COMPUTER SCIENCES AND SOFTWARE ENGINEERING

AUBURN UNVERSITY

COMP 2710

**Software Construction**

Fall 2014

**Lab 4**

**Distribution Messaging System**

**Due: November 19th, 2014**

**Analysis**

Use Cases:

1. Create a bank account of any type
   1. Savings
   2. Checking
   3. Money Market
   4. CD account
2. Deposit funds into an account
3. Withdraw funds from an account, based on the rules for that account. The withdraw function should return an integer indicating the status (either ok or insufficient funds for the withdrawal to take place).
4. Transfer funds from one account (of any type) to another. A transfer is a withdrawal from one account and a deposit into the other. Since transfer can be done at any time with any type of account, the withdrawal function in the classes must be virtual.
5. Inquiry on the balance of an account of any type
6. Print all the transactions recorded in the bank transaction system
7. Print all the deposits and withdrawals (with account name and types), and the final total Bank balance of all deposits minus the withdrawal and fees.

## Basic Flow: Menu Option (1)

|  |  |
| --- | --- |
| Use Case 1 - Menu Option (1) | |
| Description | User chooses to select menu option  (1) and create an account. They may choose to create any type of account. |
| Assumptions | User successfully creates an account. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. User selects choice 1. 2. Name is collected and stored. |
| Variation(s) | User may choose to create one of the following accounts after name is collected:   1. Savings 2. Checking 3. Money Market 4. CD Account |
| Precondition 1 | There is a user |
| Precondition 2 | User selects menu option (1) |
| Precondition 3 | User Selects Valid choice |
| Post condition 1 | An appropriate account has been created. |

## Alternate Flow: Create Savings

|  |  |
| --- | --- |
| Use Case 1A - Menu Option 1 [Savings] | |
| Description | User chooses to create a savings account. |
| Assumption | A new saving account is successfully created. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. User chooses to create savings account. 2. SavingAccount object is created. 3. Pointer to BankAccount object is successfully stored in allAccounts. |
| Variation(s) | None |
| Precondition 1 | User chooses Savings |
| Post condition 1 | Savings account created |

**Alternate Flow: Create Checking**

|  |  |
| --- | --- |
| Use Case 1B - Menu Option 1 [Checking] | |
| Description | User chooses to create a checking account. |
| Assumption | A new checking account is successfully created. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. User chooses to create checking account. 2. CheckingAccount object is created. 3. Pointer to BankAccount object is successfully stored in allAccounts. |
| Variation(s) | None |
| Precondition 1 | User chooses Checking |
| Post condition 1 | Checking account created |

## Alternate Flow: Create Money Market

|  |  |
| --- | --- |
| Use Case 1C - Menu Option 1 [Money Market] | |
| Description | User chooses to create a money market account. |
| Assumption | A new money market account is successfully created. |
| Actor(s) | User  Aubie Bank |
| Steps | 1. User chooses to create money market account. 2. MoneyMarketAccount object is created. 3. Pointer to BankAccount object is successfully stored in allAccounts. |
| Variation(s) | None |
| Precondition 1 | User chooses Money Market |
| Post condition 1 | Money Market account created |

**Alternate Flow: Create CD**

|  |  |
| --- | --- |
| Use Case 1D - Menu Option 1 [CD] | |
| Description | User chooses to create a CD account. |
| Assumption | A new CD account is successfully created. |
| Actor(s) | User  Aubie Bank |
| Steps | 1. User chooses to create CD account. 2. CDAccount object is created. 3. Pointer to BankAccount object is successfully stored in allAccounts. |
| Variation(s) | None |
| Precondition 1 | User chooses CD |
| Post condition 1 | CD account created |

## Basic Flow: Menu Option (2)

|  |  |
| --- | --- |
| Use Case 2 - Menu Option (2) | |
| Description | User chooses to select menu option (2) and deposit funds into an account. |
| Assumptions | User successfully deposits funds. |
| Actor(s) | User  Aubie Bank |
| Steps | 1. User selects choice two 2. User enters account type 3. User enters deposit amount 4. User’s funds are deposited |
| Variation(s) | None. |
| Precondition 1 | There is a user |
| Precondition 2 | User selects menu option (2) |
| Precondition 3 | Account exists |
| Precondition 3 | User Selects Valid choice |
| Post condition 1 | Funds have been deposited into the appropriate account. |

**Basic Flow: Menu Option (3)**

|  |  |
| --- | --- |
| Use Case 3 - Menu Option (3) | |
| Description | User chooses to select menu option (3) and withdraw funds from an account. |
| Assumptions | User successfully withdraws funds. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. User selects choice three 2. User enters account type 3. User enters withdrawal amount 4. User’s funds are withdrawn |
| Variation(s) | User may choose to withdraw from one of the following accounts   1. Savings 2. Checking 3. Money Market 4. CD Account |
| Precondition 1 | There is a user |
| Precondition 2 | User selects menu option (3) |
| Precondition 3 | User Selects Valid choice |
| Precondition 4 | Account exists |
| Post condition 1 | Funds have been withdrawn from the appropriate account. |

## Alternate Flow: Withdraw Savings

|  |  |
| --- | --- |
| Use Case 3A - Menu Option 3 [Savings] | |
| Description | User chooses to withdrawal from a savings account. |
| Assumption | Funds are successfully withdrawn. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. Checks balance 2. Remove funds if possible    1. No restrictions 3. Notifies user if not |
| Variation(s) | None |
| Precondition 1 | User chooses Savings |
| Precondition 2 | Account exists |
| Post condition 1 | Funds are withdrawn |
| Post condition 2 | User is notified of insufficient funds. |

**Alternate Flow: Withdraw Checking**

|  |  |
| --- | --- |
| Use Case 3B - Menu Option 3 [Checking] | |
| Description | User chooses to withdrawal from a checking account. |
| Assumption | Funds are successfully withdrawn. |
| Actor(s) | User  Aubie Bank |
| Steps | 1. Checks balance 2. Remove funds if possible    1. If balance < $500 add FEE 3. Notifies user if not |
| Variation(s) | None |
| Precondition 1 | User chooses checking |
| Precondition 2 | Account exists |
| Post condition 1 | Funds are withdrawn |
| Post condition 2 | User is notified of insufficient funds. |

## Alternate Flow: Withdraw Money Market

|  |  |
| --- | --- |
| Use Case 3C - Menu Option 3 [Money Market] | |
| Description | User chooses to withdrawal from a money market account. |
| Assumption | Funds are successfully withdrawn. |
| Actor(s) | User  Aubie Bank |
| Steps | 1. Checks balance 2. Remove funds if possible    1. If user has withdrawn more than twice in time frame, charge FEE 3. Notifies user if not |
| Variation(s) | None |
| Precondition 1 | User chooses money market |
| Precondition 2 | Account exists |
| Post condition 1 | Funds are withdrawn |
| Post condition 2 | User is notified of insufficient funds. |

|  |  |
| --- | --- |
| Use Case 3D - Menu Option 3 [CD] | |
| Description | User chooses to withdrawal from a CD account. |
| Assumption | Funds are successfully withdrawn. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. Checks balance 2. Remove funds if possible    1. Charge PENALTY 3. Notifies user if not |
| Variation(s) | None |
| Precondition 1 | User chooses CD |
| Precondition 2 | Account exists |
| Post condition 1 | Funds are withdrawn |
| Post condition 2 | User is notified of insufficient funds. |

**Alternate Flow: Withdraw CD**

## Basic Flow: Menu Option (4)

|  |  |
| --- | --- |
| Use Case 4 - Menu Option (4) | |
| Description | User chooses to select menu option  (4) and transfer funds |
| Assumptions | User successfully transfers funds. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. User selects choice four 2. User enters sender account name 3. User enters sender account type 4. User enters receiver account name 5. User enters receiver account type 6. User’s funds are transferred |
| Variation(s) | None |
| Precondition 1 | There is a user |
| Precondition 2 | Accounts exist |
| Precondition 3 | User Selects Valid choice |
| Post condition 1 | Funds have been withdrawn from the appropriate account. |

**Basic Flow: Menu Option (5)**

|  |  |
| --- | --- |
| Use Case 5 - Menu Option (5) | |
| Description | User chooses to select menu option  (5) and inquiry an account. |
| Assumptions | User successfully inquiries an account. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. Account name is collected 2. Account type is collected 3. Balance is displayed |
| Variation(s) | None |
| Precondition 1 | There is a user |
| Precondition 2 | Account exists |
| Precondition 3 | User Selects valid choice |
| Post condition 1 | Inquiry has been displayed. |

## Basic Flow: Menu Option (6)

|  |  |
| --- | --- |
| Use Case 6 - Menu Option (6) | |
| Description | User chooses to select menu option  (6) and print all transactions |
| Assumptions | All transactions are successfully displayed. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. Print out each transaction in the transaction map |
| Variation(s) | None |
| Precondition 1 | There is a user |
| Precondition 2 | Transactions exists |
| Precondition 3 | User Selects valid choice |
| Post condition 1 | All transactions have been displayed |

**Basic Flow: Menu Option (7)**

|  |  |
| --- | --- |
| Use Case 6 - Menu Option (7) | |
| Description | User chooses to select menu option (7) and display bank balance |
| Assumptions | Bank balance is successfully displayed. |
| Actor(s) | User, Aubie Bank |
| Steps | 1. Find each bank account 2. Add all deposits 3. Subtract all withdrawals 4. Calculate bank balance 5. Display bank balance |
| Variation(s) | None |
| Precondition 1 | There is a user |
| Precondition 2 | Bank accounts exist |
| Precondition 3 | User Selects valid choice |
| Post condition 1 | All bank balances have been calculated |
| Post condition 2 | All bank balances have been displayed |

**Design**

**Class Diagram**



**Data Flow Diagram**

**Testing**

1. **System Welcome:**
   1. Check if the initial system welcome is printed correctly.
2. **User Operation:**
   1. Ensure user can pick a choice.
   2. Ensure choice is valid.
   3. Ensure that all objects are successfully created, and all data properly stored into them.
3. **Quit the System:** 
   1. Check if the program posts a thank you message and then exits.