

Object Oriented Programming Fall 2020

Semester Project

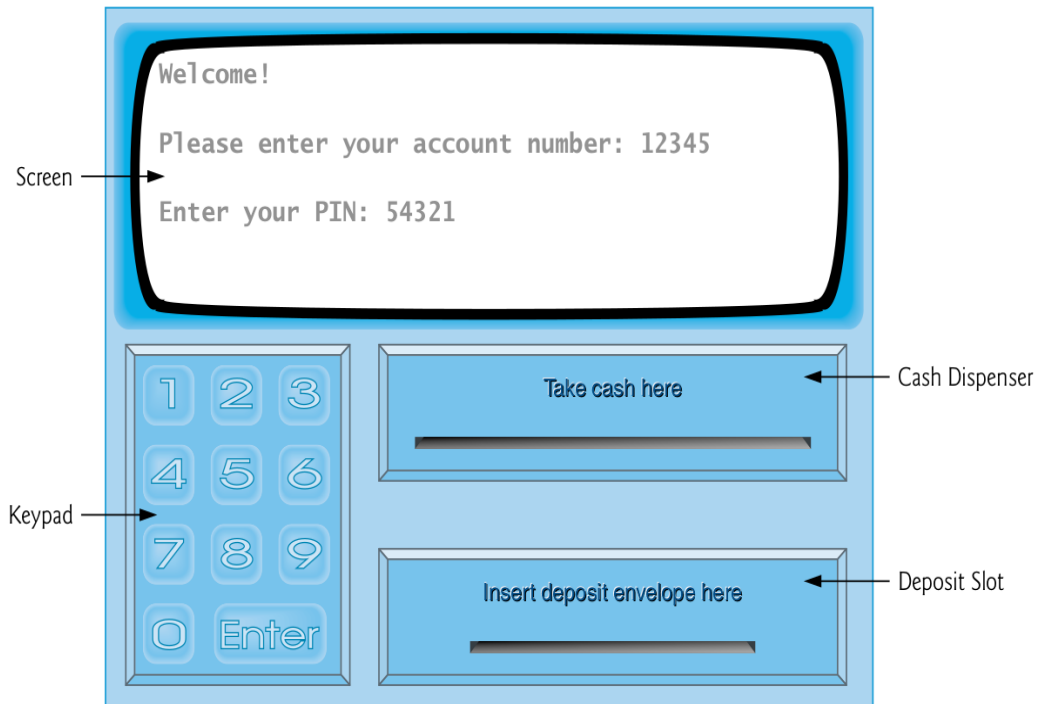


Fig. 25.1 | Automated teller machine user interface.

Course Instructors

Dr. Subhan Ullah, Dr. Abdul Waheed khan, Mr. Jawad Hassan.

Design Deadline (Class Diagram): 16 December ,2020

Deadline Implementation: 26 December,2020

Project weightage: 10%

Instructions

Make sure that you read and understand each and every instruction. If you have any questions or comments, you are encouraged to discuss your problems with your colleagues (and instructors) on google classroom.

Plagiarism is strongly forbidden and will be very strongly punished. If we find that you have copied from someone else or someone else has copied from you (with or without your knowledge) both of you will be punished. You will be awarded (straight zero in the project — which can eventually result in your failure) and appropriate action as recommended by the Disciplinary Committee (DC can even award a straight F in the subject) will be taken.

Try to understand and do the project yourself even if you are not able to complete the project. Note that you will be mainly awarded on your effort not on the basis whether you have completed the project or not.

No Marks will be given if you do not submit your class diagram and if you do not use the object oriented design principles you have learned during the course.

Imagination Powers: Use your imaginative powers to make this as interesting and appealing as you can think of. An excellent solution can get you bonus.

Combine all your work in one .zip file.

Name the .zip file as ROLL-NUM SECTION.zip (e.g. 19i-0001 B.zip).

Submit the .zip file on Google Classroom.

You must follow the submission instructions to the letter, as failing to do so will get you a zero in the Project.

Examining the ATM Requirements Document:

We begin our design process by presenting a requirements document that specifies the ATM system's overall purpose and what it must do. Throughout the case study, we refer to the requirements document to determine what functionality the system must include.

Requirements Document:

A local bank intends to install a new automated teller machine (ATM) to allow users (i.e., bank customers) to perform basic financial transactions (Fig. 25.1). Each user can have only one account at the bank. ATM users should be able to view their account balance, withdraw cash (take money out of an account) and deposit funds (i.e., place money into an account).

The objective of project is to make a Bank ATM Machine application. The user interface of the automated teller machine contains the following hardware components:

- a screen that displays messages to the user
- a keypad that receives numeric input from the user
- a cash dispenser that dispenses cash to the user and
- a deposit slot that receives deposit envelopes from the user.

The cash dispenser begins each day loaded with 500 \$20 bills. [Note: Owing to the limited scope of this case study, certain elements of the ATM described here do not accurately mimic those of a real ATM. For example, a real ATM typically contains a device that reads a user's account number from an ATM card, whereas this ATM asks the user to type an account number using the keypad. A real ATM also usually prints a receipt at the end of a session, but all output from this ATM appears on the screen.]

The bank wants you to develop software to perform the financial transactions initiated by bank customers through the ATM. The bank will integrate the software with the ATM's hardware at a later time. The software should encapsulate the functionality of the hardware devices (e.g., cash dispenser, deposit slot) within software components, but it need not concern itself with how these devices perform their duties. The ATM hardware has not been developed yet, so instead of writing your software to run on the ATM, you should develop a first version of the software to run on a personal computer. This version should use the computer's monitor to simulate the ATM's screen, and the computer's key- board to simulate the ATM's keypad.

An ATM session consists of authenticating a user (i.e., proving the user's identity) based on an account number and personal identification number (PIN), followed by creating and executing financial transactions. To authenticate a user and perform transactions, the ATM must interact with the bank's account information database. [Note: A database is an organized collection of data stored on a computer.] For each bank account, the database stores an account number, a PIN and a balance indicating the amount of money in the account. [Note: For simplicity, we assume that the bank plans to build only one ATM, so we do not need to worry about multiple ATMs accessing this database at the same time. Furthermore, we assume that the bank does not make any changes to the information in the database while a user is accessing the ATM. Also, any business system like an ATM faces reasonably complicated security issues that go well beyond the scope of a first- or second- semester computer science course. We make the simplifying assumption, however, that the bank trusts the ATM to access and manipulate the information in the database without significant security measures.]

Upon first approaching the ATM, the user should experience the following sequence of events (shown in Fig. 25.1):

1. The screen displays a welcome message and prompts the user to enter an account number.
2. The user enters a five-digit account number, using the keypad.

3. The screen prompts the user to enter the PIN (personal identification number) associated with the specified account number.
4. The user enters a five-digit PIN, using the keypad.
5. If the user enters a valid account number and the correct PIN for that account, the screen displays the main menu (Fig. 25.2). If the user enters an invalid account number or an incorrect PIN, the screen displays an appropriate message, then the ATM returns to Step 1 to restart the authentication process.

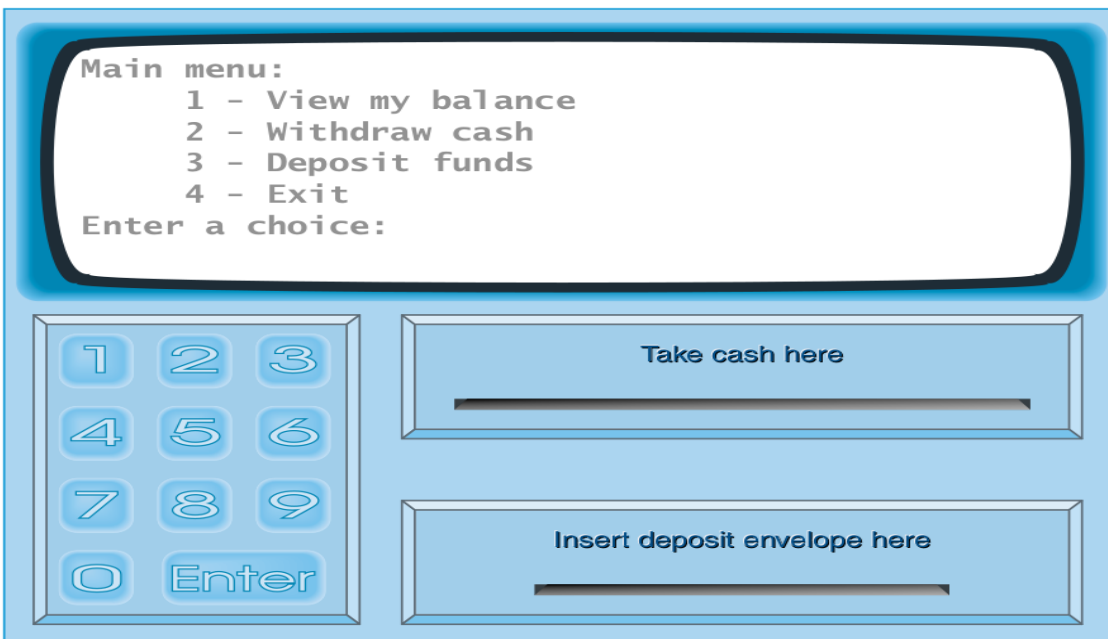


Fig. 25.2 | ATM main menu.

After the ATM authenticates the user, the main menu (Fig. 25.2) displays a numbered option for each of the three types of transactions: balance inquiry (option 1), withdrawal (option 2) and deposit (option 3). The main menu also displays an option that allows the user to exit the system (option 4). The user then chooses either to perform a transaction (by entering 1, 2 or 3) or to exit the system (by entering 4). If the user enters an invalid option, the screen displays an error message, then redisplay to the main menu.

If the user enters 1 to make a balance inquiry, the screen displays the user's account balance. To do so, the ATM must retrieve the balance from the bank's database.

The following actions occur when the user enters 2 to make a withdrawal:

1. The screen displays a menu (shown in Fig. 25.3) containing standard withdrawal amounts: \$20 (option 1), \$40 (option 2), \$60 (option 3), \$100 (option 4) and \$200 (option 5). The menu also contains an option to allow the user to cancel the transaction (option 6).
2. The user enters a menu selection (1–6) using the keypad.
3. If the withdrawal amount chosen is greater than the user's account balance, the screen displays a message stating this and telling the user to choose a smaller amount. The ATM then returns to Step 1. If the withdrawal amount chosen is less than or equal to the user's account balance (i.e., an acceptable withdrawal amount), the ATM proceeds to Step 4. If the user

chooses to cancel the transaction (option 6), the ATM displays the main menu (Fig. 25.2) and waits for user input.

4. If the cash dispenser contains enough cash to satisfy the request, the ATM proceeds to Step 5. Otherwise, the screen displays a message indicating the problem and telling the user to choose a smaller withdrawal amount. The ATM then returns to Step 1.

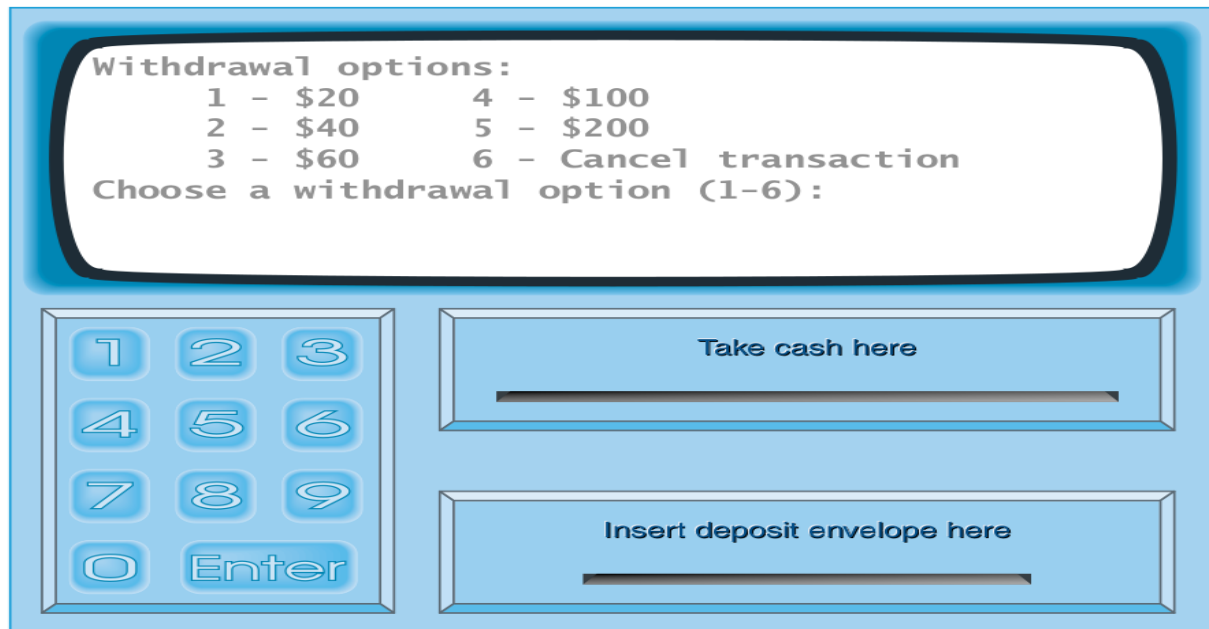


Fig. 25.3 | ATM withdrawal menu.

5. The ATM debits (i.e., subtracts) the withdrawal amount from the user's account balance in the bank's database.

6. The cash dispenser dispenses the desired amount of money to the user.

7. The screen displays a message reminding the user to take the money.

The following actions occur when the user enters 3 (while the main menu is displayed) to make a deposit:

1. The screen prompts the user to enter a deposit amount or to type 0 (zero) to cancel the transaction.

2. The user enters a deposit amount or 0, using the keypad. [Note: The keypad does not contain a decimal point or a dollar sign, so the user cannot type a real dollar amount (e.g., \$1.25). Instead, the user must enter a deposit amount as a number of cents (e.g., 125). The ATM then divides this number by 100 to obtain a number representing a dollar amount (e.g., $125 \div 100 = 1.25$).]

3. If the user specifies a deposit amount, the ATM proceeds to Step 4. If the user chooses to cancel the transaction (by entering 0), the ATM displays the main menu (Fig. 25.2) and waits for user input.

4. The screen displays a message telling the user to insert a deposit envelope into the deposit slot.

5. If the deposit slot receives a deposit envelope within two minutes, the ATM credits (i.e., adds) the deposit amount to the user's account balance in the bank's database. This money is not immediately available for withdrawal. The bank first must physically verify the amount of cash in the deposit envelope, and any checks in the envelope must clear (i.e., money must be

transferred from the check writer's account to the check recipient's account). When either of these events occurs, the bank appropriately updates the user's balance stored in its database. This occurs independently of the ATM system. If the deposit slot does not receive a deposit envelope within this time period, the screen displays a message that the system has canceled the transaction due to inactivity. The ATM then displays the main menu and waits for user input.

After the system successfully executes a transaction, the system should redisplay the main menu (Fig. 25.2) so that the user can perform additional transactions. If the user chooses to exit the system (option 4), the screen should display a thank you message, then display the welcome message for the next user.

Designing the ATM System

Before begin the ATM system's design. A system is a set of components that interact to solve a problem. To perform the ATM system's designated tasks, our ATM system has a user interface (Fig. 25.1), contains software that executes financial transactions and interacts with a database of bank account information. System structure describes the system's objects and their interrelationships. System behavior describes how the system changes as its objects interact with one another. Every system has both structure and behavior designers must specify both. There are several distinct types of system structures and behaviors. For example, the interactions among objects in the system differ from those between the user and the system, yet both constitute a portion of the system behavior.

Expected output of project:

1.Before writing code read the scenario carefully that help you for identifying the classes required to build the ATM system.

Note: classes identify irrelevant to scenario given will be marked zero so try to write code related to scenario.

2.Create class diagram with operations and attributes as mentioned in scenario given, show generalization / inheritance, all types of UML Associations. Whatever applies in this scenario), and also show aggregation, and compositions if applicable.

3. All the information must be stored into filing system. Instead of using databases You must have to make your project perfect by introducing filing in it. Save the transactions of a person in a file, read the balance from there, deposit balance there. By filing you can make a perfect ATM system.

4.Design Proper Graphical User Interface for ATM screens and STL as Bonus marks.

5.Your project must cover all 4 pillars of object oriented programming including polymorphism, encapsulation, inheritance and data abstraction.