmail.com”. User only can enter 11 digits phone No and first digit always start with “0” as we are living in Pakistan. User can only enter 13 digits CNIC and first digit is always “3” as we are living in Pakistan. Email password can consist of 7 digits and it’s a combination of both digits and characters. Atm Id consist of 11 digits and password must be 4 digits. Employee salary start from 15,000 🡪 70,000. Employee Id consists of 3 digits. You can not enter same phone no, email, cnic, employee Id for other users.

**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 include association, encapsulation, inheritance and polymorphism. I have used this programming paradigm in my project.

* **Encapsulation**

In Object-oriented programming, encapsulation is making object instance variables private that are kept in classes. This is for security so no one can directly change these variables or access these variables. To modify or set their values we used getter and setter, different function and constructors.

I used object instance variables as private in each class to implement encapsulation that is OOP concept. To access these variables I used getter’s and setter’s. And I set these variables value using different function and constructors. For Example in User class Name, Email, Password, Phone NO, Address, CNIC.

* **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 multiple places.

1. Account-Transaction Association: This association connects individual transactions to the specific accounts they are related to. Each transaction is linked to one particular account.
2. Customer-Loan Association: This association connects customers with their loan records. It helps track loans taken by individual customers.
3. Total Balance Aggregation: Calculating the total balance across all accounts in the bank.
4. Account Composition: An account in the bank management system composed of multiple attributes and methods, such as account number, account holder details, balance, transaction history, etc. These attributes and methods represented as individual components that together form the complete account object.
5. Transaction Composition: A transaction is a composed of elements like transaction ID, amount, involved accounts, and other relevant data.

**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 Transactions, account, ATM and Customer in procedural programming which is rectified in OOP. Now the lists of Customer and his account, ATM and transactions details are within the class of Customer.

* **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 my management system.

1. User class is a parent class. Admin, Employee and Customer class are derived from this class. Admin, Employee and Customer are three users. So, that is the reason I have applied the concept of inheritance here as they are inheriting user object which contains name, password, Email, CNIC, Role, Address, Phone No and id form the User 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.

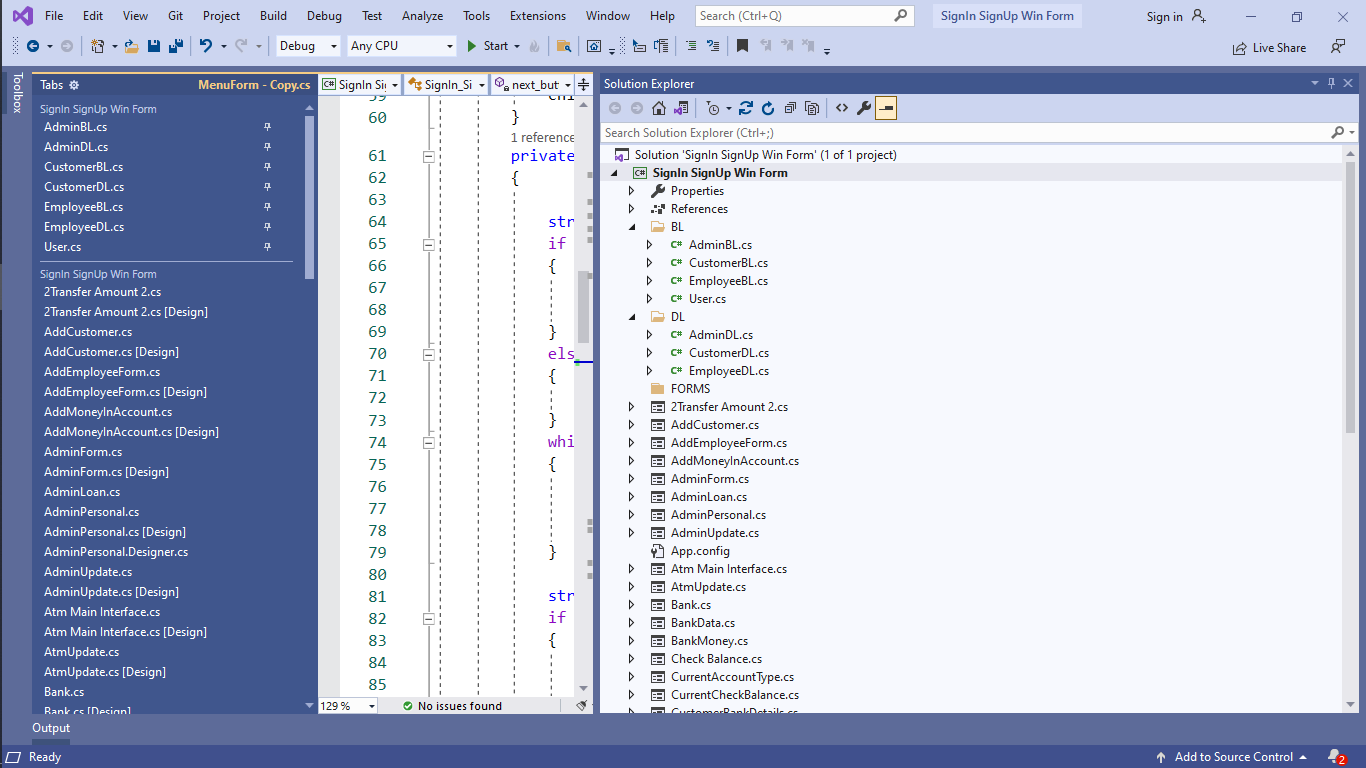
The other type of Polymorphism is Static Polymorphism. I have used this polymorphism for the constructors and also in finding specific atm Id, password and then verify it.

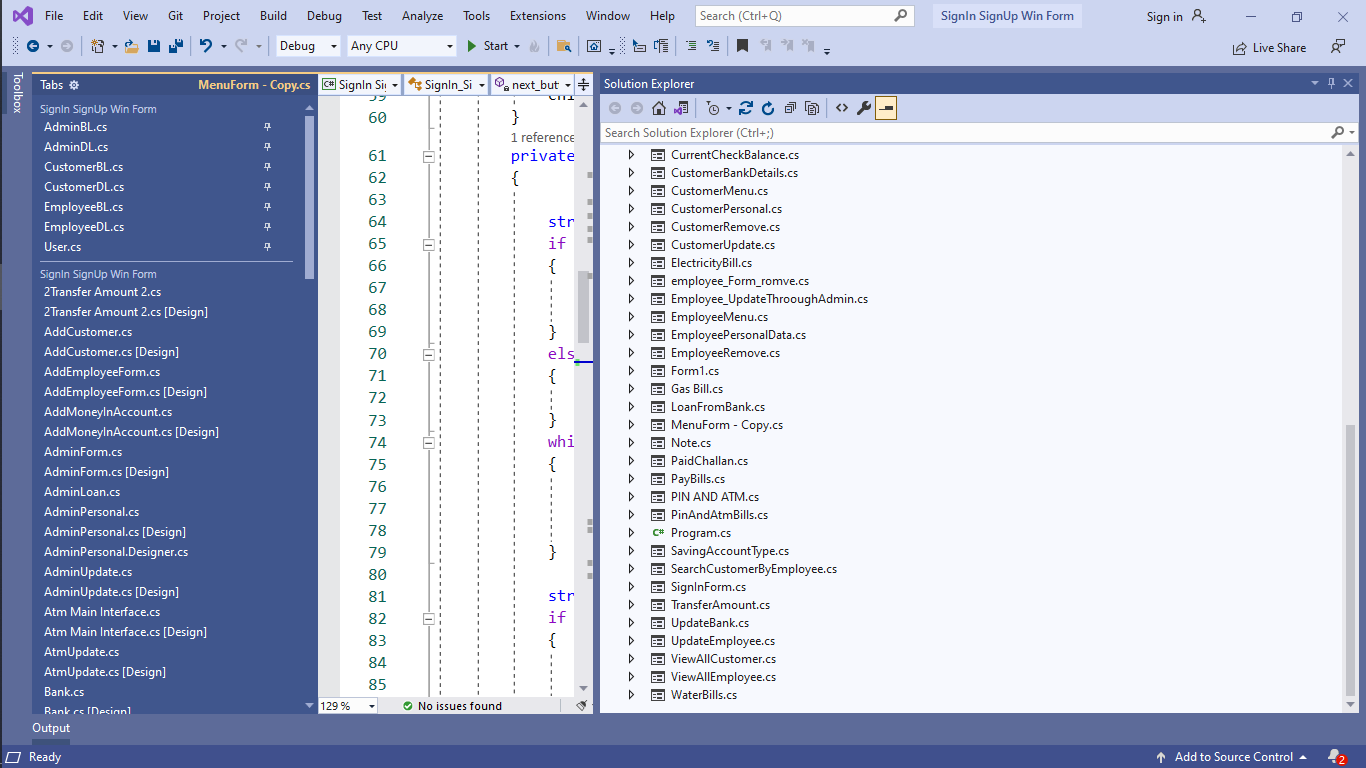
**Advantage**

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 require 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 Users (Credentials), Employee, Admin, and 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 Admin and Customer and Employee. 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. I Also include many validations functions in this layer also like Email, Phone No, Atm Id, CNIC validations.

* **User Interface or User Control (UI/UC)**

This folder contains all the code for Forms for interaction with user.

**Class Details**

* **User**

A User class is used for generalizing the three classes of Admin, Employee and Customer. It is a parent class of these three classes. It contains the common attributes of these three classes. User class contains strings of username, password, address, phone no, Cnic 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 six arguments are used when sign-up functionality is used. The other constructor is a user defined default constructor.

* **Admin**

This class inherits the attributes and behavior of the parent class User. It also includes Bank Name, Email Bank, Bank Branch Id, Bank Address, Bank Money, Bank telephone. It also has 2 constructors one constructor for sign in and other is a default. It also has a function that set the data of Bank.

* **Customer**

Customer class is a child class of User class. It inherits the attributes and behavior of the parent class. All the attributes of the Customer class are also private. It also has attributes like Atm Id, Atm pin, amount and account type. There is a one function that set these attributes of customer.

* **Employee**

Employee is a child class of User also. It inherits the attributes and behavior of the parent class. Employee also has Employee Id, salary and join date. These are attributes set by a function that only present in Employee Class.

**Conclusion**

In conclusion, the Bank Management System is built using the object-oriented programming approach. Its key functionalities include the CRUD operations. Important concepts object-oriented concepts such as encapsulation, 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:**

**BL:**

**User Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using BMS.DL;

namespace BMS.BL

{

class User

{

protected string email;

protected string password;

protected string role;

protected string name;

protected long phoneNo;

protected string address;

protected long cnic;

public User(string aEmail, string aPassword,long aCNIC,string aAddress,long aPhoneNo,string aName)

{

Email = aEmail;

Password = aPassword;

role = "admin";

CNIC = aCNIC;

Address = aAddress;

PhoneNo = aPhoneNo;

Name = aName;

}

public User()

{

}

public string Email

{

get { return email; }

set

{

email = value;

}

}

public string Password

{

get { return password; }

set

{

password = value;

}

}

public string Role

{

get { return role; }

set

{

role = value;

}

}

public string Name

{

get { return name; }

set

{

name = value;

}

}

public long PhoneNo

{

get {return phoneNo; }

set

{

phoneNo = value;

}

}

public string Address

{

get { return address; }

set

{

address = value;

}

}

public long CNIC

{

get { return cnic; }

set

{

cnic = value;

}

}

}

}

**Admin Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using BMS.BL;

using BMS.DL;

namespace BMS.BL

{

// Child Class Of User that is a admin...................................

class AdminBL : User

{

public AdminBL(string aEmail, string aPassword, long aCNIC, string aAddress, long aPhoneNo, string aName) : base(aEmail, aPassword, aCNIC, aAddress, aPhoneNo, aName)

{

}

public AdminBL()

{

}

private string bankName = "Null";

private string bankEmail = "NULL";

private string bankBranchId = "000";

private long bankNo = 0000000000;

private string bankAddress = "Null";

private long bankMoney;

public void setBankDetails(string aBankName, string aBankEmail, string aBankBranchId, string aBankAddress, long aBankNo,long aBankMoney)

{

BankName = aBankName;

BankEmail = aBankEmail;

BankBranchID = aBankBranchId;

BankAdddress = aBankAddress;

BankNo = aBankNo;

BankMoney = aBankMoney;

}

public string BankName

{

get { return bankName; }

set

{

bankName = value;

}

}

public string BankEmail

{

get { return bankEmail; }

set

{

bankEmail = value;

}

}

public string BankBranchID

{

get { return bankBranchId; }

set

{

bankBranchId = value;

}

}

public long BankNo

{

get { return bankNo; }

set

{

bankNo = value;

}

}

public string BankAdddress

{

get { return bankAddress; }

set

{

bankAddress = value;

}

}

public long BankMoney

{

get { return bankMoney; }

set

{

bankMoney = value;

}

}

}

}

**Customer Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BMS.BL

{

class CustomerBL : User

{

private string atmId;

private int atmPassword;

private string accountType;

private long amount = 0;

public void SetAtmForCustomer(string aAtmId, int aAtmPassword, string aAcountType, long aAmount)

{

AtmId = aAtmId;

AtmPassword = aAtmPassword;

AccountType = aAcountType;

Amount = aAmount;

}

public long Amount

{

get { return amount; }

set

{

amount = value;

}

}

public string AtmId

{

get { return atmId; }

set

{

atmId = value;

}

}

public int AtmPassword

{

get { return atmPassword; }

set

{

atmPassword = value;

}

}

public string AccountType

{

get { return accountType; }

set

{

accountType = value;

}

}

public CustomerBL(string aEmail, string aPassword, long aCNIC, string aAddress, long aPhoneNo, string aName, string aRole)

{

email = aEmail;

password = aPassword;

cnic = aCNIC;

address = aAddress;

phoneNo = aPhoneNo;

name = aName;

role = aRole;

}

public CustomerBL()

{

}

}

}

**Employee Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BMS.BL

{

// Child Class Of Employee...................................

class EmployeeBL : User

{

public EmployeeBL(string aEmail, string aPassword, long aCNIC, string aAddress, long aPhoneNo, string aName, string aRole)

{

email = aEmail;

password = aPassword;

cnic = aCNIC;

address = aAddress;

phoneNo = aPhoneNo;

name = aName;

role = aRole;

}

private string employeeId;

private long salary;

private DateTime joinDate;

public string EmployeeId

{

get { return employeeId; }

set

{

employeeId = value;

}

}

public long Salary

{

get { return salary; }

set

{

salary = value;

}

}

public DateTime JoinDate

{

get { return joinDate; }

set

{

joinDate = value;

}

}

public void setExtraEmployeeInformation(string aEmployeeId, long aSalary, DateTime aJoinDate)

{

EmployeeId = aEmployeeId;

Salary = aSalary;

JoinDate = aJoinDate;

}

public EmployeeBL()

{

}

}

}

**DL:**

**AdminDL Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

using System.Threading.Tasks;

using BMS.BL;

namespace BMS.DL

{

//Admin List.....................................................

class ListsDLUser

{

//Sign Up File Handling ...........................

public static void signUp\_File(string path, string aEmail, string aPassword, string aName, string aAddress, long aPhoneNo, long aCnic)

{

if (File.Exists(path))

{

File.Delete(path);

}

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(aEmail + "," + aPassword + "," + aName + "," + aAddress + "," + aPhoneNo + "," + aCnic + "," + "ADMIN");

file.Flush();

file.Close();

}

//Store Bank Data In file ......................

static public void Store\_Bank\_Data\_Into\_File(string path,AdminBL a)

{

if (File.Exists(path))

{

File.Delete(path);

}

StreamWriter file = new StreamWriter(path,true);

file.Write(a.BankName + "," + a.BankEmail + "," + a.BankAdddress + "," + a.BankBranchID + "," + a.BankNo + "," + a.BankMoney);

file.Flush();

file.Close();

}

//Load Data From File For Bank............

static public void load\_bank\_data()

{

AdminBL aL = new AdminBL();

string path = "Bank\_Data.txt";

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

AdminBL a = new AdminBL();

while ((record = fileVariable.ReadLine()) != null)

{

string[] spiltRecord = record.Split(',');

a.BankName = spiltRecord[0];

a.BankEmail = spiltRecord[1];

a.BankAdddress = spiltRecord[2];

a.BankBranchID = spiltRecord[3];

a.BankNo = long.Parse(spiltRecord[4]);

a.BankMoney = long.Parse(spiltRecord[5]);

ListsDLUser.setlistOfBank(a);

}

fileVariable.Close();

}

}

//File Load Data...........................................

static public void loadAdminData()

{

User u1 = new User();

string path = "SignIn\_SignUp.txt";

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

string[] spiltRecord = record.Split(',');

u1.Email = spiltRecord[0];

u1.Password = spiltRecord[1];

u1.Name = spiltRecord[2];

u1.Address = spiltRecord[3];

u1.PhoneNo = long.Parse(spiltRecord[4]);

u1.CNIC = long.Parse(spiltRecord[5]);

u1.Role = spiltRecord[6];

ListsDLUser.setList(u1);

}

fileVariable.Close();

}

}

private static List<AdminBL> listOfBankData = new List<AdminBL>();

public static List<AdminBL> getlistOfBank()

{

return listOfBankData;

}

public static void setlistOfBank(AdminBL A)

{

listOfBankData.Insert(0,A);

}

private static List<User> listofAdminandManager = new List<User>();

public static List<User> getListOfAdminandManager()

{

return listofAdminandManager;

}

public static int getListCount()

{

return listofAdminandManager.Count();

}

public static void setList(User U)

{

listofAdminandManager.Add(U);

}

// Check Email Format......................................................

public static bool checkEmail(string email)

{

char[] Mail = { '@', 'g', 'm', 'a', 'i', 'l', '.', 'c', 'o', 'm' };

int lengthOfEmail = email.Length - 11;

int countEmail = email.Length - 1;

int i = 9;

bool flag = false;

while (countEmail != lengthOfEmail)

{

if (email[countEmail] == Mail[i])

{

flag = true;

}

else

{

flag = false;

break;

}

countEmail--;

i--;

}

return flag;

}

//Check Bank Phone No ...................................

public static bool checkPhoneNoForBank(long aPhoneNo)

{

long pNo = aPhoneNo, b = 0;

while (pNo != 0)

{

pNo = pNo / 10;

b++;

if (pNo == 0)

{

if (b != 9)

{

return false;

}

else if (b == 9)

{

return true;

}

}

}

return false;

}

//Check Phone No ...................................

public static bool checkPhoneNo(long aPhoneNo)

{

long firstDigit;

firstDigit = aPhoneNo / 100000000000;

long pNo = aPhoneNo, b = 0;

while (pNo != 0)

{

pNo = pNo / 10;

b++;

if (pNo == 0)

{

if (b != 10 && firstDigit != 0)

{

return false;

}

else if (b == 10 && firstDigit == 0)

{

return true;

}

}

}

return false;

}

//Check CNIC for Sign up ........................................

public static bool checkCNIC(long aCNIC)

{

long firstDigit = aCNIC / 1000000000000;

long cnic = aCNIC, b = 0;

while (cnic != 0)

{

cnic = cnic / 10;

b++;

if (cnic == 0)

{

if (b != 13 && firstDigit != 3)

{

return false;

}

else if (b == 13 && firstDigit == 3)

{

return true;

}

}

}

return false;

}

// Finding Index function to reomve manager ............

public static EmployeeBL indexOfUser(string eMail)

{

foreach (var user in EmployeeDL.getListofEmloyee())

{

if (eMail == user.Email)

{

return user;

}

}

return null;

}

//Finding Customer Specific...................................

public static CustomerBL indexOfUser(string eMail, string extra)

{

foreach (var user in CustomerDL.getListofCustomer())

{

if (eMail == user.Email)

{

return user;

}

}

return null;

}

//

public static string checkUser(string aEmail, string aPassword)

{

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (a.Email == aEmail && a.Password == aPassword)

{

return a.Role;

}

}

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (a.Email == aEmail && a.Password == aPassword)

{

return a.Role;

}

EmployeeDL.specificIndexForEmployee++;

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (a.Email == aEmail && a.Password == aPassword)

{

return a.Role;

}

CustomerDL.specificIndexForCustomer++;

}

return " ";

}

//Check Phone No Already Existed.........................

public static bool checkPhonNoAlreadyExisted(long aPhoneNo)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

return true;

}

//Check CNIC Already Existed..............................

public static bool checkCNICAlreadyExisted(long aCNIC)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

return true;

}

}

}

**EmployeeDL Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

using System.Threading.Tasks;

using BMS.BL;

namespace BMS.DL

{

// List Of Employee .....................

class EmployeeDL

{

public static void add\_EmployeesInTo\_file(string path, string aEmail, string aPassword, string aName, string aAddress, long aPhoneNo, long aCnic,string aEmployeeID,long employeeSalary,DateTime aInput)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(aEmail + "," + aPassword + "," + aName + "," + aAddress + "," + aPhoneNo + "," + aCnic + "," + "EMPLOYEE" + "," +aEmployeeID + "," + employeeSalary + "," + aInput);

file.Flush();

file.Close();

}

public static void add\_EmployeesInTo\_file2(string path,List<EmployeeBL> a1)

{

if(File.Exists(path))

{

File.Delete(path);

}

StreamWriter file = new StreamWriter(path, true);

foreach(var a in a1)

{

file.WriteLine(a.Email + "," + a.Password + "," + a.Name + "," + a.Address + "," + a.PhoneNo + "," + a.CNIC + "," + "EMPLOYEE" + "," + a.EmployeeId + "," + a.Salary + "," + a.JoinDate);

}

file.Flush();

file.Close();

}

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;

}

static public void loadEmployeeData()

{

string path = "Employee.txt";

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

string[] spiltRecord = record.Split(',');

string Email = spiltRecord[0];

string Password = spiltRecord[1];

string Name = spiltRecord[2];

string Address = spiltRecord[3];

long PhoneNo = long.Parse(spiltRecord[4]);

long CNIC = long.Parse(spiltRecord[5]);

string Role = spiltRecord[6];

string EmployeeId = spiltRecord[7];

long Salary = long.Parse(spiltRecord[8]);

DateTime JoinDate = DateTime.Parse(spiltRecord[9]);

EmployeeBL a1 = new EmployeeBL(Email, Password, CNIC, Address, PhoneNo, Name, "EMPLOYEE");

a1.setExtraEmployeeInformation(EmployeeId, Salary, JoinDate);

setListOfEmployee(a1);

}

fileVariable.Close();

}

}

public static bool checkEmployeeId(string aEmployeeId)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (a.EmployeeId == aEmployeeId)

{

return false;

}

}

return true;

}

// date of joining .............

static public bool DOJ(string input)

{

DateTime date;

if (DateTime.TryParseExact(input, "ddMMyy", null, System.Globalization.DateTimeStyles.None, out date))

{

int day = date.Day;

int month = date.Month;

int year = date.Year;

//Console.WriteLine($"Day: {day}");

//Console.WriteLine($"Month: {month}");

//Console.WriteLine($"Year: {year}");

return true;

}

else

{

return false;

}

}

public static int specificIndexForEmployee = 0;

private static List<EmployeeBL> listOfEmployee = new List<EmployeeBL>();

public static List<EmployeeBL> getListofEmloyee()

{

return listOfEmployee;

}

public static void setListOfEmployee(EmployeeBL U)

{

listOfEmployee.Add(U);

}

//search......................

private static List<EmployeeBL> listOfEmployeeForSearch = new List<EmployeeBL>();

public static List<EmployeeBL> getlistOfEmployeeForSearch()

{

return listOfEmployeeForSearch;

}

public static void setlistOfEmployeeForSearch(EmployeeBL u)

{

listOfEmployeeForSearch.Add(u);

}

static public bool employeeToRemove(string email)

{

foreach (var employee in getListofEmloyee())

{

if (email == employee.Email)

{

EmployeeDL.getListofEmloyee().Remove(employee);

return true;

}

}

return false;

}

static public EmployeeBL employee\_remove(string email)

{

foreach (var a in getListofEmloyee())

{

if (email == a.Email)

{

return a;

}

}

return null;

}

static public void checkEmployee(string email)

{

foreach (var a in getListofEmloyee())

{

if (email == a.Email)

{

setlistOfEmployeeForSearch(a);

}

}

}

static public void updateName(string aName)

{

int i = 0;

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (specificIndexForEmployee == i)

{

a.Name = aName;

}

i++;

}

}

static public void updateEmail(string aEmail)

{

int i = 0;

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (specificIndexForEmployee == i)

{

a.Email = aEmail;

}

i++;

}

}

static public void updateCNIC(long aCNIC)

{

int i = 0;

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (specificIndexForEmployee == i)

{

a.CNIC = aCNIC;

}

i++;

}

}

static public void updatePhnoNo(long aPhoneNo)

{

int i = 0;

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (specificIndexForEmployee == i)

{

a.PhoneNo = aPhoneNo;

}

i++;

}

}

static public void updateAddress(string aAddress)

{

int i = 0;

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (specificIndexForEmployee == i)

{

a.Address = aAddress;

}

i++;

}

}

static public void updatePassword(string aPassword)

{

int i = 0;

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (specificIndexForEmployee == i)

{

a.Password = aPassword;

}

i++;

}

}

public static long transactionReport()

{

long total = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

total = a.Amount + total;

}

return total;

}

// Check ATM ID Format........................................

public static bool atmIDFormat(string aATMID)

{

if (aATMID.Substring(4, 1) != "-" || aATMID.Substring(9, 1) != "-")

{

return false;

}

return true;

}

//Finding Customer Specific...................................

public static CustomerBL indexOfUser(string eMail)

{

foreach (var user in CustomerDL.getListofCustomer())

{

if (eMail == user.Email)

{

return user;

}

}

return null;

}

// Check Email Format......................................................

public static bool checkEmail(string email)

{

char[] Mail = { '@', 'g', 'm', 'a', 'i', 'l', '.', 'c', 'o', 'm' };

int lengthOfEmail = email.Length - 11;

int countEmail = email.Length - 1;

int i = 9;

bool flag = false;

while (countEmail != lengthOfEmail)

{

if (email[countEmail] == Mail[i])

{

flag = true;

}

else

{

flag = false;

break;

}

countEmail--;

i--;

}

return flag;

}

//Check Phone No ...................................

public static bool checkPhoneNo(long aPhoneNo)

{

long firstDigit;

firstDigit = aPhoneNo / 100000000000;

long pNo = aPhoneNo, b = 0;

while (pNo != 0)

{

pNo = pNo / 10;

b++;

if (pNo == 0)

{

if (b != 10 && firstDigit != 0)

{

return false;

}

else if (b == 10 && firstDigit == 0)

{

return true;

}

}

}

return false;

}

//Check Phone No Already Existed.........................

public static bool checkPhonNoAlreadyExisted(long aPhoneNo)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

return true;

}

//Check CNIC Already Existed..............................

public static bool checkCNICAlreadyExisted(long aCNIC)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

return true;

}

//Check Atm ID Password......................................

public static bool atmPasswordLength(int aATMPassword)

{

int CATMpassword = aATMPassword;

int p = 0;

while (CATMpassword != 0)

{

CATMpassword = CATMpassword / 10;

p++;

if (CATMpassword == 0)

{

if (p != 4)

{

return false;

}

else if (p == 4)

{

return true;

}

}

}

return true;

}

//check IdAlready Existed....................................

public static bool checkATMIdAlreadyExisted(string atmId)

{

foreach (var a in CustomerDL.getListofCustomer())

{

if (atmId == a.AtmId)

{

return false;

}

}

return true;

}

//Check User That Already Exists..............

public static bool checkUserAlreadyExisted(string eemail)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (eemail == a.Email)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (eemail == a.Email)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (eemail == a.Email)

{

return false;

}

}

return true;

}

public static bool checkCNIC(long aCNIC)

{

long firstDigit = aCNIC / 1000000000000;

long cnic = aCNIC, b = 0;

while (cnic != 0)

{

cnic = cnic / 10;

b++;

if (cnic == 0)

{

if (b != 13 && firstDigit != 3)

{

return false;

}

else if (b == 13 && firstDigit == 3)

{

return true;

}

}

}

return false;

}

public static void deleteEmployeeFromList(EmployeeBL Temp)

{

for (int index = 0; index < listOfEmployee.Count; index++)

{

if (listOfEmployee[index].Name == Temp.Name && listOfEmployee[index].Password == Temp.Password && listOfEmployee[index].CNIC == Temp.CNIC && listOfEmployee[index].CNIC == Temp.CNIC && listOfEmployee[index].Email == Temp.Email && listOfEmployee[index].JoinDate == Temp.JoinDate && listOfEmployee[index].Role == Temp.Role && listOfEmployee[index].Salary == Temp.Salary)

{

listOfEmployee.RemoveAt(index);

}

}

}

}

}

**CustomerDL Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

using System.Threading.Tasks;

using BMS.BL;

namespace BMS.DL

{

class CustomerDL

{

static public long dailyAmount = 10000;

static public long loanDail = 20000;

static public int atmINdex;

public static void deleteCustomerFromList(CustomerBL Temp)

{

for (int index = 0; index < listOfCustomer.Count; index++)

{

if (listOfCustomer[index].Name == Temp.Name && listOfCustomer[index].Password == Temp.Password && listOfCustomer[index].Email == Temp.Email)

{

listOfCustomer.RemoveAt(index);

}

}

}

public static bool verifyATM(string atm,int pin)

{

if(getListofCustomer()[specificIndexForCustomer].AtmId == atm && getListofCustomer()[specificIndexForCustomer].AtmPassword == pin)

{

return true;

}

return false;

}

public static string verifyATM(string atm)

{

if (getListofCustomer()[specificIndexForCustomer].AtmId == atm)

{

return "same";

}

foreach(var a in getListofCustomer())

{

if(a.AtmId == atm)

{

return "exist";

}

}

return "false";

}

public static int indexForATM(string atm)

{

int i = 0;

foreach (var a in getListofCustomer())

{

if (a.AtmId == atm)

{

return i;

}

i++;

}

return -1;

}

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;

}

static public void loadCustomersData()

{

string path = "Customer.txt";

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

while ((record = fileVariable.ReadLine()) != null)

{

string[] spiltRecord = record.Split(',');

string Email = spiltRecord[0];

string Password = spiltRecord[1];

string Name = spiltRecord[2];

string Address = spiltRecord[3];

long PhoneNo = long.Parse(spiltRecord[4]);

long CNIC = long.Parse(spiltRecord[5]);

string Role = spiltRecord[6];

string AtmId = spiltRecord[7];

int AtmPassword = int.Parse(spiltRecord[8]);

string AccountType = spiltRecord[9];

long Amount = long.Parse(spiltRecord[10]);

CustomerBL u1 = new CustomerBL(Email, Password, CNIC, Address, PhoneNo, Name, "CUSTOMER");

u1.SetAtmForCustomer(AtmId,AtmPassword,AccountType,Amount);

setListOfCustomer(u1);

}

fileVariable.Close();

}

}

public static void add\_CustomersInTo\_file2(string path, List<CustomerBL> cl)

{

if (File.Exists(path))

{

File.Delete(path);

}

StreamWriter file = new StreamWriter(path, true);

foreach (var c in cl)

{

file.WriteLine(c.Email + "," + c.Password + "," + c.Name + "," + c.Address + "," + c.PhoneNo + "," + c.CNIC + "," + "CUSTOMER" + "," + c.AtmId + "," + c.AtmPassword + "," + c.AccountType + "," + c.Amount);

}

file.Flush();

file.Close();

}

public static void add\_CustomersInTo\_file(string path,CustomerBL c)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(c.Email + "," + c.Password + "," + c.Name + "," + c.Address + "," + c.PhoneNo + "," + c.CNIC + "," + "CUSTOMER" + "," + c.AtmId+ "," + c.AtmPassword + "," + c.AccountType + "," + c.Amount);

file.Flush();

file.Close();

}

//Check user for sign In......................

public static int specificIndexForCustomer = 0;

private static List<CustomerBL> listOfCustomer = new List<CustomerBL>();

public static List<CustomerBL> getListofCustomer()

{

return listOfCustomer;

}

public static void setListOfCustomer(CustomerBL U)

{

listOfCustomer.Add(U);

}

//search......................

private static List<CustomerBL> listOfCustomerForSearch = new List<CustomerBL>();

public static List<CustomerBL> getlistOfCustomerForSearch()

{

return listOfCustomerForSearch;

}

public static void setlistOfECustomerForSearch(CustomerBL u)

{

listOfCustomerForSearch.Add(u);

}

static public bool customerToRemove(string email)

{

foreach (var a in getListofCustomer())

{

if (email == a.Email)

{

getListofCustomer().Remove(a);

return true;

}

}

return false;

}

static public void checkCustomer(string email)

{

foreach(var a in getListofCustomer())

{

if(email == a.Email)

{

setlistOfECustomerForSearch(a);

}

}

}

//Update Name ..........................

static public bool updateNameOfCustomer(string aName)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.Name = aName;

return true;

}

i++;

}

return false;

}

static public void updateEmailOfCustomer(string aEmail)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.Email = aEmail;

}

i++;

}

}

static public void updateCNICForCustomer(long aCNIC)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.CNIC = aCNIC;

}

i++;

}

}

static public void updatePhnoNoForCustomer(long aPhoneNo)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.PhoneNo = aPhoneNo;

}

i++;

}

}

static public void updateAddressofCustomer(string aAddress)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.Address = aAddress;

}

i++;

}

}

static public void updatePasswordForCustomer(string aPassword)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.Password = aPassword;

}

i++;

}

}

//Update ATM ID...................................

public static bool updateATMIDOfCustomer(string aATMID)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.AtmId = aATMID;

return true;

}

i++;

}

return false;

}

//Update ATM Password...................................

public static bool updateATMPasswordForCustomer(int aAtmPassword)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.AtmPassword = aAtmPassword;

return true;

}

i++;

}

return false;

}

//Update Account Type...................................

public static bool updateAccountOfCustomer(string aAccountType)

{

int i = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

if (specificIndexForCustomer == i)

{

a.AccountType = aAccountType;

return true;

}

i++;

}

return false;

}

// Check ATM ID Format........................................

public static bool atmIDFormat(string aATMID)

{

if (aATMID.Substring(4, 1) != "-" || aATMID.Substring(9, 1) != "-")

{

return false;

}

return true;

}

//Check Atm Id and Atm Password For Transactions....................................

public static string atmTransactions(string aAtmId, long aAtmPassword)

{

foreach (var a in CustomerDL.getListofCustomer())

{

if (aAtmId == a.AtmId && aAtmPassword == a.AtmPassword)

{

return a.AccountType;

}

}

return null;

}

public static long atmTransactions(long aAtmPassword, string aAtmId)

{

foreach (var a in CustomerDL.getListofCustomer())

{

if (aAtmId == a.AtmId && aAtmPassword == a.AtmPassword)

{

return a.Amount;

}

}

return 0;

}

public static void atmTransactions(long aAtmPassword, string aAtmId, long aAmount1)

{

foreach (var a in CustomerDL.getListofCustomer())

{

if (aAtmId == a.AtmId && aAtmPassword == a.AtmPassword)

{

a.Amount = a.Amount + aAmount1;

}

}

}

public static void atmTransactions(long aAmount1, long aAtmPassword, string aAtmId)

{

foreach (var a in CustomerDL.getListofCustomer())

{

if (aAtmId == a.AtmId && aAtmPassword == a.AtmPassword)

{

a.Amount = a.Amount - aAmount1;

}

}

}

//Check User That Already Exists..............

public static bool checkUserAlreadyExisted(string eemail)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (eemail == a.Email)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (eemail == a.Email)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (eemail == a.Email)

{

return false;

}

}

return true;

}

// Check Email Format......................................................

public static bool checkEmail(string email)

{

char[] Mail = { '@', 'g', 'm', 'a', 'i', 'l', '.', 'c', 'o', 'm' };

int lengthOfEmail = email.Length - 11;

int countEmail = email.Length - 1;

int i = 9;

bool flag = false;

while (countEmail != lengthOfEmail)

{

if (email[countEmail] == Mail[i])

{

flag = true;

}

else

{

flag = false;

break;

}

countEmail--;

i--;

}

return flag;

}

//Check CNIC for Sign up ........................................

public static bool checkCNIC(long aCNIC)

{

long firstDigit = aCNIC / 1000000000000;

long cnic = aCNIC, b = 0;

while (cnic != 0)

{

cnic = cnic / 10;

b++;

if (cnic == 0)

{

if (b != 13 && firstDigit != 3)

{

return false;

}

else if (b == 13 && firstDigit == 3)

{

return true;

}

}

}

return false;

}

//Check CNIC Already Existed..............................

public static bool checkCNICAlreadyExisted(long aCNIC)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (aCNIC == a.CNIC)

{

return false;

}

}

return true;

}

//Check Phone No ...................................

public static bool checkPhoneNo(long aPhoneNo)

{

long firstDigit;

firstDigit = aPhoneNo / 100000000000;

long pNo = aPhoneNo, b = 0;

while (pNo != 0)

{

pNo = pNo / 10;

b++;

if (pNo == 0)

{

if (b != 10 && firstDigit != 0)

{

return false;

}

else if (b == 10 && firstDigit == 0)

{

return true;

}

}

}

return false;

}

//Check Phone No Already Existed.........................

public static bool checkPhonNoAlreadyExisted(long aPhoneNo)

{

foreach (var a in EmployeeDL.getListofEmloyee())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

foreach (var a in ListsDLUser.getListOfAdminandManager())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

foreach (var a in CustomerDL.getListofCustomer())

{

if (aPhoneNo == a.PhoneNo)

{

return false;

}

}

return true;

}

//check IdAlready Existed....................................

public static bool checkATMIdAlreadyExisted(string atmId)

{

foreach (var a in CustomerDL.getListofCustomer())

{

if (atmId == a.AtmId)

{

return false;

}

}

return true;

}

//Check Atm ID Password......................................

public static bool atmPasswordLength(int aATMPassword)

{

int CATMpassword = aATMPassword;

int p = 0;

while (CATMpassword != 0)

{

CATMpassword = CATMpassword / 10;

p++;

if (CATMpassword == 0)

{

if (p != 4)

{

return false;

}

else if (p == 4)

{

return true;

}

}

}

return true;

}

}

}

**Forms Code**

**Form1:**

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 BMS.BL;

using BMS.DL;

using System.Drawing.Drawing2D;

using System.Runtime.InteropServices;

using System.Windows.Forms;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class Form1 : Form

{

private Rectangle buttonOriginalrectangle;

private Rectangle originalFormSize;

public Form1()

{

InitializeComponent();

ListsDLUser.loadAdminData();

CustomerDL.loadCustomersData();

EmployeeDL.loadEmployeeData();

ListsDLUser.load\_bank\_data();

ApplyRoundedRegion(this);

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void button\_Login\_Click(object sender, EventArgs e)

{

if(ListsDLUser.getListCount() == 0)

{

this.Hide();

MenuForm signUp = new MenuForm();

signUp.Show();

}

else

{

MessageBox.Show("Admin Already SignUp !!!");

}

}

private void txt\_UserEmail\_TextChanged(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

//originalFormSize = this.Size;

//buttonOriginalrectangle = new Rectangle(button\_SignIn.Location.X);

}

private void button\_SignIn\_Click(object sender, EventArgs e)

{

this.Hide();

SignInForm s = new SignInForm();

s.Show();

}

private void button\_exit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void ApplyRoundedRegion(Form form)

{

GraphicsPath path = new GraphicsPath();

int radius = 20; // Adjust this value to change the rounding radius

// Create a rounded rectangle path

Rectangle rect = new Rectangle(0, 0, form.Width, form.Height);

path.AddArc(rect.X, rect.Y, radius \* 2, radius \* 2, 180, 90);

path.AddLine(rect.X + radius, rect.Y, rect.Right - radius \* 2, rect.Y);

path.AddArc(rect.X + rect.Width - radius \* 2, rect.Y, radius \* 2, radius \* 2, 270, 90);

path.AddLine(rect.Right, rect.Y + radius \* 2, rect.Right, rect.Bottom - radius \* 2);

path.AddArc(rect.X + rect.Width - radius \* 2, rect.Y + rect.Height - radius \* 2, radius \* 2, radius \* 2, 0, 90);

path.AddLine(rect.Right - radius \* 2, rect.Bottom, rect.X + radius \* 2, rect.Bottom);

path.AddArc(rect.X, rect.Y + rect.Height - radius \* 2, radius \* 2, radius \* 2, 90, 90);

path.AddLine(rect.X, rect.Bottom - radius \* 2, rect.X, rect.Y + radius \* 2);

form.Region = new Region(path);

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

}

}

**Sign IN Form:**

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 BMS.DL;

using BMS.BL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class SignInForm : Form

{

public SignInForm()

{

InitializeComponent();

}

private void ClearDataFromForm()

{

emailtxt.Text = "";

emailpasswordtxt.Text = "";

}

private void next\_button\_Click(object sender, EventArgs e)

{

}

private void back\_button\_Click(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void next\_button\_Click\_1(object sender, EventArgs e)

{

CustomerDL.dailyAmount = 10000;

CustomerDL.specificIndexForCustomer = 0;

EmployeeDL.specificIndexForEmployee = 0;

string email = emailtxt.Text;

if (string.IsNullOrEmpty(emailtxt.Text.Trim()))

{

errorProviderEmail.SetError(emailtxt, "Email Is Required");

return;

}

else

{

errorProviderEmail.SetError(emailtxt, string.Empty);

}

if (string.IsNullOrEmpty(emailpasswordtxt.Text.Trim()))

{

errorProviderEmail.SetError(emailpasswordtxt, "Password Is Required");

return;

}

else

{

errorProviderEmail.SetError(emailpasswordtxt, string.Empty);

}

//while((string.IsNullOrEmpty(emailtxt.Text)))

//{

// MessageBox.Show("Email Can not Empty ");

// return;

//}

string password = emailpasswordtxt.Text;

string role = ListsDLUser.checkUser(email, password);

role = role.ToUpper();

if (role == "ADMIN")

{

MessageBox.Show("SignIn Successfully as Admin!!!");

this.Hide();

AdminForm a = new AdminForm();

a.Show();

}

else if (role == "EMPLOYEE")

{

MessageBox.Show("SignIn Successfully as Employee!!!");

this.Hide();

EmployeeMenu em = new EmployeeMenu();

em.Show();

}

else if (role == "CUSTOMER")

{

CustomerMenu cM = new CustomerMenu();

cM.Show();

this.Hide();

}

else

{

MessageBox.Show("User Not Found !!!");

}

ClearDataFromForm();

}

private void back\_button\_Click\_1(object sender, EventArgs e)

{

this.Close();

Form1 f1 = new Form1();

f1.Show();

}

private void button1\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

private void emailtxt\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Sign UP Form:**

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.Text.RegularExpressions;

using BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class MenuForm : Form

{

public MenuForm()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void Name\_TextChanged(object sender, EventArgs e)

{

}

private void emailtxt\_TextChanged(object sender, EventArgs e)

{

}

private void addresstxt\_TextChanged(object sender, EventArgs e)

{

}

private void phnotxt\_TextChanged(object sender, EventArgs e)

{

}

private void cnictxt\_TextChanged(object sender, EventArgs e)

{

}

private void ClearDataFromForm()

{

emailtxt.Text = "";

passwortxt.Text = "";

nametxt.Text = "";

addresstxt.Text = "";

phnotxt.Text = "";

cnictxt.Text = "";

}

private void next\_button\_Click(object sender, EventArgs e)

{

string email = emailtxt.Text;

if (string.IsNullOrEmpty(emailtxt.Text.Trim()))

{

errorProviderEmail.SetError(emailtxt, "Email Is Required");

return;

}

else

{

errorProviderEmail.SetError(emailtxt, string.Empty);

}

while ((ListsDLUser.checkEmail(email) == false))

{

MessageBox.Show("Email Is not in a Correct Format ");

return;

}

string password = passwortxt.Text;

if (string.IsNullOrEmpty(passwortxt.Text.Trim()))

{

errorProviderEmail.SetError(passwortxt, "Password Is Required");

return;

}

else

{

errorProviderEmail.SetError(passwortxt, string.Empty);

}

string name = nametxt.Text;

string address = addresstxt.Text;

if (string.IsNullOrEmpty(addresstxt.Text.Trim()))

{

errorProviderEmail.SetError(addresstxt, "Address is Required");

return;

}

else

{

errorProviderEmail.SetError(addresstxt, string.Empty);

}

string phoneNo = (phnotxt.Text);

if (string.IsNullOrEmpty(phnotxt.Text.Trim()))

{

errorProviderEmail.SetError(phnotxt, "Phone No is Required");

return;

}

else

{

errorProviderEmail.SetError(phnotxt, string.Empty);

}

while ((ListsDLUser.checkPhoneNo(Convert.ToInt64(phoneNo)) == false))

{

MessageBox.Show("length of PhNo 11 Digits and First digit is \"0\" as in Pakistan ");

return;

}

long Phone\_no = Convert.ToInt64(phoneNo);

string CNIC = (cnictxt.Text);

if (string.IsNullOrEmpty(cnictxt.Text.Trim()))

{

errorProviderEmail.SetError(cnictxt, "CNIC is Required");

return;

}

else

{

errorProviderEmail.SetError(cnictxt, string.Empty);

}

while ((ListsDLUser.checkCNIC(Convert.ToInt64(CNIC)) == false))

{

MessageBox.Show("length of CNIC 13 Digits and First digit is \"3\" as in Pakistan ");

return;

}

long c\_n\_I\_C = Convert.ToInt64(CNIC);

User u = new User(email,password,c\_n\_I\_C,address,Phone\_no,name);

ListsDLUser.setList(u);

string path = "SignIn\_SignUp.txt";

ListsDLUser.signUp\_File(path, email, password, name, address, Phone\_no, c\_n\_I\_C);

MessageBox.Show("Admin Added Successfully!!!");

ClearDataFromForm();

this.Close();

Form1 f = new Form1();

f.Show();

}

public bool isValidName(string n)

{

Regex check = new Regex(@"^([A-Z][a-z-A-z]+)$");

bool valid = false;

valid = check.IsMatch(n);

if(valid == true)

{

return valid;

}

else

{

MessageBox.Show("Name Format is Not a Correct");

return valid;

}

}

private void back\_button\_Click(object sender, EventArgs e)

{

this.Close();

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void MenuForm\_Load(object sender, EventArgs e)

{

}

}

}

**Transfer Amount Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class \_2Transfer\_Amount\_2 : Form

{

public \_2Transfer\_Amount\_2()

{

InitializeComponent();

}

private void \_2Transfer\_Amount\_2\_Load(object sender, EventArgs e)

{

currentAmountBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

long amount = Convert.ToInt64(transferAmountBox.Text);

while (amount > CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount)

{

MessageBox.Show("Low Money!!!");

transferAmountBox.Text = "";

return;

}

CustomerDL.getListofCustomer()[CustomerDL.atmINdex].Amount = amount + CustomerDL.getListofCustomer()[CustomerDL.atmINdex].Amount;

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount - amount;

currentAmountBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

transferAmountBox.Text = "";

MessageBox.Show("Transfer Successfully!!!");

CustomerDL.add\_CustomersInTo\_file2("Customer.txt", CustomerDL.getListofCustomer());

}

private void button3\_Click(object sender, EventArgs e)

{

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

}

}

**Add Customers Form:**

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.Text.RegularExpressions;

using BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AddCustomer : Form

{

public AddCustomer()

{

InitializeComponent();

}

private void SubmitButton\_Click(object sender, EventArgs e)

{

string email = emailBox.Text;

while(CustomerDL.checkUserAlreadyExisted(email) == false)

{

MessageBox.Show("Email Already Existed!!! Plzz Enter Unique Email ");

emailBox.Text = "";

return;

}

while ((CustomerDL.checkEmail(email) == false))

{

MessageBox.Show("Email Is not in a Correct Format ");

emailBox.Text = "";

return;

}

string emailPassword = emailPassBox.Text;

string name = NameBox.Text;

long cnic = Convert.ToInt64(CnicBox.Text);

while (CustomerDL.checkCNIC(cnic) == false)

{

MessageBox.Show("CNIC must be 13 digit and first digit must be \"3\" as in Pakistan ");

CnicBox.Text = "";

return;

}

while(CustomerDL.checkCNICAlreadyExisted(cnic) == false)

{

MessageBox.Show("CNIC Already Existed!!! Plzz Enter Unique CNIC");

CnicBox.Text = "";

return;

}

long phoneNo = Convert.ToInt64(phonNoBox.Text);

while (CustomerDL.checkPhoneNo(phoneNo) == false)

{

MessageBox.Show("Phone No must be 11 digit and first digit must be 0 as in Pakistan ");

phonNoBox.Text = "";

return;

}

while(CustomerDL.checkPhonNoAlreadyExisted(phoneNo) == false)

{

MessageBox.Show("Phone No Already Existed!!! Plzz Enter Unique Phone No ");

phonNoBox.Text = "";

return;

}

string atm = ATMIdBox.Text;

while(CustomerDL.checkATMIdAlreadyExisted(atm) == false)

{

MessageBox.Show("ATM ID already in used Plzz Enter Unique Id!!!");

ATMIdBox.Text = "";

return;

}

int atmPassword = Convert.ToInt32(AtmPasswordBox.Text);

while(CustomerDL.atmPasswordLength(atmPassword) == false)

{

MessageBox.Show("ATM password Length Must be 4 Plzz Enter 4 Digit PIN!!!");

AtmPasswordBox.Text = "";

return;

}

string accountType = acountTypeCombo.Text;

string address = addressBox.Text;

long amount = 0;

CustomerBL c1 = new CustomerBL(email, emailPassword, cnic, address, phoneNo, name, "CUSTOMER");

c1.SetAtmForCustomer(atm,atmPassword,accountType,amount);

CustomerDL.setListOfCustomer(c1);

string path = "Customer.txt";

CustomerDL.add\_CustomersInTo\_file(path, c1);

MessageBox.Show("Data Save Successfully ");

this.Hide();

AddCustomer ac = new AddCustomer();

ac.Show();

}

public bool isValidName(string n)

{

Regex check = new Regex(@"^([A-Z][a-z-A-z]+)$");

bool valid = false;

valid = check.IsMatch(n);

if (valid == true)

{

return valid;

}

else

{

MessageBox.Show("Name Format is Not a Correct");

return valid;

}

}

private void tableLayoutPanel5\_Paint(object sender, PaintEventArgs e)

{

}

private void label11\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

EmployeeMenu em = new EmployeeMenu();

em.Show();

this.Hide();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void AddCustomer\_Load(object sender, EventArgs e)

{

}

private void tableLayoutPanel2\_Paint(object sender, PaintEventArgs e)

{

}

private void tableLayoutPanel4\_Paint(object sender, PaintEventArgs e)

{

}

}

}

**Add Employee Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AddEmployeeForm : Form

{

string path = "Employee.txt";

public AddEmployeeForm()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Form1 home = new Form1();

this.Close();

home.Show();

}

private void SubmitButton\_Click(object sender, EventArgs e)

{

string emailEmployee = emailBox.Text;

while (EmployeeDL.checkUserAlreadyExisted(emailEmployee) == false)

{

MessageBox.Show("Email Already Existed!!! Plzz Enter Unique Email ");

emailBox.Text = "";

return;

}

while ((EmployeeDL.checkEmail(emailEmployee) == false))

{

MessageBox.Show("Email Is not in a Correct Format ");

emailBox.Text = "";

return;

}

string emailPassword = emailPassBox.Text;

string nameEmployee = NameBox.Text;

long cnic = Convert.ToInt64(CnicBox.Text);

while (EmployeeDL.checkCNIC(cnic) == false)

{

MessageBox.Show("CNIC must be 13 digit and first digit must be \"3\" as in Pakistan ");

CnicBox.Text = "";

return;

}

while (EmployeeDL.checkCNICAlreadyExisted(cnic) == false)

{

MessageBox.Show("CNIC Already Existed!!! Plzz Enter Unique CNIC");

CnicBox.Text = "";

return;

}

long phoneNo = Convert.ToInt64(phonNoBox.Text);

while (EmployeeDL.checkPhoneNo(phoneNo) == false)

{

MessageBox.Show("Phone No must be 11 digit and first digit must be 0 as in Pakistan ");

phonNoBox.Text = "";

return;

}

while (EmployeeDL.checkPhonNoAlreadyExisted(phoneNo) == false)

{

MessageBox.Show("Phone No Already Existed!!! Plzz Enter Unique Phone No ");

phonNoBox.Text = "";

return;

}

string employeeId = employeeIdBox.Text;

while(EmployeeDL.checkEmployeeId(employeeId) == false)

{

MessageBox.Show("Employee Id Already Existed");

employeeIdBox.Text = "";

return;

}

long employeeSalary = Convert.ToInt64(employeeSalaryBox.Text);

while(employeeSalary < 17000 || employeeSalary > 70000)

{

MessageBox.Show("Set Salary Between 17000 to 70000$");

employeeSalaryBox.Text = "";

return;

}

string address = AddressBox.Text;

EmployeeBL a1 = new EmployeeBL(emailEmployee, emailPassword, cnic, address, phoneNo, nameEmployee, "EMPLOYEE");

a1.setExtraEmployeeInformation(employeeId, employeeSalary, join);

EmployeeDL.setListOfEmployee(a1);

string path = "Employee.txt";

EmployeeDL.add\_EmployeesInTo\_file(path, emailEmployee, emailPassword, nameEmployee, address, phoneNo, cnic, employeeId, employeeSalary, join);

MessageBox.Show("Data Save Successfully ");

this.Hide();

AddEmployeeForm student = new AddEmployeeForm();

student.Show();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

AdminForm menu = new AdminForm();

this.Hide();

menu.Show();

}

private void label16\_Click(object sender, EventArgs e)

{

Form1 home = new Form1();

this.Close();

home.Show();

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

}

private void tableLayoutPanel5\_Paint(object sender, PaintEventArgs e)

{

}

}

}

**Add Money In account Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AddMoneyInAccount : Form

{

public AddMoneyInAccount()

{

InitializeComponent();

}

private void AddMoneyInAccount\_Load(object sender, EventArgs e)

{

(CurrentBox.Text) = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

remainBox.Text = CustomerDL.dailyAmount.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

while (Convert.ToInt64(AmountBox.Text) > CustomerDL.dailyAmount)

{

MessageBox.Show("Limit Exceeded!!!");

AmountBox.Text = "";

return;

}

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount + Convert.ToInt64(AmountBox.Text);

MessageBox.Show("Amount Added Successfully!!!");

CustomerDL.dailyAmount = CustomerDL.dailyAmount - Convert.ToInt64(AmountBox.Text);

remainBox.Text = CustomerDL.dailyAmount.ToString();

CurrentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

AmountBox.Text = "";

CustomerDL.add\_CustomersInTo\_file2("Customer.txt",CustomerDL.getListofCustomer());

}

private void button1\_Click(object sender, EventArgs e)

{

Atm\_Main\_Interface aI = new Atm\_Main\_Interface();

aI.Show();

this.Close();

}

private void remainBox\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Admin Forms:**

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 BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AdminForm : Form

{

public AdminForm()

{

InitializeComponent();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void AdminForm\_Load(object sender, EventArgs e)

{

//dataGridView1.DataSource = ListsDLUser.getListOfAdminandManager();

}

private void label3\_Click\_1(object sender, EventArgs e)

{

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

private void label3\_Click\_2(object sender, EventArgs e)

{

AddEmployeeForm ae1 = new AddEmployeeForm();

this.Hide();

ae1.Show();

}

private void label4\_Click(object sender, EventArgs e)

{

ViewAllEmployee vE = new ViewAllEmployee();

vE.Show();

this.Hide();

}

private void label7\_Click(object sender, EventArgs e)

{

ViewAllCustomer vC = new ViewAllCustomer();

vC.Show();

this.Hide();

}

private void monthCalendar1\_DateChanged(object sender, DateRangeEventArgs e)

{

}

private void label16\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 f = new Form1();

f.Show();

}

private void label9\_Click(object sender, EventArgs e)

{

this.Hide();

EmployeeRemove eR = new EmployeeRemove();

eR.Show();

}

private void label11\_Click(object sender, EventArgs e)

{

AdminPersonal aP = new AdminPersonal();

aP.Show();

this.Hide();

}

private void label10\_Click(object sender, EventArgs e)

{

CustomerRemove cR = new CustomerRemove();

cR.Show();

this.Hide();

}

private void label13\_Click(object sender, EventArgs e)

{

if(ListsDLUser.getlistOfBank().Count == 0)

{

Bank b = new Bank();

b.Show();

this.Hide();

}

else

{

MessageBox.Show("Bank Data Already Added!!!");

}

}

private void label15\_Click(object sender, EventArgs e)

{

BankData bD = new BankData();

bD.Show();

this.Hide();

}

private void label6\_Click(object sender, EventArgs e)

{

Employee\_UpdateThrooughAdmin eA = new Employee\_UpdateThrooughAdmin();

eA.Show();

this.Hide();

}

private void label12\_Click(object sender, EventArgs e)

{

AdminUpdate aU = new AdminUpdate();

aU.Show();

this.Hide();

}

private void label14\_Click(object sender, EventArgs e)

{

UpdateBank uB = new UpdateBank();

uB.Show();

this.Hide();

}

private void label2\_Click(object sender, EventArgs e)

{

AdminLoan aL = new AdminLoan();

aL.Show();

this.Hide();

}

private void notebox\_Click(object sender, EventArgs e)

{

Note n = new Note();

n.Show();

this.Hide();

}

}

}

**Admin Loan Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AdminLoan : Form

{

public AdminLoan()

{

InitializeComponent();

}

private void AdminLoan\_Load(object sender, EventArgs e)

{

long money = 0;

foreach (var a in CustomerDL.getListofCustomer())

{

money = a.Amount + money;

}

(moneyBox.Text) = money.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

}

}

**Admin Personal Form:**

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 BMS.DL;

using BMS.BL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AdminPersonal : Form

{

public AdminPersonal()

{

InitializeComponent();

}

private void AdminPersonal\_Load(object sender, EventArgs e)

{

dataGridView1.Rows.Add(ListsDLUser.getListOfAdminandManager()[0].Name, ListsDLUser.getListOfAdminandManager()[0].PhoneNo, ListsDLUser.getListOfAdminandManager()[0].CNIC, ListsDLUser.getListOfAdminandManager()[0].Address, ListsDLUser.getListOfAdminandManager()[0].Email, ListsDLUser.getListOfAdminandManager()[0].Password);

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

}

}

**Admin Update Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AdminUpdate : Form

{

public AdminUpdate()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string Name = NameBox.Text;

string Email = EmailBox.Text;

string password = PasswordBox.Text;

string PhoneNo = (PhoneNoBox.Text);

string CNIC = (CNICBox.Text);

string address = AddressBox.Text;

long phNO = Convert.ToInt64(PhoneNo);

long cnic = Convert.ToInt64(CNIC);

ListsDLUser.getListOfAdminandManager()[0].Name = Name;

ListsDLUser.getListOfAdminandManager()[0].Email = Email;

ListsDLUser.getListOfAdminandManager()[0].Password = password;

ListsDLUser.getListOfAdminandManager()[0].PhoneNo = phNO;

ListsDLUser.getListOfAdminandManager()[0].CNIC = cnic;

ListsDLUser.getListOfAdminandManager()[0].Address = address;

ListsDLUser.signUp\_File("SignIn\_SignUp.txt",Email,password,Name,address,phNO,cnic);

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

this.Hide();

AdminForm aF = new AdminForm();

aF.Show();

}

private void backButton\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

private void AdminUpdate\_Load(object sender, EventArgs e)

{

NameBox.Text = ListsDLUser.getListOfAdminandManager()[0].Name;

EmailBox.Text = ListsDLUser.getListOfAdminandManager()[0].Email;

PasswordBox.Text = ListsDLUser.getListOfAdminandManager()[0].Password;

PhoneNoBox.Text = ListsDLUser.getListOfAdminandManager()[0].PhoneNo.ToString();

CNICBox.Text = ListsDLUser.getListOfAdminandManager()[0].CNIC.ToString();

AddressBox.Text = ListsDLUser.getListOfAdminandManager()[0].Address;

}

}

}

**ATM Main Interface Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class Atm\_Main\_Interface : Form

{

public Atm\_Main\_Interface()

{

InitializeComponent();

}

private void Atm\_Main\_Interface\_Load(object sender, EventArgs e)

{

if(CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AccountType == "Saving")

{

label1.Text = "Your account Type is Saving so You only can add money";

}

else if (CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AccountType == "Current")

{

label1.Text = " Your account Type is Current so You can Perform Multiple Operations";

}

}

private void label3\_Click(object sender, EventArgs e)

{

CustomerMenu cM = new CustomerMenu();

cM.Show();

this.Hide();

}

private void label2\_Click(object sender, EventArgs e)

{

if (CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AccountType == "Current")

{

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

else if (CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AccountType == "Saving")

{

SavingAccountType cT = new SavingAccountType();

cT.Show();

this.Hide();

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

}

**ATM Update Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class AtmUpdate : Form

{

public AtmUpdate()

{

InitializeComponent();

}

private void typeCombo\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void AtmUpdate\_Load(object sender, EventArgs e)

{

IdBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AtmId;

passwordBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AtmPassword.ToString();

typeCombo.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AccountType.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AtmId = IdBox.Text;

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AtmPassword = Convert.ToInt32(passwordBox.Text);

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AccountType = typeCombo.Text;

CustomerDL.add\_CustomersInTo\_file2("Customer.txt",CustomerDL.getListofCustomer());

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

clear();

}

private void clear()

{

IdBox.Text = "";

passwordBox.Text = "";

typeCombo.Text = "";

}

private void button3\_Click(object sender, EventArgs e)

{

CustomerMenu cM = new CustomerMenu();

cM.Show();

this.Close();

}

}

}

**Bank Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class Bank : Form

{

public Bank()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

private void button2\_Click(object sender, EventArgs e)

{

AdminBL a = new AdminBL();

string BankName = BankNameBox.Text;

string BankEmail = BankEmailBox.Text;

string BankBranchID = BankBranchBox.Text;

long BankNo = Convert.ToInt64(BankNoBox.Text);

string BankAdddress = BankAddressBox.Text;

long loan = Convert.ToInt64(moneyLoanBox.Text);

a.setBankDetails(BankName, BankEmail, BankBranchID, BankAdddress, BankNo,loan);

ListsDLUser.setlistOfBank(a);

string path = "Bank\_Data.txt";

ListsDLUser.Store\_Bank\_Data\_Into\_File(path,a);

MessageBox.Show("Data Save Successfully!!!");

this.Hide();

AdminForm aF = new AdminForm();

aF.Show();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Bank Data Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class BankData : Form

{

public BankData()

{

InitializeComponent();

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void BankData\_Load(object sender, EventArgs e)

{

try

{

dataGridView1.Rows.Add(ListsDLUser.getlistOfBank()[0].BankName, ListsDLUser.getlistOfBank()[0].BankEmail, ListsDLUser.getlistOfBank()[0].BankAdddress, ListsDLUser.getlistOfBank()[0].BankNo, ListsDLUser.getlistOfBank()[0].BankBranchID, ListsDLUser.getlistOfBank()[0].BankMoney);

}

catch

{

MessageBox.Show("Bank Data is not Added Yet!!!");

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void Backbtt\_Click(object sender, EventArgs e)

{

Form1 f = new Form1();

f.Show();

this.Hide();

}

}

}

**Bank Money Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class BankMoney : Form

{

public BankMoney()

{

InitializeComponent();

}

private void BankMoney\_Load(object sender, EventArgs e)

{

long money = 0;

foreach(var a in CustomerDL.getListofCustomer())

{

money = a.Amount + money;

}

(moneyBox.Text) = money.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

EmployeeMenu af = new EmployeeMenu();

af.Show();

this.Hide();

}

}

}

**Check Balance Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class Check\_Balance : Form

{

public Check\_Balance()

{

InitializeComponent();

}

private void Check\_Balance\_Load(object sender, EventArgs e)

{

amountBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

SavingAccountType cT = new SavingAccountType();

cT.Show();

this.Hide();

}

}

}

**Current Account Type Form:**

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 SignIn\_SignUp\_Win\_Form

{

public partial class CurrentAccountType : Form

{

public CurrentAccountType()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

CurrentCheckBalance cB = new CurrentCheckBalance();

cB.Show();

this.Hide();

}

private void button1\_Click(object sender, EventArgs e)

{

AddMoneyInAccount aA = new AddMoneyInAccount();

this.Hide();

aA.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Atm\_Main\_Interface aI = new Atm\_Main\_Interface();

aI.Show();

this.Hide();

}

private void button4\_Click(object sender, EventArgs e)

{

PIN\_AND\_ATM pA = new PIN\_AND\_ATM();

pA.Show();

this.Hide();

}

private void button3\_Click(object sender, EventArgs e)

{

PayBills pB = new PayBills();

pB.Show();

this.Hide();

}

}

}

**Customer Menu Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class CustomerMenu : Form

{

public CustomerMenu()

{

InitializeComponent();

}

private void label2\_Click(object sender, EventArgs e)

{

CustomerPersonal cP = new CustomerPersonal();

cP.Show();

this.Hide();

}

private void label4\_Click(object sender, EventArgs e)

{

CustomerBankDetails cD = new CustomerBankDetails();

cD.Show();

this.Hide();

}

private void label6\_Click\_1(object sender, EventArgs e)

{

AtmUpdate aU = new AtmUpdate();

aU.Show();

this.Hide();

}

private void label3\_Click\_1(object sender, EventArgs e)

{

CustomerUpdate cU = new CustomerUpdate();

cU.Show();

this.Hide();

}

private void label5\_Click(object sender, EventArgs e)

{

Atm\_Main\_Interface aI = new Atm\_Main\_Interface();

aI.Show();

this.Hide();

}

private void label7\_Click(object sender, EventArgs e)

{

Form1 f1 = new Form1();

f1.Show();

this.Hide();

}

}

}

**Customer Personal Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class CustomerPersonal : Form

{

public CustomerPersonal()

{

InitializeComponent();

}

private void CustomerPersonal\_Load(object sender, EventArgs e)

{

dataGridView1.Rows.Add(CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Name, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Email, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Password, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].PhoneNo, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].CNIC, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Address, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AtmId, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AtmPassword, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].AccountType, CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount);

}

private void button1\_Click(object sender, EventArgs e)

{

CustomerMenu cM = new CustomerMenu();

cM.Show();

this.Hide();

}

}

}

**Customer Update Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class CustomerUpdate : Form

{

public CustomerUpdate()

{

InitializeComponent();

}

private void CustomerUpdate\_Load(object sender, EventArgs e)

{

NameBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Name;

EmailBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Email;

PasswordBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Password;

PhoneNoBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].PhoneNo.ToString();

CNICBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].CNIC.ToString();

AddressBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Address;

}

private void button2\_Click(object sender, EventArgs e)

{

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Name = NameBox.Text;

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Email = EmailBox.Text;

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Password = PasswordBox.Text;

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].CNIC = Convert.ToInt64(CNICBox.Text);

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Address = AddressBox.Text;

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].PhoneNo = Convert.ToInt64(PhoneNoBox.Text);

CustomerDL.add\_CustomersInTo\_file2("Customer.txt",CustomerDL.getListofCustomer());

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

clear();

}

private void clear()

{

NameBox.Text = "";

EmailBox.Text = "";

PasswordBox.Text = "";

CNICBox.Text = "";

PhoneNoBox.Text = "";

AddressBox.Text = "";

}

private void button1\_Click(object sender, EventArgs e)

{

CustomerMenu cM = new CustomerMenu();

cM.Show();

this.Hide();

}

}

}

**Electricity Bill Form:**

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 BMS.DL;

using BMS.BL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class ElectricityBill : Form

{

public ElectricityBill()

{

InitializeComponent();

}

private void ElectricityBill\_Load(object sender, EventArgs e)

{

currentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

if (Convert.ToInt64(gasBox.Text) > CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount)

{

MessageBox.Show("Low Money!!!");

MessageBox.Show("Are You sure you enter correct bill price");

ElectricityBill gB = new ElectricityBill();

gB.Show();

this.Hide();

}

else

{

long amount = Convert.ToInt64(gasBox.Text);

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount - amount;

MessageBox.Show("Payment Successfull!!!!");

currentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

CustomerDL.add\_CustomersInTo\_file2("Customer.txt", CustomerDL.getListofCustomer());

PayBills pB = new PayBills();

pB.Show();

this.Hide();

}

}

private void button2\_Click(object sender, EventArgs e)

{

PayBills pB = new PayBills();

pB.Show();

this.Hide();

}

}

}

**Employee Menu Form:**

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 SignIn\_SignUp\_Win\_Form

{

public partial class EmployeeMenu : Form

{

public EmployeeMenu()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

this.Hide();

AddCustomer ac = new AddCustomer();

ac.Show();

}

private void label8\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 f = new Form1();

f.Show();

}

private void label3\_Click(object sender, EventArgs e)

{

SearchCustomerByEmployee sE = new SearchCustomerByEmployee();

sE.Show();

this.Hide();

}

private void label7\_Click(object sender, EventArgs e)

{

BankData bD = new BankData();

bD.Show();

this.Hide();

}

private void label6\_Click(object sender, EventArgs e)

{

UpdateEmployee uE = new UpdateEmployee();

uE.Show();

this.Hide();

}

private void label4\_Click(object sender, EventArgs e)

{

BankMoney bM = new BankMoney();

bM.Show();

this.Hide();

}

private void label5\_Click(object sender, EventArgs e)

{

EmployeePersonalData eD = new EmployeePersonalData();

eD.Show();

this.Hide();

}

}

}

**Employee Personal Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class EmployeePersonalData : Form

{

public EmployeePersonalData()

{

InitializeComponent();

}

private void EmployeePersonalData\_Load(object sender, EventArgs e)

{

dataGridView1.Rows.Add(EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Email, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Name, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Password, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Address, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].PhoneNo, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].CNIC, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].EmployeeId, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Salary, EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].JoinDate);

}

private void button1\_Click(object sender, EventArgs e)

{

EmployeeMenu eM = new EmployeeMenu();

eM.Show();

this.Hide();

}

}

}

**Gas Bill Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class Gas\_Bill : Form

{

public Gas\_Bill()

{

InitializeComponent();

}

private void Gas\_Bill\_Load(object sender, EventArgs e)

{

currentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

if(Convert.ToInt64(gasBox.Text) > CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount)

{

MessageBox.Show("Low Money!!!");

MessageBox.Show("Are You sure you enter correct bill price");

Gas\_Bill gB = new Gas\_Bill();

gB.Show();

this.Hide();

}

else

{

long amount = Convert.ToInt64(gasBox.Text);

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount - amount;

MessageBox.Show("Payment Successfull!!!!");

currentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

CustomerDL.add\_CustomersInTo\_file2("Customer.txt",CustomerDL.getListofCustomer());

PayBills pB = new PayBills();

pB.Show();

this.Hide();

}

}

private void button2\_Click(object sender, EventArgs e)

{

PayBills pB = new PayBills();

pB.Show();

this.Hide();

}

}

}

**Loan Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class LoanFromBank : Form

{

public LoanFromBank()

{

InitializeComponent();

}

private void LoanFromBank\_Load(object sender, EventArgs e)

{

CurrentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

remainBox.Text = CustomerDL.loanDail.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

while (Convert.ToInt64(AmountBox.Text) > CustomerDL.loanDail)

{

MessageBox.Show("Limit Exceeded!!!");

AmountBox.Text = "";

return;

}

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount + Convert.ToInt64(AmountBox.Text);

CustomerDL.loanDail = CustomerDL.loanDail - Convert.ToInt64(AmountBox.Text);

MessageBox.Show("Amount Added Successfully!!!");

ListsDLUser.getlistOfBank()[0].BankMoney = ListsDLUser.getlistOfBank()[0].BankMoney - Convert.ToInt64(AmountBox.Text);

remainBox.Text = CustomerDL.loanDail.ToString();

CurrentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

AmountBox.Text = "";

CustomerDL.add\_CustomersInTo\_file2("Customer.txt", CustomerDL.getListofCustomer());

}

private void button1\_Click(object sender, EventArgs e)

{

SavingAccountType cT = new SavingAccountType();

cT.Show();

this.Close();

}

}

}

**Note Loan Form:**

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 SignIn\_SignUp\_Win\_Form

{

public partial class Note : Form

{

public Note()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

}

}

**Pay Bills Form:**

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 SignIn\_SignUp\_Win\_Form

{

public partial class PayBills : Form

{

public PayBills()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Gas\_Bill gB = new Gas\_Bill();

gB.Show();

this.Hide();

}

private void button3\_Click(object sender, EventArgs e)

{

ElectricityBill eB = new ElectricityBill();

eB.Show();

this.Hide();

}

private void button2\_Click(object sender, EventArgs e)

{

WaterBills wB = new WaterBills();

wB.Show();

this.Hide();

}

private void button4\_Click(object sender, EventArgs e)

{

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

}

}

**PIN AND ATM Form:**

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 BMS.DL;

using BMS.BL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class PIN\_AND\_ATM : Form

{

public PIN\_AND\_ATM()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string atmId = IdBox.Text;

int password = Convert.ToInt32(PINBox.Text);

bool flag = CustomerDL.verifyATM(atmId,password);

if(flag == false)

{

MessageBox.Show("Not Verify Enter ATM ID and PIN again");

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

else if(flag == true)

{

TransferAmount tA = new TransferAmount();

tA.Show();

this.Hide();

}

}

private void button2\_Click(object sender, EventArgs e)

{

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

}

}

**PIN and Atm Bills Form:**

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 BMS.DL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class PinAndAtmBills : Form

{

public PinAndAtmBills()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string atmId = IdBox.Text;

int password = Convert.ToInt32(PINBox.Text);

bool flag = CustomerDL.verifyATM(atmId, password);

if (flag == false)

{

MessageBox.Show("Not Verify Enter ATM ID and PIN again");

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

else if (flag == true)

{

PayBills pB = new PayBills();

pB.Show();

this.Hide();

}

}

private void button2\_Click(object sender, EventArgs e)

{

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

}

}

**Saving Account Type Form:**

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 SignIn\_SignUp\_Win\_Form

{

public partial class SavingAccountType : Form

{

public SavingAccountType()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

Atm\_Main\_Interface cM = new Atm\_Main\_Interface();

cM.Show();

this.Hide();

}

private void button1\_Click(object sender, EventArgs e)

{

LoanFromBank lB = new LoanFromBank();

lB.Show();

this.Hide();

}

private void button2\_Click(object sender, EventArgs e)

{

AddMoneyInAccount aA = new AddMoneyInAccount();

aA.Show();

this.Hide();

}

private void button4\_Click(object sender, EventArgs e)

{

Check\_Balance cB = new Check\_Balance();

cB.Show();

this.Hide();

}

}

}

**Search Customer By Employee Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class SearchCustomerByEmployee : Form

{

public SearchCustomerByEmployee()

{

InitializeComponent();

}

private void label16\_Click(object sender, EventArgs e)

{

EmployeeMenu eM = new EmployeeMenu();

eM.Show();

this.Hide();

}

private void label2\_Click(object sender, EventArgs e)

{

Form1 f1 = new Form1();

f1.Show();

this.Hide();

}

private void SearchCustomerByEmployee\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = CustomerDL.getListofCustomer();

}

private void searchButton\_Click(object sender, EventArgs e)

{

string customerEmail = customerEmailBox.Text;

CustomerDL.checkCustomer(customerEmail);

dataGridView1.Refresh();

dataGridView1.DataSource = CustomerDL.getlistOfCustomerForSearch();

}

public void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = CustomerDL.getlistOfCustomerForSearch();

dataGridView1.Refresh();

}

private void resetButton\_Click(object sender, EventArgs e)

{

customerEmailBox.Clear();

//List<EmployeeBL> employee = EmployeeDL.getlistOfEmployeeForSearch();

for (int i = 0; i < CustomerDL.getlistOfCustomerForSearch().Count; i++)

{

CustomerDL.getlistOfCustomerForSearch().RemoveAt(i);

}

dataGridView1.DataSource = null;

viewAllCustomerLoad(this, null);

dataGridView1.DataSource = CustomerDL.getListofCustomer();

}

private void viewAllCustomerLoad(object sender, EventArgs e)

{

refreshGrid();

}

private void refreshGrid()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = CustomerDL.getListofCustomer();

dataGridView1.Refresh();

}

}

}

**Transfer Amount Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class TransferAmount : Form

{

public TransferAmount()

{

InitializeComponent();

}

private void TransferAmount\_Load(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string atm = atmBox.Text;

string check = CustomerDL.verifyATM(atm);

CustomerDL.atmINdex = CustomerDL.indexForATM(atmBox.Text);

if(check == "same")

{

MessageBox.Show("OOPs!!! Sender and Reciever Both are the same This Transaction can not be perormed");

atmBox.Text = "";

}

else if(check == "false")

{

MessageBox.Show("OOPs!!! Reciever not found This Transaction can not be perormed");

atmBox.Text = "";

}

else

{

\_2Transfer\_Amount\_2 t = new \_2Transfer\_Amount\_2();

t.Show();

this.Hide();

}

}

private void button2\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

CurrentAccountType cT = new CurrentAccountType();

cT.Show();

this.Hide();

}

}

}

**Update Bank Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class UpdateBank : Form

{

public UpdateBank()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string name = NameBox.Text;

string email = EmailBox.Text;

string bankId = IdBox.Text;

string no = PhoneNoBox.Text;

long bankPhone = Convert.ToInt64(no);

string address = addresBox.Text;

string lo = textBox1.Text;

long money = Convert.ToInt64(lo);

AdminBL a = new AdminBL();

a.setBankDetails(name, email, bankId, address, bankPhone, money);

ListsDLUser.setlistOfBank(a);

string path = "Bank\_Data.txt";

ListsDLUser.Store\_Bank\_Data\_Into\_File(path, a);

MessageBox.Show("Update Successfully!!!");

this.Hide();

AdminForm aF = new AdminForm();

aF.Show();

}

private void backButton\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void AddressBox\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Update Employee Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class UpdateEmployee : Form

{

public UpdateEmployee()

{

InitializeComponent();

}

private void UpdateEmployee\_Load(object sender, EventArgs e)

{

MessageBox.Show("You can not Update Your Salary Id and Date Of Joining Only Admin Can Update it You can Only Update Other Personal Information");

emailbtt.Text = EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Email;

passwordbtt.Text = EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Password;

Namebtt.Text = EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Name;

addressBtt.Text = EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Address;

phnobtt.Text = Convert.ToString(EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].PhoneNo);

cnicbtt.Text = Convert.ToString(EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].CNIC);

salaryBtt.Text = Convert.ToString(EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Salary);

idbtt.Text = Convert.ToString(EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].EmployeeId);

dateBox.Text = EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].JoinDate.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Name = Namebtt.Text;

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Email = emailbtt.Text;

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Password = passwordbtt.Text;

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].CNIC = Convert.ToInt64(cnicbtt.Text);

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].PhoneNo = Convert.ToInt64(phnobtt.Text);

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Address = addressBtt.Text;

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].EmployeeId = idbtt.Text;

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].Salary = Convert.ToInt64(salaryBtt.Text);

EmployeeDL.getListofEmloyee()[EmployeeDL.specificIndexForEmployee].JoinDate = Convert.ToDateTime(dateBox.Text);

EmployeeDL.add\_EmployeesInTo\_file2("Employee.txt", EmployeeDL.getListofEmloyee());

MessageBox.Show("Updated Successfully");

clear();

}

private void clear()

{

Namebtt.Text = "";

emailbtt.Text = "";

passwordbtt.Text = "";

cnicbtt.Text = "";

phnobtt.Text = "";

addressBtt.Text = "";

idbtt.Text = "";

salaryBtt.Text = "";

dateBox.Text = "";

}

private void button1\_Click(object sender, EventArgs e)

{

EmployeeMenu eM = new EmployeeMenu();

eM.Show();

this.Hide();

}

}

}

**View All Customers Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class ViewAllCustomer : Form

{

public ViewAllCustomer()

{

InitializeComponent();

}

private void ViewAllCustomer\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = CustomerDL.getListofCustomer();

}

private void customerGV\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

private void label16\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

private void label2\_Click(object sender, EventArgs e)

{

Form1 f1 = new Form1();

f1.Show();

this.Hide();

}

private void customerEmailBox\_TextChanged(object sender, EventArgs e)

{

}

private void searchButton\_Click(object sender, EventArgs e)

{

string customerEmail = customerEmailBox.Text;

CustomerDL.checkCustomer(customerEmail);

dataGridView1.Refresh();

dataGridView1.DataSource = CustomerDL.getlistOfCustomerForSearch();

}

private void resetButton\_Click(object sender, EventArgs e)

{

customerEmailBox.Clear();

//List<EmployeeBL> employee = EmployeeDL.getlistOfEmployeeForSearch();

for (int i = 0; i < CustomerDL.getlistOfCustomerForSearch().Count; i++)

{

CustomerDL.getlistOfCustomerForSearch().RemoveAt(i);

}

dataGridView1.DataSource = null;

viewAllCustomerLoad(this, null);

dataGridView1.DataSource = CustomerDL.getListofCustomer();

}

private void viewAllCustomerLoad(object sender, EventArgs e)

{

refreshGrid();

}

private void refreshGrid()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = CustomerDL.getListofCustomer();

dataGridView1.Refresh();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}

**Employees Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class ViewAllEmployee : Form

{

public ViewAllEmployee()

{

InitializeComponent();

}

private void ViewAllEmployee\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = EmployeeDL.getListofEmloyee(); // introspection

}

private void label3\_Click(object sender, EventArgs e)

{

Form1 f1 = new Form1();

f1.Show();

this.Hide();

}

private void tableLayoutPanel2\_Paint(object sender, PaintEventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void searchButton\_Click(object sender, EventArgs e)

{

string employeeEmail = employeeEmailBox.Text;

EmployeeDL.checkEmployee(employeeEmail);

dataGridView1.Refresh();

List<EmployeeBL> employee = EmployeeDL.getlistOfEmployeeForSearch();

dataGridView1.DataSource = employee;

}

public void dataBind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = EmployeeDL.getlistOfEmployeeForSearch();

dataGridView1.Refresh();

}

private void label2\_Click\_1(object sender, EventArgs e)

{

Form1 f = new Form1();

f.Show();

this.Hide();

}

private void label16\_Click(object sender, EventArgs e)

{

AdminForm aF = new AdminForm();

aF.Show();

this.Hide();

}

private void resetButton\_Click(object sender, EventArgs e)

{

employeeEmailBox.Clear();

List<EmployeeBL> employee = EmployeeDL.getlistOfEmployeeForSearch();

for (int i = 0; i < EmployeeDL.getlistOfEmployeeForSearch().Count; i++)

{

EmployeeDL.getlistOfEmployeeForSearch().RemoveAt(i);

}

dataGridView1.DataSource = null;

viewAllemployeeLoad(this,null);

dataGridView1.DataSource = EmployeeDL.getListofEmloyee();

}

private void viewAllemployeeLoad(object sender, EventArgs e)

{

refreshGrid();

}

private void refreshGrid()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = EmployeeDL.getListofEmloyee();

dataGridView1.Refresh();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}

**Water Bills Form:**

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 BMS.BL;

using BMS.DL;

namespace SignIn\_SignUp\_Win\_Form

{

public partial class WaterBills : Form

{

public WaterBills()

{

InitializeComponent();

}

private void WaterBills\_Load(object sender, EventArgs e)

{

currentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

if (Convert.ToInt64(gasBox.Text) > CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount)

{

MessageBox.Show("Low Money!!!");

MessageBox.Show("Are You sure you enter correct bill price");

WaterBills wB = new WaterBills();

wB.Show();

this.Hide();

}

else

{

long amount = Convert.ToInt64(gasBox.Text);

CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount - amount;

MessageBox.Show("Payment Successfull!!!!");

currentBox.Text = CustomerDL.getListofCustomer()[CustomerDL.specificIndexForCustomer].Amount.ToString();

CustomerDL.add\_CustomersInTo\_file2("Customer.txt", CustomerDL.getListofCustomer());

PayBills pB = new PayBills();

pB.Show();

this.Hide();

}

}

}

}

END