SOFTWARE DESIGN & ARCHITECTURE

(Lecture-6)

USAMA MUSHARAF

LECTURER (Department of Computer Science)

FAST-NUCES PESHAWAR

CONTENTS

- Case Study with in depth Analysis & Design,

CASE STUDY

EXAMINING THE REQUIREMENT DOCUMENT

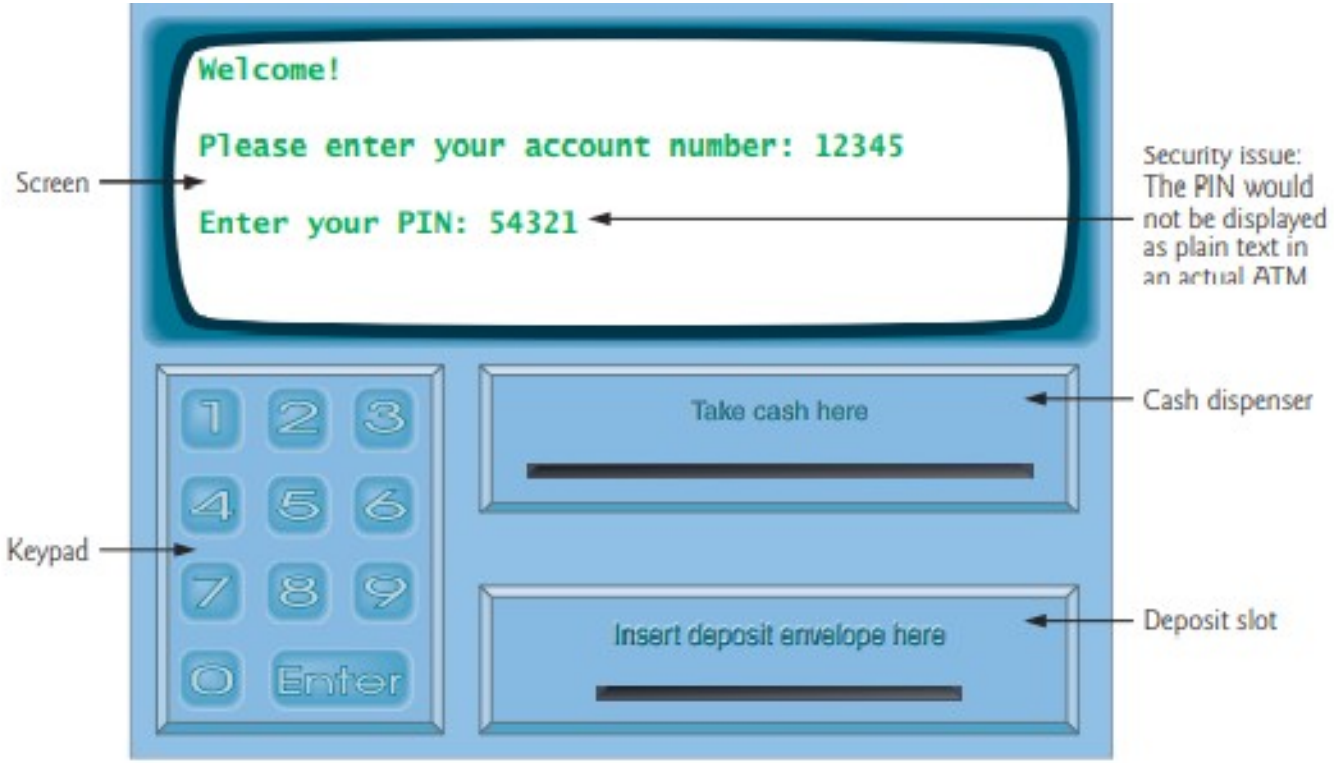
Requirement Document

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

Requirement Document

The user interface of the automated teller machine contains

- 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



Requirement Document

- 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 to run on a personal computer.
  - This version should use the computer’s monitor to simulate the ATM’s screen, and the computer’s keyboard to simulate the ATM’s keypad.

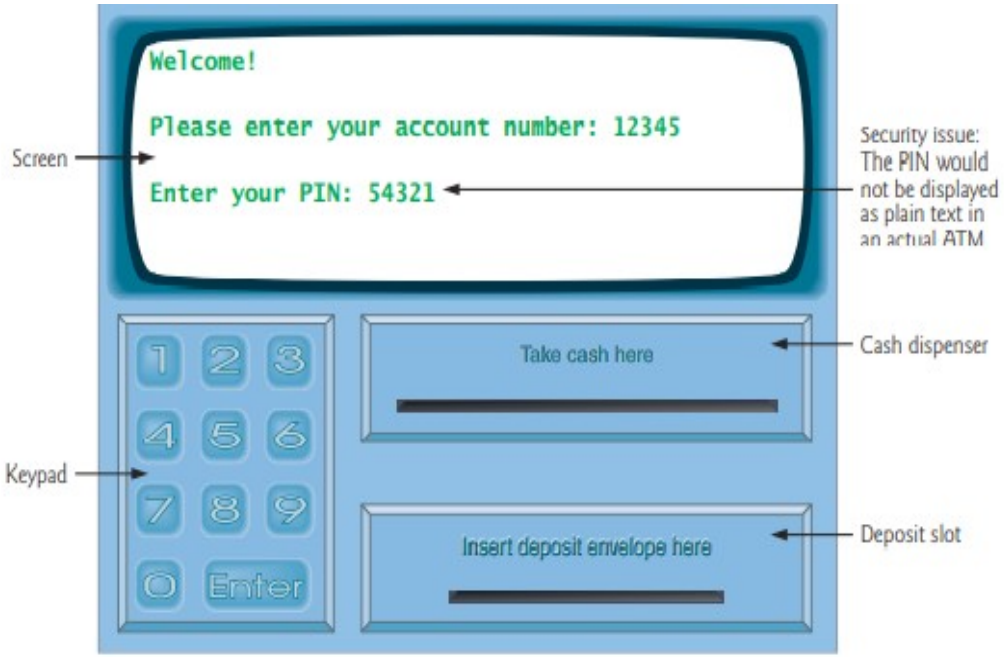
Requirement Document

- 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 (i.e., 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: We assume that the bank plans to build only one ATM, so we need not 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.

Requirement Document

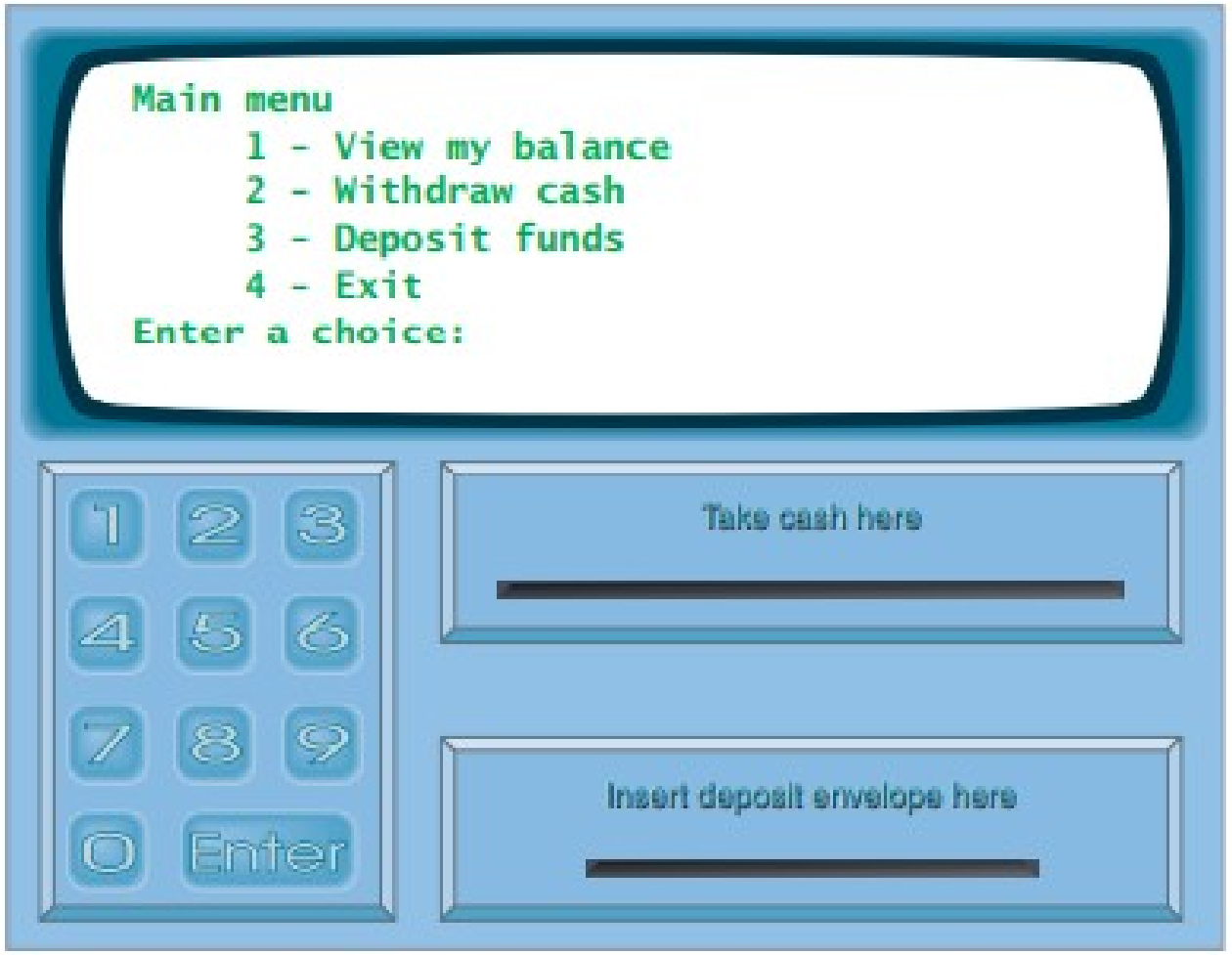
Upon first approaching the ATM (assuming no one is currently using it), the user should experience the following sequence of events.

- 1. The screen displays Welcome! 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.



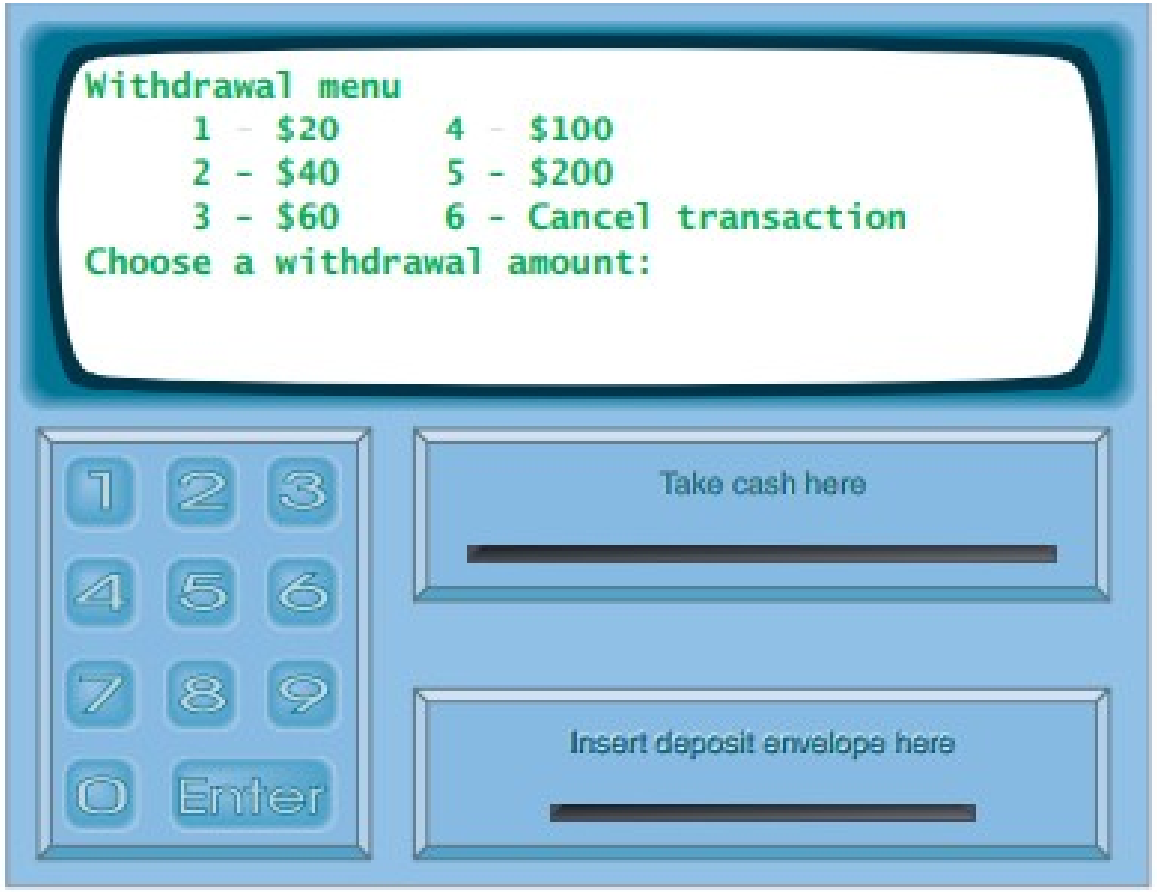
Requirement Document

- 5. If the user enters a valid account number and the correct PIN for that account, the screen displays the main menu. 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.



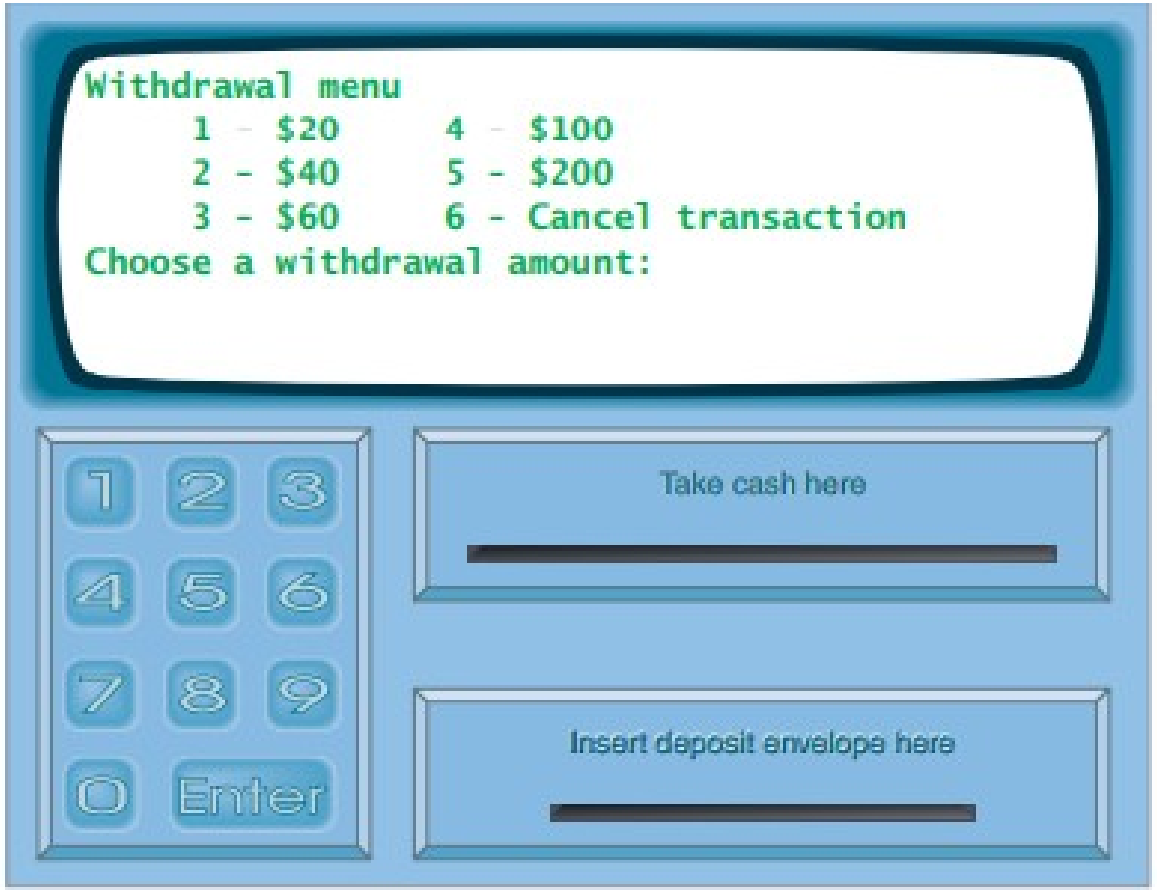
Requirement Document

- The screen displays a menu 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).
- The user enters a menu selection using the keypad.
- 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 amount), the ATM proceeds to Step 4. If the user chooses to cancel the transaction (option 6), the ATM displays the main menu and waits for user input



Requirement Document

- If the cash dispenser contains enough cash, 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.
- The ATM debits the withdrawal amount from the user's account in the bank's database (i.e., subtracts the withdrawal amount from the user's account balance).
- The cash dispenser dispenses the desired amount of money to the user.
- The screen displays a message reminding the user to take the money.



Requirement Document

The following steps describe the actions that occur when the user enters 3 (when viewing the main menu) to make a deposit:

- 1. The screen prompts the user to enter a deposit amount or type 0 (zero) to cancel.
- 2. The user enters a deposit amount or 0 using the keypad.
- 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 and waits for user input.
- 4. The screen displays a message telling the user to insert a deposit envelope.

Requirement Document

- If the deposit slot receives a deposit envelope within two minutes, the ATM credits the deposit amount to the user's account in the bank's database (i.e., adds the deposit amount to the user's account balance). 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, it should return to the main menu so that the user can perform additional transactions. If the user exits the system, the screen should display a thank you message, then display the welcome message for the next user.

ANALYZING ATM SYSTEM

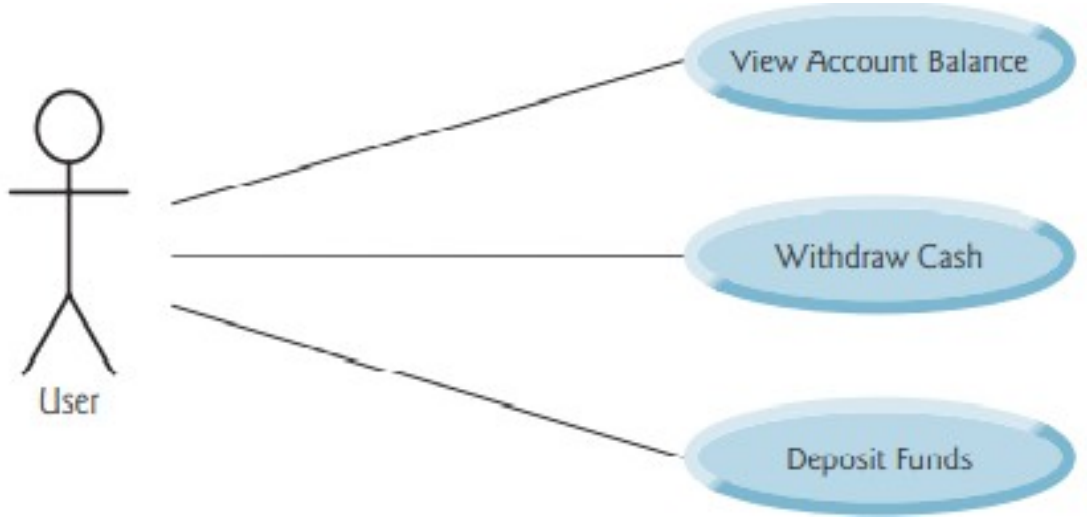
Analysis

- The requirements document (presented earlier) describes the requirements of our ATM system in sufficient detail that you need not go through an extensive analysis stage.
- If requirements are not in sufficient details, then use
  - Use Case
  - Activity
  - Swim Lane

  to capture requirements. (already discussed).

Use Case of ATM Case Study

- To capture what a proposed system should do, developers often employ a technique known as use case modeling. This process identifies the use cases of the system, each representing a different capability that the system provides to its clients.
- For example, ATMs typically have several use cases, such as “View Account Balance,” “Withdraw Cash,” “Deposit Funds,” “Transfer Funds Between Accounts”.



DESIGNING ATM SYSTEM

Designing

- A system is a set of components that interact to solve a problem. For example, to perform the ATM system's designated tasks, our ATM system has a user interface (as shown in requirement document), and 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.

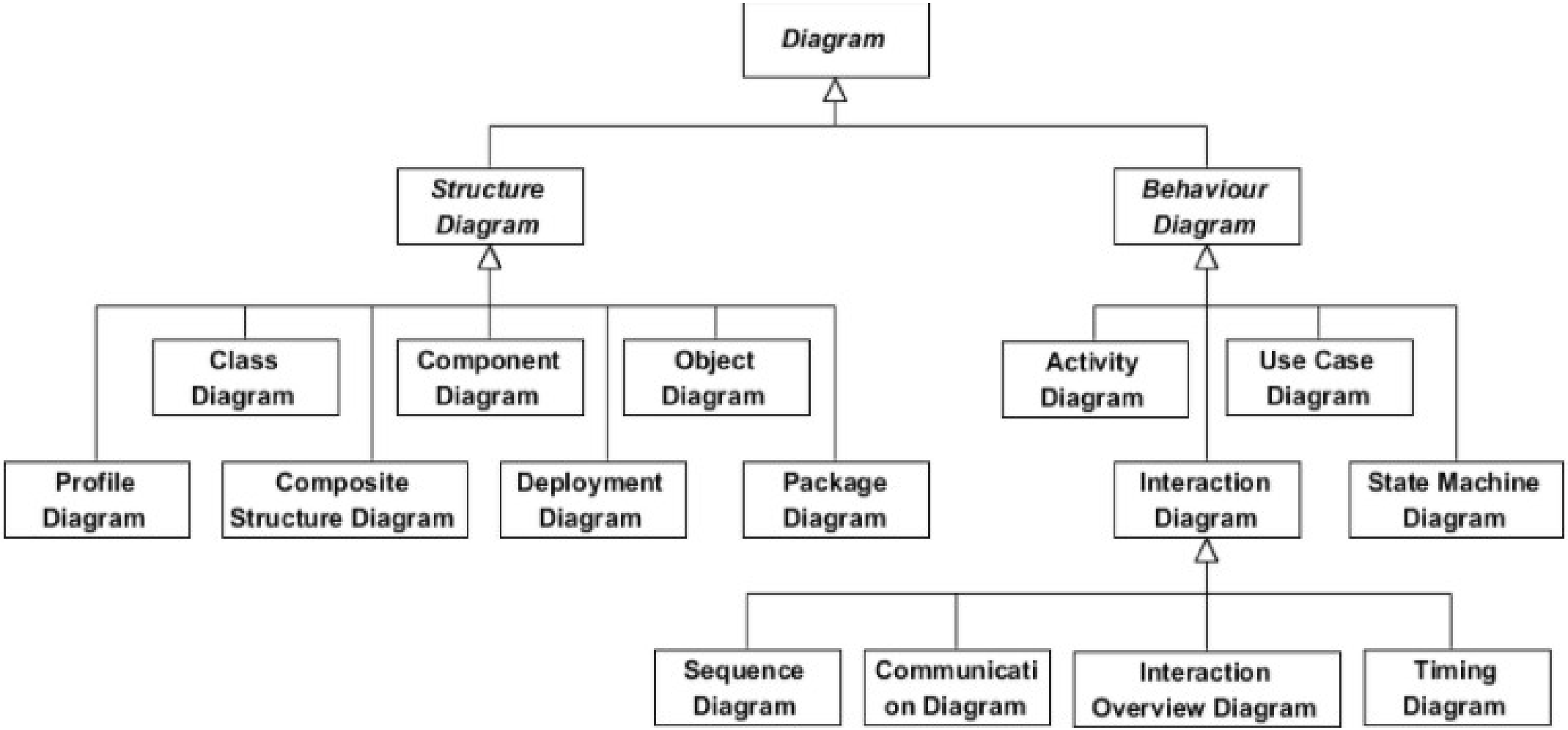


Diagram
Structure Diagram
Behaviour Diagram
Class Diagram
Component Diagram
Object Diagram
Activity Diagram
Use Case Diagram
Profile Diagram
Composite Structure Diagram
Deployment Diagram
Package Diagram
Interaction Diagram
State Machine Diagram
Sequence Diagram
Communicati on Diagram
Interaction Overview Diagram
Timing Diagram

UML

- **Use Case Diagrams,** model the interactions between a system and its external entities (actors) in terms of use cases (system capabilities, such as “View Account Balance,” “Withdraw Cash” and “Deposit Funds”.
- **Class diagrams,** model the classes, or “building blocks,” used in a system. Each noun or “thing” described in the requirements document is a candidate to be a class in the system (e.g., Account, Keypad). Class diagrams help us specify the structural relationships between parts of the system.
  - For example, the ATM system class diagram will specify that the ATM is physically composed of a screen, a keypad, a cash dispenser and a deposit slot.

UML

- **State machine diagrams,** model the ways in which an object changes state. An object's state is indicated by the values of all its attributes at a given time. When an object changes state, it may behave differently in the system.
  - For example, after validating a user's PIN, the ATM transitions from the "user not authenticated" state to the "user authenticated" state, at which point it allows the user to perform financial transactions (e.g., view account balance, withdraw cash, deposit funds).
- **Activity diagrams,** model an object's activity—is workflow (sequence of events) during program execution. An activity diagram models the actions the object performs and specifies the order in which it performs them.
  - For example, an activity diagram shows that the ATM must obtain the balance of the user's account (from the bank's account information database) before the screen can display the balance to the user.

UML

- **Communication diagrams** (called collaboration diagrams in earlier versions of the UML) model the interactions among objects in a system, with an emphasis on what interactions occur..
  - For example, the ATM must communicate with the bank's account information database to retrieve an account balance.

- **Sequence diagrams** also model the interactions among the objects in a system, but unlike communication diagrams, they emphasize when interactions occur.
  - For example, the screen prompts the user to enter a withdrawal amount before cash is dispense.

IDENTIFYING CLASSES IN A REQUIREMENT DOCUMENT

Identifying Classes in a System

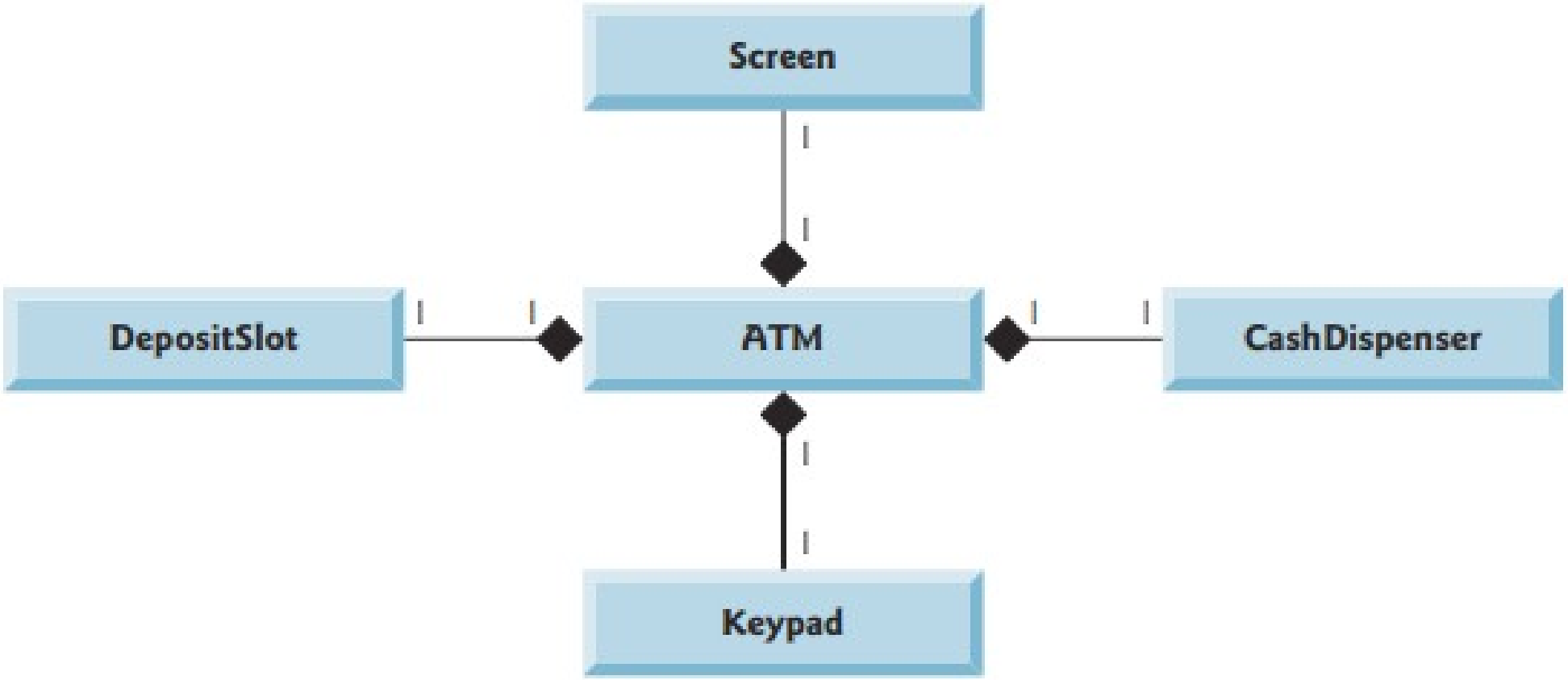
Nouns and noun phrases in the ATM requirements document			
bank	money / funds	account number	ATM
screen	PIN	user	keypad
bank database	customer	cash dispenser	balance inquiry
transaction	$20 bill / cash	withdrawal	account
deposit slot	deposit	balance	deposit envelope

Identifying Classes in a System

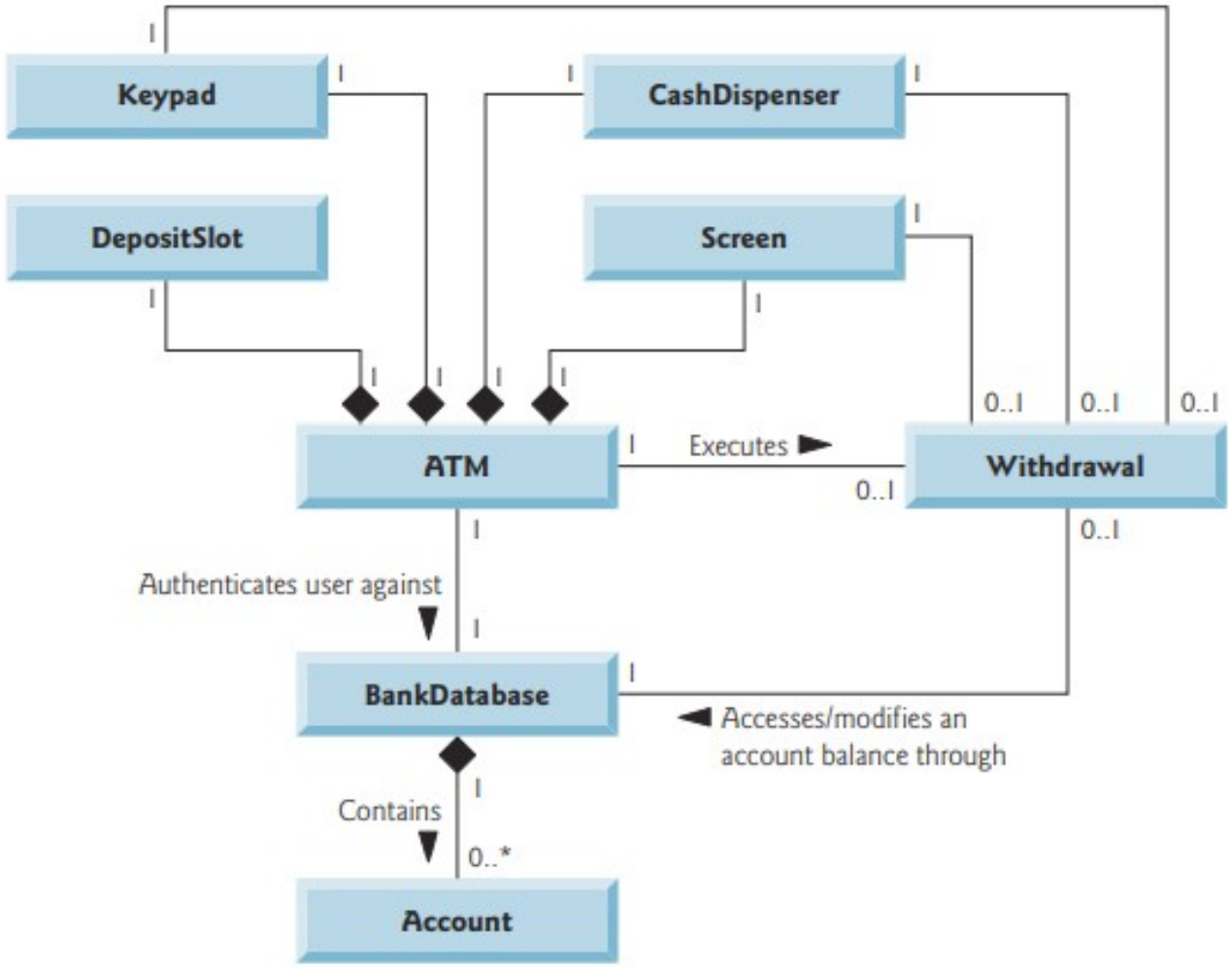
- ATM
- Screen
- Keypad
- Cash dispenser
- Deposit slot
- Account
- Bank database
- Balance inquiry
- Withdrawal
- Deposit

MODELING CLASSES

Modeling



Modeling


Keypad
CashDispenser
DepositSlot
Screen
ATM
Executes
Withdrawal
Authenticates user against
BankDatabase
Accesses/modifies an account balance through
Contains
Account
1
0..1
0..*

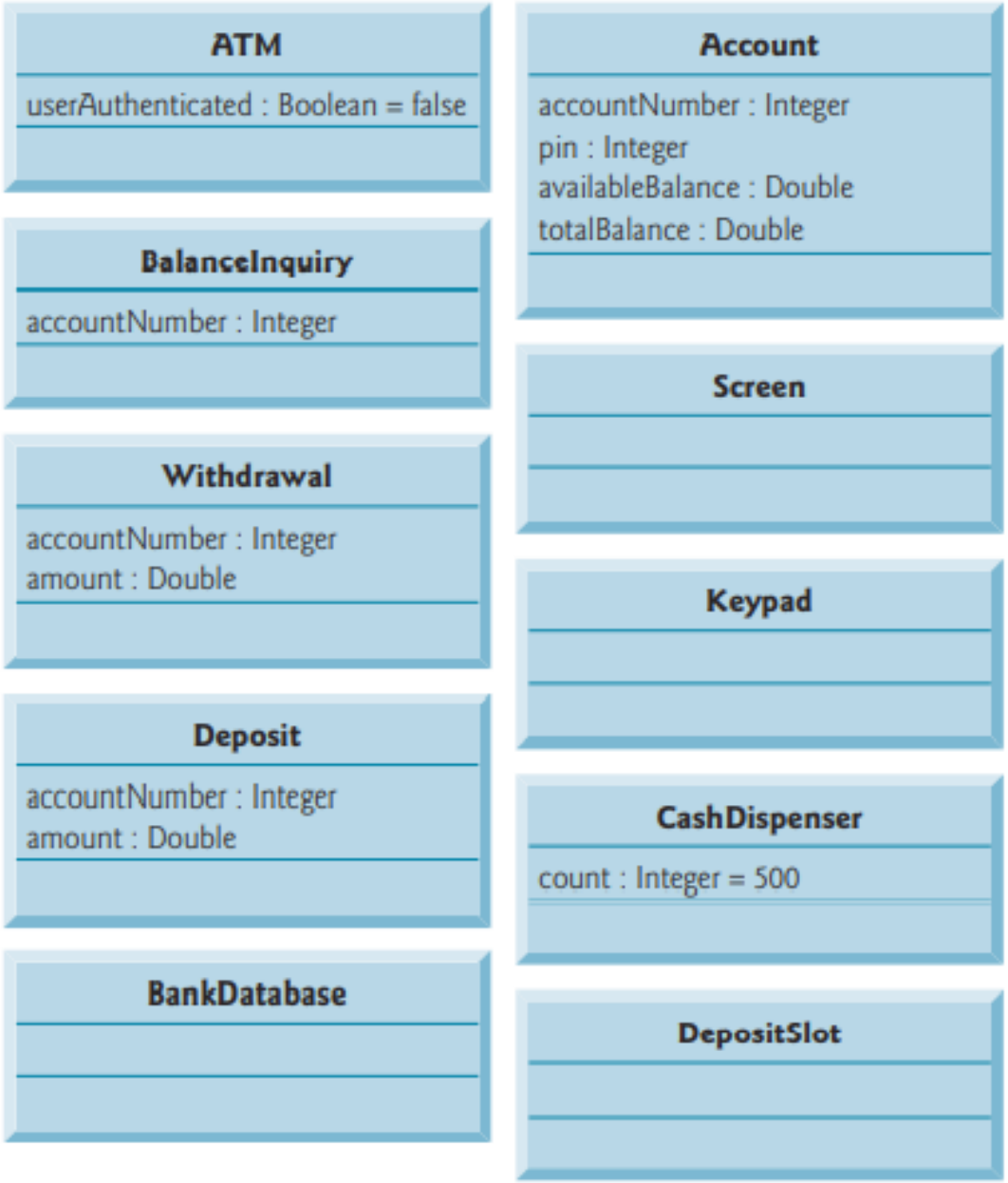
IDENTIFYING CLASS ATTRIBUTES

identifying class attributes

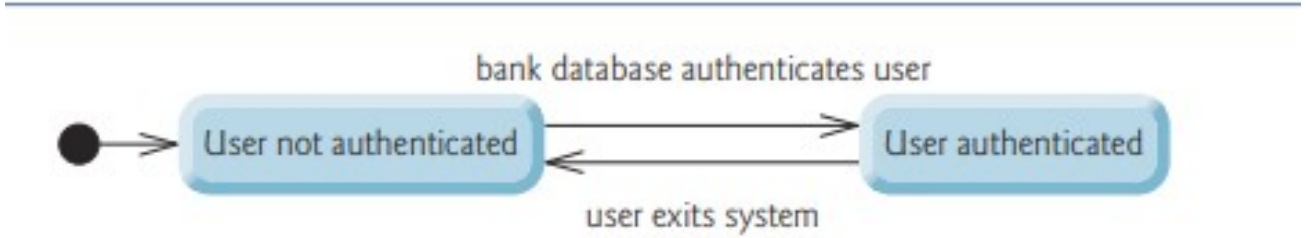
- Classes have attributes (data) and operations (behaviors).

- Account No
- Amount
- PIN
- Balance
- User Authentication

- Classes Attributes



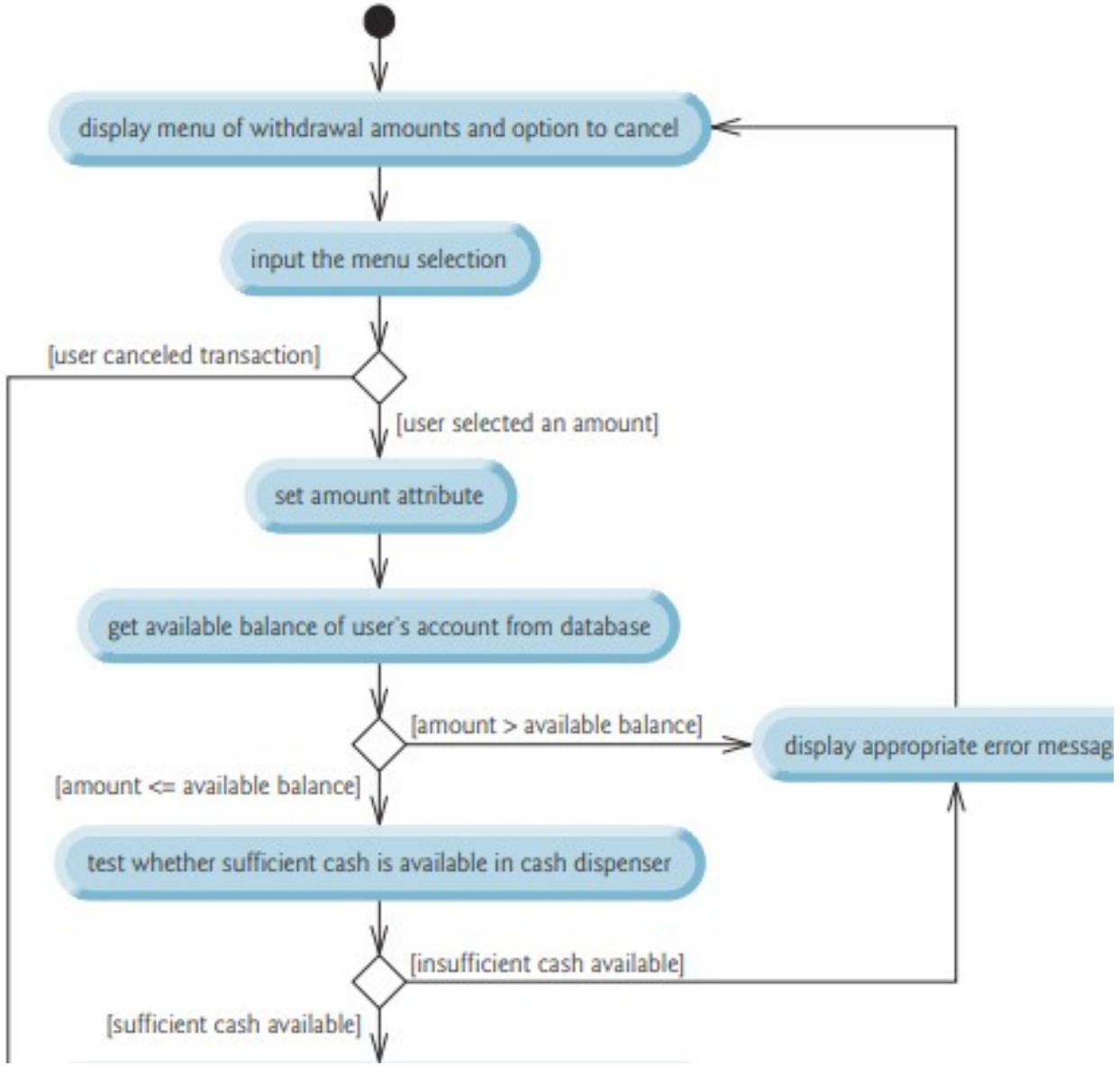
IDENTIFYING OBJECTS STATES & ACTIVITIES

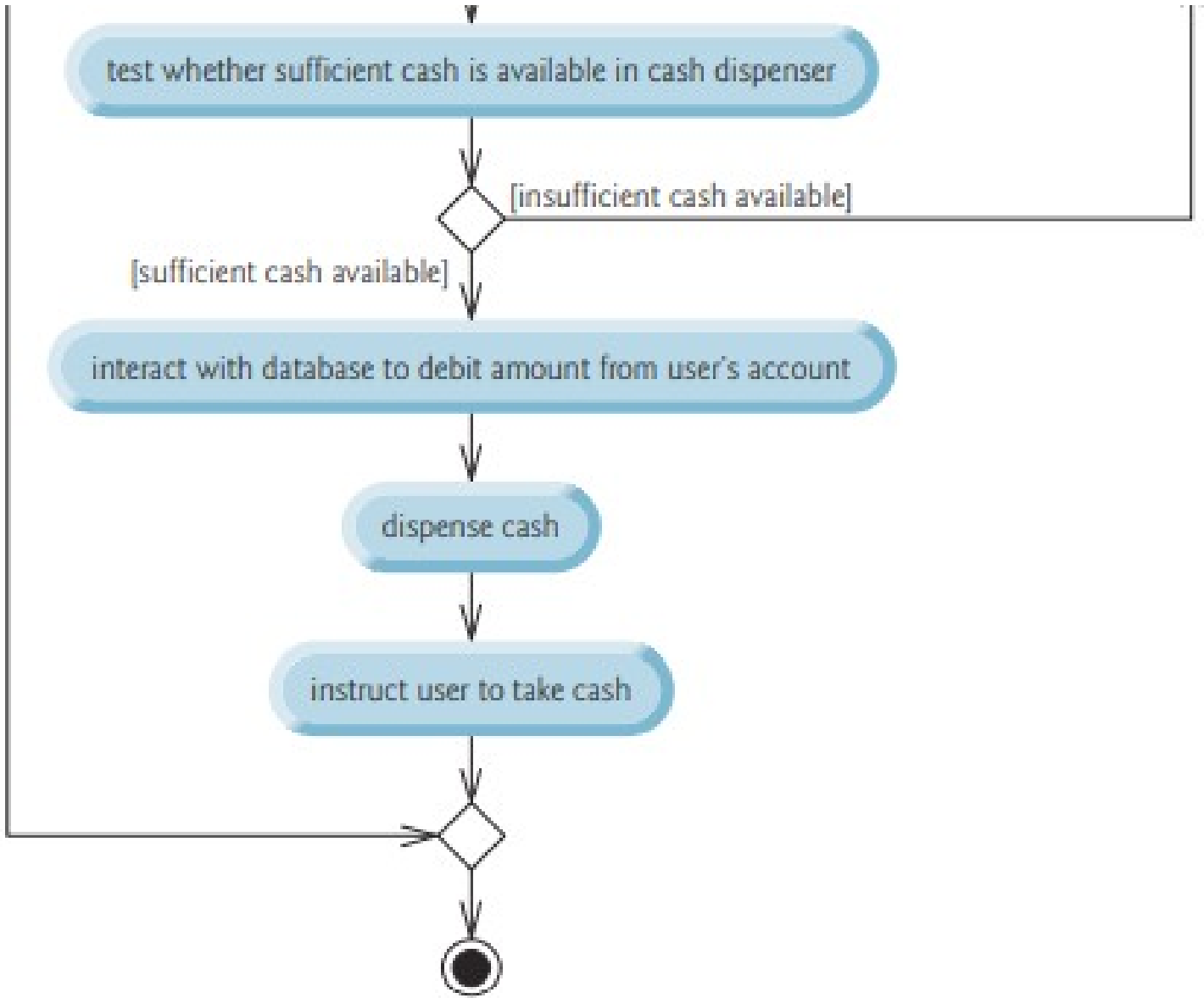


State Machine Diagrams

- Each object in a system goes through a series of states. An object’s state is indicated by the values of its attributes at a given time.
- State machine diagrams (commonly called state diagrams) model several states of an object and show under what circumstances the object changes state.

Activity Diagram of Withdraw transaction



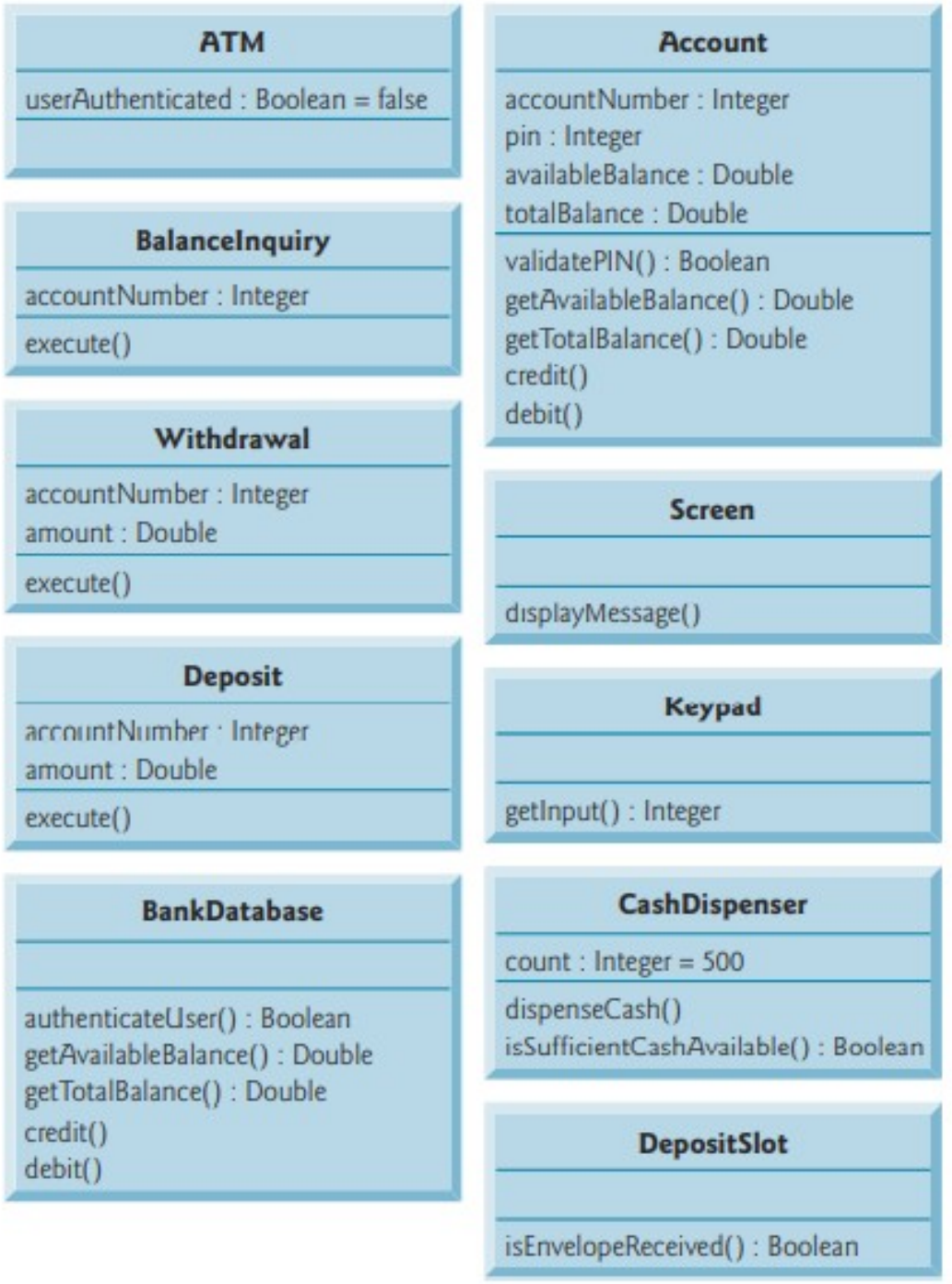

test whether sufficient cash is available in cash dispenser
[insufficient cash available]
[sufficient cash available]
interact with database to debit amount from user's account
dispense cash
instruct user to take cash

IDENTIFYING CLASS OPERATIONS

An operation is a service that objects of a class provide to clients (users) of the class.

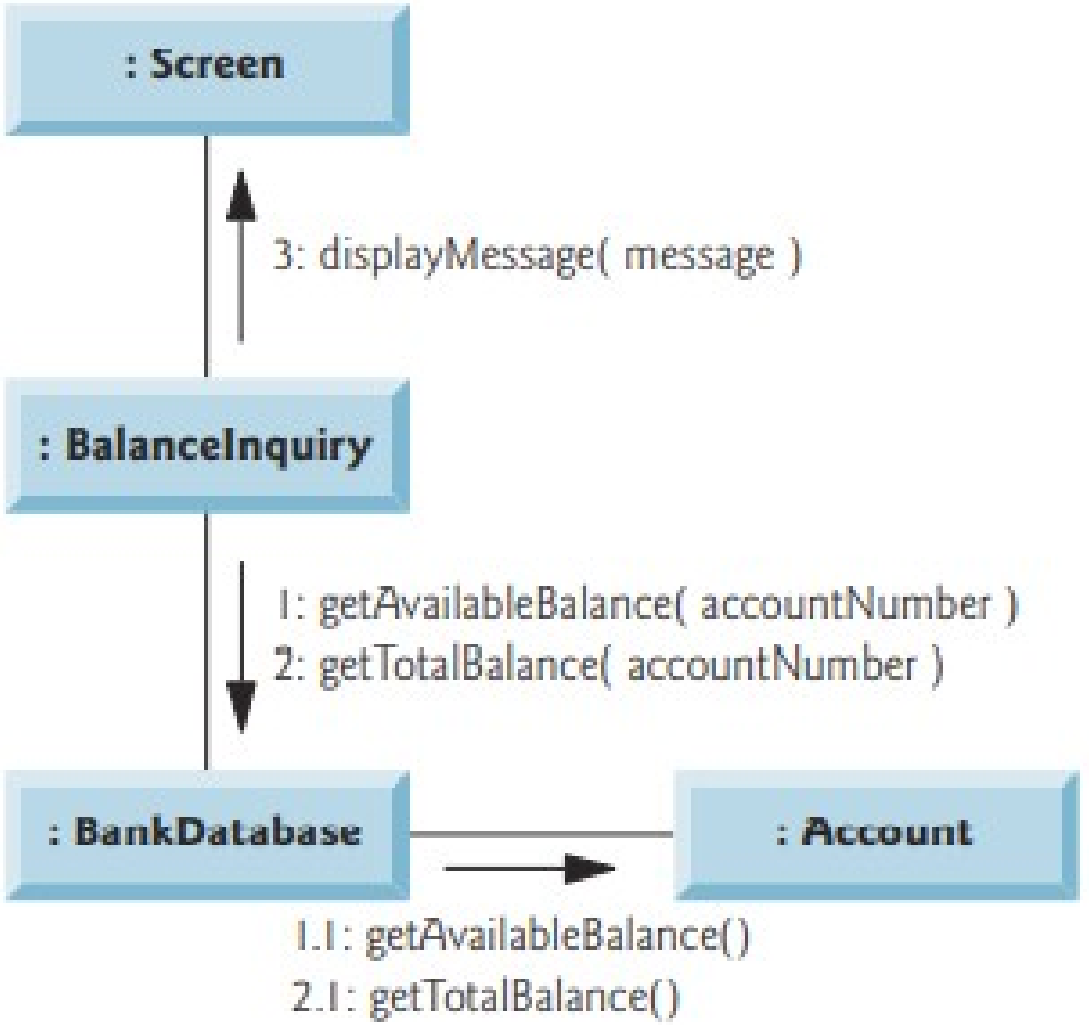
ATM	executes financial transactions
Balance Inquiry	-
Withdrawal	-
Deposit	-
Bank Database	authenticates a user, retrieves an account balance, credits a deposit amount to an account, debits a withdrawal amount from an account .
Account	retrieves an account balance, credits a deposit amount to an account, debits a withdrawal amount from an account
Screen	displays a message to the user Keypad receives numeric input from the user
Keyboard	Keypad receives numeric input from the user
Cash Dispenser	dispenses cash, indicates whether it contains enough cash to satisfy a withdrawal request
Deposit Slot	receives a deposit envelope

Modeling Operations



COLLABORATION AMONG OBJECTS

Communication Diagram



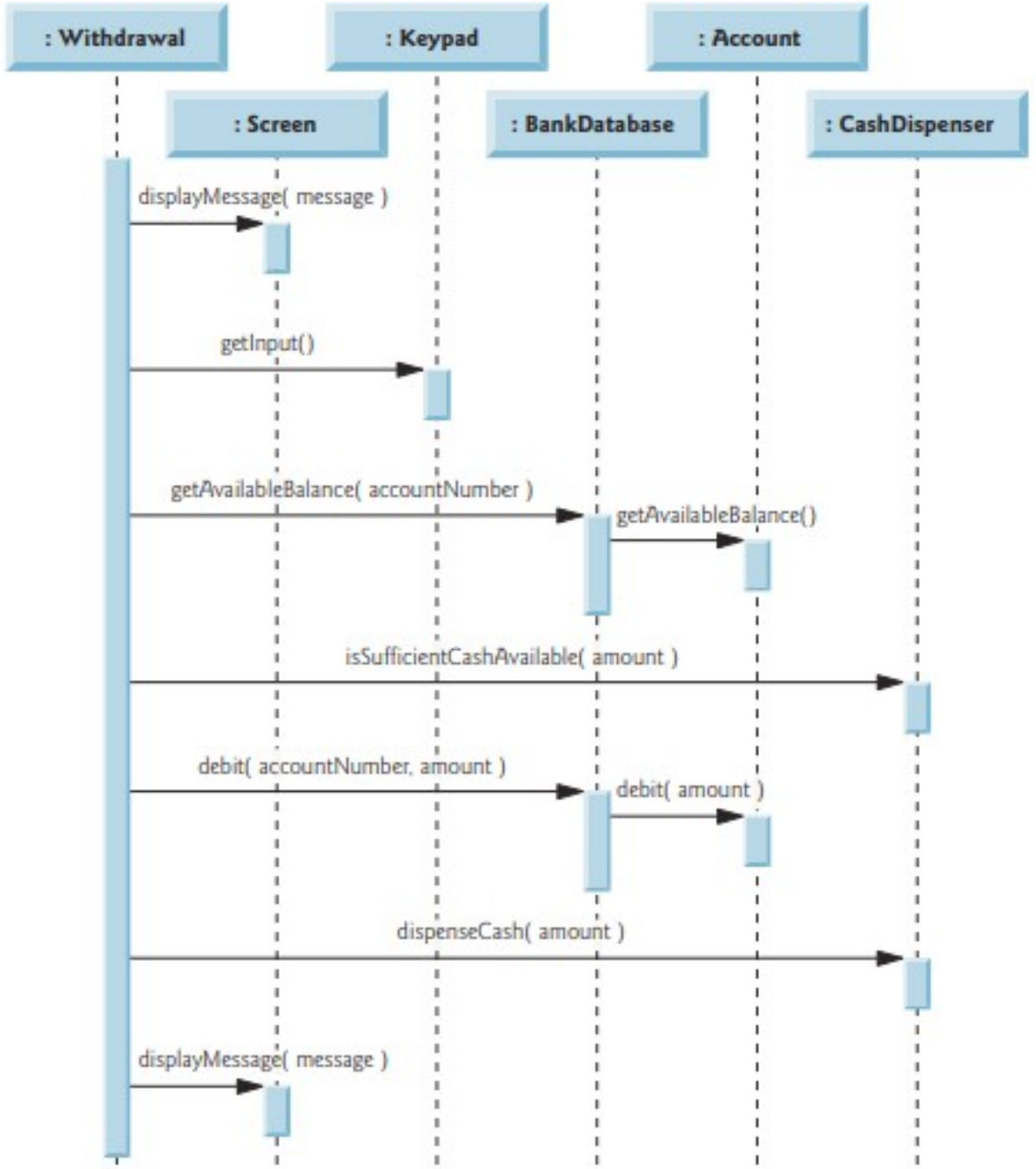
Collaborations in ATM system

Object of Class	Send the message	to an object of class
ATM	• displayMessage • getInput • authenticateUser • execute • execute • execute	• Screen • Keypad • BankDatabase • BalanceInquiry • Withdrawal • Deposit
BalanceInquiry	• getAvailableBalance • getTotalBalance • displayMessage	• BankDatabase • BankDatabase • Screen
Withdraw	• displayMessage • getInput • getAvailableBalance • isSufficientCashAvailabl • debit • dispenseCash	• Screen • Keypad • BankDatabase • CashDispenser • BankDatabase • CashDispenser

Collaborations in ATM system

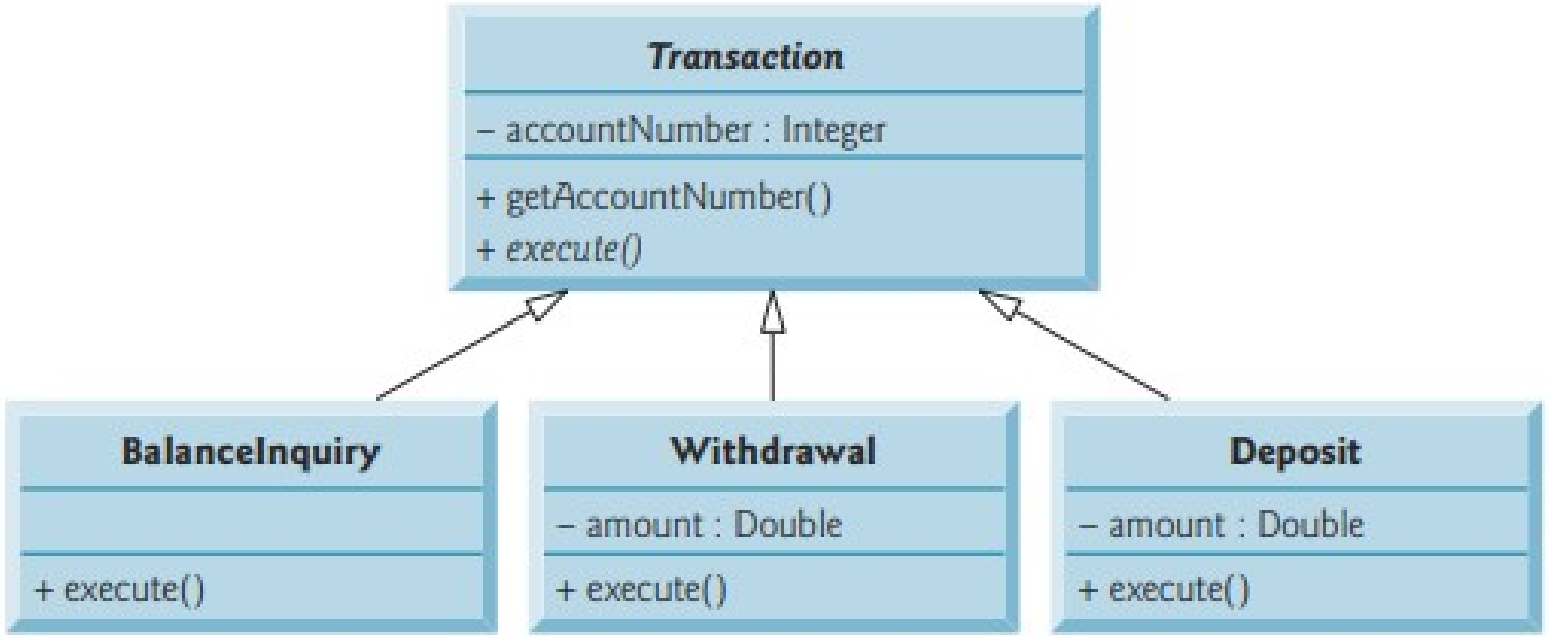
Object of Class	Send the message	to an object of class
Deposit	• displayMessage • getInput • isEnvelopeReceived • credit	• Screen • Keypad • DepositSlot • BankDatabase
BankDatabase	• validatePIN • getAvailableBalance • getTotalBalance • debit • credit	• Account • Account • Account • Account • Account

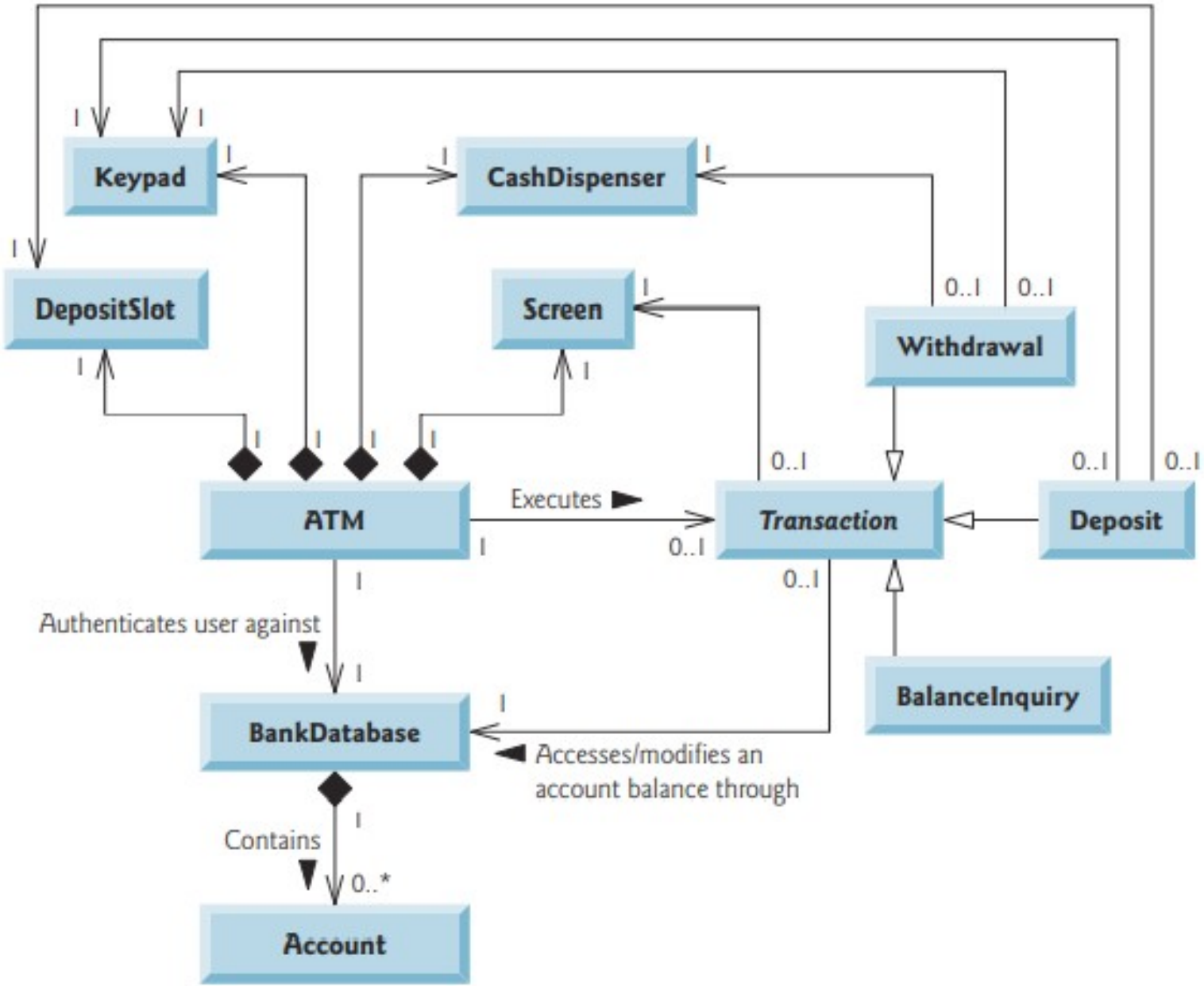
Sequence Diagram



INCORPORATING INHERITANCE & POLYMORPHISM

Incorporating inheritance & polymorphism




Keypad
CashDispenser
DepositSlot
Screen
Withdrawal
ATM
Executes
Transaction
Deposit
Authenticates user against
BalanceInquiry
BankDatabase
Accesses/modifies an account balance through
Contains
Account
1
0..1
0..*

HAVE A GOOD DAY!