JNCT

LNCT Group of Colleges

Object-Oriented Programming Methodology Case Study

Banking System

Phase 3

Team: HelloWorld

Mentor 3: Prof Bhawna D
Mentor 2: Prof Kalpana M
Mentor 1: Dr A. Pathre

Team Members:

- Vishwasjeet Kr Gupta (Leader- 0131CS221225)
- Sunil Rathore (0131CS221208)
- Ram Vinay Kumar (LE)
- Vivek Agarwal (0131CS221226)

Department of Computer Science and Engineering

Object-Oriented Programming Methodology Case Study- Banking System

Contents:

- Summary of First Phase
- Summary of Second Phase
- Input-Output Flow
- Program Structure
- Git Hub

Object-Oriented Programming Methodology Case Study - Banking System

Brief Of First Phase

Objective

In the first phase, we presented our proposal for this case study. We found our objectives to create a class for customers and banks and perform the different transactions to check balance, deposit cash, withdraw cash, and transfer money. We proposed to create an easy to use and unique system.

Studies

We studied the best banking system applications available in the market. We observed all the pros and cons of those software and through our case study, we are proposing our system with some unique features which are not there in these existing applications.

Input-Output Analysis

We analyzed all the input and output requirements of a banking system. As banking service requires users' confidential data, we need to keep it safe and secure. The user's credentials should be verified before any transaction because unauthorized transactions may lead to fraud.

Object-Oriented Programming Methodology Case Study- Banking System

Brief Of Second Phase

✓ Class Diagram

In the second phase of this case study we presented the class diagram to visually represent the structure and relationships of classes.

✓ Use Case Diagram

To illustrate the interactions between system actors and functionalities, outline system behavior and user interactions for system modeling.

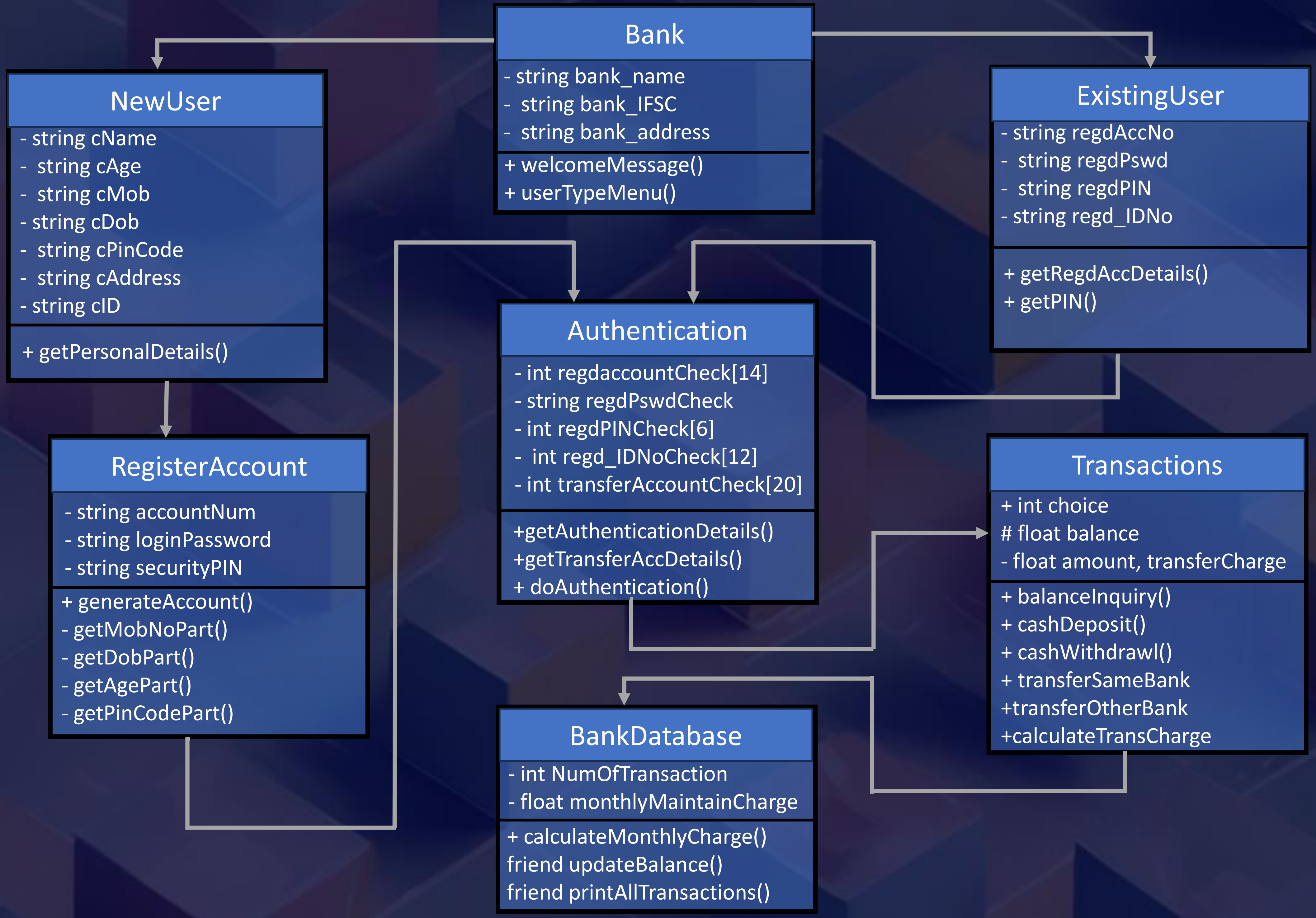
✓ Input Output Flow

To outline the data's journey into and out of a system, depicting its processing stages and interactions for system comprehension and analysis.

Hello World

Object-Oriented Programming Methodology Case Study - Banking System

Class-Object Diagram



Bank

- string bank_name
- string bank_IFSC
- string bank_address

+ welcomeMessage()
+ userTypeMenu()

NewUser

- string cName
- string cAge
- string cMob
- string cDob
- string cPinCode
- string cAddress
- string cID

+ getPersonalDetails()

ExistingUser

- string regdAccNo
- string regdPswd
- string regdPIN
- string regd_IDNo

+ getRegdAccDetails()
+ getPIN()

RegisterAccount

- string accountNum
- string loginPassword
- string securityPIN

+ generateAccount()
- getMobNoPart()
- getDobPart()
- getAgePart()
- getPinCodePart()

Authentication

- int regdaccountCheck[14]
- string regdPswdCheck
- int regdPINCheck[6]
- int regd_IDNoCheck[12]
- int transferAccountCheck[20]

+getAuthenticationDetails()
+getTransferAccDetails()
+ doAuthentication()

Transactions

+ int choice
# float balance
- float amount, transferCharge

+ balanceInquiry()
+ cashDeposit()
+ cashWithdrawl()
+ transferSameBank
+transferOtherBank
+calculateTransCharge

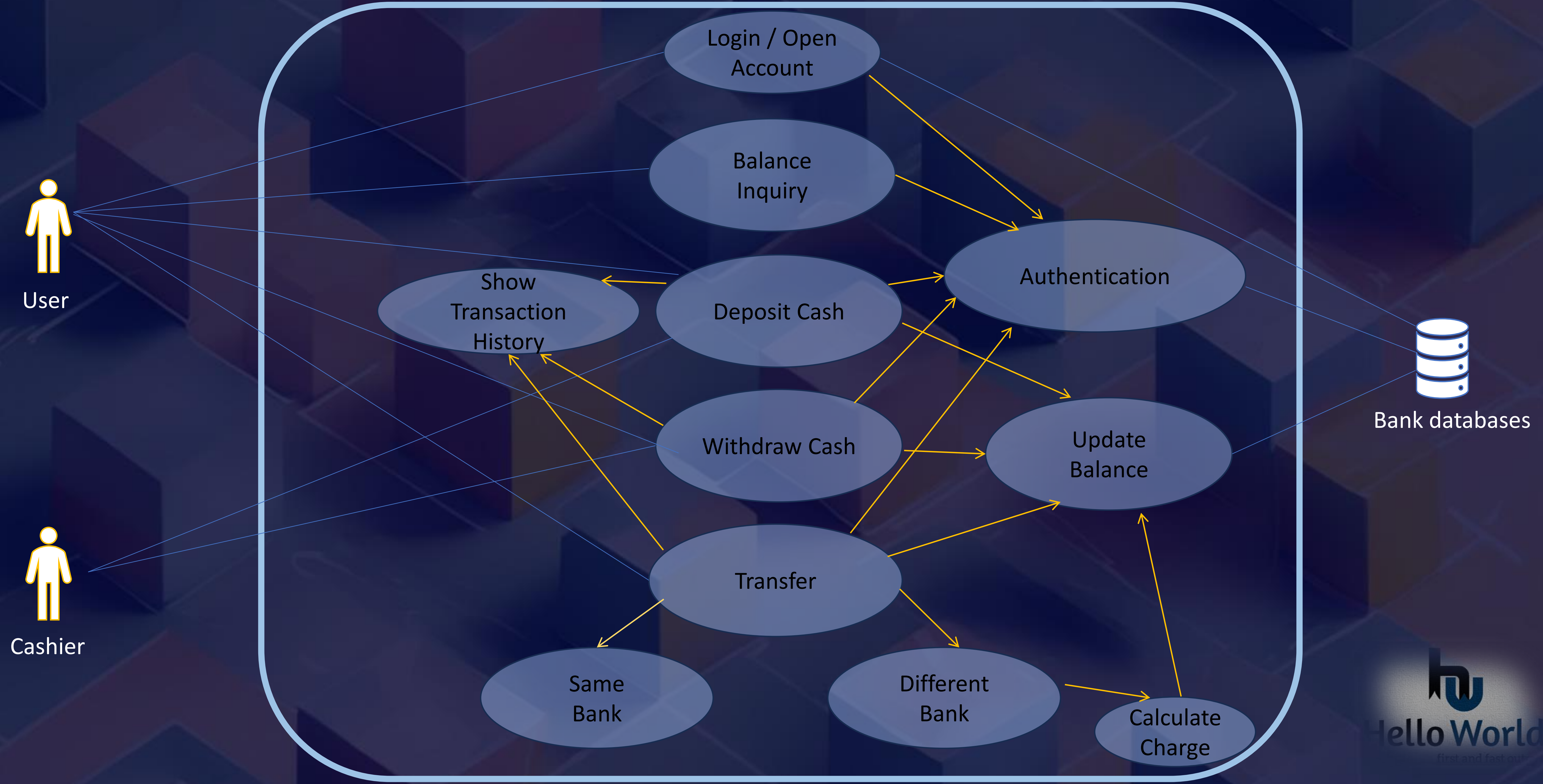
BankDatabase

- int NumOfTransaction
- float monthlyMaintainCharge

+ calculateMonthlyCharge()
friend updateBalance()
friend printAllTransactions()

Object-Oriented Programming Methodology Case Study - Banking System

Use-Case Diagram



Object Oriented Programming Methodology Case Study - Banking System

Input-Output Flow

Front Page

Option 1:New User

India Bank Limited
Hello Sir/Madam
Please select the option from the following menu to proceed:

1. New User
2. Existing User

Thank you for your interest in our bank.
Please fill the form given below to open your account :
1. Your Good Name: Rahul Kumar
2. DOB(DD-MM-YYYY): 02-03-2000
3. Mob No.: 9131555689
4. ID card No: 5678 4589 9590 (Aadhar/ Voter Id)
5. PAN: CUAPG035M
6. Age: 23
7. Address: Sector A, Indrapuri
8. PIN Code: 465647

Your Account is successfully created and your unique account number is generated:

A/c No.: 57020368923647

Please create your login password and transaction PIN to complete the account opening process:
Login Password: Rk0203@590
(min 8 char max 12)
PIN(6Digits): 689547

Congratulations your account is registered with us with the following details:

Name: Rahul Kumar
A/c No.: 57020368923647
IFSC: IBL005710
DOB: 02-03-2000
Mob No.: 9131555689
User Name: rk68935m
Address: Sector A, Indrapuri,
Pin-465647

Object Oriented Programming Methodology Case Study - Banking System

Input-Output Flow

Option 2:Existing User/New User After Account Opening

```
Please login with your credentials
to proceed:

User Name: rk68935m
Password: Rk0203@590

Authenticating…
Login Successful
You can proceed now…
```

```
Welcome back Mr/Mrs. Rahul
Kumar

Thank you for  logging in, your
registered details are:

Name: Rahul Kumar
A/c No.: 57020368923647
IFSC: IBL005710
DOB: 02-03-2000
Mob No.: 9131555689
Address: Sector A, Indrapuri,
              Pin-465647
```

Object Oriented Programming Methodology Case Study - Banking System

Input-Output Flow

Service Menu

```
Choose the services
needed:
1. Check Balance
2. Deposit Cash
3. Withdrawal Cash
4. Transfer money to the
same bank
5. Transfer money to
another Bank
6. Exit
 Enter your choice: 1

Available Balance: 0
```

```
Choose the services
needed:
1. Check Balance
2. Deposit Cash
3. Withdrawal Cash
4. Transfer money to same
the bank
5. Transfer money to other
Bank
6. Exit
Enter your choice: 2

Enter Amount: 50000
Cash Deposit Successful
Available Balance: 50000
```

```
Choose the services
needed:
1. Check Balance
2. Deposit Cash
3. Withdrawal Cash
4. Transfer money to the
same bank
5. Transfer money to the
other Bank
6. Exit
Enter your choice: 3

Enter Amount: 10000
Enter PIN: 689547
Cash Withdrawal Successful
Available Balance: 40000
```

```
Choose the services
needed:
1. Check Balance
2. Deposit Cash
3. Withdrawal Cash
4. Transfer money to the
same bank
5. Transfer to the other
Bank(Charges applied)
6. Exit
Enter your choice: 4

Enter Amount: 5000
Enter PIN: 689547
Money Transfer Successful
Available Balance: 35000
```

```
Choose the services
needed:
1. Check Balance
2. Deposit Cash
3. Withdrawal Cash
4. Transfer money to the
same bank
5. Transfer to the other
Bank (Charges applied)
6. Exit
Enter your choice: 5

Enter Amount: 10000
Enter PIN: 689547
Money Transfer Successful
Charge Deducted: ₹5
Available Balance: 24995
```

Object Oriented Programming Methodology Case Study - Banking System

Input-Output Flow

Transaction History

```
Your Transaction Details:
     Date               Transaction Type        Amount        Charge          Balance
1. 05/12/2023            Deposit                50000           0               50000
2. 05/12/2023            Withdrawal             10000           0               40000
3. 05/12/2023            Transfer                5000           0               35000
4. 05/12/2023            Transfer(other bank)   10000           5               24995
```

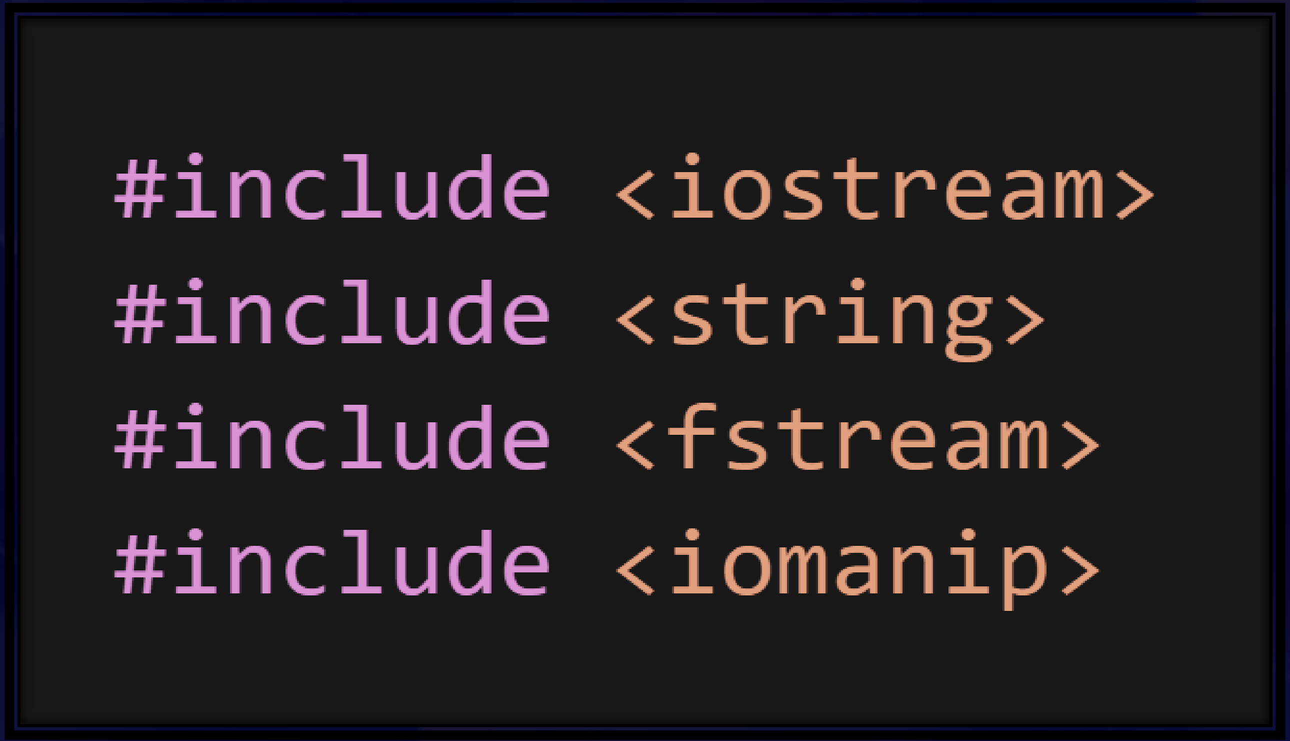
```
Account Summary:
Name: Rahul Kumar
A/c No.: 57020368923647
IFSC: IBL005710

Maintenance Charge: 0
Total Balance: 24995
```

Object-Oriented Programming Methodology Case Study - Banking System

Program Structure

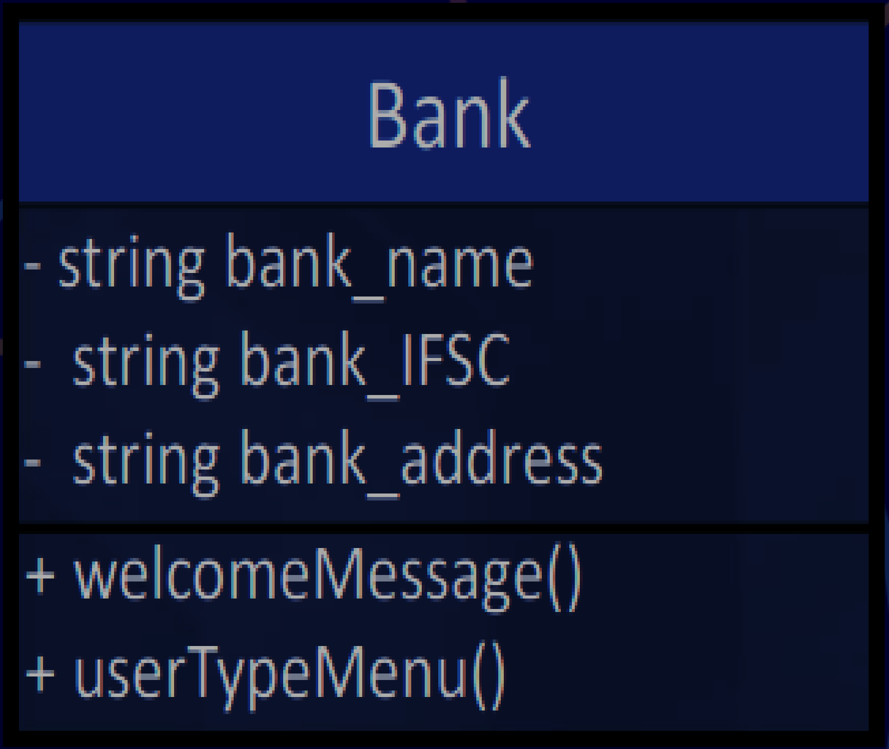
Header Files



```
#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>
```

Object Oriented Programming Methodology Case Study - Banking System

Program Structure



Bank

- string bank_name
- string bank_IFSC
- string bank_address

+ welcomeMessage()
+ userTypeMenu()

Bank class

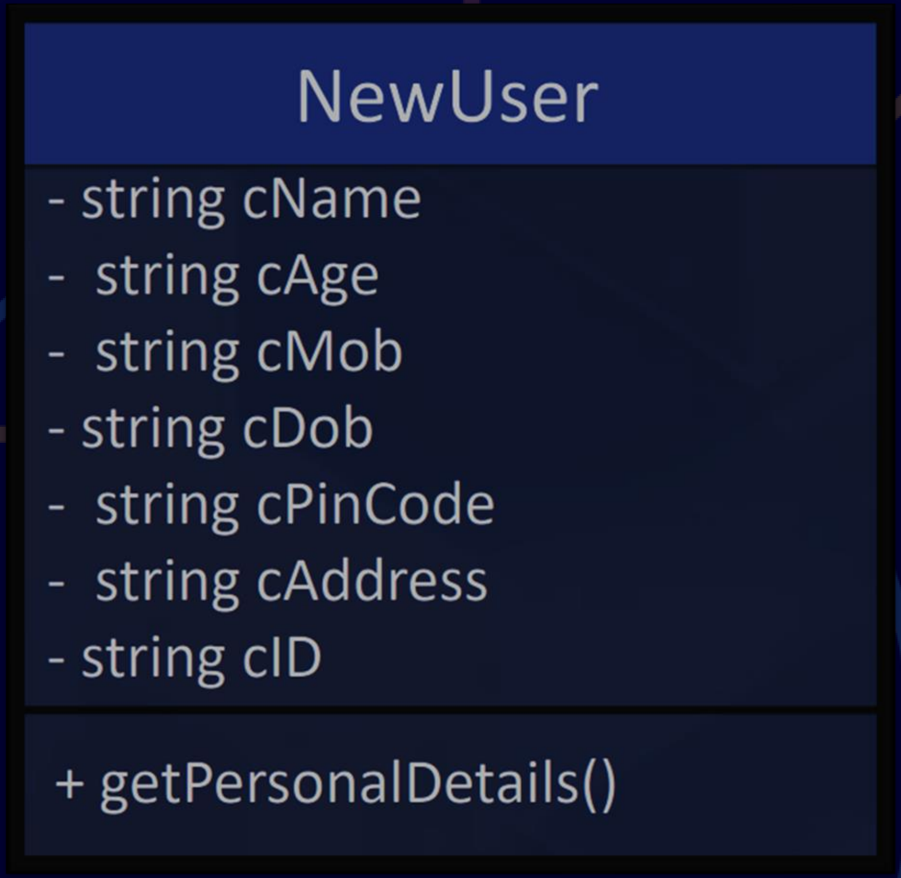
The Bank class stores the details of the bank like the bank's name, IFSC code, and Address. It has three data members and two member functions. Data members are kept private to ensure security and unauthorized access.

Data member bank_name stores the name of the bank, bank_IFSC stores the IFSC Code, and bank_address stores the address of the bank.

Member function welcomeMessage() displays the welcome screen and front page and the member function userTypeMenu() displays the first menu to select the user type.

Object Oriented Programming Methodology Case Study - Banking System

Program Structure



NewUser

- string cName
- string cAge
- string cMob
- string cDob
- string cPinCode
- string cAddress
- string cID

+ getPersonalDetails()

NewUser class

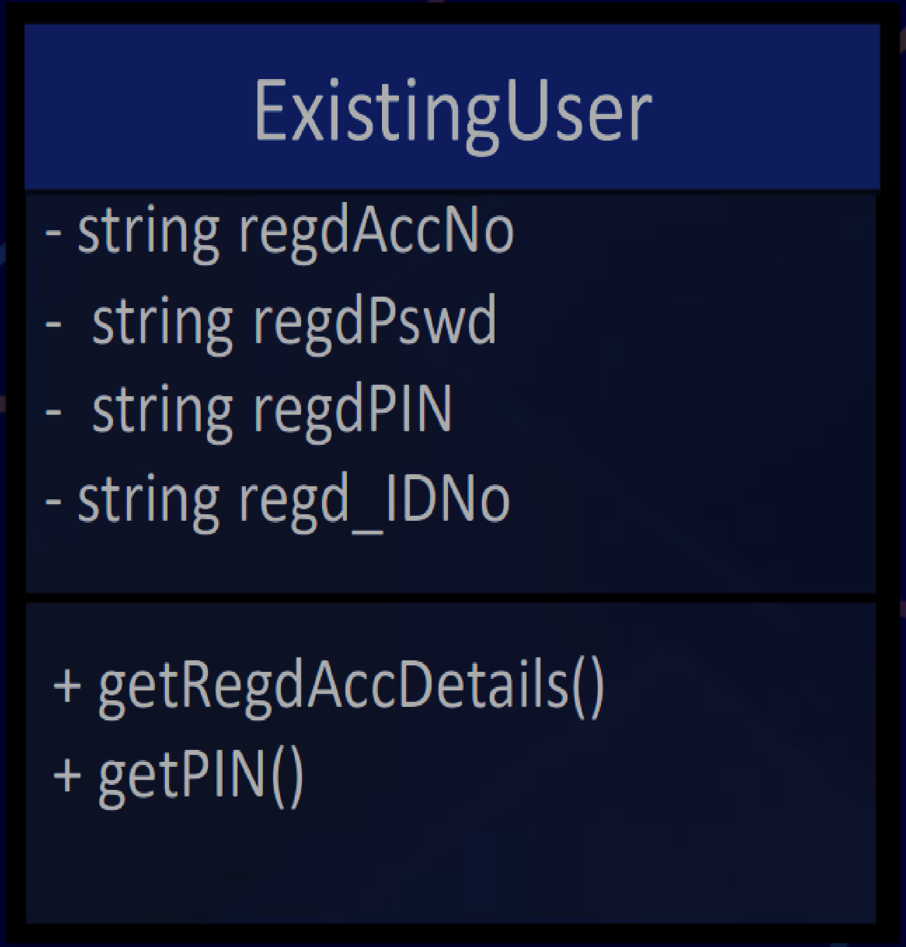
The NewUser class takes the user details as input and stores it through private data members. It has seven data members and one member function.

- string cName	: User's name
- string cAge	: User's Age
- string cMob	: User's mobile number
- string cDob	: User's Date of birth
- string cPinCode	: User's PIN code
- string cAddress	: User's Address
- string cID	: User's Identity card number

Member function +getPersonalDetails() - Takes input from the users and stores it in private data members that are already declared.

Object Oriented Programming Methodology Case Study - Banking System

Program Structure



ExistingUser

- string regdAccNo
- string regdPswd
- string regdPIN
- string regd_IDNo

+ getRegdAccDetails()
+ getPIN()

ExistingUser

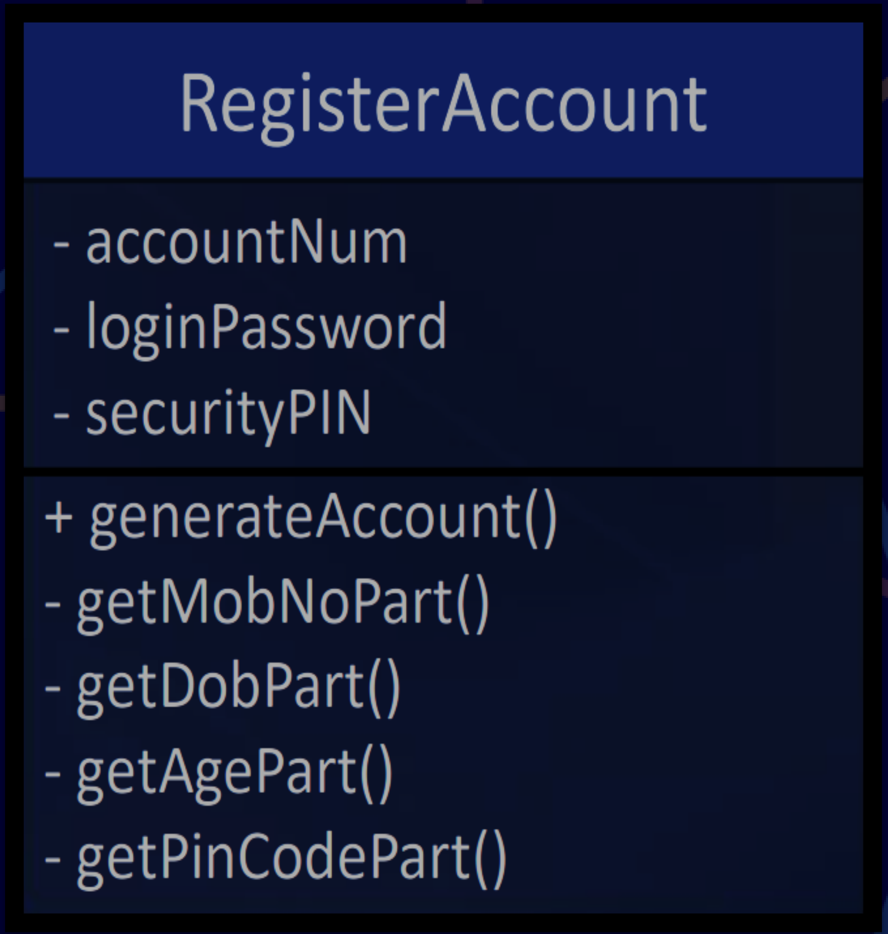
The ExistingUser class will act for the registered user. It is inherited from the bank class and it has four data members and two member functions member function.

- string regdAccNo : Registered User Account Number
- string regdPswd : Registered User Login Password
- string regdPIN : Registered User PIN
- string regd_IDNo : Registered User Identity Card Number

Member function +getRegdAccDetails() Takes input from the registered users and stores it in private data members. These data will be used in the authentication process in the child class. +getPIN() function will take the transaction PIN from the user.

Object Oriented Programming Methodology Case Study - Banking System

Program Structure



RegisterAccount

- accountNum
- loginPassword
- securityPIN

+ generateAccount()
- getMobNoPart()
- getDobPart()
- getAgePart()
- getPinCodePart()

RegisterAccount

The RegisterAccount class will act for the new user. It is inherited from the NewUser class. This class will be used to open a new bank account. It has three data members and fifteen member functions member function.

- string accountNum : Generated account number
- string loginPassword : Password created by user
- string securityPIN : Transaction PIN created by user

Member Functions

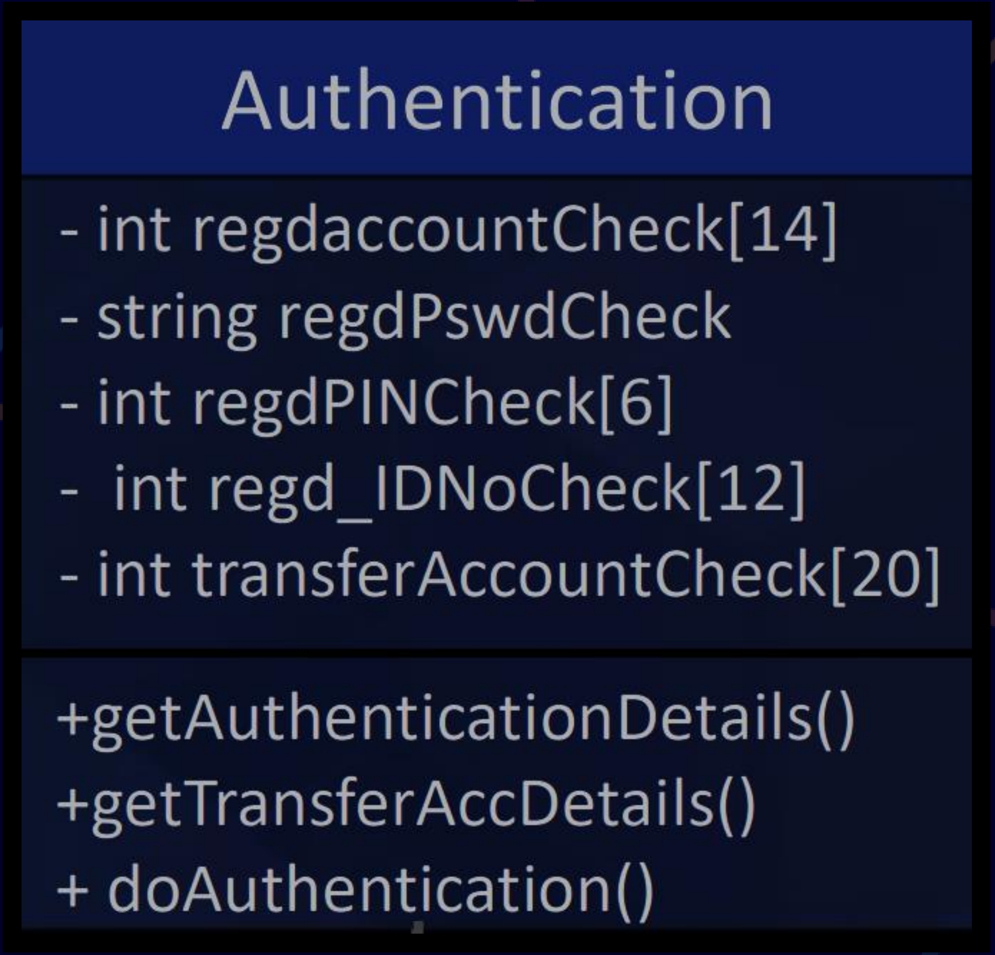
+ generateAccount() : To generate an account number for the new user.

- getMobNoPart()
- getDobPart()
- getAgePart()
- getPinCodePart()

To get combination of digits for generating a unique Account number.

Object Oriented Programming Methodology Case Study - Banking System

Program Structure



Authentication

- int regdaccountCheck[14]
- string regdPswdCheck
- int regdPINCheck[6]
- int regd_IDNoCheck[12]
- int transferAccountCheck[20]

+getAuthenticationDetails()
+getTransferAccDetails()
+ doAuthentication()

Authentication

The Authentication class will act for both type of users . It has two parent classes , RegisterAcccount and ExitsingUser.

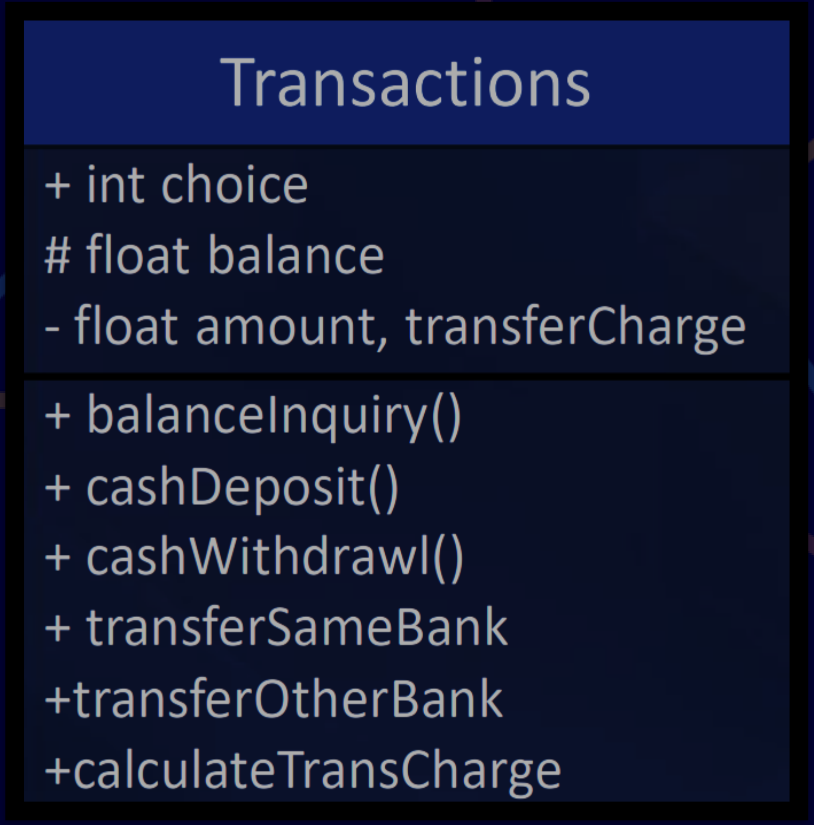
- int regdaccountCheck[14]   : Account number for authentication.
- string regdPswdCheck   : Password for Authentication.
- int regdPINCheck[6]   : Transaction Pin for Authentication.
- int regd_IDNoCheck[12]   : Identity card number for Authentication.
- int transferAccountCheck[20] : Account number Authentication for
money
transfer .

<u>Member Function :</u>

+getAuthenticationDetails() : Get details from the user for Authentication
+getTransferAccDetails()   : Get Account details for money transfer.
+ doAuthentication()   : To verify all the given details.

Object Oriented Programming Methodology Case Study - Banking System

Program Structure



Transactions

+ int choice
# float balance
- float amount, transferCharge

+ balanceInquiry()
+ cashDeposit()
+ cashWithdrawl()
+ transferSameBank
+transferOtherBank
+calculateTransCharge

Transactions

The Transactions class will act for both types of users. It is inherited from the Authentication class. This class will be used to perform all the transactions for the users. It has Four data members and Six member functions.

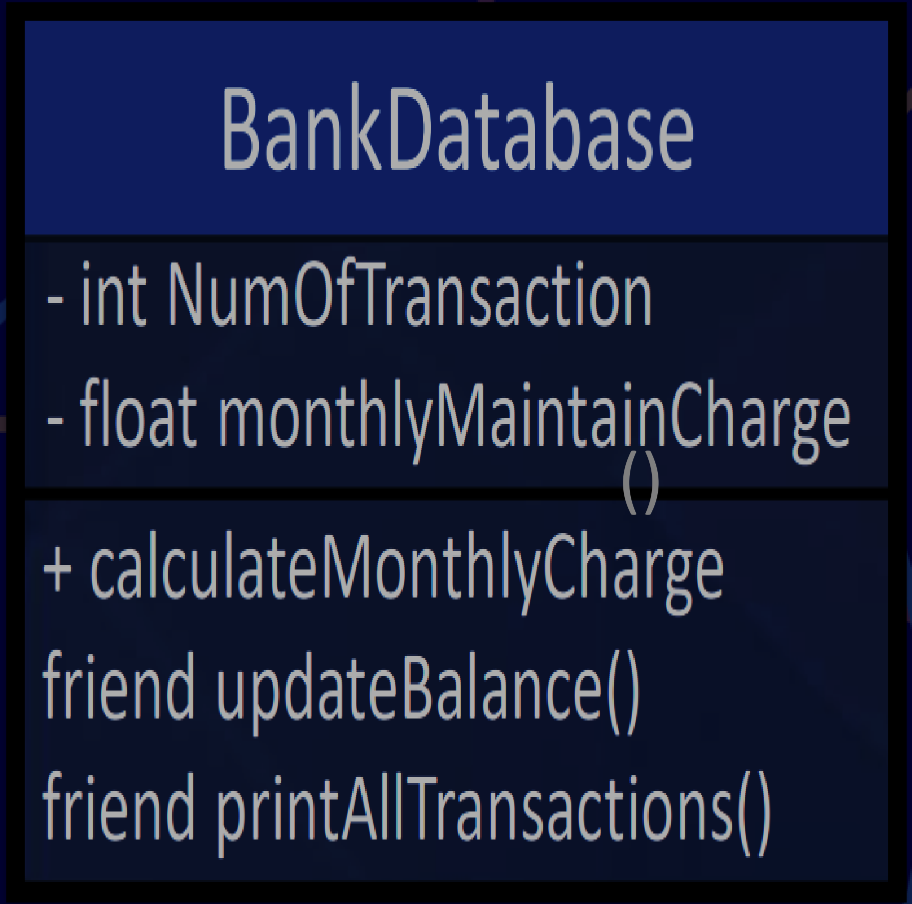
+ int choice : selecting the option from the list of services.
# float balance : Account opening balance.
- float amount : Transaction amount for Deposit, Withdrawal, and Transfer.
- transferCharge : Calculated transfer charge.

Member Function :

+ balanceInquiry() : To check account balance.
+ cashDeposit() : To deposit cash.
+ cashWithdrawl() : To withdraw cash.
+ transferSameBank : To transfer money to same bank.
+transferOtherBank : To transfer money to other bank .
+calculateTransCharge : To calculate transfer charges.

Object Oriented Programming Methodology Case Study - Banking System

Program Structure



BankDatabase

- int NumOfTransaction
- float monthlyMaintainCharge

()

+ calculateMonthlyCharge

friend updateBalance()

friend printAllTransactions()

BankDatabase

The BankDatabase class holds all the data of the user securely and it has attributes to keep the record of transactions. It has two data member and three member function.

- int NumOfTransaction : Store number of transactions.

- float monthlyMaintainCharge : Monthly maintenance charge for users.

Member function :

+ calculateMonthlyCharge : To calculate monthly charges.

friend updateBalance() :To update user's account balance.

friend printAllTransactions() : To print all transaction details.

Object Oriented Programming Methodology Case Study - Banking System

Git Hub

Click to go to the link:

HelloWorld-Banking-System

Object Oriented Programming Methodology Case Study - Banking System

Team: HelloWorld

Vishwasjeet Kr Gupta

(0131CS221225)
(Team Leader)

Sunil Rathore

(0131CS221208)
(Team Member)

Ram Vinay

(LE)
(Team Member)

Vivek Agrawal

(0131CS221226)
(Team Member)