BANK MANAGEMENT SYSTEM

Zack Bank

PROJECT REPORT

This document contains the report of banking system to elaborate the project.

Zafeer Hafeez
Fa-2020-BSCS-022

Table of Contents

Introduction

Banking System is developed in C++ to replaced manual System by a computerized system. This system allows to create a new account and allows to deposit and withdrawal amount facilities. Name of this banking system is **Zack**.

Goal

To design and build a banking management system in C++ using learned Object-Oriented techniques.

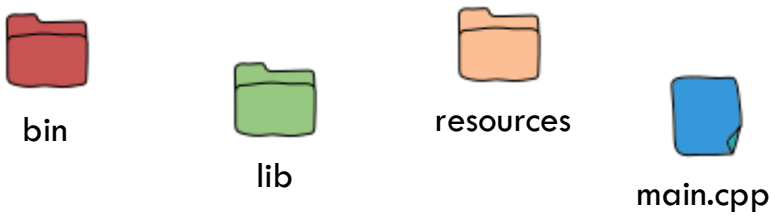
Deployment

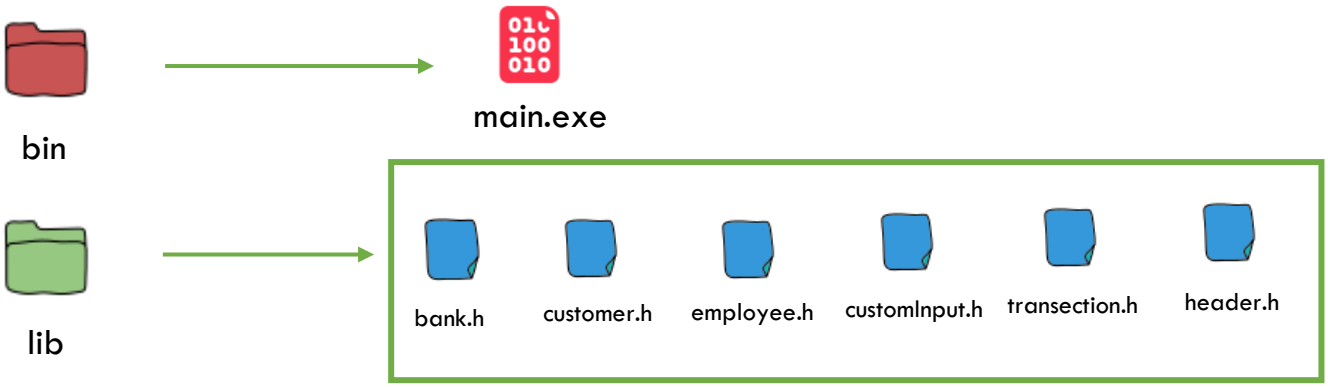
The built system will be deployed in a bank, so should work from both user and bank employee's perspectives.

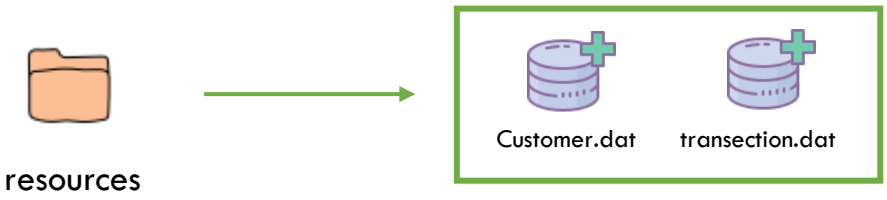
Project Explanation

File Management

We are using the following naming system for our project:







Execution of Program

```
*****************************
          Zack Bank
*****************************
1. Login as Bank Employee.
2. Login as Bank Customer.
0. Exit
```

Login

If we choose 1 then it will ask for employee credentials which are

Name: user

Password: zack123

Interesting thing is, the password input. **passwdInput()** is responsible for this functionality and it is defined in customInput.h.

```
*****************************
          Zack Bank
*****************************
1. Login as Bank Employee.
2. Login as Bank Customer.
0. Exit

Name: user
Password: *******
```

```
#pragma once

void passwdInput(char *password){
  int index=0;
  char character;

  while(character≠'\r' && index<100){
    character = getch();
    if(character≠'\r'){
        if(character≠'\b' && character≠'\t'){
            cout << "*";
            password[index] = character;
            index++;
        }
        else if(index≠0 && character≠'\t'){
            index--;
            password[index]= '\0';
            cout << '\b' << " " << '\b';
        }
    }
  }
  password[index] = '\0';
  cout << endl;
}
```

Employee

menuHandler()

```
*****************************
          Empoyee
*****************************
1. New Account
2. Update Account
3. Remove Account
4. Get Infomation of an account
5. Display All Accounts
6. Withdraw Money
7. Deposit Money
0. Logout
```

```
/***********************
        menu()
***********************/
void EmployeeManagement::menu(){
  banner();
  cout << "1. New Account" << endl;
  cout << "2. Update Account" << endl;
  cout << "3. Remove Account" << endl;
  cout << "4. Get Infomation of an account" << endl;
  cout << "5. Display All Accounts" << endl;
  cout << "6. Withdraw Money" << endl;
  cout << "7. Deposit Money" << endl;
  cout << "0. Logout" << endl;

  do{
  choice = getch();
  }
  while( choice≠'1' && choice≠'2' && choice≠'3'
        && choice≠'4' && choice≠'5' && choice≠'6'
       && choice≠'7' && choice≠'0');
}
```

New Account

```
void CustomerManagement::newRegistration(){
  int ct, at;

  cin.clear();
  cin.ignore(124, '\n');
  cout << "Enter Name: "; cin.getline(customer.name,20);
  cout << "Enter CNIC: "; cin.getline(customer.cnic,16);
  cout << "Enter Address: "; cin.getline(customer.address,100);
  cout << "Enter Telephone: "; cin.getline(customer.telephoneNumber,14);
  cout << "Enter Date Of Birth: "; cin.getline(customer.dateOfBirht,15);
  cout << "1. Master Card"<< endl;
  cout << "2. Visa Card"<< endl;
  cout << "3. Local Card"<< endl;
  cin.clear();
  cin.ignore(124, '\n');
  cout << "Enter Credit Card Type: "; cin >> ct;
  cout << "1. Current Account"<< endl;
  cout << "2. Saving Account"<< endl;
  cin.clear();
  cin.ignore(124, '\n');
  cout << "Enter Account Type: "; cin >> at;
  cin.clear();
  cin.ignore(124, '\n');
  cout << "Enter Password: "; cin.getline(customer.password, 20);
```

```
    switch(ct){
      case MasterCard:
        customer.intrestRate = 18;
        customer.limit = 50000;
        customer.credtCardType=MasterCard;
        break;
      case VisaCard:
        customer.intrestRate = 10;
        customer.limit = 100000;
        customer.credtCardType=VisaCard;

        break;
      case 3:
        customer.intrestRate = 25;
        customer.limit = 10000;
        customer.credtCardType=LocalCard;
        break;
    }

    customer.cardNumber = cardNumGenerator();
    Sleep(10);
    customer.accountType = static_cast<AccountTypes>(at);

    customer.accountNumber = accountNumGenerator();

    customer.balance = 0;


  // Output to the file
  ofstream outFile(FILENAME, ios::app);

  if(outFile.is_open()){
    outFile.write((char*)&customer, sizeof(Customer));
    outFile.close();
  }
  else{
    cout << "Error: unable to open file" << endl;
  }
}
```

```
Enter Name: Husnain Haider
Enter CNIC: 3520234543453
Enter Address: 12A 90 Lahore
Enter Telephone: +923345434567
Enter Date Of Birth: 13 JUN 1990
1. Master Card
2. Visa Card
3. Local Card

Enter Credit Card Type: 1
1. Current Account
2. Saving Account
Enter Account Type: 1
Enter Password: 12345
```

Display All Accounts

```
| Name          | Account Number | CNIC          | Telephone     | Date of Birth | Credit Card    | Card Type   |
| Husnain Haider | 1993947919    | 3520234543453 | +923345434567 | 13 JUN 1990   | 1878280384     | Master Card |
```

```
void CustomerManagement::listAllAccounts(){
  system("cls");
  Customer tmpCustomer;
  ifstream inFile(FILENAME, ios::in);

  if(inFile.is_open()){
    userListHeader();

    while(inFile.read((char*)&tmpCustomer, sizeof(Customer))){
      output(tmpCustomer);
    }
    userListFooter();
    inFile.close();
  }
}
```

Update Account

```
void CustomerManagement::updateInfo(){
  int an,pos, at;

  cin.clear();
  cin.ignore(124, '\n');
  cout << "Enter Account Number: "; cin >> an;

  fstream file(FILENAME, ios::in | ios::out | ios::binary);

  while(!file.eof()){

    pos = file.tellg();
    file.read((char*)&customer, sizeof(Customer));

    if(customer.accountNumber == an){
      char choice;

      cout << "1. Name" << endl;
      cout << "2. Address" << endl;
      cout << "3. Telephone Number" << endl;
      cout << "4. Account Type" << endl;
      cout << "0. Back" << endl;
      cout << "Choose: "; cin >> choice;
  switch(choice){
    case '1':
      cin.clear();
      cin.ignore(124, '\n');
      cout << "Enter Name: "; cin.getline(customer.name, 20);
      break;
    case '2':
      cin.clear();
      cin.ignore(124, '\n');
      cout << "Enter Address: "; cin.getline(customer.address, 100);
      break;
    case '3':
      cin.clear();
      cin.ignore(124, '\n');
      cout << "Enter Telephone: "; cin.getline(customer.telephoneNumber, 14);
      break;
    case '4':
      cout << "1. Current Account: " << endl;
      cout << "2. Saving Account: " << endl;
      cin.clear();
      cin.ignore(124, '\n');
      cout << "Enter Account Type: "; cin >> at;
      customer.accountType = static_cast<AccountTypes>(at);
      break;
  }
```

```
            file.seekp(pos);

            file.write((char*)&customer, sizeof(Customer));

            file.close();

            break;
        }
    }
    file.close();
}
```

```
Enter Account Number: 1993947919
1. Name
2. Address
3. Telephone Number
4. Account Type
0. Back
Choose: 1
Enter Name: Zafeer Hafeez
```

Delete Account

```
void CustomerManagement::removeCustomer(){
  int an;
  char TMPFILENAME[] = "resources/tmp.dat";

  cout << "Enter Account Number: "; cin >> an;

  ifstream inFile(FILENAME);
  ofstream outFile(TMPFILENAME);

  while(inFile.read((char*)&customer, sizeof(Customer))){

    if(customer.accountNumber==an){
      cout << "Deleted Sucessfully!" << endl;
    }
    else{
      outFile.write((char*)&customer, sizeof(Customer));
    }
  }
  inFile.close();
  outFile.close();

  remove(FILENAME);
  rename(TMPFILENAME, FILENAME);
}
```

Get Information

```
void CustomerManagement::getInfo(int accountNum){
  ifstream inFile(FILENAME);

  while(!inFile.eof()){
    inFile.read((char*)&customer, sizeof(Customer));

    if(customer.accountNumber == accountNum){
      system("cls");
      cout << "Name: " << customer.name << endl;
      cout << "Date of Birth: " << customer.dateOfBirht << endl;
      cout << "Phone: " << customer.telephoneNumber << endl;
      cout << "CNIC: " << customer.cnic << endl;
      cout << "Address: " << customer.address << endl;
      cout << "Account Type: " << ((customer.accountType==CurrentAccount)?
                                   "Current Account": "Saving Account") << endl;
      cout << "Account Number: " << customer.accountNumber << endl;
      cout << "Credit Card Type: " << ((customer.credtCardType==MasterCard)?"Master Card":
                                       (customer.credtCardType==VisaCard)?"Visa Card": "Local Card")
                                    << endl;
      cout << "Credit Card Number: " << customer.cardNumber << endl;
      cout << "Balance: " << customer.balance << endl;
      cout << "Limit: " << customer.limit << endl;
      cout << "Intrest Rate: " << customer.intrestRate << "%" << endl;
      cout << "Password: " << customer.password << endl;

      inFile.close();
      return;
    }
  }
}
```

```
Name: Zafeer Hafeez
Date of Birth: 13 OCT 2009
Phone: +923345654567
CNIC: 3520254345673
Address: 132A 99 Lahore
Account Type: Current Account
Account Number: 1993947919
Credit Card Type: Visa Card
Credit Card Number: 1878280384
Balance: 0
Limit: 100000
Intrest Rate: 10%
Password: 12345
```

Deposit Amount

```
void CustomerManagement::deposit(){
  int money, an, pos, isSucessful=0;
  char state[10];
  cout << "Enter Account Number: "; cin >> an;
  cout << "Enter Money: "; cin >> money;

  fstream file(FILENAME, ios::in | ios::out | ios::binary);

  if(file.is_open()){
    file.seekp(0);

    while(!file.eof()){
      pos = file.tellg();
      file.read((char*)&customer, sizeof(Customer));
      if(customer.accountNumber==an){
        customer.balance += money;
        file.seekp(pos);

        file.write((char*)&customer, sizeof(Customer));
        TransectionManagement transection;

        strcpy(state, "deposited");
        transection.write(customer.accountNumber, money, state);

        cout << "Deposited Sucessfully!" << endl;

        file.close();
        isSucessful=1;
        break;
      }
    }
```

```
    if(!isSucessful){
      cout << "Unable to Deposit." << endl;
    }
    file.close();
  }
}
```

```
Enter Account Number: 1993947919
Enter Money: 20000
Deposited Sucessfully!
```

Withdraw Money

```
1. Checkbook
2. Credit Card
Enter Credit Card Number: 1993947919
Enter Ammount: 2000
```

Customer

menuHandler()

```
*********************************************************
 Name: Zafeer Hafeez              Balance: Rs. 20000
 Account Number: 1993947919       Credit Card:1878280384
*********************************************************
1. Mini Statement
2. Withdraw Money Via Credit Card
3. Withdraw Money Via Check Book
4. Balance Inquiry
5. Transection History
6. Account Information
0. Logout
```

Mini Statement

```
void getTrans(int accountNum, int limit=10){
  system("cls");
  int count=1;
  ifstream inFile(FILENAME);

  tranHeader();

  if(limit==10){
    while(inFile.read((char*)&transection, sizeof(Transection))){
      if(transection.accountNumber==accountNum && count<=limit){
        output(transection);
        count++;
      }
    }
  }else{
    while(inFile.read((char*)&transection, sizeof(Transection))){
      if(transection.accountNumber==accountNum){
        output(transection);
      }
    }
  }
  tranFooter();
  inFile.close();
}
```

```
 _____________________________________________
| Account No.        | Amount      | Status     |
|____________________|_____________|____________|
| 1993947919         | 20000       | deposited  |
|_____________________________________________|
```

All other options are same as in employee.

Conclusion

This project can be a good alternative to manual system of banking and can be even more effective than the old one. We can add more features in future which relief both the customers and the employees.