using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EVALUATION

{

class Customer

{

private static int accountNumberCounter = 100000000;

private static Random random = new Random();

public int AccountNumber { get; set; }

public string CustomerName { get; set; }

public string AccountType { get; set; }

public double Balance { get; set; }

public double MinimumBalance { get; set; }

public string MobileNumber { get; set; }

public string EmailId { get; set; }

public int ATM\_Pin { get; set; }

public string PANNumber { get; set; }

public Customer(string customerName, string accountType, double balance, double minimumBalance, string mobileNumber, string emailId, string panNumber = "")

{

AccountNumber = accountNumberCounter++;

CustomerName = customerName;

AccountType = accountType;

Balance = balance;

MinimumBalance = minimumBalance;

MobileNumber = mobileNumber;

EmailId = emailId;

ATM\_Pin = random.Next(1000, 9999);

PANNumber = panNumber;

}

}

class Bank

{

private List<Customer> customers;

public Bank()

{

customers = new List<Customer>();

}

public void AddCustomer(string customerName, string accountType, double balance, double minimumBalance, string mobileNumber, string emailId, string panNumber)

{

customers.Add(new Customer(customerName, accountType, balance, minimumBalance, mobileNumber, emailId, panNumber));

Console.WriteLine("Customer added successfully.");

}

public void UpdateCustomerDetails(int accountNumber, string mobileNumber, string emailId)

{

Customer customer = customers.Find(c => c.AccountNumber == accountNumber);

if (customer == null)

{

Console.WriteLine("Account number not found.");

return;

}

customer.MobileNumber = mobileNumber;

customer.EmailId = emailId;

Console.WriteLine("Customer details updated successfully.");

}

public void DeleteCustomer(int accountNumber)

{

Customer customer = customers.Find(c => c.AccountNumber == accountNumber);

if (customer == null)

{

Console.WriteLine("Account number not found.");

return;

}

customers.Remove(customer);

Console.WriteLine("Customer deleted successfully.");

}

public void DisplayAllCustomers()

{

Console.WriteLine("List of all customers in the bank:");

foreach (Customer customer in customers)

{

Console.WriteLine("Account Number: {0}", customer.AccountNumber);

Console.WriteLine("Customer Name: {0}", customer.CustomerName);

Console.WriteLine("Account Type: {0}", customer.AccountType);

Console.WriteLine("Balance: {0}", customer.Balance);

Console.WriteLine("Minimum Balance: {0}", customer.MinimumBalance);

Console.WriteLine("Mobile Number: {0}", customer.MobileNumber);

Console.WriteLine("Email Id: {0}", customer.EmailId);

Console.WriteLine("PAN ID: {0}", customer.PANNumber);

Console.WriteLine("ATM Pin: {0}", customer.ATM\_Pin);

Console.WriteLine();

}

}

class Program

{

static void Main(string[] args)

{

Bank bank = new Bank();

int choice;

int num;

Console.WriteLine("------ATM TRANSACTIONS-------");

Console.WriteLine("Are you");

Console.WriteLine("1. Admin");

Console.WriteLine("2. Customer");

num = int.Parse(Console.ReadLine());

if (num == 1)

{

do

{

Console.WriteLine("1. Add a new customer");

Console.WriteLine("2. Update customer details");

Console.WriteLine("3. Delete a customer");

Console.WriteLine("4. Display all customers");

Console.WriteLine("5. Display customer details of a specific customer");

Console.WriteLine("6. Exit");

Console.WriteLine("Enter your choice: ");

choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine("Enter customer name: ");

string customerName = Console.ReadLine();

Console.WriteLine("Enter account type (Savings/Current): ");

string accountType = Console.ReadLine();

Console.WriteLine("Enter balance: ");

double balance = double.Parse(Console.ReadLine());

Console.WriteLine("Enter minimum balance: ");

double minimumBalance = double.Parse(Console.ReadLine());

Console.WriteLine("Enter mobile number: ");

string mobileNumber = Console.ReadLine();

Console.WriteLine("Enter email id: ");

string emailId = Console.ReadLine();

Console.WriteLine("Enter PAN No: ");

string panNumber = Console.ReadLine();

Console.WriteLine("Account number is : ");

bank.AddCustomer(customerName, accountType, balance, minimumBalance, mobileNumber, emailId, panNumber);

break;

case 2:

Console.WriteLine("Enter account number: ");

int accountNumber = int.Parse(Console.ReadLine());

Console.WriteLine("Enter new mobile number: ");

mobileNumber = Console.ReadLine();

Console.WriteLine("Enter new email id: ");

emailId = Console.ReadLine();

bank.UpdateCustomerDetails(accountNumber, mobileNumber, emailId);

break;

case 3:

Console.WriteLine("Enter account number: ");

accountNumber = int.Parse(Console.ReadLine());

bank.DeleteCustomer(accountNumber);

break;

case 4:

bank.DisplayAllCustomers();

break;

case 5:

Console.WriteLine("Enter account number: ");

accountNumber = int.Parse(Console.ReadLine());

Customer customer = bank.customers.Find(c => c.AccountNumber == accountNumber);

if (customer == null)

{

Console.WriteLine("Account number not found.");

break;

}

Console.WriteLine("Account Number: {0}", customer.AccountNumber);

Console.WriteLine("Customer Name: {0}", customer.CustomerName);

Console.WriteLine("Account Type: {0}", customer.AccountType);

Console.WriteLine("Balance: {0}", customer.Balance);

Console.WriteLine("Minimum Balance: {0}", customer.MinimumBalance);

Console.WriteLine("Mobile Number: {0}", customer.MobileNumber);

Console.WriteLine("Email Id: {0}", customer.EmailId);

Console.WriteLine("ATM Pin: {0}", customer.ATM\_Pin);

break;

case 6:

Console.WriteLine("Exiting the program...");

break;

default:

Console.WriteLine("Invalid choice. Please try again.");

break;

}

} while (choice != 6);

}

else if (num == 2)

{

int amount = 1000, deposit, withdraw;

int choice1, pin = 0, x = 0;

Console.WriteLine("Enter Your Pin Number ");

pin = int.Parse(Console.ReadLine());

while (true)

{

Console.WriteLine("\*\*\*\*\*\*\*\*Welcome to ATM Service\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("1. Check Balance\n");

Console.WriteLine("2. Withdraw Cash\n");

Console.WriteLine("3. Deposit Cash\n");

Console.WriteLine("4. Quit\n");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

Console.WriteLine("Enter your choice: ");

choice1 = int.Parse(Console.ReadLine());

switch (choice1)

{

case 1:

Console.WriteLine("\n YOUR BALANCE IN Rs : {0} ", amount);

break;

case 2:

Console.WriteLine("\n ENTER THE AMOUNT TO WITHDRAW: ");

withdraw = int.Parse(Console.ReadLine());

if (withdraw % 100 != 0)

{

Console.WriteLine("\n PLEASE ENTER THE AMOUNT IN MULTIPLES OF 100");

}

else if (withdraw > (amount - 500))

{

Console.WriteLine("\n INSUFFICENT BALANCE");

}

else

{

amount = amount - withdraw;

Console.WriteLine("\n\n PLEASE COLLECT CASH");

Console.WriteLine("\n YOUR CURRENT BALANCE IS {0}", amount);

}

break;

case 3:

Console.WriteLine("\n ENTER THE AMOUNT TO DEPOSIT");

deposit = int.Parse(Console.ReadLine());

amount = amount + deposit;

Console.WriteLine("YOUR BALANCE IS {0}", amount);

break;

case 4:

Console.WriteLine("\n THANK U USING ATM");

break;

}

}

Console.WriteLine("\n\n THANKS FOR USING OUT ATM SERVICE");

Console.ReadKey();

}

}

}

}

}