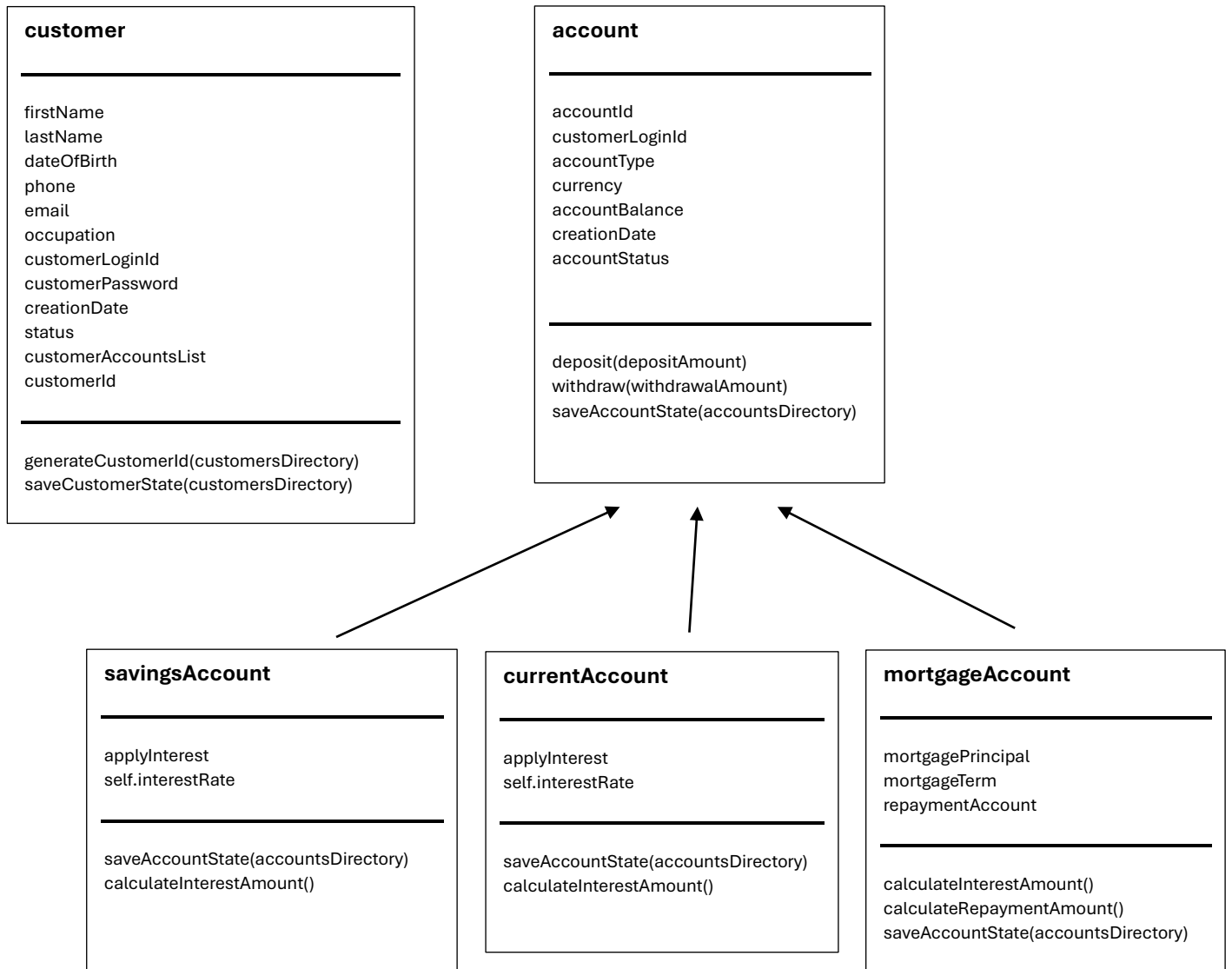


AC51002 – Software Development

Course Work 2

Name: Akachukwu Adiele

CLASS DESIGN



DEFINITIONS:

1. 'customer' Class

This class is used to represent a client of the bank. An instance of this class would have the below attributes and functions:

Attributes:

- firstName – A variable to hold the first name.
- lastName – A variable to hold the last name.
- dateOfBirth – A variable to hold the date of birth.
- phone – A variable to hold the phone number.
- email – A variable to hold the email address.
- occupation – A variable to hold the occupation.
- customerLoginId – A variable to hold the login id for the customer.
- customerPassword – A variable to hold the login password for the customer.
- creationDate – A variable to hold the date the customer record was created.
- status – A variable to hold the status of the customer (e.g. Created).
- customerAccountsList – A list to hold the account ids owned by the customer.
- customerId – A variable to hold the generated id for the customer record.

Functions:

- generateCustomerId(customersDirectory) – A method to implement a logic to generate the unique customer id.
- saveCustomerState(customersDirectory) – A method to update the customer text file for the instance of customer in question.

2. 'account' Class

This class is used to represent a client's account with the bank. An instance of this class would have the below attributes and functions:

Attributes:

- accountId – A variable to hold the unique account id.
- customerLoginId – A variable to hold the login id for the customer that owns this account.
- accountType – A variable to hold the type of account (savings, current, mortgage accounts).
- currency – A variable to hold the currency for this account.
- accountBalance – A variable to hold the account's balance.
- creationDate – A variable to hold the date the account was created.
- accountStatus – A variable to hold the status of the account.

Functions:

- deposit(depositAmount) – A method to trigger a deposit into the account
- withdraw(withdrawalAmount) – A method to trigger a withdrawal from the account
- saveAccountState(accountsDirectory) – A method to update the account text file for the instance of account in question.

The 'account' class serves as a super class for the 'savingsAccount', 'currentAccount' and 'mortgageAccount'

3. 'savingsAccount' Class

This class is used to represent a client's savings account with the bank. It is created as a subclass of the 'account' superclass.

An instance of this class would have the below attributes and functions:

Attributes:

- applyInterest – A variable to hold the decision marker to determine if interest is applied or not
- self.interestRate – A variable to hold the interest rate for savings account
- *** All other attributes of the parent class (account) are inherited.

Functions:

- saveAccountState(accountsDirectory) – A method to update the account text file for the instance of the savings account in question.
- calculateInterestAmount() – A method to calculate the interest for the current month.
- *** All other functions of the parent class (account) are inherited.

4. 'currentAccount' Class

This class is used to represent a client's current account with the bank. It is created as a subclass of the 'account' superclass.

An instance of this class would have the below attributes and functions:

Attributes:

- applyInterest – A variable to hold the decision marker to determine if interest is applied or not
- self.interestRate – A variable to hold the interest rate for current account
- *** All other attributes of the parent class (account) are inherited.

Functions:

- saveAccountState(accountsDirectory) – A method to update the account text file for the instance of the current account in question.
- calculateInterestAmount() – A method to calculate the interest for the current month.
- *** All other functions of the parent class (account) are inherited.

5. 'mortgageAccount' Class

This class is used to represent a client's mortgage account with the bank. It is created as a subclass of the 'account' superclass.

An instance of this class would have the below attributes and functions:

Attributes:

- mortgagePrincipal – A variable to hold the mortgage principal amount.
- mortgageTerm – A variable to hold the mortgage term
- repaymentAccount – A variable to hold the mortgage repayment account.
- *** All other attributes of the parent class (account) are inherited.

Functions:

- calculateInterestAmount() – A method to calculate the mortgage interest amount for the full tenure.
- calculateRepaymentAmount() – A method to calculate the mortgage repayment amount.
- saveAccountState(accountsDirectory) – A method to update the account text file for the instance of the mortgage account in question.
- *** All other functions of the parent class (account) are inherited.

TEST REPORT

❖ Test Case 1 – Registering a new customer

The 'Bank Maximus' is launched and the functionality to 'register a new customer' is tested.

```
-----  
  
Welcome to Bank Maximus  
  
1 - Login existing customer  
2 - Register new customer  
3 - Switch to Admin Menu  
# - Exit Application  
  
>>> 2
```

Select option '2'

```
-----  
  
To register as a new customer, follow the below prompts and provide accurate details:  
  
Enter preferred log in ID: aka  
Enter first name: Akachukwu  
Enter last name: Adiele  
Enter Date of Birth - Day: 24  
Enter Date of Birth - Month: 11  
Enter Date of Birth - Day: 1995  
Enter phone: 07123454321  
Enter email: aka@mail.com  
Enter occupation: Student  
Enter preferred log in Password: 
```

The required details are filled and submitted

```
Customer registered successfully  
Proceed to login  
  
-----
```

Customer is created successfully

Customer data is then saved in a text file, with the file name as the customer login id.

```
aka.txt × +  
File Edit View  
  
akachukwu, adiele, 24/11/1995, 07123454321, aka@mail.com, student, aka, 123, 27/11/2024, 21300001 | 21300002, 100001
```

❖ Test Case 2 – Customer logging in

```
-----  
Welcome to Bank Maximus  
  
1 - Login existing customer  
2 - Register new customer  
3 - Switch to Admin Menu  
# - Exit Application  
  
>>> 1
```

Option 1 is selected

```
-----  
Enter your Customer Login ID: aka  
Enter your Customer password: 1
```

Required input for login details are entered.

```
-----  
Hello Akachukwu,  
-----  
  
1 - Accounts list  
2 - Open a new account  
3 - View state of all accounts  
# - Return to Home menu  
  
>>> 
```

Login successful

❖ Test Case 3 – Opening a new account

```
-----  
Hello Akachukwu,  
-----  
  
1 - Accounts list  
2 - Open a new account  
3 - View state of all accounts  
# - Return to Home menu  
  
>>> 2
```

Select option '2' on customer menu

```
-----  
  
To open a new account, follow the below prompts and provide accurate details:  
  
Enter code to select account type:  
1 - Savings account  
2 - Current account  
3 - Mortgage account  
>>> 1
```

Select account type

```
Would you like apply interest? (Y/N): Y
```

choose if interest should be applied or not

```
Account created successfully
```

Account created succesfully

Account data is then saved in a text file, with the file name as the customer login id.

```
21300001.txt  
File Edit View  
  
21300001,aka,savings account,GBP,150.0,27/11/2024,active,Y,4,,,,,
```

❖ Test Case 4 – View account state for one account

```
-----  
Hello Akachukwu,  
-----  
  
1 - Accounts list  
2 - Open a new account  
3 - View state of all accounts  
# - Return to Home menu  
  
>>> 1
```

Select option '1' on customer menu

```
1 - 21300001  
  
Enter corresponding code for your account (or '#' to return):  
>>> 
```

Input the corresponding number code for account

```
-----  
1 - View account balance  
2 - Make a deposit  
3 - Make a withdrawal  
4 - View account state  
# - Return to customer menu  
  
>>> 4
```

Select option '4' on account menu

```
-----  
Account Id: 21300001  
Customer Login Id: aka  
Account Type: Savings Account  
Account Balance: £0.0  
Apply Interest?: Y  
Current Month's interest: £0.0  
Account Created On: 27/11/2024  
-----
```

State of the account is displayed

❖ Test Case 5 – View account state for one account

```
-----  
Hello Akachukwu,  
-----  
  
1 - Accounts list  
2 - Open a new account  
3 - View state of all accounts  
# - Return to Home menu  
  
>>> 3
```

Select option '3' on customer menu

```
-----  
Account Id: 21300001  
Customer Login Id: aka  
Account Type: Savings Account  
Account Balance: £0.0  
Apply Interest?: Y  
Current Month's interest: £0.0  
Account Created On: 27/11/2024  
-----  
  
-----  
Account Id: 21300002  
Customer Login Id: aka  
Account Type: Current Account  
Account Balance: £0.0  
Overdraft allowed: £100  
Apply Interest?: Y  
Current Month's interest: £0.0  
Account Created On: 27/11/2024  
-----
```

State of all the accounts for the customer is displayed

❖ Test Case 6 – Make deposit

```
-----  
Hello Akachukwu,  
-----  
  
1 - Accounts list  
2 - Open a new account  
3 - View state of all accounts  
# - Return to Home menu  
  
>>> 1
```

Select option '3' on customer menu

```
1 - 21300001  
2 - 21300002  
  
Enter corresponding code for your account (or '#' to return):  
>>> 1
```

Select an account

```
-----  
1 - View account balance  
2 - Make a deposit  
3 - Make a withdrawal  
4 - View account state  
# - Return to customer menu  
  
>>> 2
```

Select option '2' on account menu

```
-----  
  
Enter amount to deposit (or '#' to return):  
>>> 200  
  
Deposit of £200.0 was successful  
New account balance is 200.0
```

Deposit is completed

❖ Test Case 7 – Make withdrawal

```
-----  
Hello Akachukwu,  
-----  
  
1 - Accounts list  
2 - Open a new account  
3 - View state of all accounts  
# - Return to Home menu  
  
>>> 1
```

Select option '3' on customer menu

```
1 - 21300001  
2 - 21300002  
  
Enter corresponding code for your account (or '#' to return):  
>>> 1
```

Select an account

```
-----  
1 - View account balance  
2 - Make a deposit  
3 - Make a withdrawal  
4 - View account state  
# - Return to customer menu  
  
>>> 3
```

Select option '3' on account menu

```
-----  
  
Enter amount to withdraw (or '#' to return):  
>>> 50  
  
Withdrawal of £50.0 was successful  
New account balance is 150.0
```

Withdrawal is completed

❖ Test Case 8 – View Account Balance

```
-----  
Hello Akachukwu,  
-----  
  
1 - Accounts list  
2 - Open a new account  
3 - View state of all accounts  
# - Return to Home menu  
  
>>> 1
```

Select option '3' on customer menu

```
1 - 21300001  
2 - 21300002  
  
Enter corresponding code for your account (or '#' to return):  
>>> 1
```

Select an account

```
-----  
1 - View account balance  
2 - Make a deposit  
3 - Make a withdrawal  
4 - View account state  
# - Return to customer menu  
  
>>> 1  
  
-----
```

Select option '3' on account menu

```
-----  
Your account balance is £150.0  
-----
```

Account balance is displayed

❖ Test Case 9 – Using the admin menu to customer data

```
-----
```

```
Welcome to Bank Maximus  
  
1 - Login existing customer  
2 - Register new customer  
3 - Switch to Admin Menu  
# - Exit Application
```

```
>>> 3
```

Select option '3' to switch to admin menu

```
-----
```

```
Enter Admin ID: admin  
Enter Admin password: admin123
```

Input log in details

```
-----
```

```
1 - View Customers  
2 - View Accounts  
# - Return to Home menu
```

```
>>> 1
```

Select option '1' to view customers

```
Customers:
```

```
1 - aka
```

```
Enter corresponding code for your customer (or '#' to return):  
>>> 1
```

Select customer to view

```
-----
```

```
Name: Akachukwu Adiele  
Date of Birth: 24/11/1995  
Phone: 07123454321  
Email: aka@mail.com  
Occupation: Student  
Login Id: aka  
Joined: 27/11/2024  
Accounts: ['21300001', '21300002']
```

```
-----
```

Customer data is displayed

❖ Test Case 10 – Using the admin menu to accounts data

```
-----  
Welcome to Bank Maximus
```

```
1 - Login existing customer  
2 - Register new customer  
3 - Switch to Admin Menu  
# - Exit Application
```

```
>>> 3
```

Select option '3' to switch to admin menu

```
-----  
Enter Admin ID: admin  
Enter Admin password: admin123
```

Input log in details

```
-----  
1 - View Customers  
2 - View Accounts  
# - Return to Home menu
```

```
>>> 2
```

Select option '2' to view accounts

```
Accounts:  
1 - 21300001  
2 - 21300002  
  
Enter corresponding code for your account (or '#' to return):  
>>> 2
```

Select account to view

```
-----  
Account Id: 21300002  
Customer Login Id: aka  
Account Type: Current Account  
Account Balance: £0.0  
Overdraft allowed: £100  
Apply Interest?: Y  
Current Month's interest: £0.0  
Account Created On: 27/11/2024  
-----
```

Account data is displayed

REFLECTION

For this assessment on the banking program for 'Bank Maximus' I was able to implement all the mandatory requirements stated in the assessment brief which included the following:

- Defining classes for data used within the application.
- Providing menus to navigate for customers and admin users
- Providing options to open current accounts and savings accounts
- Providing customers the option to add interest to the accounts created.
- Providing features in the program to facilitate saving to and loading data from text files.

In addition, I implemented the optional requirements for providing a sub-menu for the bank admin and functionality for opening a mortgage account where a customer's current account is used as the repayment account.

I also put in informative comments within the program to enable ease in understanding the code and also made use of exceptions to cater for possible errors that may rise from inputs.

With respect to extra features, I did the below:

- Added redundancy functionality to check for admin profile setup and create the default login to allow access on the subsequent login attempt with the default log in
- Included overdraft feature for current accounts, hence withdrawals that exceed the account balance can still proceed if it is within the available overdraft

For usage purpose, here are default the login details for the admin section:

- Username - admin
- Password - admin123