Fall 2014 CS151 Programming Assignment 1

Instructor: Dr. Kim

ATM (Automated Teller Machine) Network

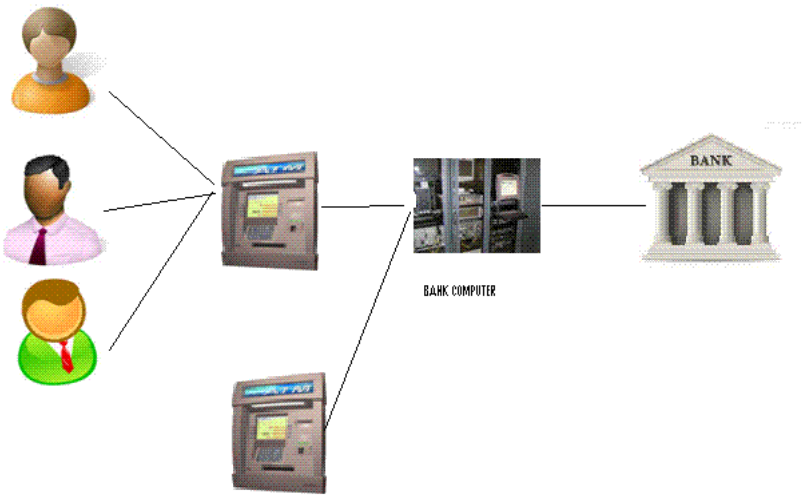
Objectives

- To exercise the analysis, design, and implementation activities of software development.
- To learn requirement analysis using use cases
- To learn noun-verb analysis and CRC technique
- To produce UML class diagram; Use a software tool to draw a UML diagram

---

In this assignment, you will design and implement an ATM network system. The analysis and design will be done by a group of two. Its implementation will be done individually. This group is only for this particular assignment 1 and does not have to be associated with your term-project group. In fact, I will form groups for this assignment while you are free to form your term project group. See the time lines and deliverables at the end of this document.

We assume two banks, each bank owns one bank computer, and each bank computer is connected two ATMs in the network. The following figure shows the part of the network that belongs to one bank.



This part of document describes entities and the mandatory requirements of ATM and Bank Computer which will be tested in grading.

Entities

- Customers get an access to an ATM machine using their cash cards and enter their own transactions. The ATM interacts the customer to gather transaction information, sends the transaction information to the bank computer for validation and processing, and dispenses cash to the customer.
- A bank manages accounts for its customers and issues cash cards authorizing access to accounts over the ATM network.

- A bank operates a bank computer system that interfaces the ATM network and the bank's account management station.
- A bank customer holds one or more accounts in the bank. For simplicity, let's limit the maximum number of accounts to two: checking and saving. That is, a customer can hold one checking or one saving or both.
- A cash card is assigned to a customer and allows the customer to access his/her accounts using an ATM machine. Each card contains a bank id and a card number. The card number determines the accounts that the card can access. A card does not have to authorize access to all of a customer's accounts. We assume that each card is owned by a single customer and thus simultaneous use of the same card from different ATMs will not be considered.

Requirements of ATM

General Requirements

1. Initializing each ATM in the network with maximum amount of cash a customer (a cash card) can withdraw per day, maximum amount of cash a customer (a cash card) can withdraw per transaction, the minimum cash in the ATM to permit a transaction, and the total fund in the ATM at start of day.
2. If the ATM doesn't have enough cash, no card is accepted and an error message is displayed.

Authorization Requirements

3. The ATM has to check if the entered card is valid by checking its expiration date. We assume that the card is readable and the card is valid if it is not expired. If it not valid, display error message and return card to the customer.
4. If the card is valid, the ATM reads the bank id and card number. Initiate authorization dialog. The card number should be logged.
5. Authorization dialog: The user is requested to enter his password. The ATM verifies the bank id and password with the bank computer. The ATM receives the result of authorization (accept/reject) from bank. There are two different error message for the reject authorization: wrong password and the card of the bank is not supported by the ATM. For the reject authorization, the relevant error message is displayed and card is returned to the customer.
6. If authorization is accepted, start transaction dialog.
7. If a card was entered more than three times in a row at any ATM and the password was wrong each time, the card is kept by the ATM. A message will be displayed that the customer should call the bank.

Requirements to process transactions

8. For simplicity, we assume that the ATM offers only withdrawal function. When authorization is successfully completed, the customer can withdraw money by entering an amount. If the amount is too much, that is, the amount is not within the pre-defined transaction limit, redo the transaction. Otherwise, the ATM starts the transaction by sending request to the bank computer.
9. If the bank computer responses that the transaction can be done, the money is dispensed and the amount is logged against the card number. Also, response should be sent to bank for money dispensed. If the transaction is not successful, an error message should be displayed. In both cases, the customer should take the card.

Requirements of the Bank Computer for the ATM

Authorization Requirements

10. When the bank computer gets a request from the ATM to verify an account, the bank computer checks if the the bank code is valid. It sends the result to the ATM.
11. The bank computer checks if the the password is valid for a valid cash card. If it is not valid, it sends the result to the ATM.
12. If it is a valid cash card and a valid password, the bank computer will send a message to the ATM that everything is OK.

Requirements to process transaction

13. Ater the bank computer gets a request to process a transaction (to withdraw an amount) from the ATM, the bank computer checks if the bank account has enough money for the transaction.
14. If the amount exceeds the limit, the transaction will fail and the bank computer will send an error message to the ATM.
15. If transaction is successful, the amount is reduced from the bank account, a success message is sent to the ATM, and the customer can get money dispensed from the ATM.

Time Lines and Deliverables

- Thursday, August 28: This assignment is posted. Read the assignment description carefully to prepare to conduct requirement analysis and design process as a team.
- Tuesday, September 2 (Sections 2 and 4) and Wednesday, September 3 (Section 1): I will form a group of two in class.
- Due: **Thursday, September 11 (Sections 2 and 4) and Friday, September 12 (Section 1)**: **Each team**submits a set of hard copies of the following in class: (I need only one set per team.)
    1. A set of use cases
    2. A set of CRC cards
    3. UML class diagram
- Due: **Saturday, September 20 11:59pm**: **Each student** submit hw1.zip through the course web site SUBMISSION link. Do not share any part of your code with anyone.) **Do not zip your workspace and/or subdirectories. Zip required files only without any directory hiearchy. Any violation of this requirement will impose -2 points of penalty on your grade.** hw1.zip is supposed to contain
    1. Required Java source files (.java) to run the program. Each source must be fully documented with javadoc comments. Name the class with the main method ATMSystem.
    2. input.txt that contains all the inputs you would enter through console to run the program. The input.txt will be used to run the program using input redirection as shown below:

       ```
       %java ATMSystem < input.txt
       ```

       If you are not familiar with input redirection, read this.
    3. readme.txt that explains specific inputs and their format for each requirement. (Note that requirements are all numbered (1 ~ 15). Use the number to recognize a particular requirement. The readme.txt will be used if the program doesn't run completely using input redirection. That is, if the program abnormally terminate in the middle of execution, the readme.txt will be used to manually test each requirement one by one.) If a certain requirement is not implemented at all, explain that in readme, so that it can be skipped in testing.
- Softcopy Grading: Due to the large number of students this semester, I will randomly choose and

grade only 8 functions (out of 15). If your program doesn't perform the selected function(s) in grading, the corresponding score will not be given.