

A New Reliable ATM

OOPT Phase 2040

Design

Project Team T6

201411140 권성완

201511247 김선정

201510436 허윤아

201510285 조수빈

Date

2018-05-13

Activity 2041. Design Real Use Cases

Use Case	1. Deposit
Actor	Customer
Purpose	Deposit cash into account by bankbook or check card, or credit card
Overview	Customer inputs card or bankbook, and cash, to deposit cash into account or credit card.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R2.1, R2.3, R3.2, R4.1 Use case : Print Transaction Receipt, Print Error, Insert, Update Server Information
Pre-Requisites	N/A
UI Widget	Window 1, 2-1, 2-2
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer selects deposit menu from the basic screen. 2. (S) Prints out "Input check card or bankbook or credit card". 3. (A) Customer inputs credit card, check card or bankbook. 4. (S) Invoke "Insert". If user information is valid, ask customer to input cash. 5. (A) Customer Inputs cash in unit of 10000₩ and 50000₩. 6. (S) Check amount of cash inputted. 7. (S) If total amount of money is correct, invoke "Print transaction receipt". 8. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4 : If customer insert credit card, view loan record. Line 5 : If cash is not in unit of 10000₩ and 50000₩, notice error. Line 7 : If total amount of cash is incorrect, notice error. If error occurs over 3 times, invoke "Forced Termination". If loan record exists, loan is automatically repaid.

Use Case	2. Withdraw
Actor	Customer
Purpose	Withdraw cash from bank account
Overview	Customer inputs check card or bankbook to withdraw cash from bank account.

Type	Primary and Essential
Cross Reference	Functional Requirements : R1.2, R2.1, R2.3, R2.4, R3.1, R3.2, R3.3 Use case : Print Transaction Receipt, Print Error, Forced Termination, Request Customer's Data, Insert, Check Password
Pre-Requisites	Customer should know password for the account, and balance should be enough to withdraw.
UI Widget	Window 1, 2-1, 4, 6
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer selects withdraw menu from the basic screen. 2. (S) Prints out "Input check card or bankbook". 3. (A) Customer inputs check card or bankbook. 4. (S) Invoke "Insert". If user information is valid, ask customer to input amount of money to withdraw. 5. (A) Input amount of money in unit of 10000₩ and 50000₩ to withdraw from account. 6. (S) Ask for password for the account. 7. (A) Input password for the account. 8. (S) Invoke "Check Password". If password is correct, count numbers of bills. 9. (S) Invoke "Request Customer's Data". If balance is enough, withdraw cash and invoke "Print Transaction Receipt". 10. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 3, 8 : If information or password is incorrect over three times, invoke "Forced Termination". Line 5 : If total amount of money to withdraw is over 50000₩, customer inputs number of 50000₩ bill. Line 9 : If balance is not enough, notice error.

Use Case	3. Deposit without Bankbook
Actor	Customer
Purpose	Deposit cash into account without bankbook or check card
Overview	Customer deposits cash without bankbook or check card.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.3, R2.1, R2.3, R2.4, R3.1, R4.1 Use Case : Print Transaction Receipt, Print Error, Forced Termination, Request Customer's Data, Update Server

	Information
Pre-Requisites	Customer should know exact account number to deposit.
UI Widget	Window 1, 3, 4
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer chooses Deposit without Bankbook menu from screen. 2. (S) Ask customer to input bank account number to deposit. 3. (A) Input bank account number. 4. (S) If bank account number is valid, ask customer to input cash in unit of 10000₩ and 50000₩. 5. (S) Check total amount of cash. 6. (S) If counted right, invoke "Print Transaction Receipt". 7. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4 : if invalid, invoke "Print Error". If invalid over 3 times, invoke "Forced Termination"

Use Case	4. Transfer
Actor	Customer
Purpose	Transfer money from customer's account to another account
Overview	Customer transfers money from own account to another.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.4, R2.1, R2.2, R2.3, R2.4, R3.1, R3.2, R3.3, R4.1 Use case : Print Transaction Receipt, Take Charge, Print Error, Forced Termination, Request Customer's Data, Insert, Check Password, Update Server Information
Pre-Requisites	Customer should know password for the account.
UI Widget	Window 1, 2-1, 2-2, 3, 4, 6
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer selects transfer menu from the basic screen. 2. (S) prints out "Input check card or bankbook". 3. (A) Customer inputs check card or bankbook. 4. (S) Invoke "Insert". If user information is valid, ask customer to input bank and account number to transfer money. 5. (A) Customer inputs bank and account number to transfer money.

	<p>6. (S) If inputted information is valid, ask customer to input amount of money to transfer.</p> <p>7. (A) Customer inputs amount of money to transfer.</p> <p>8. (S) Invoke "Request Customer's Data". If balance is enough, ask for password for the account.</p> <p>9. (A) Customer inputs password for the account.</p> <p>10. (S) Invoke "Check Password". If password is correct, invoke "Print Transaction Receipt".</p> <p>11. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>Line 3, 10 : If information or password is incorrect over 3 times, invoke "Forced Termination".</p> <p>Line 5 : If customer inputted different bank's account, invoke "Take Charge".</p>

Use Case	5. Exchange
Actor	Customer
Purpose	Exchange KRW into foreign currency
Overview	Customer exchanges KRW in account into foreign currency.
Type	Primary and Essential
Cross Reference	<p>Functional Requirements : R1.5, R2.1, R2.2, R2.3, R2.4, R3.1, R3.2, R3.3, R4.1</p> <p>Use Case : Print Transaction Receipt, Take Charge, Print Error, Forced Termination, Request Customer's Data, Insert, Check Password, Update Server Information</p>
Pre-Requisites	Customer should know password for the account. ATM only handles unit of Yen, RMB, Dollar, Euro.
UI Widget	Window 1, 2-1, 4, 5, 6
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <p>1. (A) Customer selects exchange menu from basic screen.</p> <p>2. (S) Print out "input check card or bankbook".</p> <p>3. (A) Customer inputs check card or bankbook.</p> <p>4. (S) Invoke "Insert". If user information is valid, print out list of countries available.</p> <p>5. (A) Customer selects country to exchange money.</p> <p>6. (S) Print out "input amount of money to exchange".</p> <p>7. (A) Customer inputs amount of money to exchange.</p>

	<p>8. (S) Calculate total amount of money based on exchange rate and ask for password.</p> <p>9. (A) Customer inputs password.</p> <p>10. (S) Invoke "Check Password". If password is correct, withdraw cash in foreign currency.</p> <p>11. (S) If withdrew correctly, invoke "Print Transaction Receipt".</p> <p>12. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>Line 3, 9 : If information or password is incorrect over 3 times, invoke "Forced Termination".</p> <p>Line 10 : Invoke "Take Charge". Charge is deducted from balance.</p>

Use Case	6. Loan
Actor	Customer
Purpose	Loan cash by credit card
Overview	Customer loans cash using credit card. There is loan limit.
Type	Primary and Essential
Cross Reference	<p>Functional Requirements : R1.6, R2.1, R2.2, R2.3, R2.4, R3.1, R3.2, R3.3, R3.4, R4.1</p> <p>Use Case : Print Transaction Receipt, Take Charge, Print Error, Forced Termination, Request Customer's Data, Insert, Check Password, Check Credit, Update Server Information</p>
Pre-Requisites	Customer should know password for the credit card.
UI Widget	Window 1,4,6
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <p>1. (A) Customer chooses loan menu from basic screen.</p> <p>2. (S) Print out "Input credit card".</p> <p>3. (A) Customer inputs credit card.</p> <p>4. (S) Invoke "Insert". If user information is valid, Invoke "Check Credit". Print out "input amount of money to loan".</p> <p>5. (A) Customer inputs amount of money to loan in unit of 10000₩ and 50000₩.</p> <p>6. (S) print out "input password for the credit card".</p> <p>7. (A) Customer inputs password for the credit card.</p> <p>8. (S) Invoke "Check Password". If password is correct, count number of bills.</p> <p>9. (S) Invoke "Request Customer's Data". If total amount of</p>

	<p>money to loan is under credit card limit, withdraw cash and invoke "Print Transaction Receipt".</p> <p>10. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>Line 5 : If total amount of money to withdraw is over 50000₩, customer inputs number of 50000₩ bill.</p> <p>Line 4, 8 : If information or password is incorrect over 3 times, invoke "Forced Termination".</p> <p>Line 9 : : Invoke "Take Charge".</p>

Use Case	7. Check balance
Actor	Customer
Purpose	Check balance of account
Overview	Customer checks balance of account.
Type	Primary and Essential
Cross Reference	<p>Functional Requirements : R1.7, R2.3, R2.4, R3.1, R3.2, R3.3</p> <p>Use Case : Print Error, Forced Termination, Request Customer's Data, Insert, Check Password</p>
Pre-Requisites	Customer should know password for the account.
UI-Widget	Window 1,2-1,4
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> 1. (A) Customer chooses check balance menu from basic screen. 2. (S) Print out "input check card or bankbook". 3. (A) Customer inputs check card or bankbook. 4. (S) Invoke "Insert". If user information is valid, print out "input password for the account". 5. (A) Customer inputs password for the account. 6. (S) Invoke "Check Password". If password is correct, print out recent transactional information(under 100) and balance after each transactional process. 7. (A) Customer input 'OK' button. 8. (S) Return to basic screen.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4, 6 : If information or password is incorrect over 3 times, invoke "Forced Termination".

Use Case	8. Pay Utility Bill
Actor	Customer
Purpose	Pay utility bill by giro bill
Overview	Customer pay utility bill by giro bill and account.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.8, R2.1, R2.3, R2.4, R3.1, R3.2, R3.3, R4.1 Use Case : Print Transaction Receipt, Print Error, Forced Termination, Request Customer's Data, Insert, Check Password, Update Server Information
Pre-Requisites	Customer should know password for the account, and have giro bill.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer chooses pay utility bill menu from basic screen. 2. (S) Print out "input giro bill". 3. (A) Customer inputs giro bill. 4. (S) Invoke "Insert". If giro bill is valid, print out "input check card or bankbook". 5. (A) Customer inputs check card or bankbook. 6. (S) Invoke "Insert". If user information is valid, print out "input password for the account". 7. (A) Customer inputs password for the account. 8. (S) Invoke "Check Password". If password is correct, invoke "Request Customer's Data". 9. (S) If balance is enough to pay utility bill, invoke "Print Transaction Receipt". 10. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.
Alternative Courses of Events	N/A
UI Widget	Window 1, 2-1, 2-2, 4
Exceptional Courses of Events	Line 4, 6, 8 : If information or password is incorrect over 3 times, invoke "Forced Termination". Line 9 : Is balance is not enough, notice error.

Use Case	9. Print Transaction Receipt
Actor	(None)
Purpose	Check if transaction is successfully finished by printing out Transaction Receipt

Overview	System prints out transaction receipt to check if transaction is successfully finished.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.8, R2.1, R2.3, R4.1 Use Case : Deposit, Withdraw, Deposit without Bankbook, Transfer, Exchange, Loan, Pay Utility Bill, Print Error, Update Server Information
Pre-Requisites	Transaction ends
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Transaction ends successfully. 2. (S) Print Transaction Receipt.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1 : If doesn't end successfully, notice error.

Use Case	10. Take Charge
Actor	(None)
Purpose	Take charge during transaction
Overview	System takes charge when customer transfers, exchanges, loans.
Type	Secondary
Cross Reference	Functional Requirements : R1.4, R1.5, R1.6, R2.2, R3.4, R4.1 Use Case : Transfer, Exchange, Loan, Check Credit, Update Server Information
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) During transferring, exchanging, and loaning take charge. 2. (S) During loaning, invoke "Check Credit". If credit rating is high, this use case is ignored.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	11. Print Error
Actor	(None)
Purpose	Print error during transaction
Overview	System prints out various error messages during transaction.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7,

	R1.8, R2.3, R2.4, R3.1, R3.2 Use Case : Deposit, Withdraw, Deposit without Bankbook, Transfer, Exchange, Loan, Check Balance, Pay Utility Bill, Forced Termination, Request Customer's Data, Insert
Pre-Requisites	Error occurred during transaction
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) This use case occurs if information Customer inputted is different from information in server. 2. (S) If this use case occurs more than 3 times, invoke "Forced Termination".
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	12. Forced Termination
Actor	(None)
Purpose	End transaction when error occurs more than 3 times
Overview	System ends transaction automatically when error except password error occurs more than 3 times.
Type	Primary and Essential
Cross Reference	Functional Requirements : R2.3, R2.4, R2.5, R3.3 Use Case : Print Error, Transaction Lock, Check Password
Pre-Requisites	Error occurred more than 3 times
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) This use case occurs if error occurred more than 3 times.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1 : If password error occurred more than 3 times, invoke "Transaction Lock".

Use Case	13. Transaction Lock
Actor	(None)
Purpose	Lock transaction by account or credit card when password error occurs more than 3 times
Overview	System locks transaction by account or credit card when password error occurs more than 3 times.
Type	Primary and Essential
Cross Reference	Functional Requirements : R2.3, R2.4, R3.3, R4.1 Use Case : Print Error, Transaction Lock, Check Password, Update

	Server Information
Pre-Requisites	Password error occurred more than 3 times
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) This use case occurs if password error occurred more than 3 times. 2 (S) If this use case occurred, invoke "Update Server Information".
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2 : Customer cannot unlock transaction by System.

Use Case	14. Request Customer's Data
Actor	(None)
Purpose	Request customer's data from external actor Offer
Overview	System requests customer's data from Offer to compare information inputted by Customer or show information to Customer.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.2, R1.4, R1.5, R1.6, R1.7, R1.8, R2.2, R3.1 Use Case : Withdraw, Transfer, Exchange, Loan, Check Balance, Pay Utility Bill, Take Charge
Pre-Requisites	Customer inputs information or request information from server.
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Request Offer to provide customer's data saved in server.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	15. Insert
Actor	Customer
Purpose	Check validation of information inputted
Overview	System checks validation of basic information inputted after selecting menu from basic screen.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.4, R1.5, R1.6, R1.7, R1.8, R2.3, R3.1, R3.2 Use Case : Deposit, Withdraw, Transfer, Exchange, Loan, Check Balance, Pay Utility Bill, Print Error, Request Customer's Data

Pre-Requisites	Customer inputs card, bankbook, giro bill after selecting menu from basic screen.
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Request basic information from server to Offer. 2. (S) Compare inputted information between provided information. 3. (S) If correct(or valid), keep going on a process.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 3 : If incorrect(or invalid), invoke "Print Error".

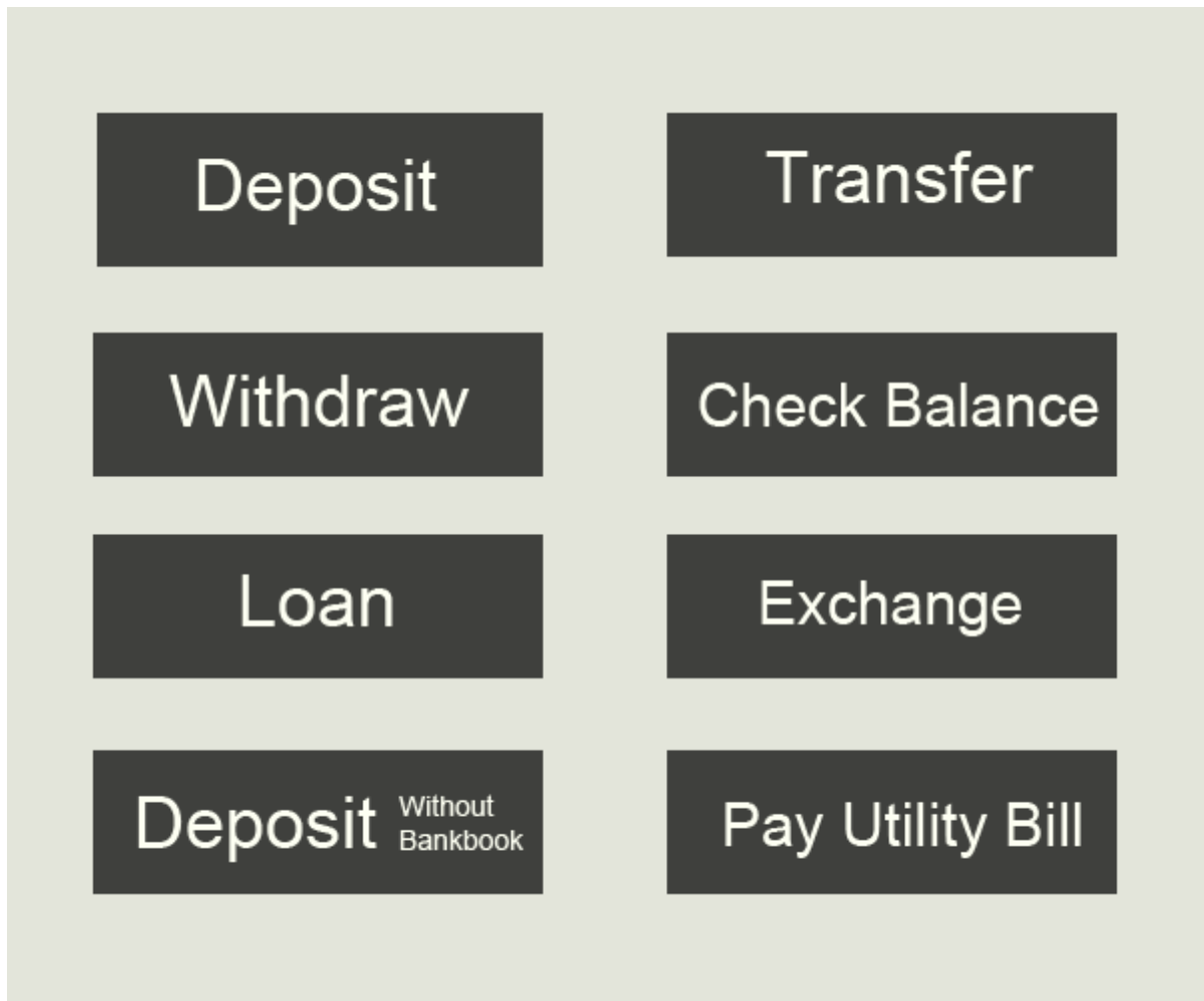
Use Case	16. Check Password
Actor	(None)
Purpose	Check password of account or credit card
Overview	System checks password of account or credit card inputted by Customer.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.2, R1.4, R1.5, R1.6, R1.7, R1.8, R2.4, R2.5, R3.3, R4.1 Use Case : Withdraw, Transfer, Exchange, Loan, Check Balance, Pay Utility Bill, Forced Termination, Transaction Lock, Update Server Information
Pre-Requisites	Customer inputs password for the account or credit card.
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Compare password inputted between server information provided. 2. (S) If incorrect, print out error message.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2 : If incorrect more than 3 times, invoke "Forced Termination", and "Transaction Lock".

Use Case	17. Check Credit
Actor	(None)
Purpose	Check credit of credit card
Overview	System checks credit of credit card inputted by Customer.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.6, R3.4 Use Case : Loan

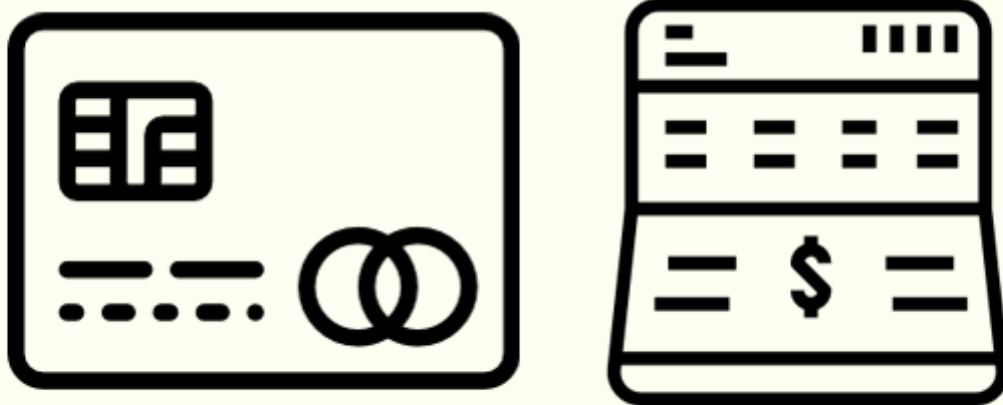
Pre-Requisites	Customer inputs credit card to loan
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Request credit information to Offer. 2. (S) Check Credit rating provided by Offer.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	18. Update Server Information
Actor	Offer
Purpose	Update server information changed during transaction
Overview	System requests Offer to update server information changed during transaction.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.8, R2.1, R2.5, R4.1 Use Case : Deposit, Withdraw, Deposit without Bankbook, Transfer, Exchange, Loan, Pay Utility Bill, Transaction Lock
Pre-Requisites	Transaction ended successfully and transaction receipt successfully printed
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Request Offer to update Server Information after Transaction. 2. (A) Offer updates server information
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Activity 2042. Define Reports, UI, and Storyboards

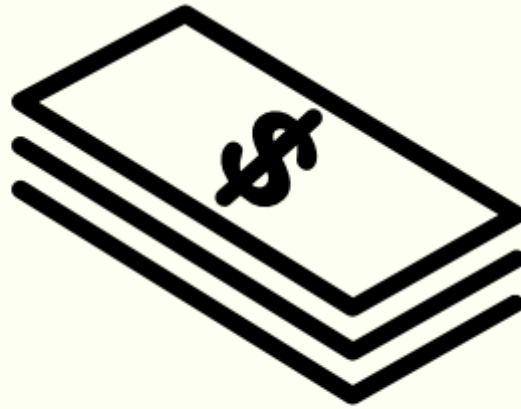


Window 1



Please Insert Card or Bankbook

Window 2-1



Please Insert Cash

Window 2-2

BANK

Shinhan



Account Num.

1	2	3	←
4	5	6	Reset
7	8	9	Enter
	0		

Window3

Input Password

*	*	—	—
---	---	---	---

1	2	3
4	5	6
7	8	9
←	0	Reset

Window 4

Select Country

USD

JPY

EUR

CNY

Window 5

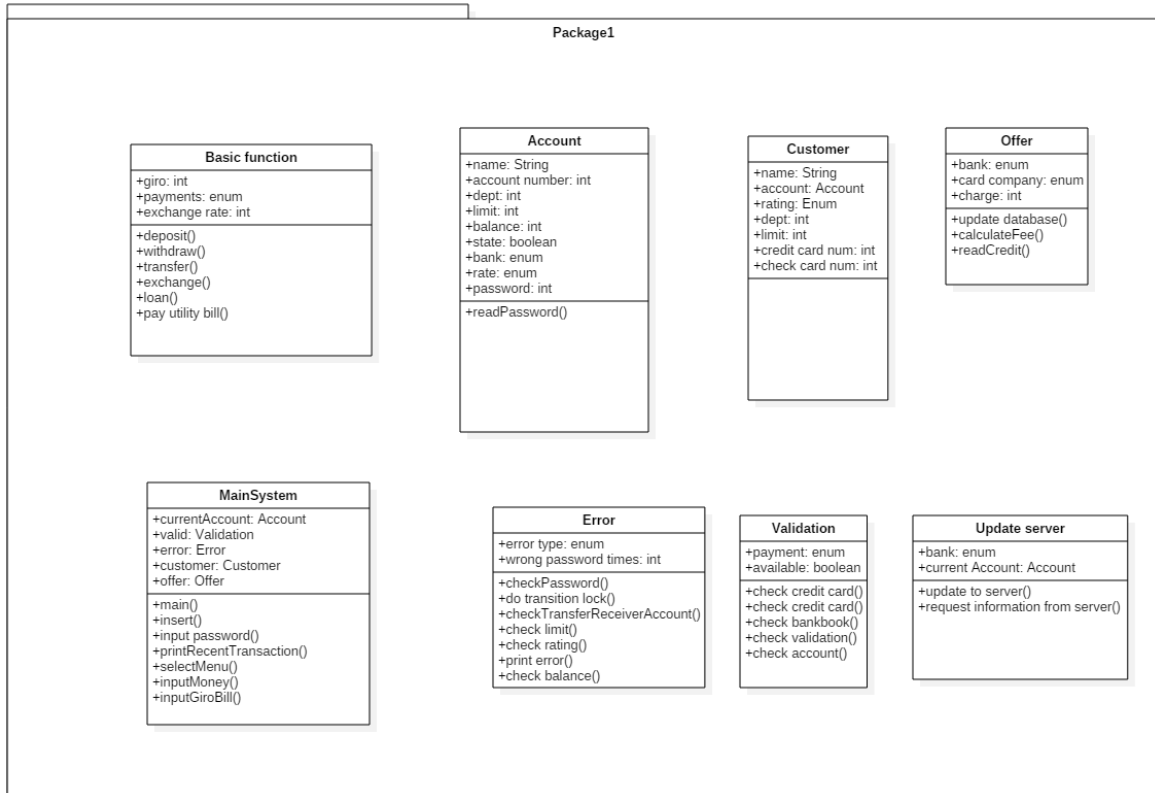
Input amount of money

₩

1	2	3	만
4	5	6	십만
7	8	9	백만
←	0	Reset	Enter

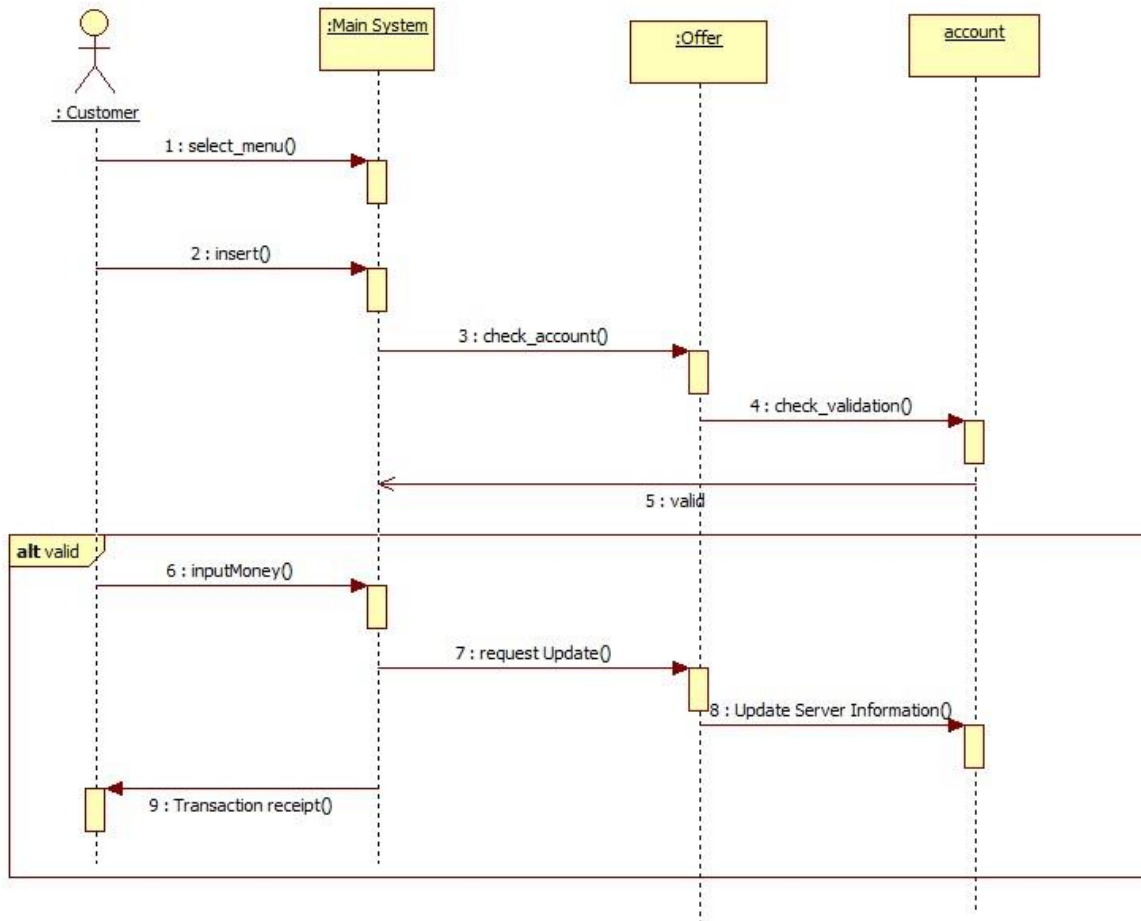
Window 6

Activity 2043. Refine System Architecture

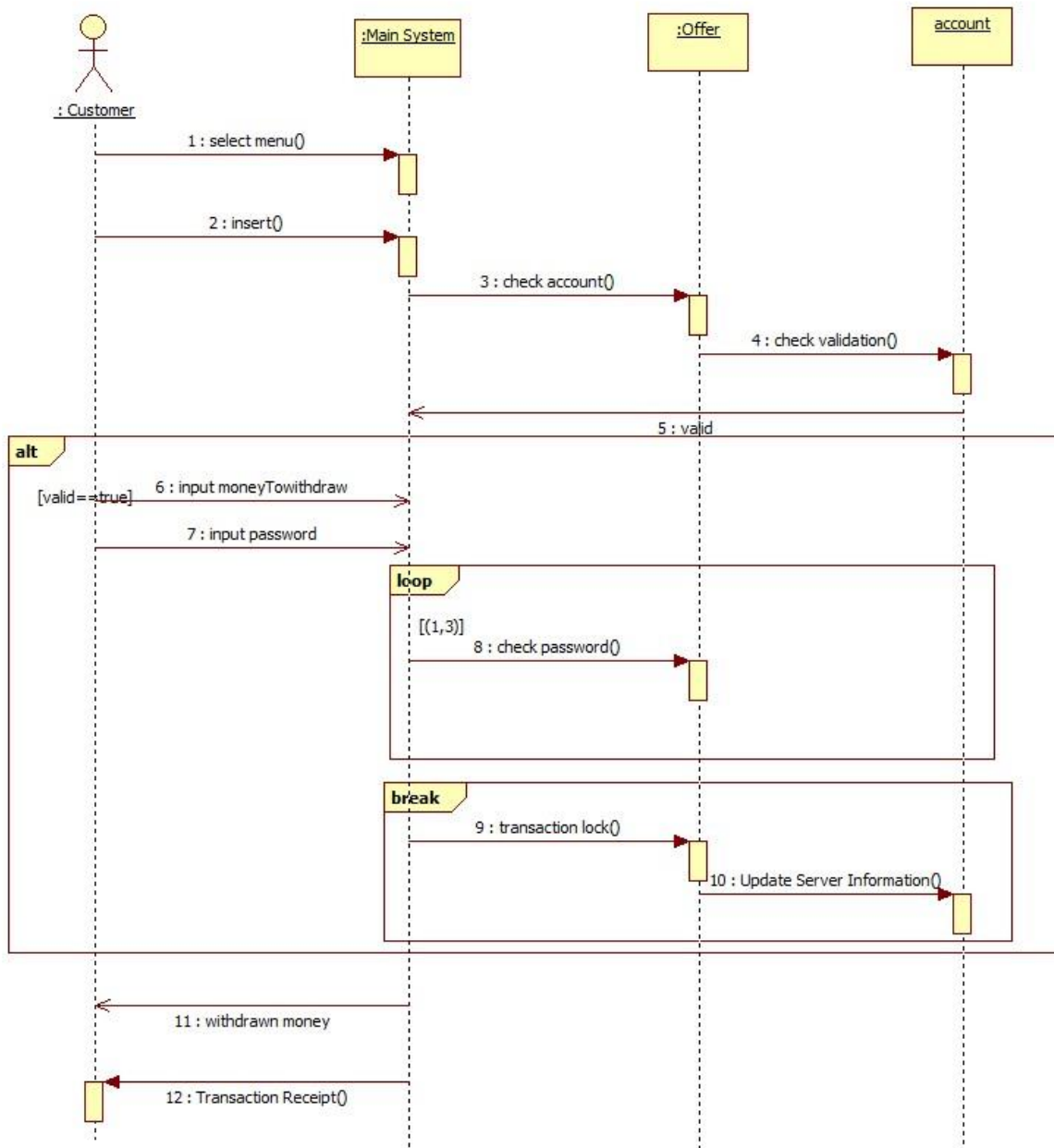


Activity 2044. Define Interaction Diagrams

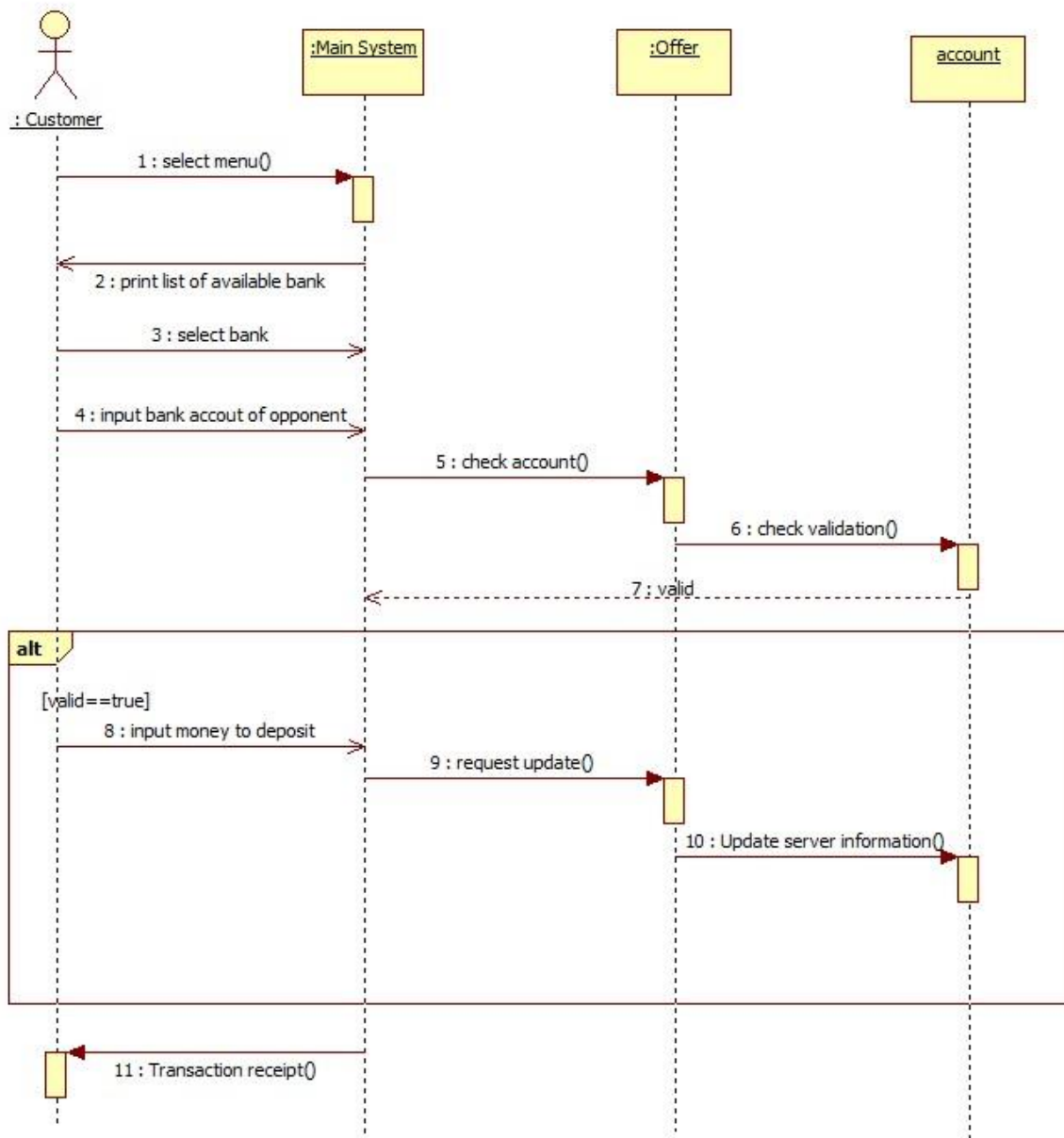
1. ;Deposit



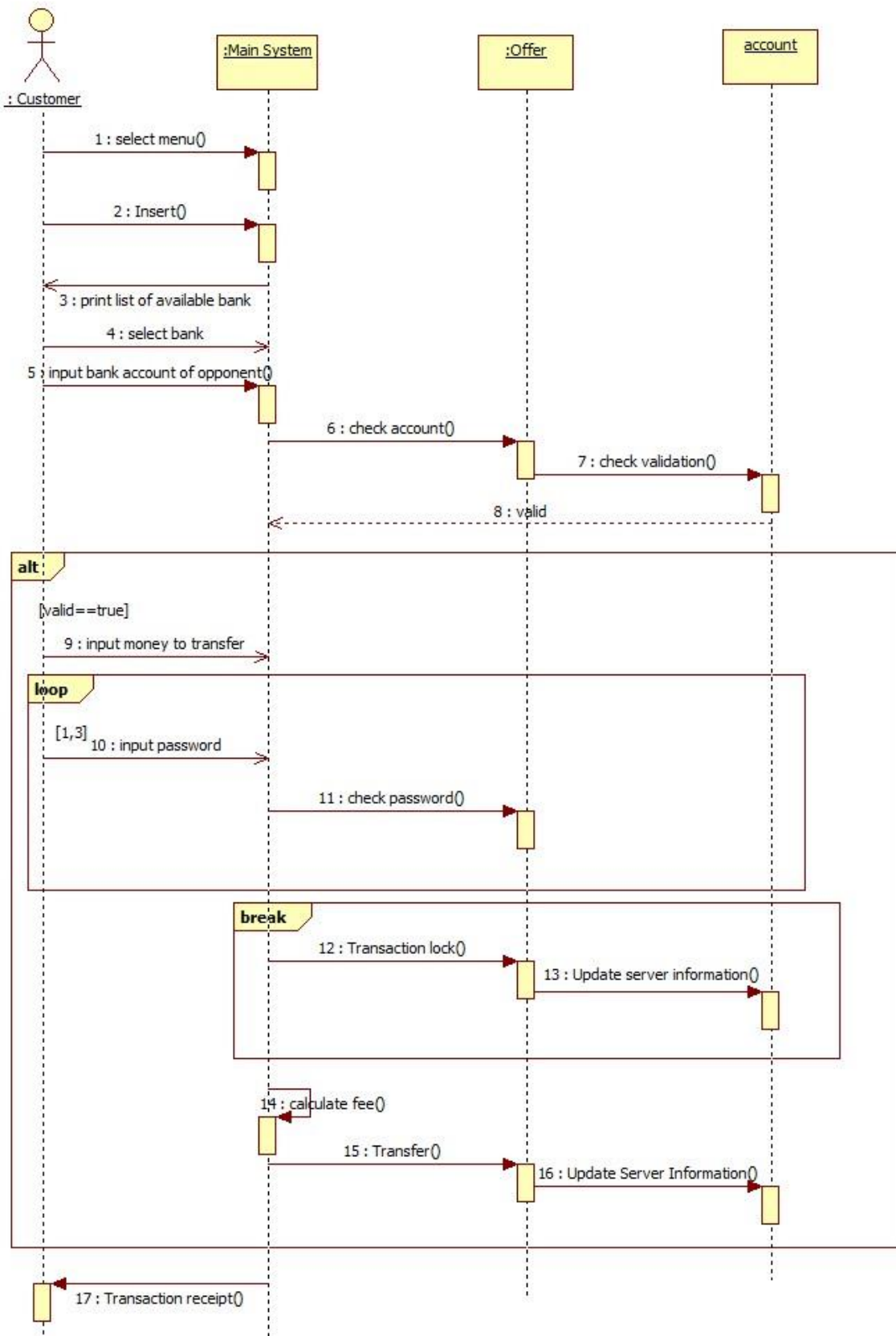
2. Withdraw



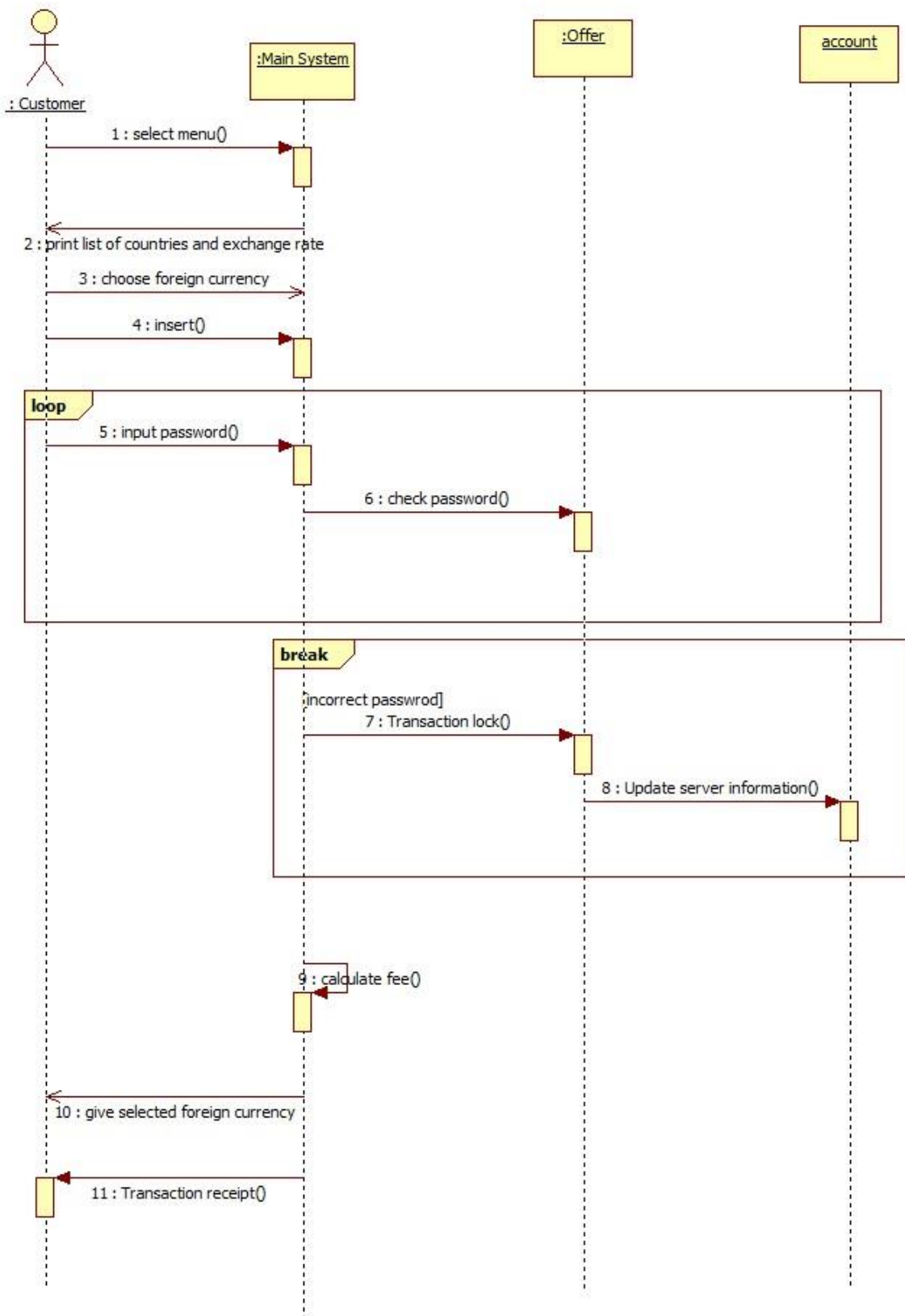
3. Deposit Without Bankbook



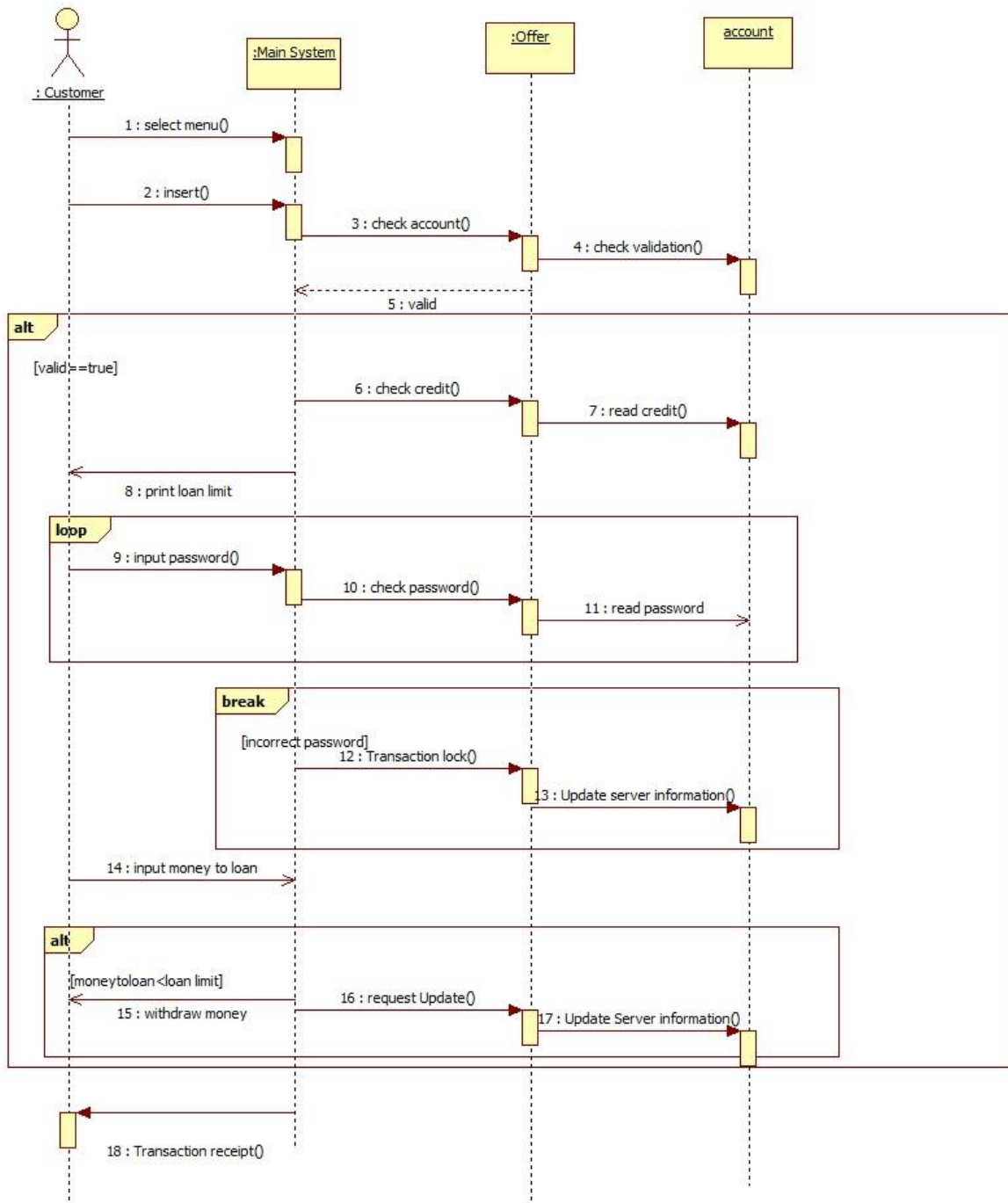
4. Transfer



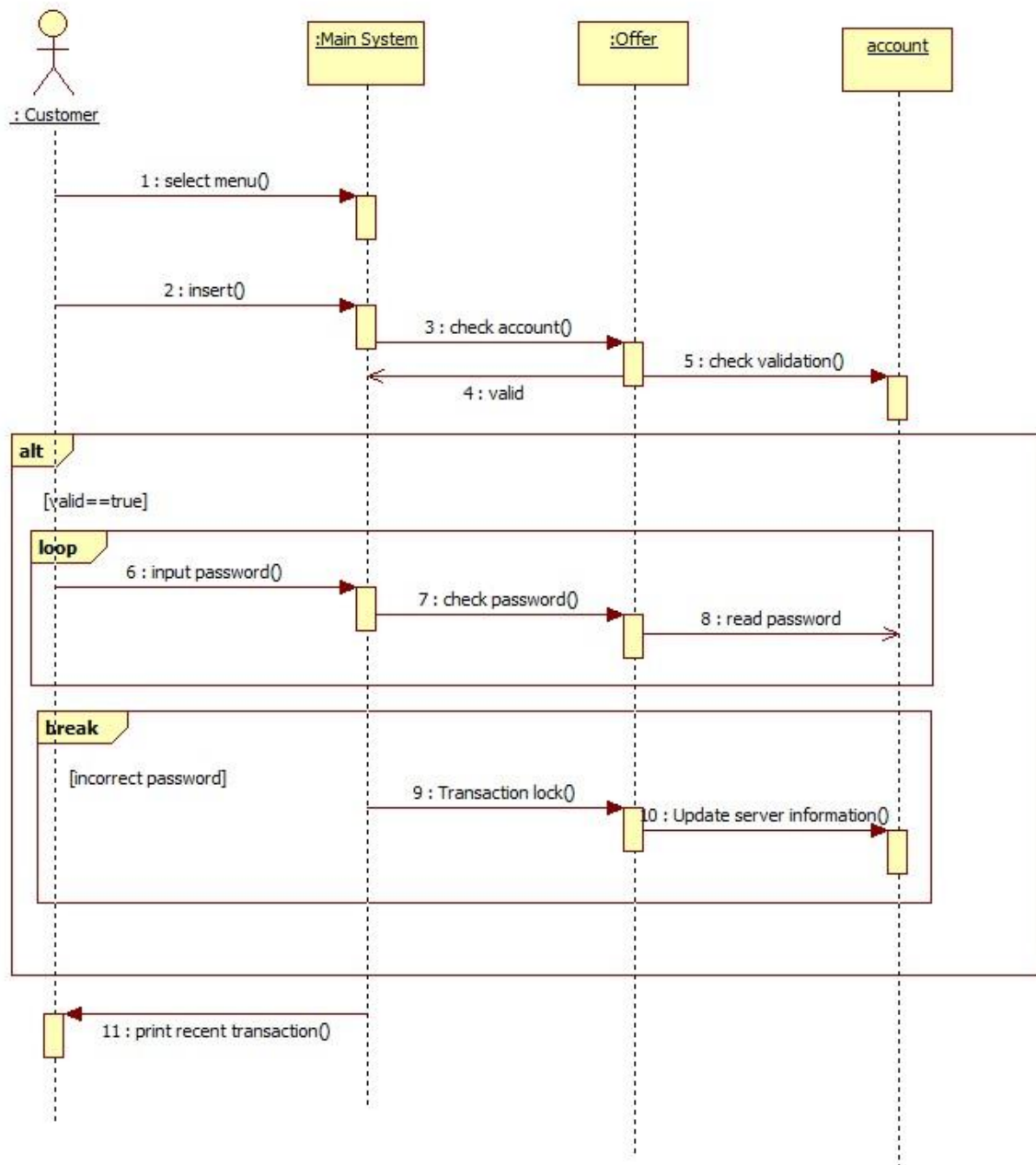
5. Exchange



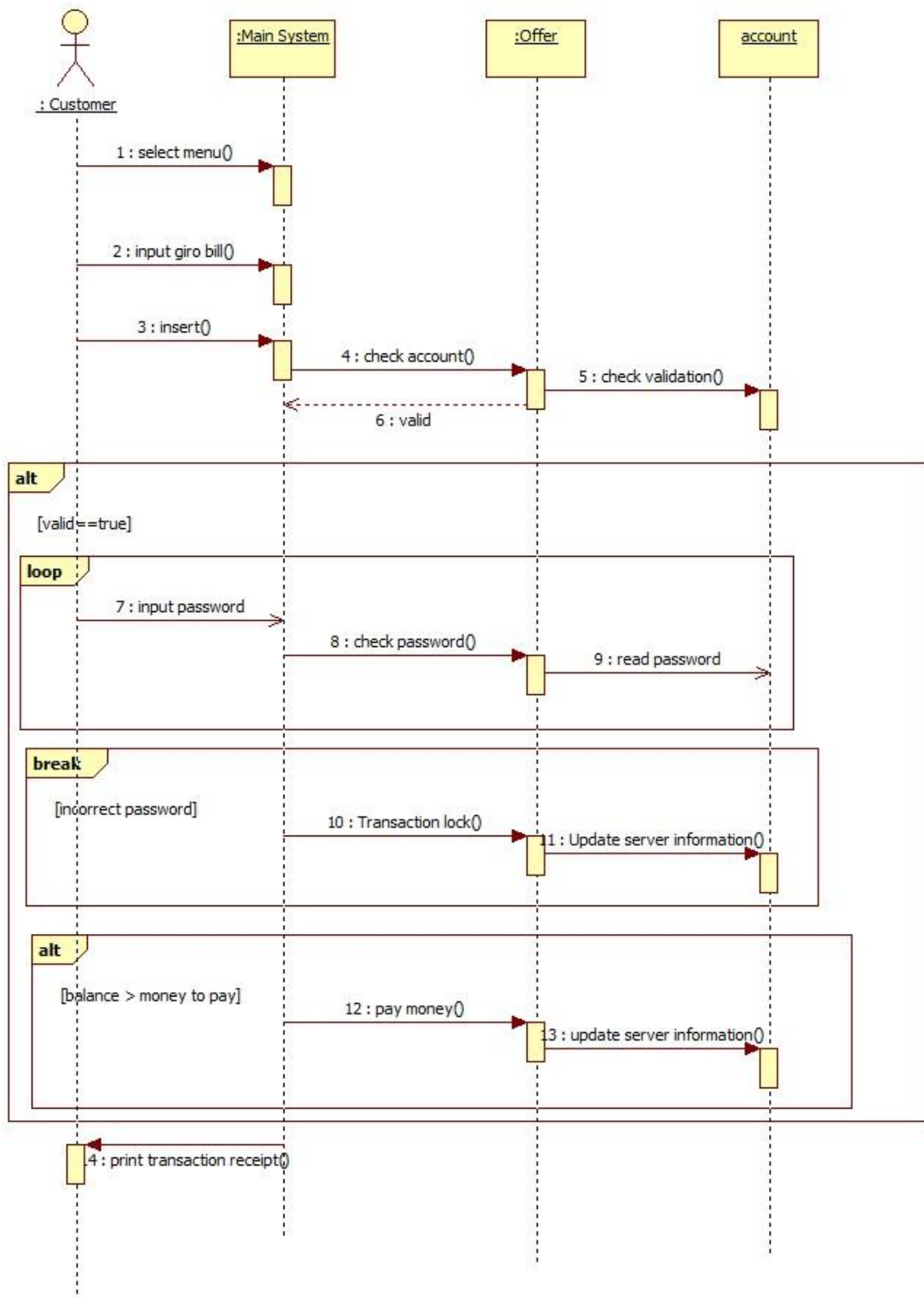
6. Loan



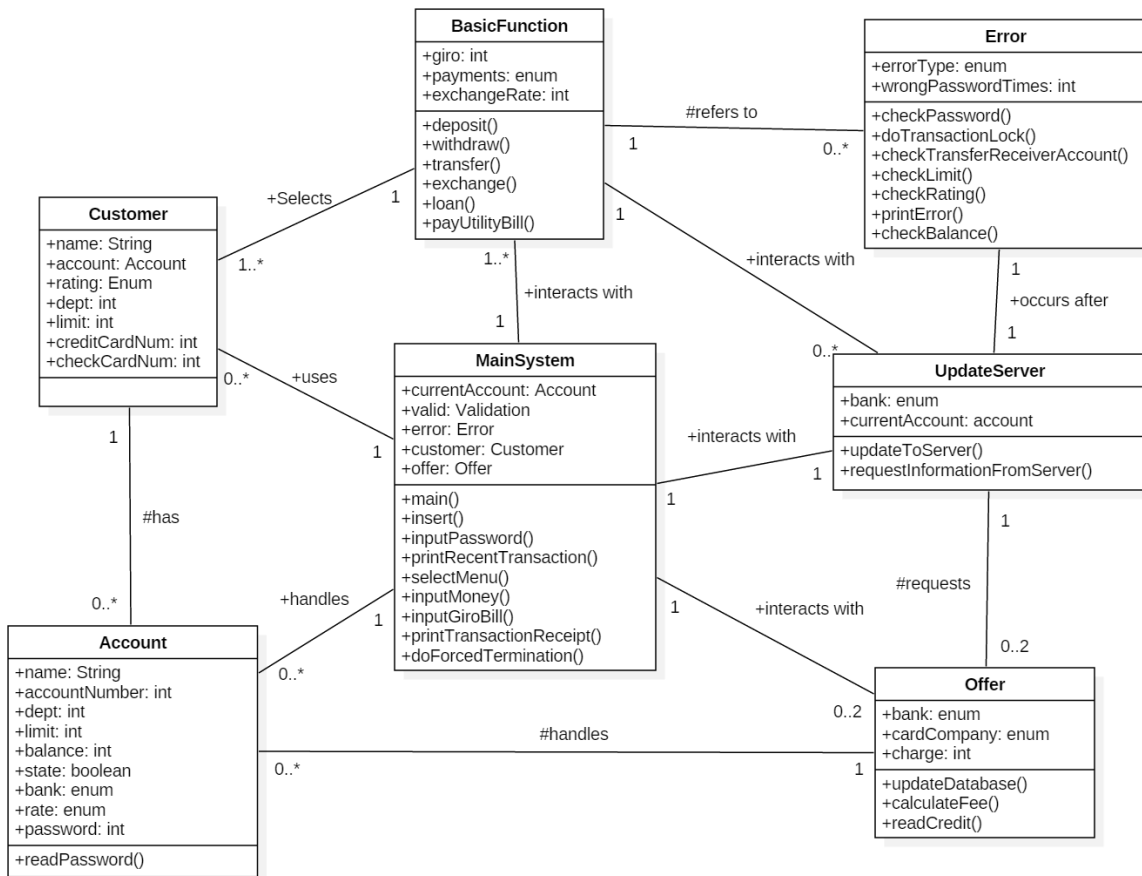
7. Check Balance



8. Pay Utility Bill



Activity 2045. Define Design Class Diagrams



Activity 2046. Design Traceability Analysis

Operation in sequence diagram	Operation in interaction diagram	Method	Class
selectDeposit()	selectMenu()	deposit()	
selectWithdraw()	insert()	withdraw()	
selectDepositWithoutBankbook()	checkAccount()	transfer()	BasicFunction
selectTransfer()	checkValidation()	exchange()	
selectExchange()	inputMoney()	loan()	
selectLoan()	inputMoneyToWithdraw()	payUtilityBill()	
selectCheckBalance()	inputPassword()	insert()	
selectPayUtilityBill()	checkPassword()	inputPassword()	
insert()	selectBank()	printRecentTransaction()	
inputMoney()	inputMoneyToDeposit()	selectMenu()	MainSystem
inputPassword()	inputBankAccountOfOpponent()	inputMoney()	
selectBank()	inputMoneyToTransfer()	inputGiroBill()	
inputBankAccountOfOpponent()	transfer()	printTransactionReceipt()	
inputMoneyToDeposit()	chooseForeignCurrency()	doForcedTermination()	
inputBankAccount()	calculateFee()	readPassword()	Account
inputMoneyToTransfer()	checkCredit()	checkPassword()	
chooseForeignCurrency()	readCredit()	doTransactionLock()	
inputAmountOfMoneyToLoan()	readPassword()	checkTransferReceiverAccount()	
inputGiroBill()	payMoney()	checkLimit()	Error
	printTransactionReceipt()	checkRating()	
	inputGiroBill()	printError()	
	requestUpdate()	checkBalance()	
	transactionLock()	updateToServer()	UpdateServer
	updateServerInformation()	requestInformationToServer()	
	inputMoneyToLoan()	updateDatabase()	
		calculateFee()	Offer
		readCredit()	

Activity 2047. Define Database Schema

(생략)