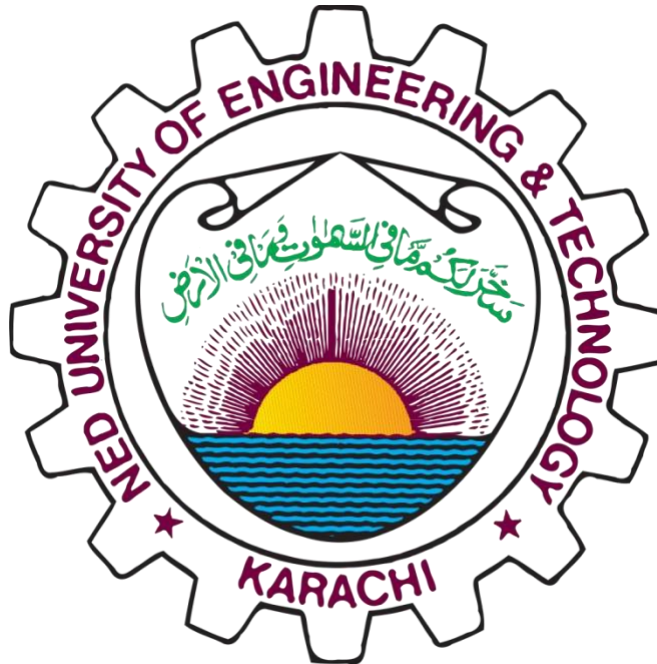


Online Bank System



F.E(CIS) CEP REPORT

Project Group ID:

Zaid Ahamed	CS-22154
Muhammed	CS-22155
Fathima Amna	CS-22156

BATCH:2022

Department of Computer and Information Systems Engineering

NED University of Engg. & Technology

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

Course Code: CS-116

Course Title: Object Oriented Programming

Complex Engineering Problem

FE Batch 2022, Spring Semester 2023

Grading Rubric

TERM PROJECT

Group Members:

Student No.	Name	Roll No.
S1		
S2		
S3		

CRITERIA AND SCALES				Marks Obtained		
				S1	S2	S3
Criterion 1: Does the application meet the desired specifications and produce the desired outputs? (CPA-1, CPA-3)						
1	2	3	4			
The application does not meet the desired specifications and is producing incorrect outputs.	The application partially meets the desired specifications and is producing incorrect or partially correct outputs.	The application meets the desired specifications but is producing incorrect or partially correct outputs.	The application meets all the desired specifications and is producing correct outputs.			
Criterion 2: How well is the code organization?						
1	2	3	4			
The code is poorly organized and very difficult to read.	The code is readable only to someone who knows what it is supposed to be doing.	Some part of the code is well organized, while some part is difficult to follow.	The code is well organized and very easy to follow.			
Criterion 3: How friendly is the application interface? (CPA-1, CPA-3)						
1	2	3	4			
The application interface is difficult to understand and use.	The application interface is easy to understand and but not that comfortable to use.	The application interface is very easy to understand and use.	The application interface is very interesting/innovative and easy to understand and use.			
Criterion 4: How does the student performed individually and as a team member? (CPA-2, CPA-3)						
1	2	3	4			
The student did not work on the assigned task.	The student worked on the assigned task, and accomplished goals partially.	The student worked on the assigned task, and accomplished goals satisfactorily.	The student worked on the assigned task, and accomplished goals beyond expectations.			
Criterion 5: Does the report adhere to the given format and requirements?						
1	2	3	4			
The report does not contain the required information and is formatted poorly.	The report contains the required information only partially but is formatted well.	The report contains all the required information but is formatted poorly.	The report contains all the required information and completely adheres to the given format.			
Total Marks:						

Table of Contents

S.No.	Page No.
1. Problem Description	4
2. Distinguishing features of project / list of object-oriented features implemented	5
3. Flow of project(output-based)	6-20
4. Class Diagram	21
5. Most challenging part while working on Project	22
6. Any new thing learned in Python while working on Project	23
7. Future expansions	24

Problem Description

This is a simple online banking system where users can hold different types of accounts as offered by the bank. The customers can perform operations according to the account type; e.g. deposit, withdraw, balance inquiry, and transferring funds. The banking system manages the accounts by debiting the fees or crediting the profits. Both the administrator and the customer can print reports on current account details.

The application contains the following:

Design: Basic Classes: Account, CheckingAccount, SavingAccount, Loan, Customer:

- A. The Account is a general account class that contains balance as instance variable, deposit, withdraw, and balanceEnquiry as instance methods.
- B. CheckingAccount is a subclass from the Account class that allows overdraft while withdrawing (making the balance go below zero up to the specified credit limit), by debiting the account balance with an overdraft fee. It has a credit limit as an instance variable.
- C. The Saving Account is a subclass from the Account class that has an interest rate as an instance variable. The system credits the balance with monthly interest based on the account balance and the interest rate.
- D. The Loan Account is a subclass from the Account class that has principal amount, interest rate, loan duration in months as instance variables. The loan balance is debited by monthly interest each month based on the interest rate and the loan balance.
- E. Allow user to create an account with some basic information like username, password, first and last names, address, etc.
- F. Each customer can have any number of accounts of any type.
 - 1. Maintain record of all account holders. Transaction history should be stored.
- G. The banking system provide the customer with an interface to access all banking operations described above and review reports about transactions and current balance.
- H. A banking administrator can print a report about all customers and their current balances.

Distinguishing features of project / list of object-oriented features implemented

Distinguishing features:

- In Saving Account, the interest adds per 1 month Automatically.
- We have used xxxx.csv files instead of using xxxx.txt files as it automatically separates the data with comma-separated values.
- We have loaded every account into one file, every transaction history into one file for the admin report, and the report for individuals can be taken per head.
- Low storage is taken and also use of time module.
- If a user creates a new account, a new account number would assign automatically.

OOP features implemented:

- Multilevel Inheritance
- Method overriding
- Exception Handling
- Composition

The flow of the project (Output based)

Main Interface

```
def Main():
    while True:
        print("#####")
        print("##### WELCOME TO ONLINE BANKING SYSTEM #####")
        print("#####")
        print("\n 01)Create An Account\n 02>Login To My Account\n 03)Bank Administrator\n 04)Exit ")
        try:
            n = int(input("\nPlease Select An Option: "))
            if n == 1:
                Create_Account()
            elif n == 2:
                Login_Account()
            elif n == 3:
                Bank_Admin()
            elif n == 4:
                break
        except:
            print("Please Enter Valid Input!")
```

Main()

1. Creating an Account

```
def create_account(self):
    self.customerID = Customer.count + 1
    Customer.count += 1
    while True:
        self.account_type = int(input(" 01)Savings Account\n 02)Checking Account\n 03)Loan Account\n Select The Type of Account: "))
        if self.account_type == 1:
            self.account_type = "Savings Account"
        elif self.account_type == 2:
            self.account_type = "Checking Account"
        elif self.account_type == 3:
            self.account_type = "Loan Account"
        else:
            print("Please Enter Correct Input!")
            continue
        self.first_name = input("Please Enter Your First Name (Must contain atleast 3 characters): ")
        self.last_name = input("Please Enter Your Last Name: ")
        self.address = input("Please Enter Your Address: ")
        self.username = input("Please Enter Your User Name (Must contain atleast 3 characters): ")
        self.password = input("Please Enter Your Password (Must contain atleast 3 characters): ")
        if len(self.first_name) < 3 or len(self.username) < 3 or len(self.password) < 3:
            print("Please Enter Valid Details!")
            continue
        break

def set_account_no(self, account_no=None):
    self.account_no = "601-501-00"+str(self.customerID)
    print("Your Account is Successfully Created")
    print("Your Account Number is:", self.account_no)

def Create_Account():
    print("Welcome To Account Creation Page")
    print("Please Provide Following Informations To Creat Account:\n")
    customer = Customer()
    customer.create_account()
    customer.customer_lists()
    customer.set_account_no()
    customer.save_details()
```

2. Login Account

I. Savings & Checking Account

```
while True:
    obj_account = Account(self.mybalance)
    obj_account.account_lists()
    My_Account.setbalance(self)
    print("Current Balance:", self.mybalance)
    print("\n 01)Deposit Money\n 02)Withdraw Money\n 03)Transfer Money\n 04)Edit Account\n 05)Print Report\n 06)Logout ")
    n1 = int(input("\nPlease Select An Option: "))
```

- Adding Monthly Interest to Savings Account

```

if self.actype == "Savings Account" and self.mybalance != 0:
    f13 = open("date.txt", "r")
    a = f13.read()
    b = a.split('/')
    if (int(z[0]) - int(b[0])) == 1 or (int(z[0]) - int(b[0])) == -11:
        savingaccount_1 = SavingAccount(self.ac_no)
        savingaccount_1.add_interest(self.mybalance, self.ac_no)
        obj_account = Account(self.mybalance)
        My_Account.setbalance(self)
    f13.close()

    f14 = open("date.txt", "w")
    f14.write(f"{y}")
    f14.close()

```

i. Withdraw Money

```

class CheckingAccount(Account):
    def __init__(self, balance, ac_no, credit_limit=5000):
        self.balance = balance
        self.ac_no = ac_no
        self.credit_limit = credit_limit

    def c_withdraw(self):
        self.cw_amount = float(input("Enter The Amount To withdraw: "))
        if self.balance + self.credit_limit >= self.cw_amount:
            if self.balance >= self.cw_amount:
                self.balance -= self.cw_amount
            else:
                self.balance -= self.cw_amount
                self.balance -= 500 #overdraft fee
        account3 = Account()
        account3.account_lists()
        for i in account3.ac_lst:
            if i[0] == self.ac_no:
                i[1] = self.balance
                if self.balance >= self.cw_amount:
                    i.append(f"Withdraw {self.cw_amount}")
                else:
                    i.append(f"Withdraw {self.cw_amount}+overdraft fee 500")
        f11 = open("Accounts.csv", "w")
        for l in account3.ac_lst:
            for m in range(len(l)):
                if m == (len(l)-1):
                    f11.write(f"{l[m]}")

elif n1 == 2:
    if self.actype == "Checking Account":
        checkingaccount_1 = CheckingAccount(self.mybalance, self.ac_no)
        checkingaccount_1.c_withdraw()
        My_Account.setbalance(self)

    else:
        obj_account.withdraw(self.ac_no)
        My_Account.setbalance(self)

```


ii. Deposit Money

```
n1 = int(input("\nPlease Select An Option: "))
if n1 == 1:
    obj_account.deposit(self.ac_no)
    My_Account.setbalance(self)

def deposit(self,ac_no):
    self.ac_no = ac_no
    self.amount = float(input("Enter The Amount To Deposit: "))
    self.balance += self.amount
    for j in self.ac_lst:
        if j[0] == self.ac_no:
            j[1] = self.balance
            j.append(f"Deposit {self.amount}")
    f7 = open("Accounts.csv","w")
    for k in self.ac_lst:
        for l in range(len(k)):
            if l == (len(k)-1):
                f7.write(f"{k[l]}")
            else:
                f7.write(f"{k[l]},")
        f7.write("\n")
    f7.close()
    Account.account_lists(self)
```

iii. Transfer to an Account

```
elif n1 == 3:
    obj_account.transfer(self.ac_no)
    My_Account.setbalance(self)

def transfer(self,ac_no):
    self.ac_no = ac_no
    self.sender = input("Enter The Sender Account No: ")
    if self.sender in self.lac:
        self.s_amount = float(input("Enter The Amount To Send: "))
        if self.balance >= self.s_amount:
            self.balance -= self.s_amount
            for n in self.ac_lst:
                if n[0] == self.ac_no:
                    n[1] = self.balance
                    n.append(f"Transferred {self.s_amount} To {self.sender}")
                if n[0] == self.sender:
                    n[1] = float(n[1])+self.s_amount
                    n.append(f"Transferred {self.s_amount} From {self.ac_no}")
                    print(f"Successfully Transferred {self.s_amount} To {n[0]}")
            f10 = open("Accounts.csv","w")
            for o in self.ac_lst:
                for p in range(len(o)):
                    if p == (len(o)-1):
                        f10.write(f"{o[p]}")
                    else:
                        f10.write(f"{o[p]},")
                f10.write("\n")
            f10.close()
            Account.account_lists(self)

        else:
            print("Insufficient balance")
```

iv. Edit Account

```
def edit_account(self,account_no,balance):
    self.account_no = account_no
    self.balance = balance
    while True:
        self.first_name = input("Please Enter Your First Name (Must contain atleast 3 characters): ")
        self.last_name = input("Please Enter Your Last Name: ")
        self.username = input("Please Enter Your User Name (Must contain atleast 3 characters): ")
        self.password = input("Please Enter Your Password (Must contain atleast 3 characters): ")
        self.address = input("Please Enter Your Address: ")
        if len(self.first_name) < 3 or len(self.username) < 3 or len(self.password) < 3:
            print("Please Enter Valid Details!")
            continue
        break

    for j in self.cst_lst:
        if j[4] == self.account_no:
            j[1] = self.first_name
            j[2] = self.last_name
            j[5] = self.username
            j[6] = self.password
            j[7] = self.address
    f11 = open("Customer_Details.csv","w")
    for k in self.cst_lst:
        f11.write(f"{k[0]},{k[1]},{k[2]},{k[3]},{k[4]},{k[5]},{k[6]},{k[7]}\n")
    f11.close()
    print("Your Account Successfully Updated")
```

v. Print Report

```
elif n1 == 5:
    obj_account.account_lists()
    for i in obj_account.ac_lst:
        if i[0] == self.ac_no:
            for j in i:
                if j == i[0]:
                    print()
                elif j == i[1]:
                    print("Balance:",j)
                    print("Transections:")
                else:
                    print(j)
```

II. Loan

```
while True:
    print("Loan Balance:",self.mybalance)
    print("\n 01)Get Loan\n 02)Pay Loan\n 03)Edit Account\n 04)Print Report\n 05)Logout ")
    n2 = int(input("\nPlease Select An Option: "))
```

- Adding Monthly Interest to Loan Account

```

self.actype == "Loan Account"
loanAccount_1 = LoanAccount(self.mybalance)
f17 = open("date.txt", "r")
a = f17.read()
b = a.split('/')
if self.mybalance != 0:
    if (int(z[0]) - int(b[0])) == 1 or (int(z[0]) - int(b[0])) == -11:
        loanAccount_1.monthly_interest(self.ac_no)
        My_Account.setbalance(self)
f17.close()

f18 = open("date.txt", "w")
f18.write(f"{y}")
f18.close()

```

i. Get Loan

```

if n2 == 1:
    loanAccount_1.get_loan(self.ac_no)
    My_Account.setbalance(self)

def get_loan(self, ac_no):
    self.ac_no = ac_no
    self.p_amount = float(input("Enter Amount:"))
    self.principal += self.p_amount
    account2 = Account()
    account2.account_lists()
    for j in account2.ac_lst:
        if j[0] == self.ac_no:
            j[1] = self.principal
            j.append(f"Loan {self.p_amount}")
    f15 = open("Accounts.csv", "w")
    for k in account2.ac_lst:
        for l in range(len(k)):
            if l == (len(k)-1):
                f15.write(f"{k[l]}")
            else:
                f15.write(f"{k[l]},")
        f15.write("\n")
    f15.close()
    Account.account_lists(self)

```

I. Pay Loan

```
elif n2 == 2:
    loanAccount_1.pay_loan(self.ac_no)
    My_Account.setbalance(self)

def pay_loan(self,ac_no):
    self.ac_no = ac_no
    self.pay_amount = float(input("Enter Amount:"))
    self.principal -= self.pay_amount
    account3 = Account()
    account3.account_lists()
    for j in account3.ac_lst:
        if j[0] == self.ac_no:
            j[1] = self.principal
            j.append(f"Payed {self.pay_amount}")
    f16 = open("Accounts.csv","w")
    for k in account3.ac_lst:
        for l in range(len(k)):
            if l == (len(k)-1):
                f16.write(f"{k[l]}")
            else:
                f16.write(f"{k[l]},")
        f16.write("\n")
    f16.close()
    Account.account_lists(self)
```

II. Edit Account

```
elif n2 == 3:
    customer2 = Customer()
    customer2.customer_lists()
    customer2.edit_account(self.ac_no,self.mybalance)
```

III. Print Report

```
elif n2 == 4:
    account4 = Account()
    account4.account_lists()
    for i in account4.ac_lst:
        if i[0] == self.ac_no:
            for j in i:
                if j == i[0]:
                    print()
                elif j == i[1]:
                    print("Balance:",j)
                    print("Transactions:")
                else:
                    print(j)
```

3. Administration Login

I. Customer Details

```
if n2 == 1:
    customer2 = Customer()
    customer2.customer_lists()
    for i in customer2.cst_lst:
        strg+=temp.format(fname=i[1],lname=i[2],atype=i[3],anum=i[4],uname=i[5],pswd=i[6],adrs=i[7])
    print(strg)
```

II. Transaction Report

```
elif n2 == 2:
    account1 = Account()
    account1.account_lists()
    print("Account Number | Balance | Transections")
    for j in account1.ac_lst:
        for k in range(len(j)):
            if k == 0:
                print(f"{j[k]} | ",end="")
            elif k == (len(j)-1):
                print(f"{j[k]}\n")
            elif k == 1:
                print(f"{j[k]} | ",end="")
            else:
                print(j[k], " | ",end=" ")
```

Outputs

Main Interface

```
#####  
##### WELCOME TO ONLINE BANKING SYSTEM #####  
#####
```

```
01)Create An Account  
02>Login To My Account  
03)Bank Administrator  
04)Exit
```

Please Select An Option: |

1. Creating an Account:

Please Select An Option: 1

Welcome To Account Creation Page

Please Provide Following Informations To Create Account:

```
01)Savings Account  
02)Checking Account  
03)Loan Account
```

Select The Type of Account Want To Create:

```
01)Savings Account  
02)Checking Account  
03)Loan Account
```

Select The Type of Account Want To Create:3

Please Enter Your First Name (Must contain atleast 3 characters): Fathima

Please Enter Your Last Name: Hafsa

Please Enter Your Address: Galle

Please Enter Your User Name (Must contain atleast 3 characters): hfs008

Please Enter Your Password (Must contain atleast 3 characters): h0f0s8

Your Account is Successfully Created

Your Account Number is: 601-501-008

2. Login Account:

I. Checking Account:

```
01)Create An Account
02)Login To My Accont
03)Bank Administrator
04)Exit
```

```
Please Select An Option: 2
##### Welcome To Login Page #####
Please Enter Your User Name: asf002
Please Enter Your Password: a0s0f2
##### Hello Ashfi Uzama Welcome To Your Account #####
Successfully Logged in
Login Time: Thu Jul 13 12:40:09 2023
Account Number: 601-501-002
Account Type: Checking Account
Current Balance: 0.0
```

```
01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout
```

```
Please Select An Option: |
```

i. Withdraw Money

```
01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout
```

```
Please Select An Option: 2
Enter The Amount To withdraw: 3000
Current Balance: -1500.0
```

ii. Deposit Money

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 1
Enter The Amount To Deposit: 2000
Current Balance: 2000.0

iii. Transfer to an Account

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 3
Enter The Sender Account No: 601-501-005
Enter The Amount To Send: 1000
Successfully Transferred 1000.0 To 601-501-005
Current Balance: 2000.0

iv. Edit Account

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 4
Please Enter Your First Name (Must contain atleast 3 characters): Mohamed
Please Enter Your Last Name: Ifthikar
Please Enter Your User Name (Must contain atleast 3 characters): ift001
Please Enter Your Password (Must contain atleast 3 characters): i0f0t1
Please Enter Your Address: Kandy
Your Account Successfully Updated

v. Print Report

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 5

Balance: -1500.0
Transections:
Deposit 2000.0
Withdraw 3000.0+overdraft fee 500

II. Saving Account:

WELCOME TO ONLINE BANKING SYSTEM #####
#####

01)Create An Account
02>Login To My Accont
03)Bank Administrator
04)Exit

Please Select An Option: 2

Welcome To Login Page

Please Enter Your User Name: ift001

Please Enter Your Password: i0f0t1

Hello Mohamed Ifthikar Welcome To Your Account

Login Time: Thu Jul 13 12:20:29 2023

Account Number: 601-501-001

Account Type: Savings Account

Current Balance: 0.0

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: |

i. Withdraw Money

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 2
Enter The Amount To withdraw: 2000
Current Balance: 3000.0

ii. Deposit Money

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 1
Enter The Amount To Deposit: 5000
Current Balance: 5000.0

iii. Transfer to an Account

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 3
Enter The Sender Account No: 601-501-005
Enter The Amount To Send: 1000
Successfully Transferred 1000.0 To 601-501-005
Current Balance: 2000.0

iv. Edit Account

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 4
Please Enter Your First Name (Must contain atleast 3 characters): Mohamed
Please Enter Your Last Name: Ifthikar
Please Enter Your User Name (Must contain atleast 3 characters): ift001
Please Enter Your Password (Must contain atleast 3 characters): i0f0t1
Please Enter Your Address: Kandy
Your Account Successfully Updated

v. Print Report

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 5

Balance: 2000.0
Transections:
Deposit 5000.0
Withdraw 2000.0
Transferred 1000.0 To 601-501-005

III. Loan:

01)Create An Account
02>Login To My Account
03)Bank Administrator
04)Exit

Please Select An Option: 2

Welcome To Login Page

Please Enter Your User Name: muh003

Please Enter Your Password: m0u0h3

Hello Muhammed Welcome To Your Account

Successfully Logged in

Login Time: Thu Jul 13 12:46:08 2023

Account Number: 601-501-003

Account Type: Loan Account

Loan Balance: 0.0

01)Get Loan
02)Pay Loan
03)Edit Account
04)Print Report
05)Logout

Please Select An Option: |

i. Get Loan

01)Get Loan
02)Pay Loan
03)Edit Account
04)Print Report
05)Logout

Please Select An Option: 1

Enter Amount:10000

Loan Balance: 10000.0

ii. Pay Loan

01)Get Loan
02)Pay Loan
03)Edit Account
04)Print Report
05)Logout

Please Select An Option: 2
Enter Amount:3500
Loan Balance: 6500.0

iii. Edit Account

01)Deposit Money
02)Withdraw Money
03)Transfer Money
04)Edit Account
05)Print Report
06)Logout

Please Select An Option: 4
Please Enter Your First Name (Must contain atleast 3 characters): Mohamed
Please Enter Your Last Name: Ifthikar
Please Enter Your User Name (Must contain atleast 3 characters): ift001
Please Enter Your Password (Must contain atleast 3 characters): i0f0t1
Please Enter Your Address: Kandy
Your Account Successfully Updated

iv. Print Report

01)Get Loan
02)Pay Loan
03)Edit Account
04)Print Report
05)Logout

Please Select An Option: 4

Balance: 6500.0
Transections:
Loan 10000.0
Payed 3500.0

3. Administrator login:

01)Create An Account
02>Login To My Account
03)Bank Administrator
04)Exit

Please Select An Option: 3

Welcome To Admin Dashboard

Please Enter The Following Details To Print Customer Details

Please Enter Your User Name: admin

Please Enter Your Password: psadmin

01)Print Customer Details
02)Print Customer Transaction Report
03)Exit

Please Select An Option: |

i. Customer Details

01)Print Customer Details
02)Print Customer Transaction Report
03)Exit

Please Select An Option: 1

First Name	Last Name	Account Type	Account Number	User Name	Password	Address
Mohamed	Ifthikar	Savings Account	601-501-001	ift001	i0f0t1	Kandy
Ashfi	Uzama	Checking Account	601-501-002	asf002	a0s0f2	Islamabad
Muhammed		Loan Account	601-501-003	muh003	m0u0h3	Colombo
Fathima	Amna	Savings Account	601-501-004	amn004	a0m0n4	Kandy
Zaid	Ahamed	Savings Account	601-501-005	zaid154	15242635	Kandy
Mohamed	Ishaq	Loan Account	601-501-006	ishaq352	321456	Hambanthotta
Umair	Lafir	Checking Account	601-501-007	umr007	u0m0r7	Kandy
Fathima	Hafsa	Loan Account	601-501-008	hfs008	h0f0s8	Galle

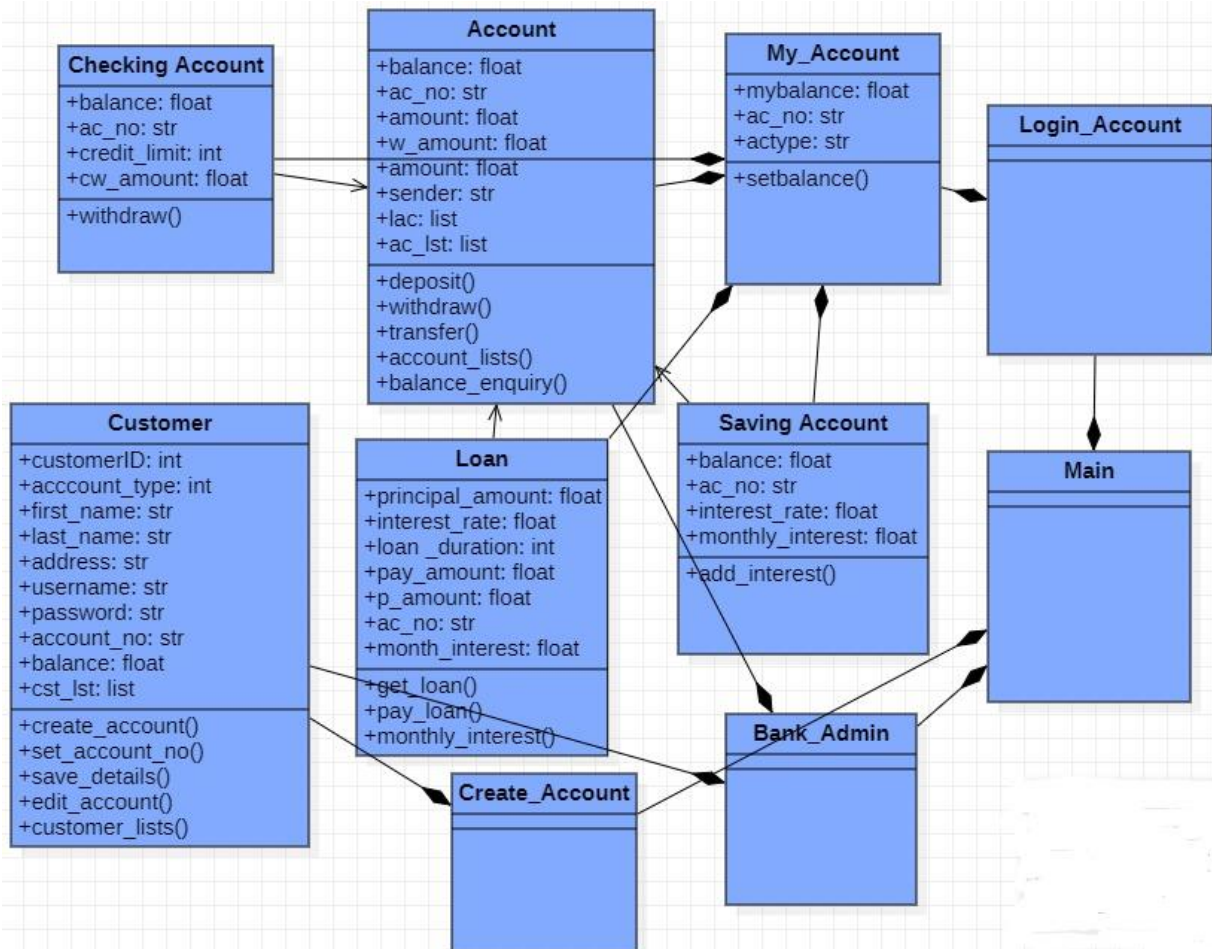
ii. Transaction report

01)Print Customer Details
02)Print Customer Transaction Report
03)Exit

Please Select An Option: 2

Account Number	Balance	Transactions
601-501-001	2008.3333333333333	Deposit 5000.0 Withdraw 2000.0 Transferred 1000.0 To 601-501-005 Interest 8.333333333333334
601-501-002	-1500.0	Deposit 2000.0 Withdraw 3000.0+overdraft fee 500
601-501-003	6527.083333333333	Loan 10000.0 Payed 3500.0 Interest 27.083333333333332
601-501-004	0	
601-501-005	1000.0	Transferred 1000.0 From 601-501-001
601-501-006	0	
601-501-008	0	

Class Diagram



The most challenging part while working on Project

- It was very hard to search and edit an account of the file because when writing the file, the accounts are being erased due 'w' option, so we overcome it by using the CSV file.
- At first it was hard to split the project among the members as the individual cannot work on their own simultaneously because of inheritance and other OOP concepts so once a member finished, we upload it to the other and he/she works from that instance.
- In Saving Account, we couldn't update the balance with interest if the account the balance lasted for a month, so we had use of time module and auto updating of interest was made.
- In real time if we deposit and started to withdraw, the code raised error saying no required balance, the file wasn't updated on run time and then we made sure how to correct this by closing each and every time after being updated.
- Some members had issue with OOP concepts at the start, but as soon as the project reached the mid likely every member had the chance of studying the concept clearly.

Any new thing learned in Python while working on Project

- Before learning the concepts of OOP, we had used procedural programming and made different .py files for every classes and then we imported each and every .py files to an interface and we had to run. But after these we shrinked the 1500 lines of code to just 700 lines and also, we implemented every code on the same .py file and we just used a simple function for interface to call.
- We learned the syntax of Multiple Inheritance, Multilevel Inheritance, Association, Abstract classes and Exception Handling.
- As use of Exception Handling, we could identify the errors caused and it was very easy for us to address the problem and it save

Future expansions

- Use GUI to improve interface as;
 - i. GUI for Customers to open a new account
 - ii. GUI for Customers to view transactions and balance for all their accounts
 - iii. GUI for an administrator to view all customer balances
- Make a web interface for the application
- Extra classes to increase functionality of the system can be done as well.

Some expansions made and couldn't finish using tkinter :

