

# Arizona Investment Bank (AIB) - Case Study

## Purpose

The purpose of this case study is for students to breakdown key concepts within a real-world context.

## Objectives

Students will be able to demonstrate the application of key concepts discussed in this unit. In particular, students will demonstrate:

- a) The identification of use cases and use case diagram from the system description and user scenarios
- b) Object identification and the construction of CRC diagrams from the use case scenarios
- c) From CRC diagrams and object extraction, identify object relationships and draw the initial class diagram
- d) Usage of sequence diagram to further analyze class interactions and identify missing methods and data abstractions in objects
- e) 1st iteration of analysis level class diagram

## 1. Arizona Investment Bank (AIB) Operations and User Scenarios

### General Operation Scenario:

AIB is a premier investment bank in Arizona. This bank provides two types of accounts to its customers: checking accounts and savings accounts. When the bank creates

accounts for customers, customers must provide their first name, last name, and home phone number. At registration, the bank assigns a unique four-digit customer ID (also known as an owner ID). Both savings and checking accounts have an account number and owner ID. Each account stores its account balance. Customers may have multiple checking and savings accounts.

Before performing any transaction, a customer must provide their customer ID for the bank to verify that the customer is a valid customer. Customers can perform following transactions: deposit money into both checking and savings accounts, withdraw money from checking accounts, and transfer money from their savings account to their checking account (money may not be withdrawn directly from a savings account.) At the time of a deposit, the customer must provide the account number and the deposit account.

### **Create Customer Scenario:**

A potential customer visiting a branch of AIB will be greeted by a bank representative and discuss the possible banking options mentioned above. If the visitor agrees to become a customer, the bank representative will create a customer account for them. At this time, the customer will provide their first name, last name, home phone number. The customer account creation process will generate a four-digit customer number assigned to the customer. The customer will then be added to the bank as an "AIB customer."

### **Create Account Scenario:**

Customers can create bank accounts (both savings and checking). When customers create an account, they must provide the customer ID, account type, and the initial deposit. The bank account creation process will generate a 5 digit account number and assign the account. The customer is related to the account through its customer ID.

### **Deposit Scenario:**

Customers can deposit money in both savings and checking accounts. To do so, the customer must provide the account number, customer ID, and the deposit amount. At this point, the bank verifies that the customer holds the bank account. If so, the deposit proceeds and the balance of the bank account will be updated.

### **Withdraw Scenario:**

Unlike deposit, withdrawal can be done only from checking accounts. The customer must provide the account number, customer ID, and the deposit amount. The bank will then verify that the customer holds the bank account and that the account is a checking account. If the withdrawal amount is less than or equal to the balance, the withdrawal will proceed and the balance of the bank account will be updated.

## Transfer Money Scenario:

Customers can transfer money from a savings account to a checking account. For this operation, the customer will be asked to enter the savings account number, checking account number and the transfer amount. Like other operations, the bank will verify if the customer holds both checking and savings accounts and that the savings account has sufficient funds to complete the transfer. If so, the transfer operation will proceed and account balances will be updated.

## 2. Identification Of Use Cases From Operation And User Scenario Descriptions

The use case diagram consists of two main elements. The use case is a main functionality of the system, and the actor is a human user or an internal or external application that interacts with the system functionality (use-case).

### Arizona Investment Bank (AIB) Detail Operation Scenario:

AIB provides two types of accounts to its **customers**: checking accounts and savings accounts. When the **bank creates accounts** for customers, customers must provide their first name, last name, and home phone number. At registration, the bank assigns a unique four-digit customer ID (also known as an owner ID). Both savings and checking accounts have an account number and owner ID. Each account stores its account balance. **Customers may have multiple checking and savings accounts.** Before performing any transaction, a customer must provide their customer ID for the bank to **verify that the customer is a valid customer**. Customers can perform following **transactions**: **deposit** money into both checking and savings accounts, **withdraw** money from checking accounts, and **transfer money** from their savings account to their checking account (money may not be withdrawn directly from a savings account.) At the time of a deposit, the customer must provide the account number and the deposit.

### Actors:

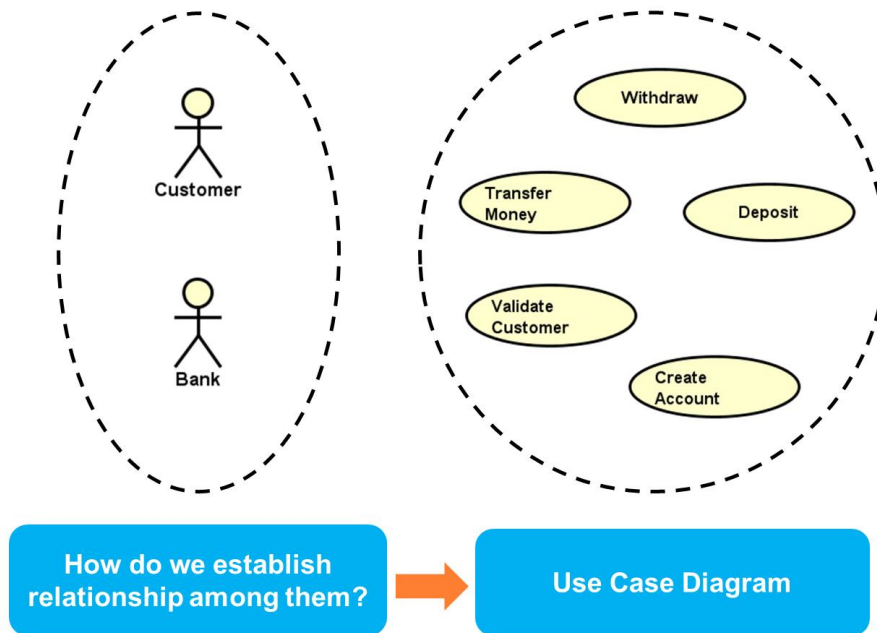
- Customer
- Bank (Bank Representative)

### Use Cases:

- create customer accounts
- deposit money
- withdraw money
- transfer money
- create customer bank accounts
- validate customer
- transactions

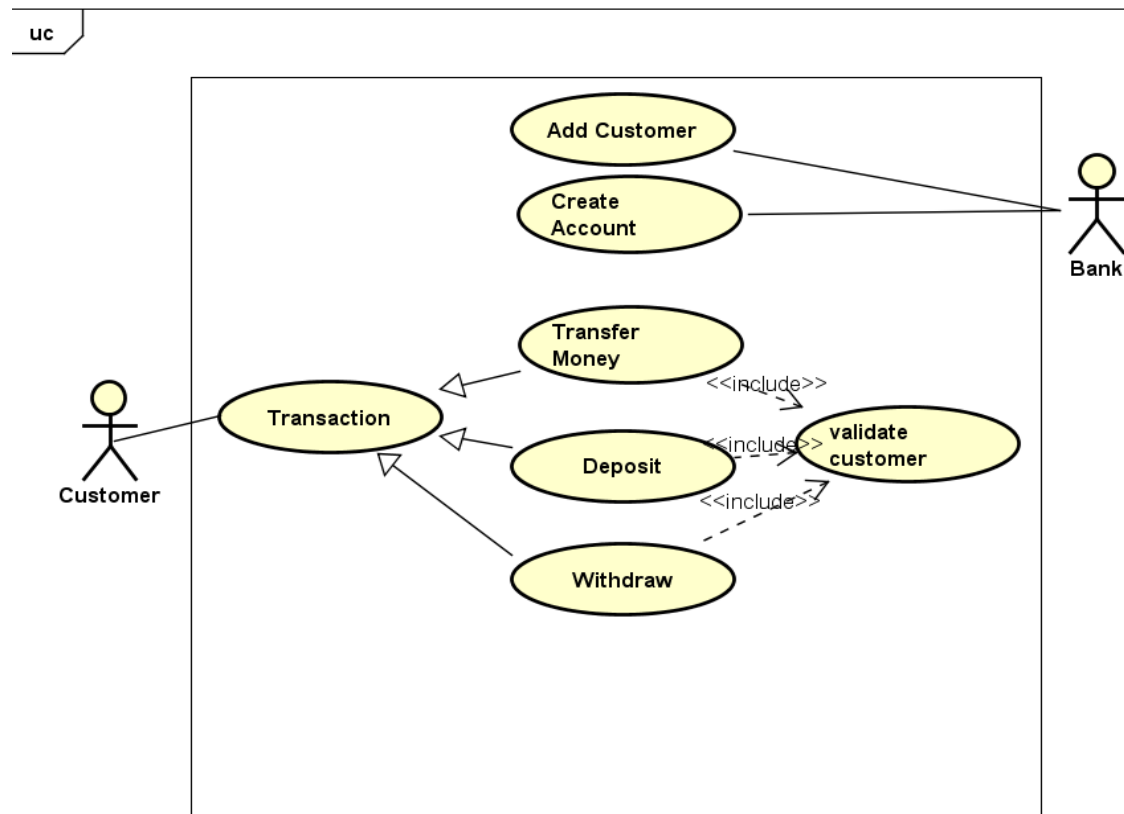
Use Case Diagram:

Initial Step:



We know that “deposit money,” “withdraw money,” and “transfer money” are transactions that customer performs and that with each transaction, the customer must be verified. We also know that “create customer accounts” and “create customer bank accounts” are performed by the bank. These relationships inform the initial use case diagram that follows:

## Use Case Diagram:



The above Use case diagram shows how actors interact with each main use case and how use cases are related to each other. Transfer money, Deposits, and Withdraw are Transactions. So they have “generalization” relationship. Each of these transactions need to validate the customer. So, the validate customer is included in all three transaction types

### 3. Object identification and CRC diagrams

Potential objects usually include Entities, Things, Occurrences (or events), Roles, Organizational Units, Places and Structure in the Use Case. Here, we use the AIB system scenario to identify potential objects as follows. Highlighted segments are potential objects

#### Arizona Investment Bank (AIB):

AIB is a premier investment bank in Arizona. This **bank** provides two types of **accounts** to its **customers**: **checking accounts** and **savings accounts**. When the **bank** creates

accounts for customers, customers must provide their first name, last name, and home phone number. At registration, the bank assigns a unique four-digit customer ID (also known as an owner ID). Both **savings and checking accounts** have an account number and owner ID. Each account stores its account balance. Customers may have multiple checking and savings accounts.

Before performing any **transaction**, a customer must provide their customer ID for the **bank** to verify that the customer is a valid customer. Customers can perform following **transactions**: deposit money into both checking and savings accounts, withdraw money from checking accounts, and transfer money from their savings account to their checking account (money may not be withdrawn directly from a savings account.) At the time of a deposit, the customer must provide the account number and the deposit account.

### Objects and their Categories:

Potential Object	Category
Customer	Role
Bank	Organizational Unit
Accounts, Checking Account, Savings Account	Things
Bank Transaction	Occurrence and Events

Boundary classes are another object category, though not listed here.

In this scenario, we must consider how the customer interacts with the Bank.

We need a Bank user interface (BankUI class).

### CRC Diagrams for each object:

<b>Class: SavingsAccount</b>	<b>Collaborators</b>  CheckingAccount
<b>Responsibilities</b>  Deposit Money Withdraw Money Transfer Money	

<b>Class: CheckingAccount</b>	<b>Collaborators</b>
<b>Responsibilities</b>  Deposit Money Withdraw money	

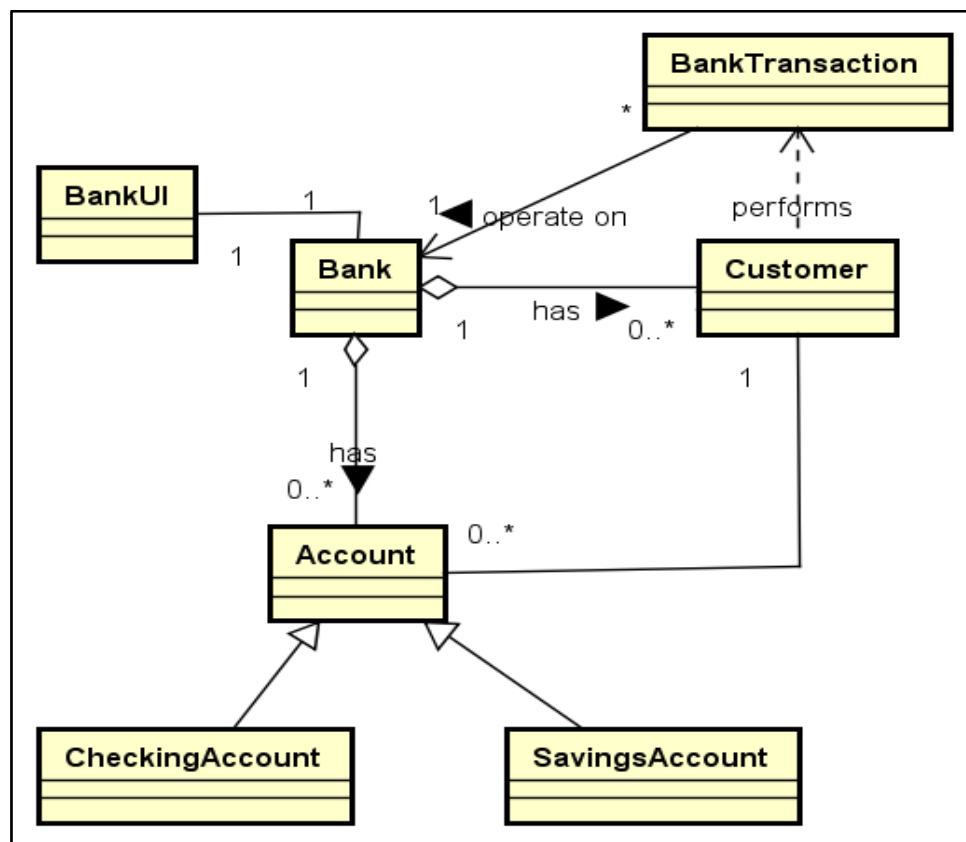
<b>Class: Customer</b>	<b>Collaborators</b>
<b>Responsibilities</b>  Deposit Money in a checking or a savings account  Withdraw Money from a checking account  Transfer money from a savings account to a checking account	

<b>Class: BankTransaction</b>	<b>Collaborators</b>
<b>Responsibilities</b>  Deposit Money Withdraw money Transfer Money Validate customer	

<b>Class: Bank</b>	<b>Collaborators</b>
<b>Responsibilities</b>  Add Account (Checking/Savings) Add Customer	

<b>Class: BankUI</b>	<b>Collaborators</b>
<b>Responsibilities</b>  Display Menu Call Add Customer Call Add Account Call Deposit Call Withdraw Call Transfer	Bank Customer

### Initial Class Diagram



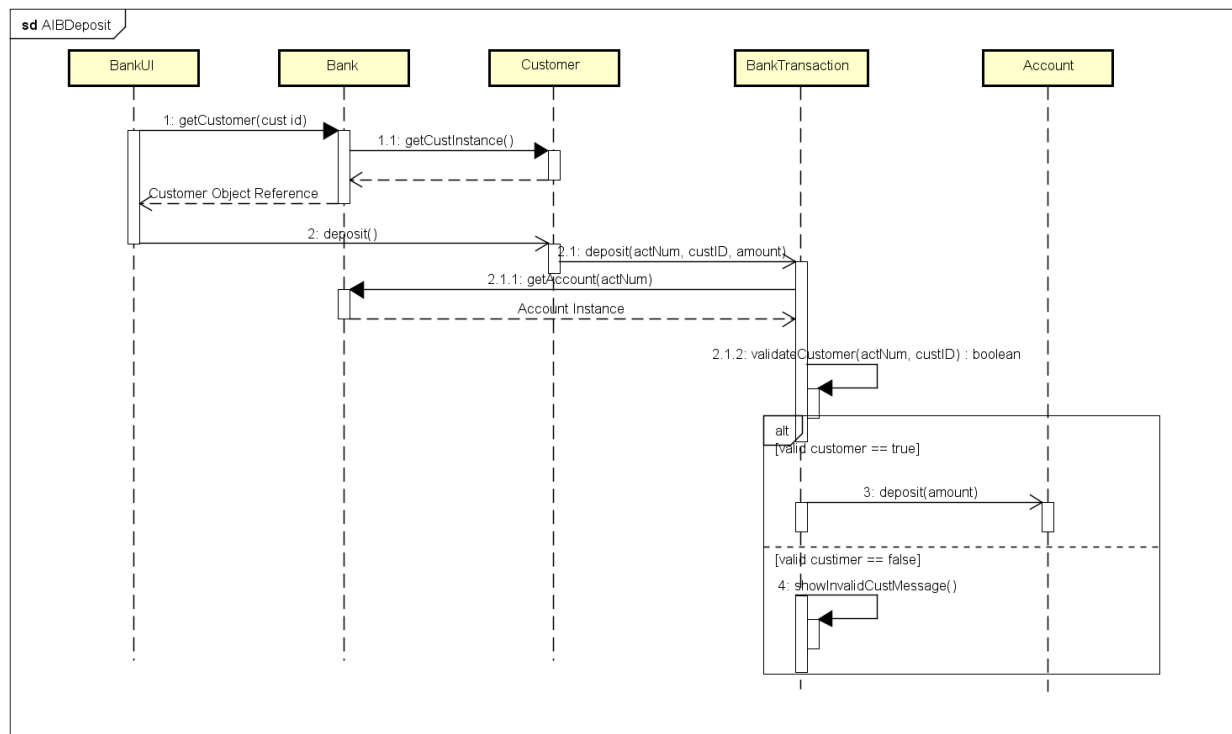
**Figure 1**  
**Initial Class Diagram (without methods and data members)**  
This reflects the object relationship in the system statically.



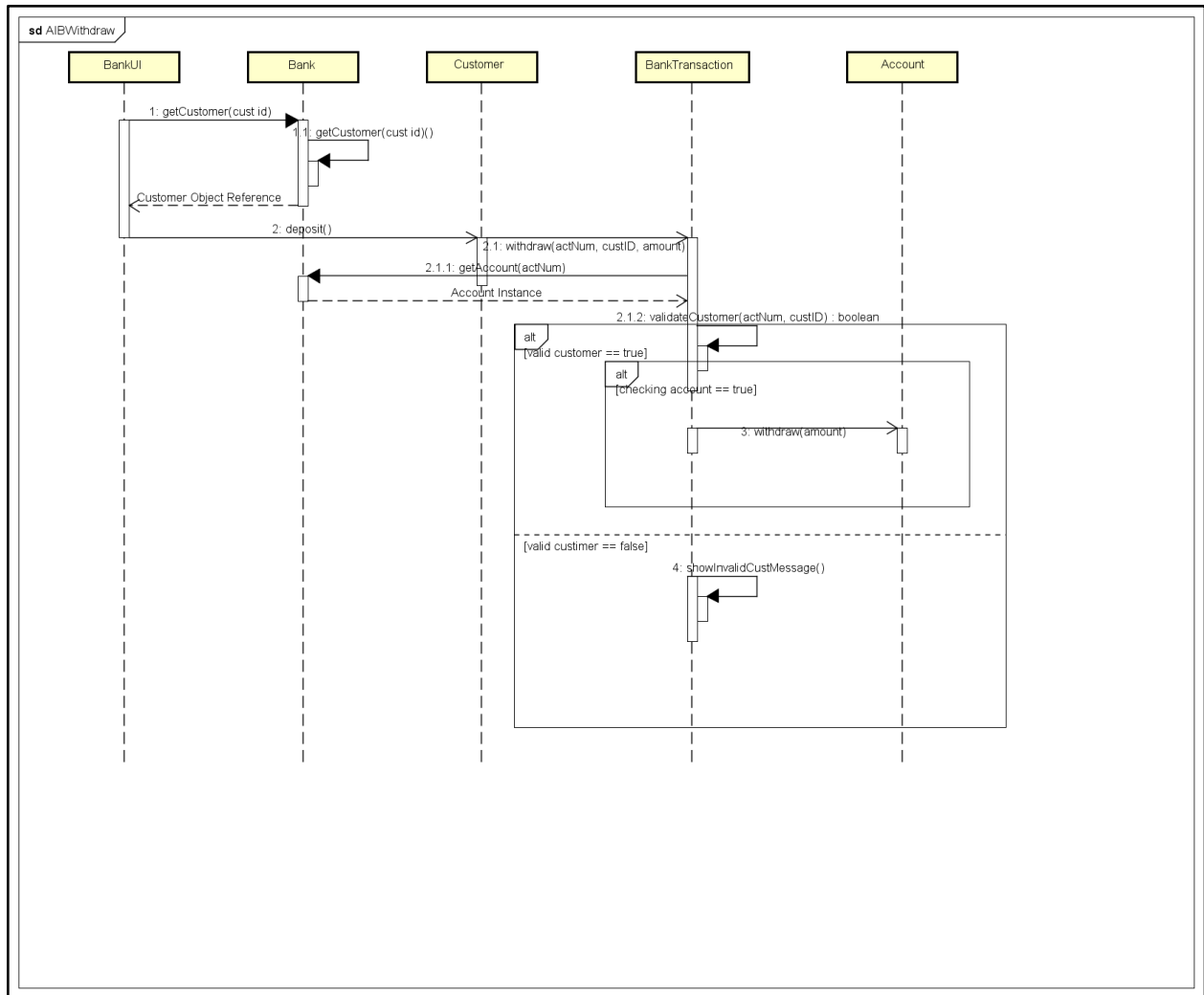
## 4. Further Elaboration of Object Interaction at Runtime: Sequence Diagrams

Sequence diagrams describe the dynamic behavior of objects. They describe interactions between objects by depicting the time ordering. In general, each use case has at least one sequence diagram.

The following two sequence diagrams demonstrate the deposit and withdrawal scenarios and how objects interact with each other while performing a deposit or withdrawal. This exercise helps designers to identify object methods, parameter passing, methods interactions at runtime while performing user scenarios in the system.

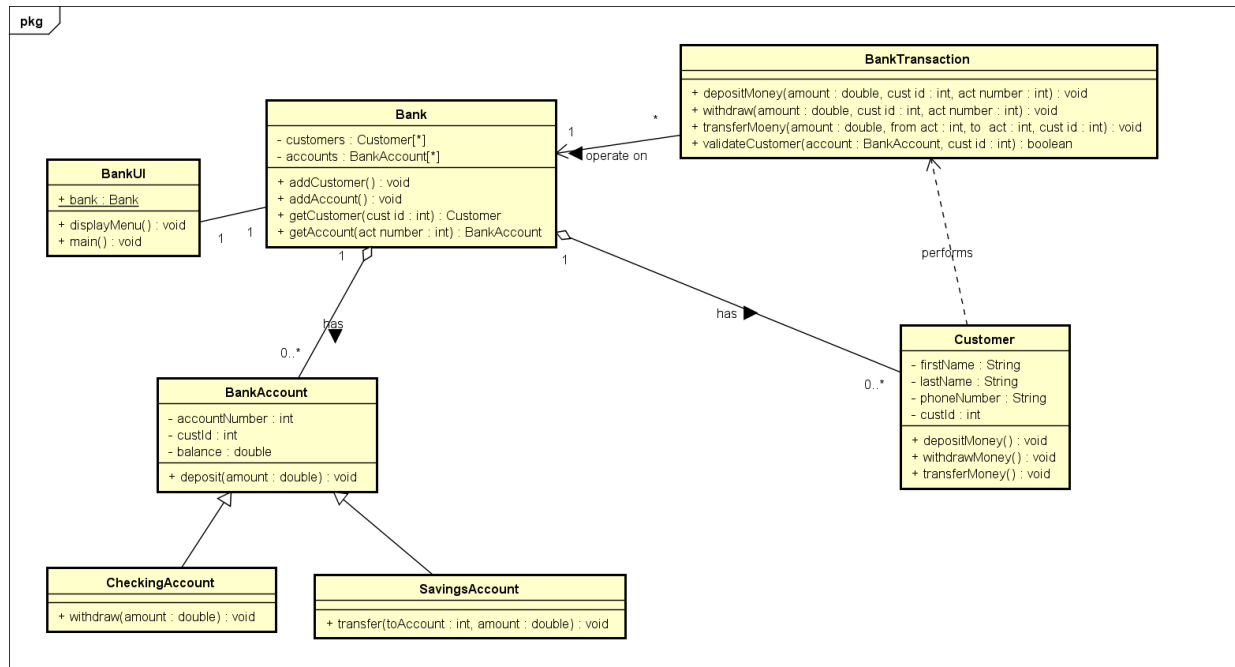


**Figure 2**  
Sequence Diagram for Deposits



**Figure 3**  
Sequence diagram for withdrawal

## 5. Analysis-level class diagram: 1st iteration



**Figure 4**  
Analysis-level class diagram

## 6. Java code implementation

Please check the source code provided. Note that the implementation is intended to demonstrate how to realize the class diagram in Figure 4, particularly class relationships. In real-world projects, this class diagram would need to be further refined by adding design and software architecture considerations.