OOP Design dengan UML

Studi Kasus ATM

Objectives

- Memahami kebutuhan perancangan dan mendokumentasikannya (*requirements documentation*)
- Mengidentifikasikan Class dan atributnya berdasarkan *requirements*.
- Mengidentifikasikan status, aktivifitas, dan operasi objek berdasarkan *requirements*.
- Menentukan kolaborasi antar objek dalam system
- Diagram UML: usecase, class, state, activity, communication, dan sequence.

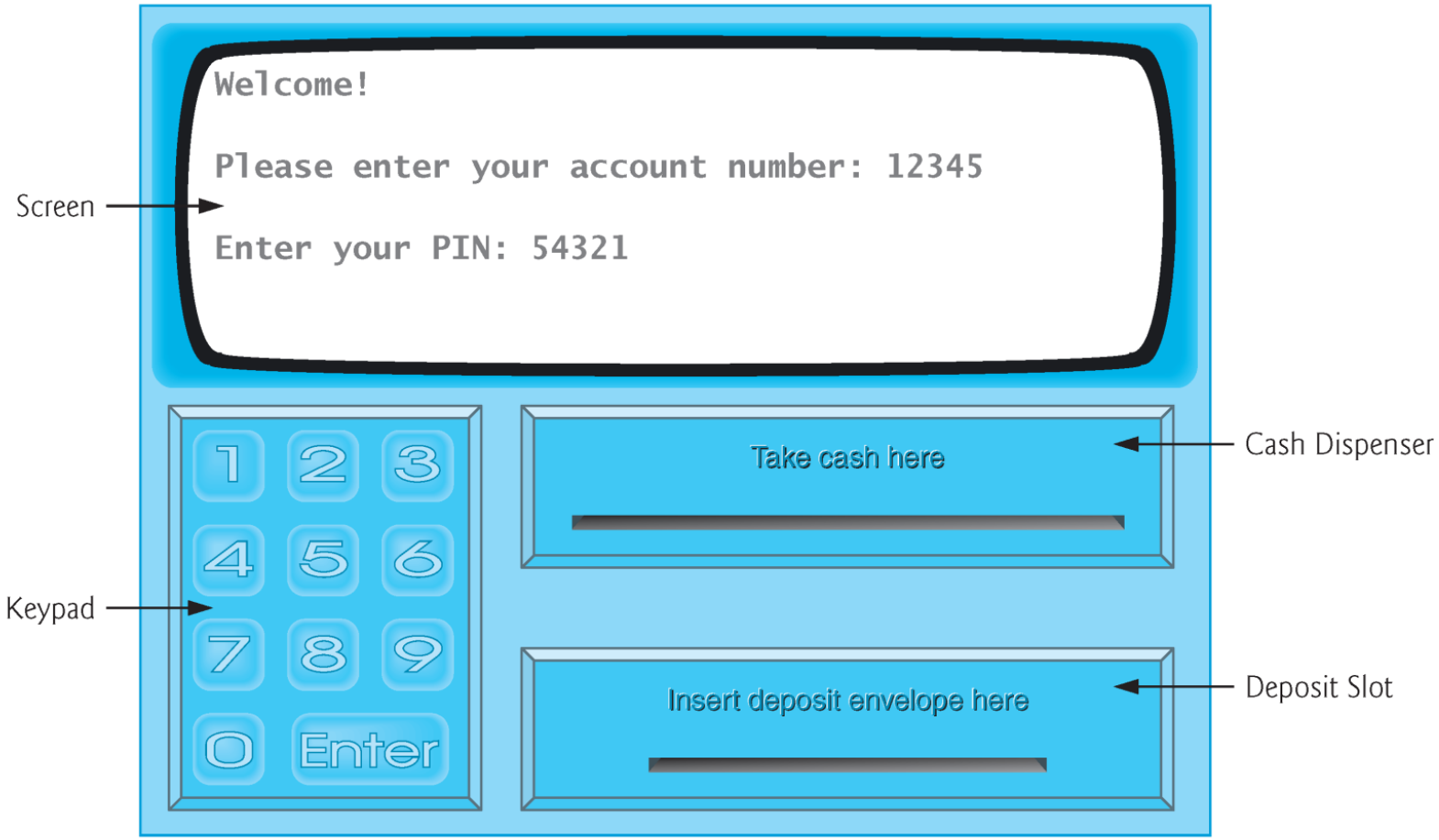


Fig. 25.1 | Automated teller machine user interface.

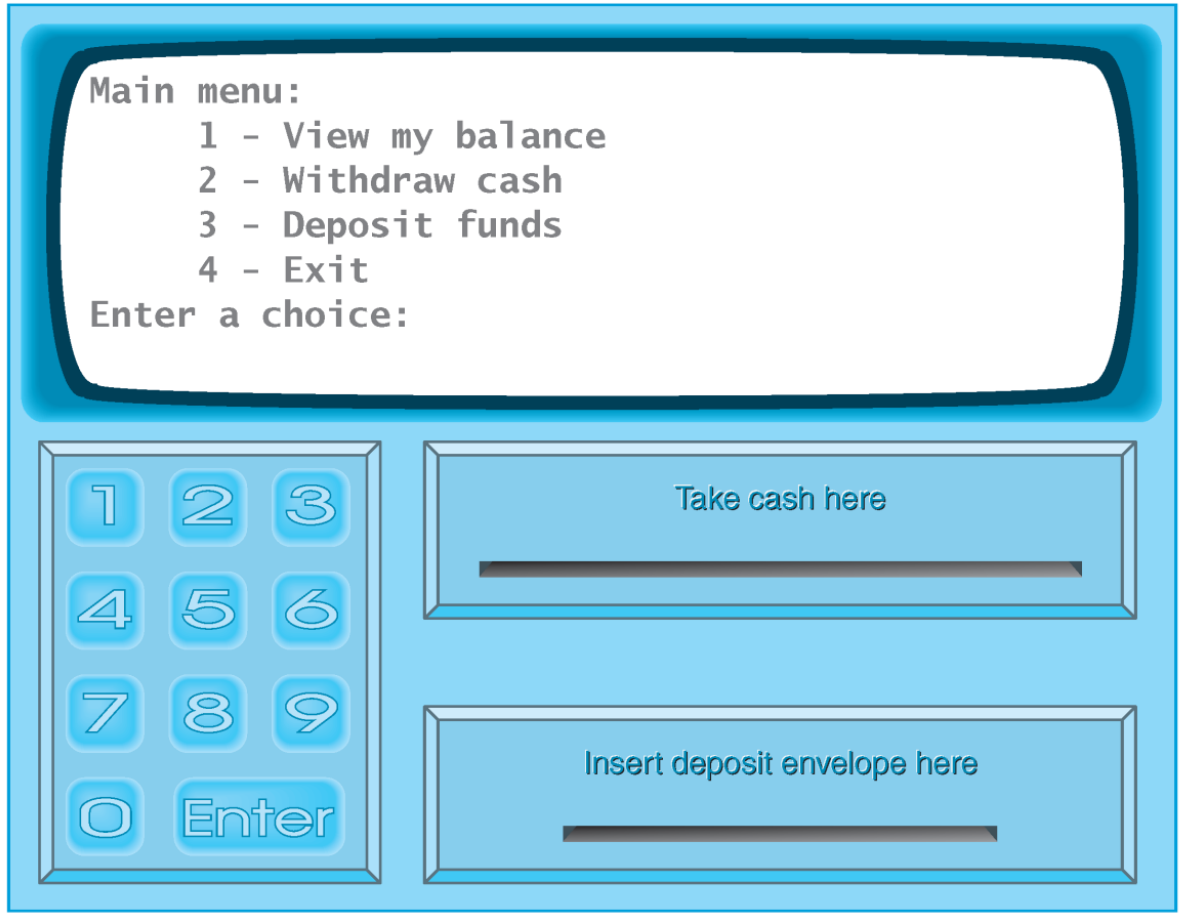


Fig. 25.2 | ATM main menu.

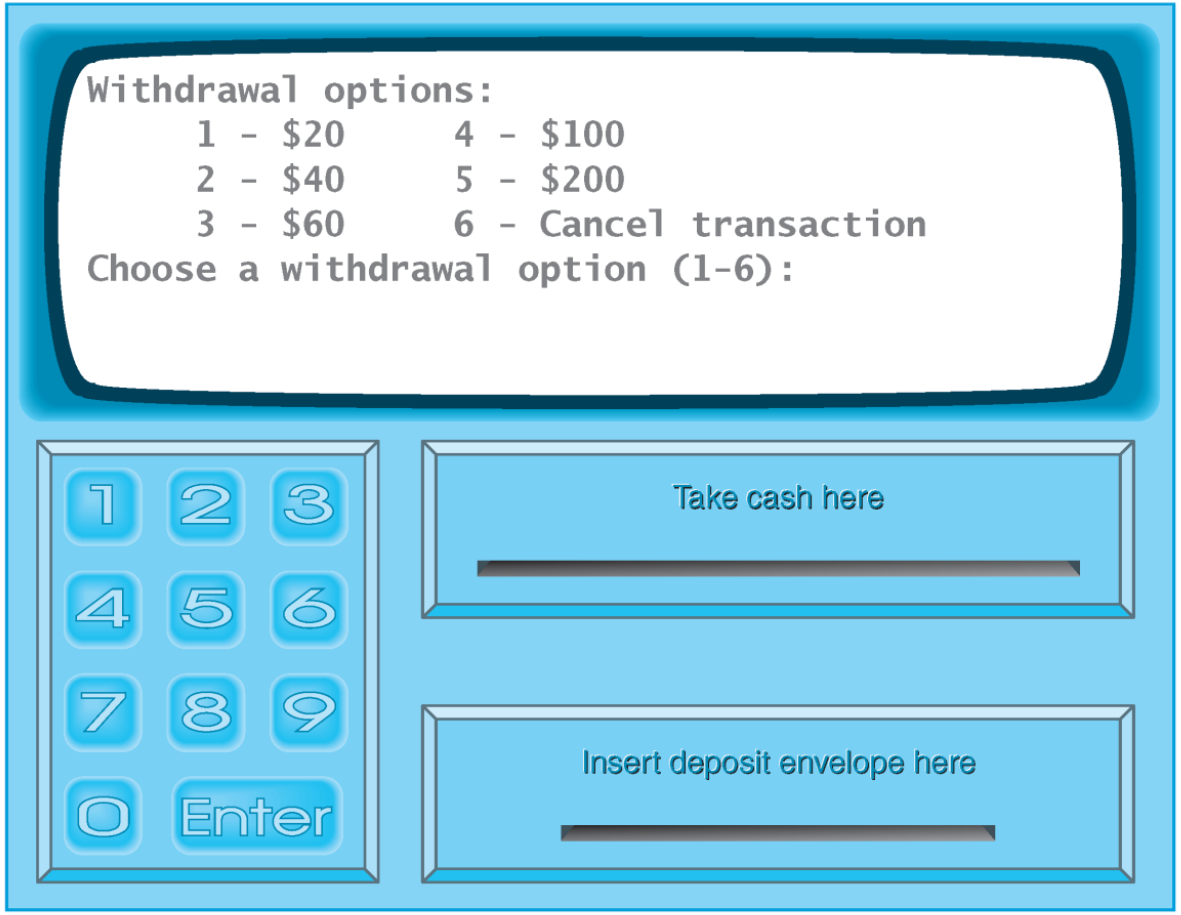


Fig. 25.3 | ATM withdrawal menu.

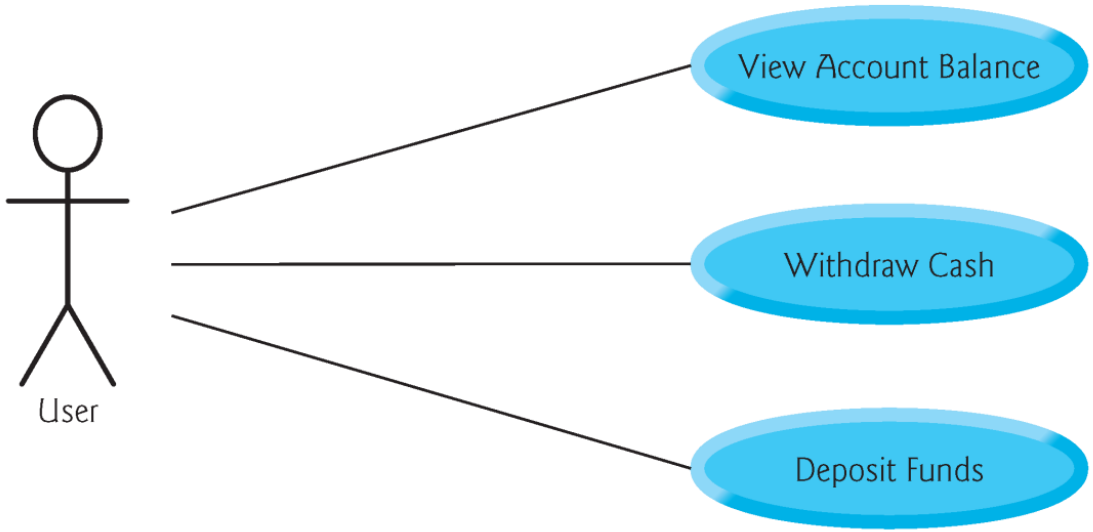


Fig. 25.4 | Use case diagram for the ATM system from the User's perspective.

Nouns and noun phrases in the requirements document			
bank	money / fund	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

Fig. 25.5 | Nouns and noun phrases in the requirements document.

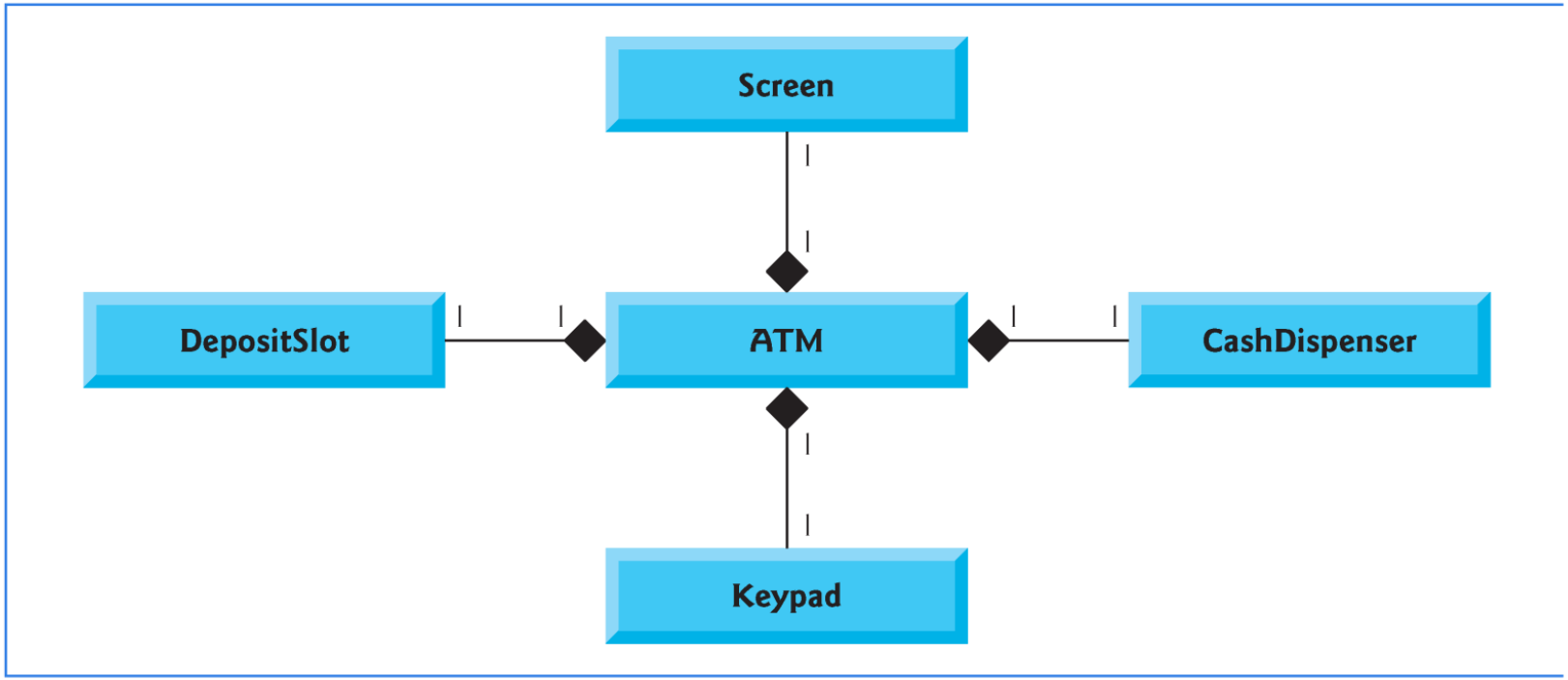
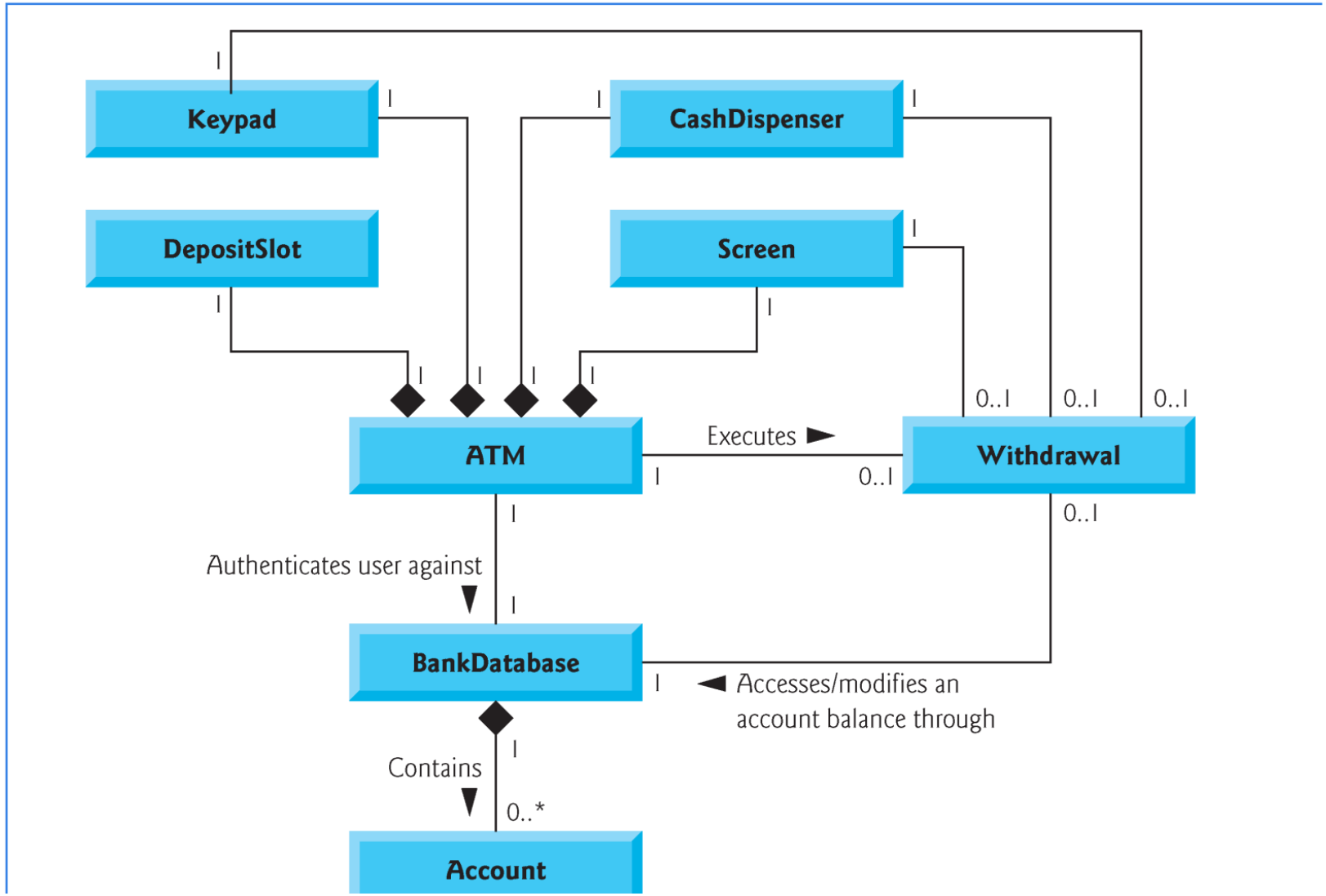


Fig. 25.9 | Class diagram showing composition relationships.


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

Class	Descriptive words and phrases
`ATM`	user is authenticated
`BalanceInquiry`	account number
`Withdrawal`	account number amount
`Deposit`	account number amount
`BankDatabase`	[no descriptive words or phrases]
`Account`	account number PIN balance
`Screen`	[no descriptive words or phrases]
`Keypad`	[no descriptive words or phrases]
`CashDispenser`	begins each day loaded with 500 $20 bills
`DepositSlot`	[no descriptive words or phrases]

Fig. 25.11 | Descriptive words and phrases from the ATM requirements.

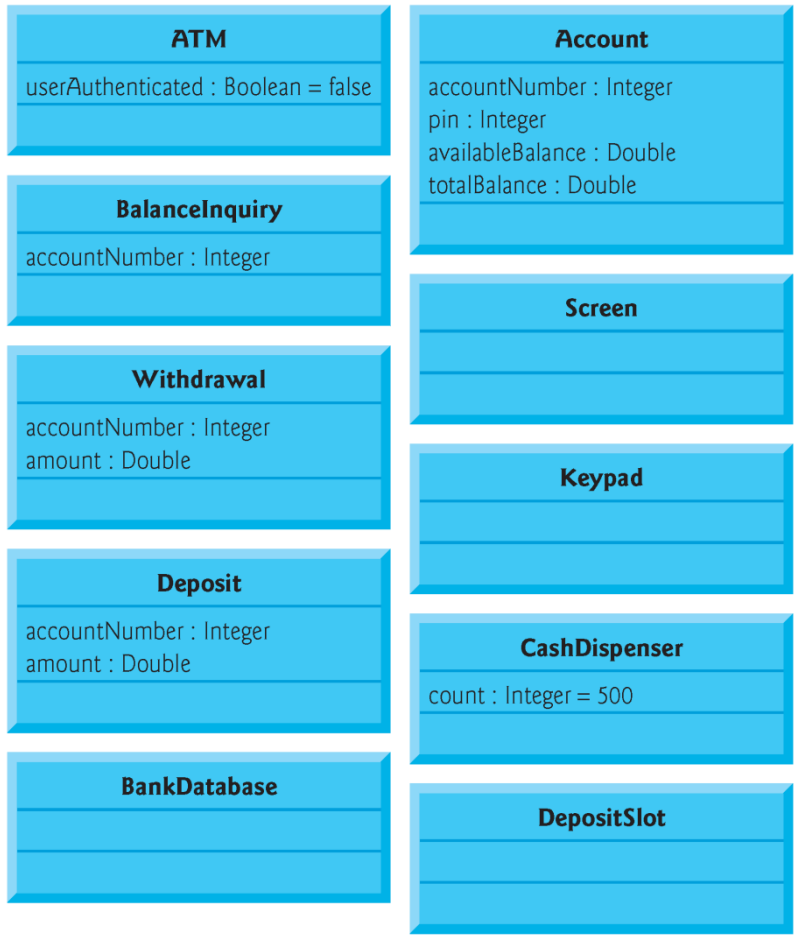


Fig. 25.12 | Classes with attributes.

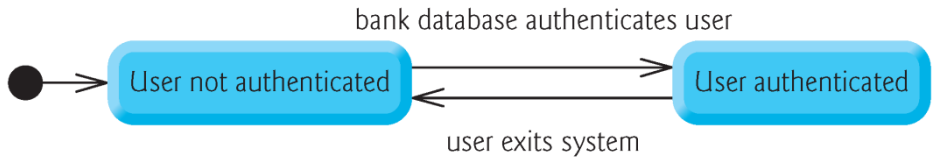


Fig. 25.13 | State diagram for the ATM object.

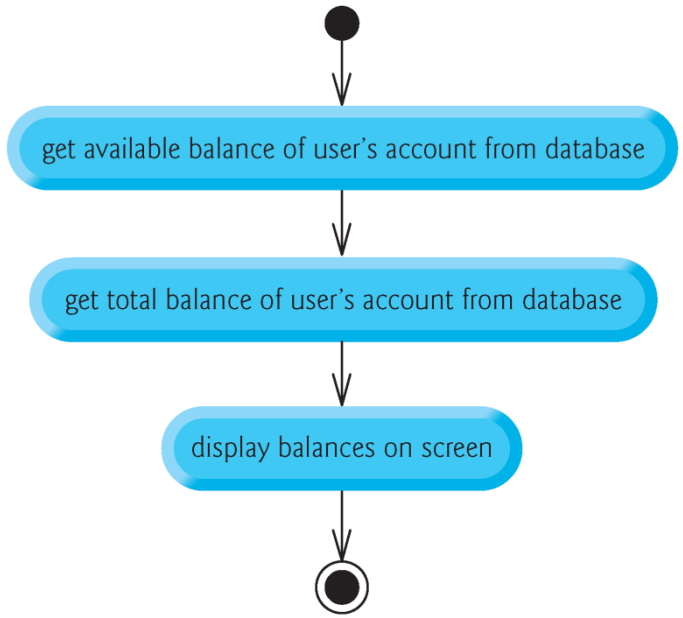


Fig. 25.14 | Activity diagram for a `BalanceInquiry` transaction.

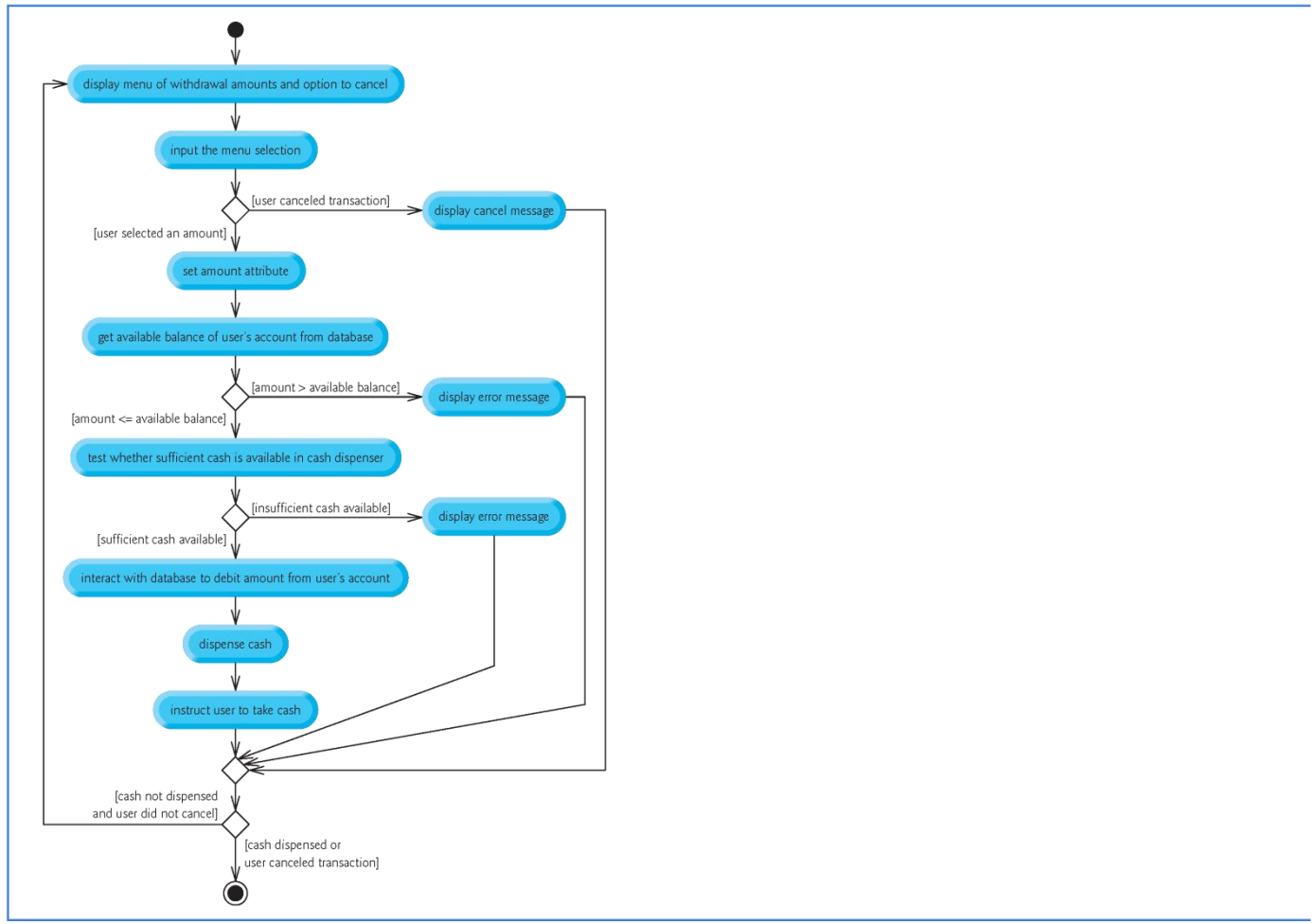


Fig. 25.15 | Activity diagram for a `Withdrawal` transaction.

Class	Verbs and verb phrases
`ATM`	executes financial transactions
`BalanceInquiry`	[none in the requirements document]
`Withdrawal`	[none in the requirements document]
`Deposit`	[none in the requirements document]
`BankDatabase`	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
`CashDispenser`	dispenses cash, indicates whether it contains enough cash to satisfy a withdrawal request
`DepositSlot`	receives a deposit envelope

Fig. 25.16 | Verbs and verb phrases for each class in the ATM system.

Fig. 25.17 | Classes in the ATM system with attributes and

An object of class…	sends the message…	to an object of class…
ATM	displayMessage	Screen
	getInput	Keypad
	authenticateUser	BankDatabase
	execute	BalanceInquiry
	execute	Withdrawal
	execute	Deposit
BalanceInquiry	getAvailableBalance	BankDatabase
	getTotalBalance	BankDatabase
	displayMessage	Screen
Withdrawal	displayMessage	Screen
	getInput	Keypad
	getAvailableBalance	BankDatabase
	isSufficientCashAvailable	CashDispenser
	debit	BankDatabase
	dispenseCash	CashDispenser

Fig. 25.22 | Collaborations in the ATM system. (Part 1 of 2.)

An object of class…	sends the message…	to an object of class…
Deposit	displayMessage	Screen
	getInput	Keypad
	isEnvelopeReceived	DepositSlot
	credit	BankDatabase
BankDatabase	validatePIN	Account
	getAvailableBalance	Account
	getTotalBalance	Account
	debit	Account
	credit	Account

Fig. 25.22 | Collaborations in the ATM system. (Part 2 of 2.)

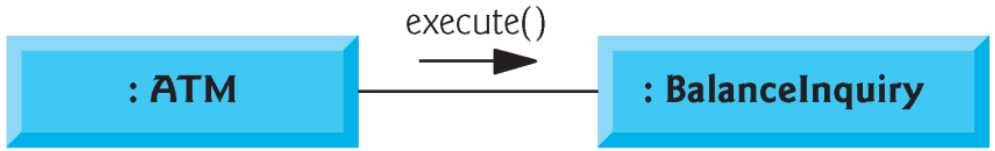


Fig. 25.23 | Communication diagram of the ATM executing a balance inquiry.

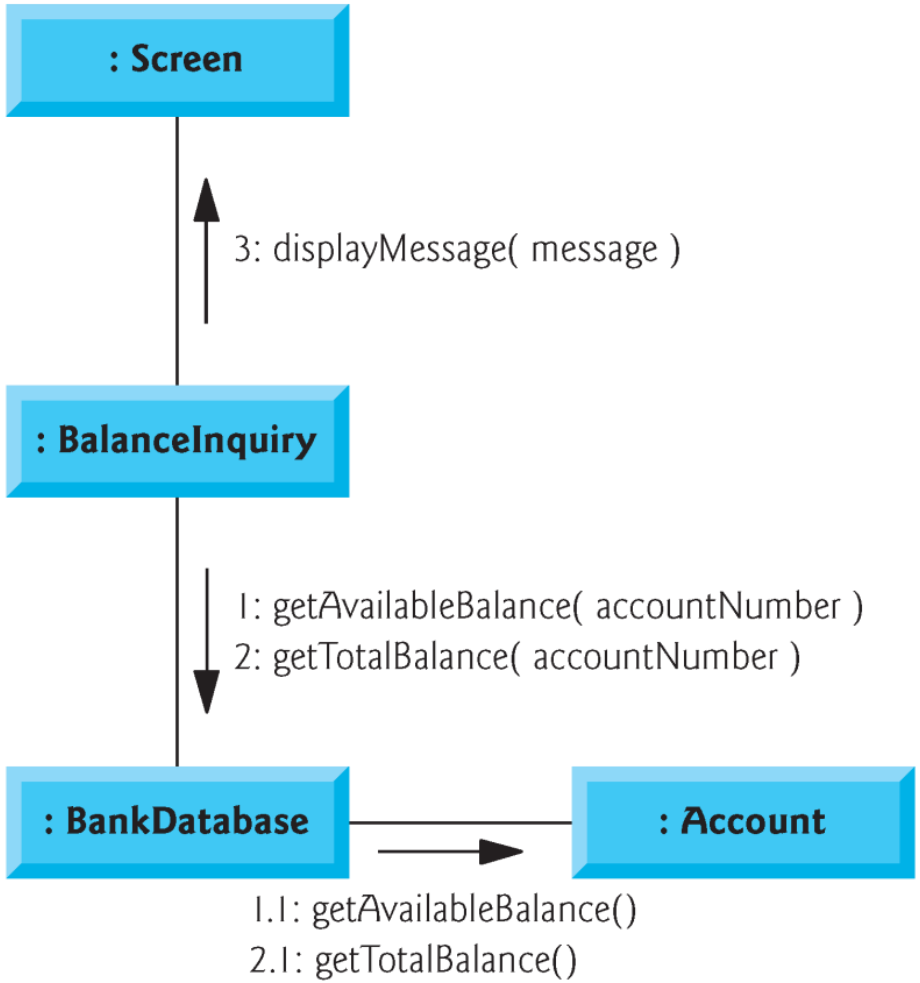


Fig. 25.24 | Communication diagram for executing a balance inquiry.

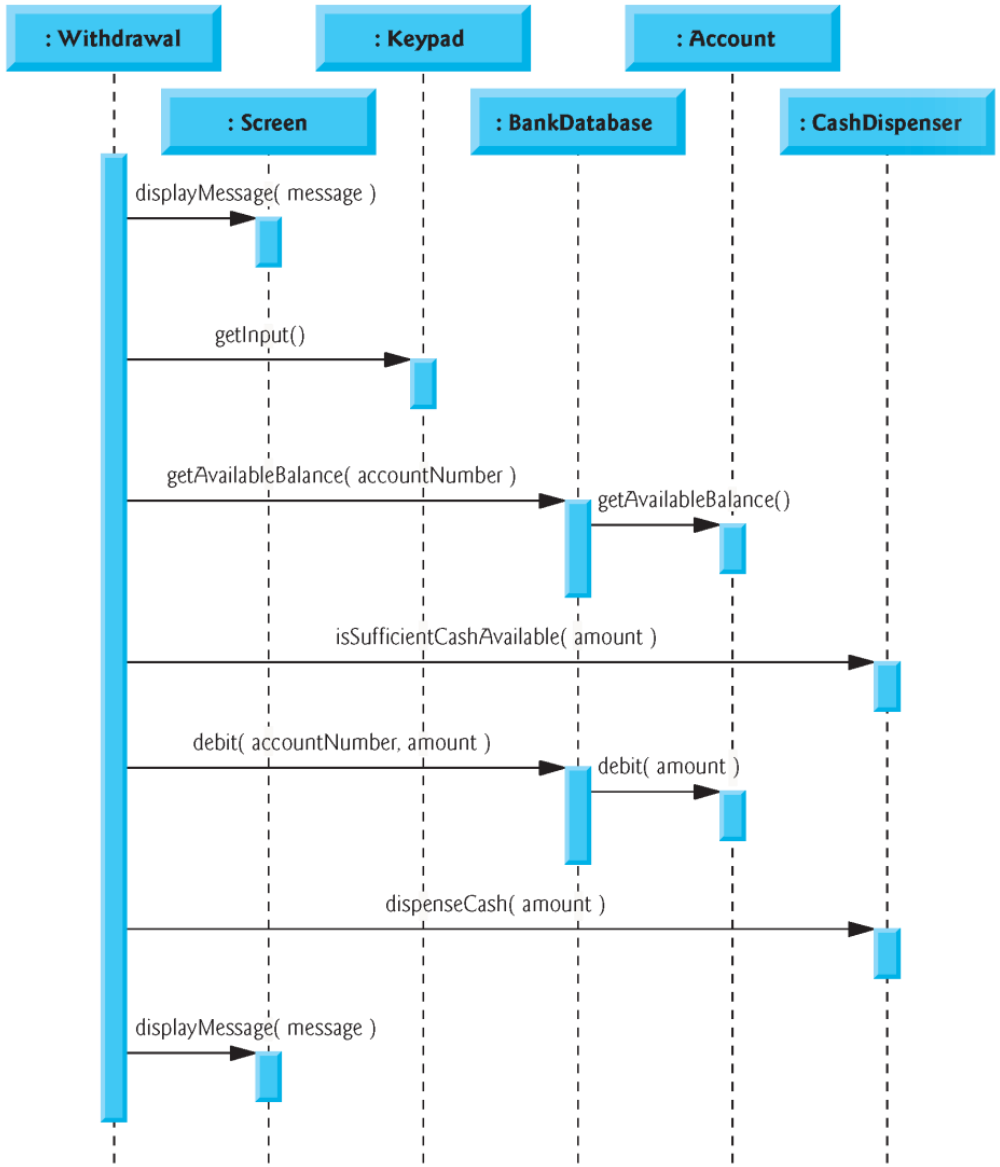


Fig. 25.25 | Sequence diagram that models a `Withdrawal` executing.

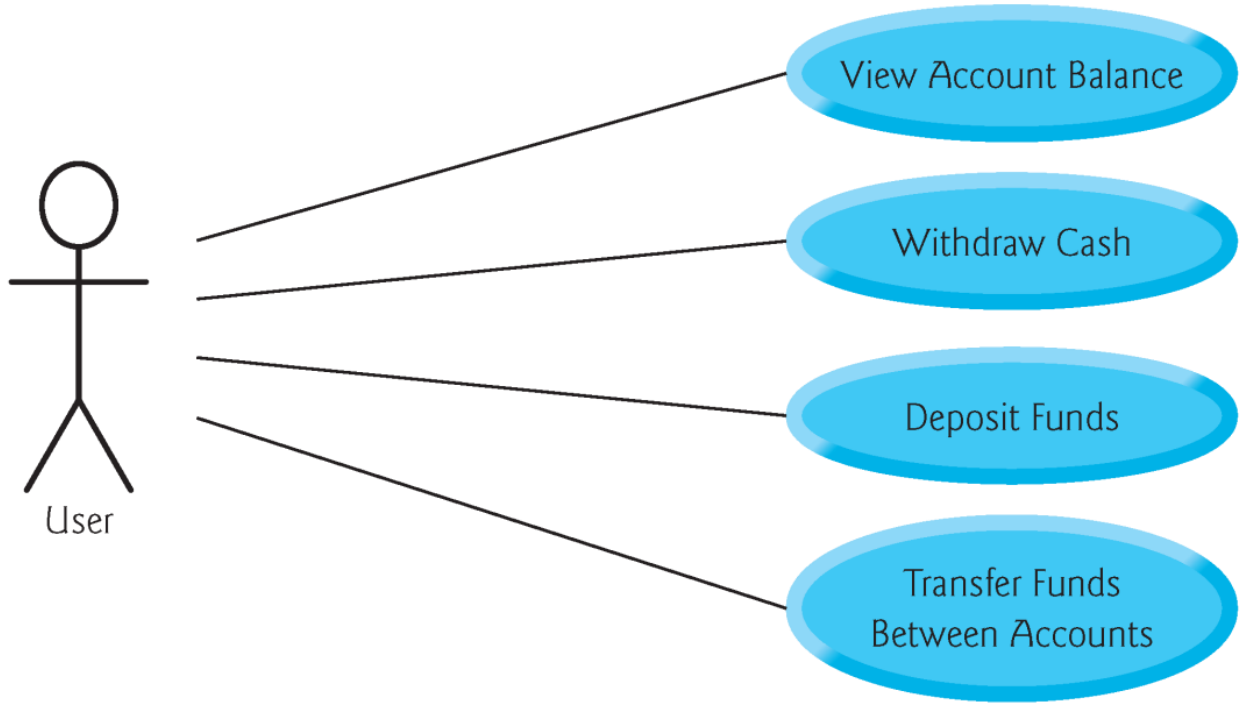


Fig. 25.26 | Use case diagram for a modified version of our ATM system that also allows users to transfer money between accounts.

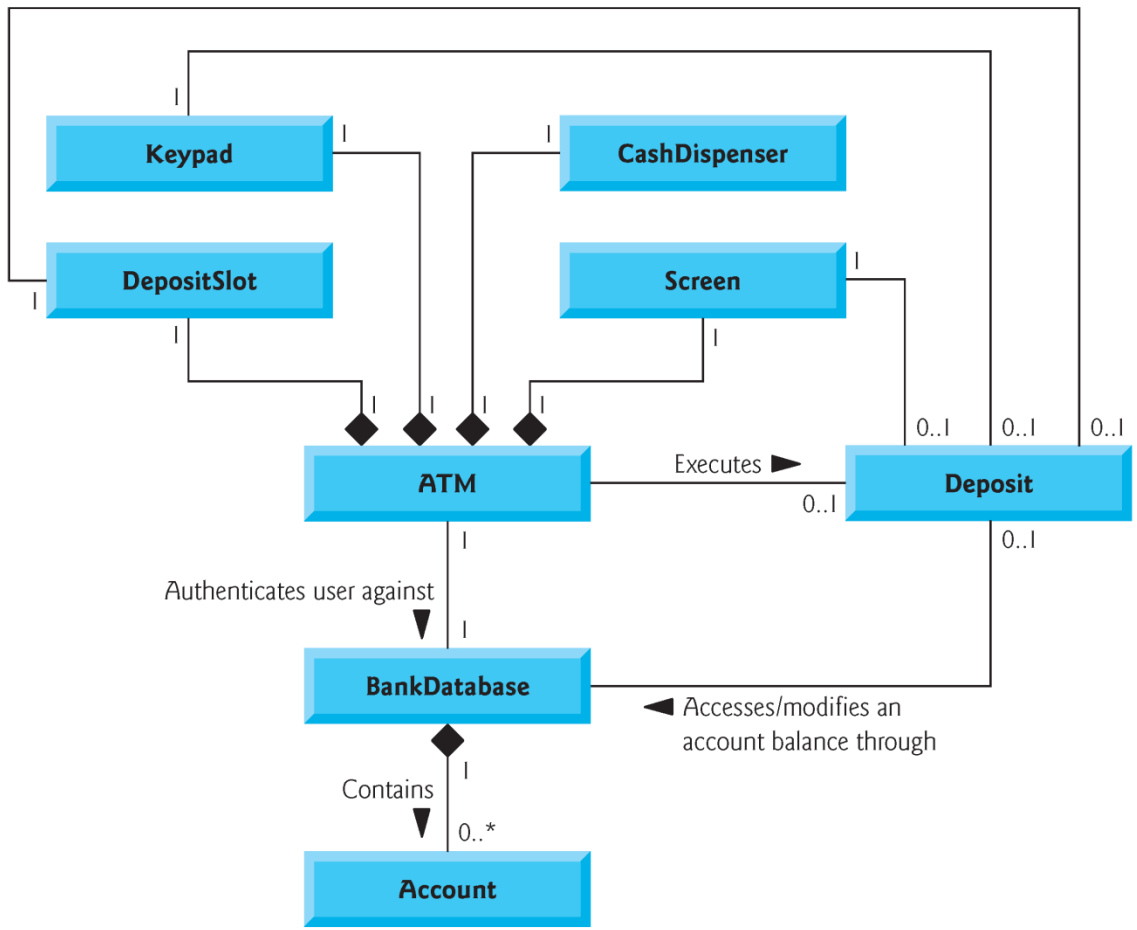


Fig. 25.28 | Class diagram for the ATM system model including

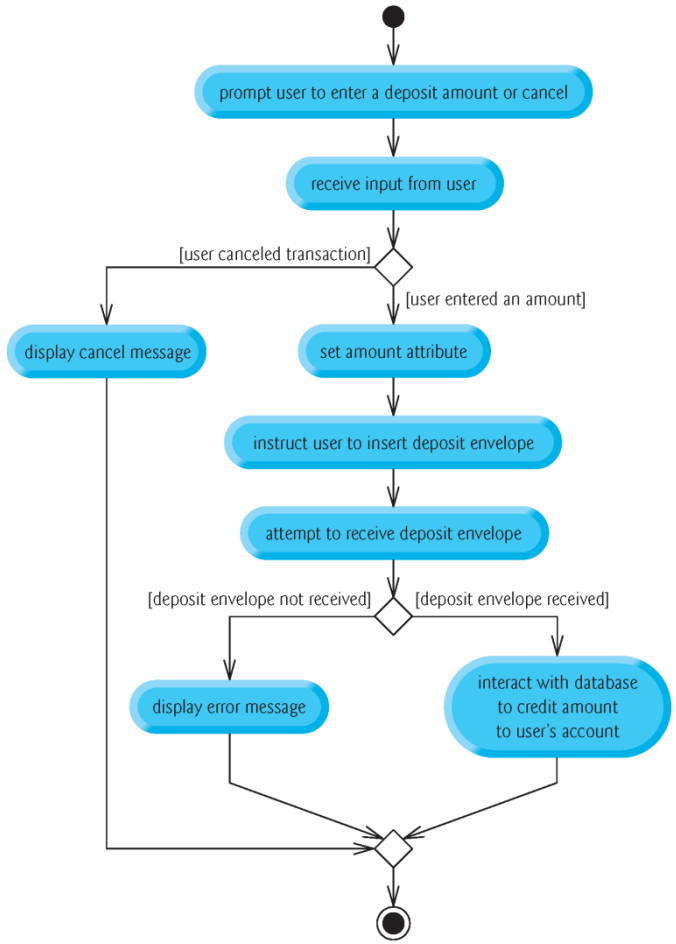


Fig. 25.29 | Activity diagram for a `Deposit` transaction.

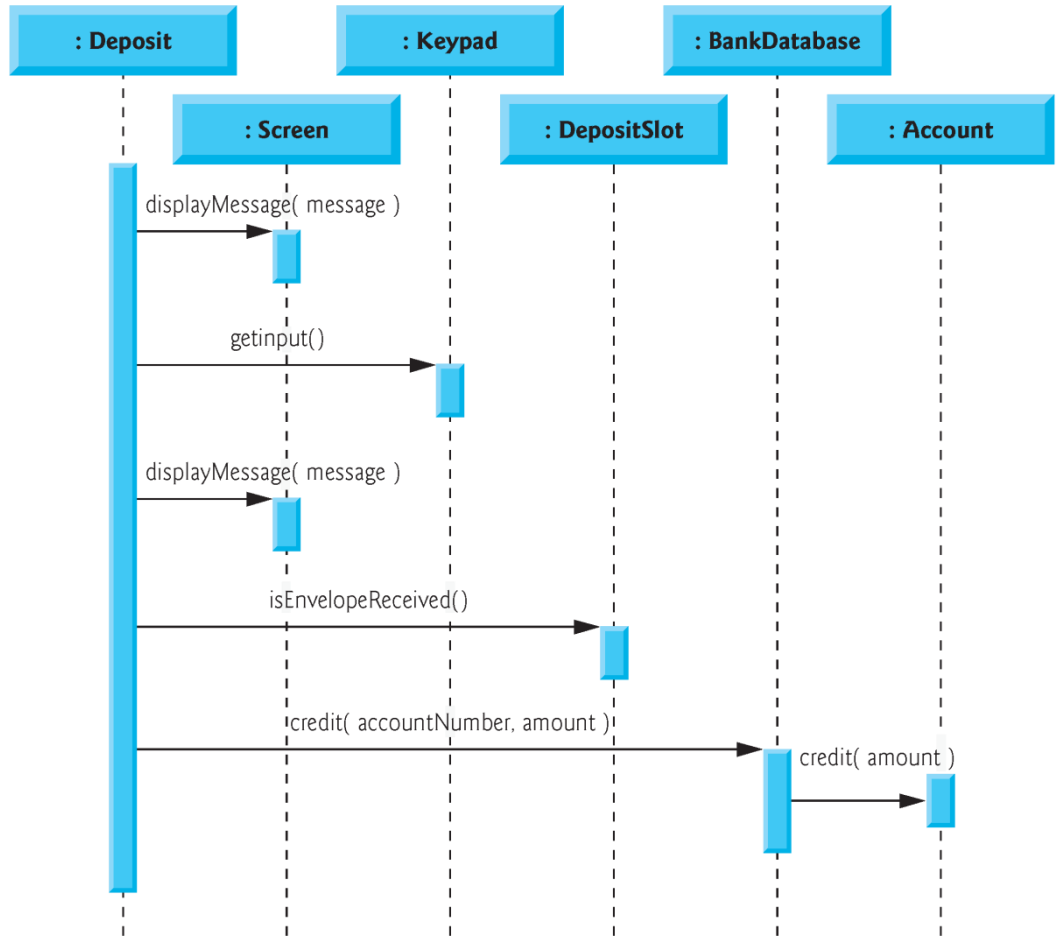


Fig. 25.30 | Sequence diagram that models a `Deposit` executing.