

A New Reliable ATM

OOPT Phase 2040

Design

Project Team T6

201411140 권성완

201511247 김선정

201510436 허윤아

201510285 조수빈

Date

2018-05-30

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, R2.2, R2.3, R2.4, R6.1, R6.2 Use case : Insert, Print Transaction Receipt, Print Error, Do Forced Termination, Check Validation, Update Database
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 "Insert check card or bankbook or credit card". 3. (A) Customer inserts credit card, check card or bankbook("Insert"). 4. (S) Invoke "Check Validation". If inserted method 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, invoke "Update Database".
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₩, print error. Line 7 : If incorrect, invoke "Print Error". If error occurs 3 times, invoke "Do Forced Termination". If loan record exists, loan is automatically repaid.

Use Case	2. 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 : R1.2, R2.2, R2.3, R2.4, R6.1, R6.2 Use Case : Print Transaction Receipt, Print Error, Do Forced Termination, Check Validation, Update Database
Pre-Requisites	Customer should know exact account number to deposit.
UI Widget	Window 1, 2-2, 3
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) Customer inputs 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, invoke "Update Database".
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4 : if invalid, invoke "Print Error". If error occurs 3 times, invoke "Do Forced Termination"

Use Case	3. 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 : R1.3, R2.1, R2.2, R2.3, R2.4, R4.1, R5.1, R6.1, R6.2 Use Case : Insert, Print Transaction Receipt, Print Error, Do Forced Termination, Check Password, Transaction Lock, Check Validation, Update Database
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 "Insert check card or bankbook".

	<p>3. (A) Customer inserts check card or bankbook("Insert").</p> <p>4. (S) Invoke "Check Validation". If inserted method is valid, ask customer to input amount of money to withdraw.</p> <p>5. (A) Customer inputs amount of money in unit of 10000₩ and 50000₩ to withdraw from account.</p> <p>6. (S) Ask for password for the account.</p> <p>7. (A) Customer inputs password for the account.</p> <p>8. (S) Invoke "Check Password". If password is correct, count numbers of bills.</p> <p>9. (S) If balance is enough, withdraw cash and invoke "Print Transaction Receipt".</p> <p>10. (S) If Transaction Receipt is successfully printed, invoke "Update Database".</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>Line 4, 8, 9 : If incorrect or invalid, or if balance is not enough, invoke "Print Error". If error occurred more than three times, invoke "Do Forced Termination". Especially if password is incorrect 3 times, additionally invoke "Transaction Lock", "Update Database".</p> <p>Line 5 : If total amount of money to withdraw is over 50000₩, customer inputs number of 50000₩ bill.</p>

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	<p>Functional Requirements : R1.4, R2.1, R2.2, R2.3, R2.4, R3.1, R4.1, R5.1, R6.1, R6.2</p> <p>Use Case : Insert, Print Transaction Receipt, Print Error, Do Forced Termination, Take Charge, Check Password, Transaction Lock, Check Validation, Update Database</p>
Pre-Requisites	Customer should know password for the account, and account to transfer money.
UI Widget	Window 1, 2-1, 3, 4, 6
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <p>1. (A) Customer selects transfer menu from the basic screen.</p> <p>2. (S) prints out "Insert check card or bankbook".</p>

	<p>3. (A) Customer inserts check card or bankbook("Insert").</p> <p>4. (S) Invoke "Check Validation". If inserted method is valid, ask customer to input bank and account number to transfer money.</p> <p>5. (A) Customer inputs bank and account number to transfer money.</p> <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) 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, invoke "Update Database".</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>Line 4, 10 : If incorrect or invalid, or if balance is not enough, invoke "Print Error." If error occurs 3 times, invoke "Do Forced Termination". Especially, if password is incorrect 3 times, additionally invoke "Transaction Lock", "Update Database".</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, R4.1, R5.1, R6.1, R6.2</p> <p>Use Case : Insert, Print Transaction Receipt, Print Error, Do Forced Termination, Take Charge, Check Password, Transaction Lock, Check Validation, Update Database</p>
Pre-Requisites	Customer should know password for the account. ATM only handles USD, JPY, CNY, EUR.
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 "insert check card or bankbook".</p>

	<p>3. (A) Customer inserts check card or bankbook("Insert").</p> <p>4. (S) Invoke "Check Validation". If inserted method 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 and balance is enough, 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, invoke "Update Database".</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>Line 4, 9 : If incorrect or invalid, or if balance is not enough, invoke "Print Error". If error occurs 3 times, invoke "Do Forced Termination". Especially, if password is incorrect 3 times, additionally invoke "Transaction Lock", "Update Database".</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, R4.1, R5.1, R6.1, R6.2</p> <p>Use Case : Loan, Insert, Print Transaction Receipt, Print Error, Do Forced Termination, Take Charge, Check Password, Transaction Lock, Check Validation, Update Database</p>
Pre-Requisites	Customer should know password for the credit card.
UI Widget	Window 1, 2-1, 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 "Insert credit card".</p> <p>3. (A) Customer inserts credit card("Insert").</p> <p>4. (S) Invoke "Check Validation". If inserted method is valid, print</p>

	<p>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) If total amount of 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, invoke "Update Database".</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 incorrect or invalid, invoke "Print Error". If error occurs 3 times, invoke "Do Forced Termination". Especially, if password is incorrect 3 times, additionally invoke "Transaction Lock", "Update Database".</p> <p>Line 9 : Invoke "Take Charge".</p>

Use Case	7. 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	<p>Functional Requirements : R1.7, R2.1, R2.2, R2.3, R2.4, R4.1, R5.1, R6.1 R6.2</p> <p>Use Case : Insert, Print Transaction Receipt, Print Error, Do Forced Termination, Check Password, Transaction Lock, Check Validation, Update Database</p>
Pre-Requisites	Customer should know password for the account, and have giro bill.
UI-Widget	Window 1, 2-1, 3, 4
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <p>1. (A) Customer chooses pay utility bill menu from basic screen.</p> <p>2. (S) Print out "insert giro bill".</p> <p>3. (A) Customer inserts giro bill("Insert").</p> <p>4. (S) Invoke "Check Validation". If giro bill is valid, print out</p>

	<p>"insert check card or bankbook".</p> <p>5. (A) Customer inserts check card or bankbook("Insert").</p> <p>6. (S) Invoke "Check Validation". If inserted method is valid, print out "input password for the account".</p> <p>7. (A) Customer inputs password for the account.</p> <p>8. (S) Invoke "Check Password". If password is correct and balance is enough to pay utility bill, invoke "Print Transaction Receipt".</p> <p>9. (S) If Transaction Receipt is successfully printed, invoke "Update Database".</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4, 6, 8 : If incorrect or invalid, or if balance is not enough, invoke "Print Error". If error occurs 3 times, invoke "Do Forced Termination". Especially, if password is incorrect 3 times, additionally invoke "Transaction Lock", "Update Database".

Use Case	8. Check balance
Actor	Customer
Purpose	Check balance of account
Overview	Customer checks balance of account.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.8, R2.1, R2.3, R2.4, R4.1, R5.1, R6.1 Use Case : Insert, Print Error, Do Forced Termination, Check Password, Check Validation
Pre-Requisites	Customer should know password for the account.
UI-Widget	Window 1, 2, 5
Typical Courses of Events	<p>(A) : Actor, (S) : System</p> <p>1. (A) Customer chooses check balance menu from basic screen.</p> <p>2. (S) Print out "insert check card or bankbook".</p> <p>3. (A) Customer inserts check card or bankbook("Insert").</p> <p>4. (S) Invoke "Check Validation". If inserted method is valid, print out "input password for the account".</p> <p>5. (A) Customer inputs password for the account.</p> <p>6. (S) Invoke "Check Password". If password is correct, print out recent transactional information(under 100) and balance after each transactional process.</p> <p>7. (A) Customer input 'OK' button.(GUI level)</p> <p>8. (S) Return to basic screen.</p>

Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4, 6 : If incorrect or invalid, invoke "Print Error". If error occurs 3 times, invoke "Do Forced Termination". Especially, if password is incorrect 3 times, additionally invoke "Transaction Lock", "Update Database".

Use Case	9. Insert
Actor	Customer
Purpose	Insert method for transaction
Overview	Customer inserts method for transaction
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R2.1, R2.3, R6.1 Use Case : Deposit, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Print Error, Check Validation
Pre-Requisites	N/A
UI-Widget	Window 2-1
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer inserts method for transaction, such as check card, bankbook, credit card, giro bill.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1 : After, invoke "Check Validation".

Use Case	10. 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, R1.7, R2.2, R2.3, R6.2 Use Case : Deposit, Deposit Without Bankbook, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Print Error, Update Database
Pre-Requisites	Transaction ends
UI-Widget	N/A

Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Transaction ends successfully. 2. (S) Print Transaction Receipt. 3. (S) If transaction receipt is printed successfully, invoke "Update Database".
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1 : If doesn't end successfully, invoke "Print Error".

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, R2.1, R2.2, R2.3, R2.4, R4.1, R5.1, R6.1, R6.2 Use Case : Deposit, Deposit Without Bankbook, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Insert, Print Transaction Receipt, Do Forced Termination, Check Password, Transaction Lock, Check Validation, Update Database
Pre-Requisites	Error occurred during transaction
UI-Widget	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) If error occurred during certain transaction, this use case is invoked. 2. (S) If this use case occurs 3 times, invoke "Do Forced Termination".
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2 : If password is incorrect 3 times, additionally invoke "Transaction Lock", "Update Database".

Use Case	12. Do Forced Termination
Actor	(None)
Purpose	Immediately end transaction when error occurs 3 times
Overview	System ends transaction automatically and immediately when error occurs 3 times.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7,

	R1.8, R2.1, R2.3, R2.4, R5.1, R6.2 Use Case : Deposit, Deposit Without Bankbook, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Insert, Print Error, Transaction Lock, Update Database
Pre-Requisites	Use case "Print Error" occurred 3 times
UI-Widget	Window 1
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) This use case occurs if "Print Error" occurred 3 times.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1 : If password error occurs 3 times, additionally invoke "Transaction Lock", "Update Database".

Use Case	13. Take Charge
Actor	(None)
Purpose	Take charge during transaction
Overview	System takes charge when customer transfers, exchanges, loans.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.4, R1.5, R1.6, R3.1 Use Case : Transfer, Exchange, Loan
Pre-Requisites	N/A
UI-Widget	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) During transferring, exchanging, and loaning, take charge.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1 : If credit rating is high, this use case is ignored.

Use Case	14. 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 : R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R2.3, R2.4, R4.1, R5.1, R6.2 Use Case : Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Print Error, Do Forced Termination, Transaction Lock, Update Database

Pre-Requisites	Customer inputs password for the account or credit card.
UI-Widget	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Compare password inputted between information stored in DB. 2. (S) If incorrect, invoke "Print Error".
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2 : If incorrect more than 3 times, invoke "Do Forced Termination", "Transaction Lock", "Update Database",

Use Case	15. Transaction Lock
Actor	(None)
Purpose	Lock transaction of account or credit card when password error occurs 3 times
Overview	System locks transaction of account or credit card when password error occurs 3 times.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R2.1, R2.3, R2.4, R4.1, R5.1, R6.2 Use Case : Deposit, Deposit Without Bankbook, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Insert, Print Error, Do Forced Termination, Check Password, Update Database
Pre-Requisites	In use case "Check Password", use case "Print Error" occurred 3 times.
UI-Widget	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) This use case occurs if password error occurred 3 times. 2 (S) If this use case occurred, invoke "Update Database".
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1 : In each error, invoke "Print Error". After error occurred 3 times, invoke "Do Forced Termination". Line 2 : Customer cannot unlock transaction by System.

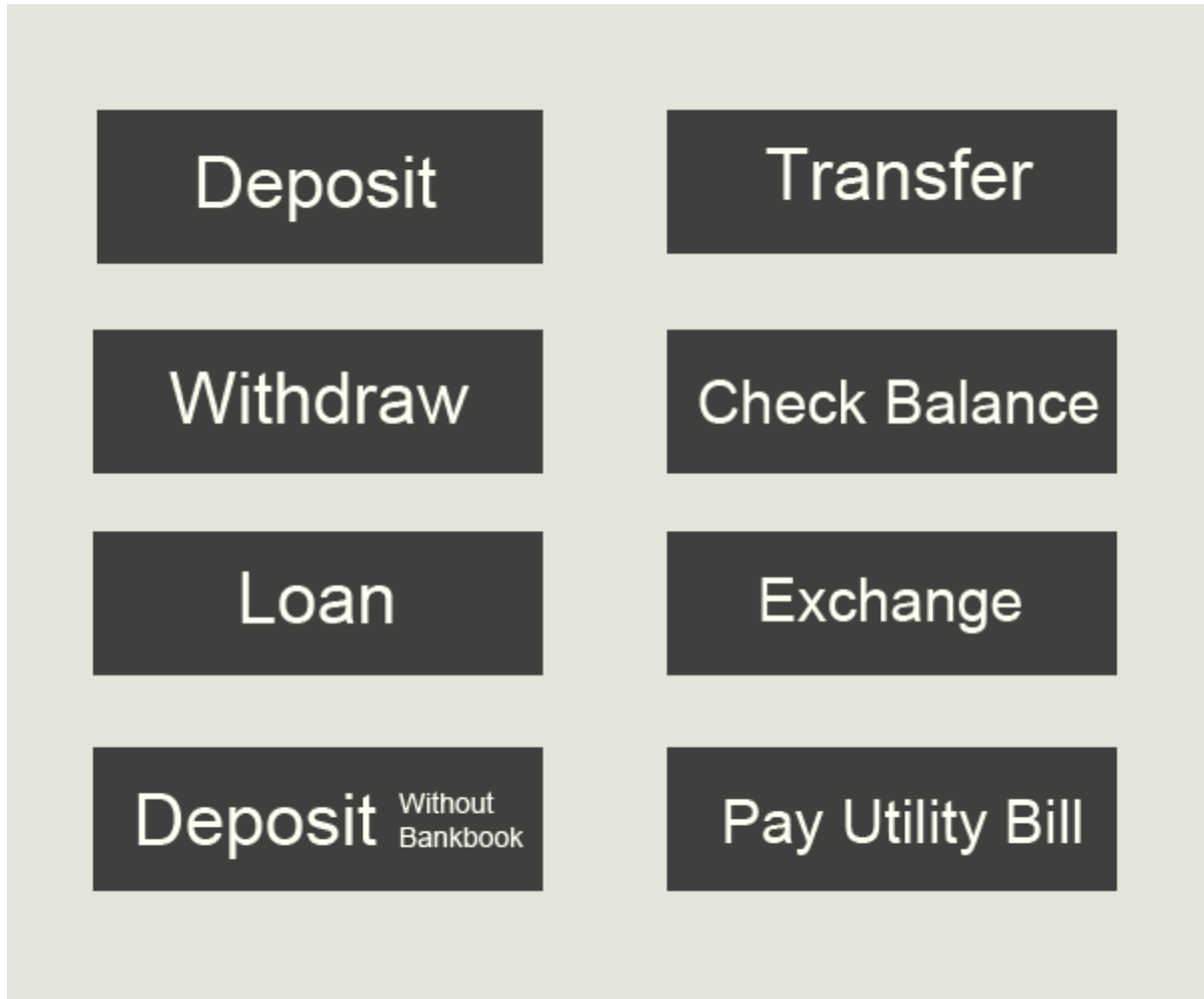
Use Case	16. Check Validation
Actor	Offer
Purpose	Check validation of inserted method

Overview	System checks validation of inserted method after selecting menu from basic screen.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R2.1, R2.3, R2.4, R5.1, R6.1 Use Case : Deposit, Deposit Without Bankbook, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Insert, Print Error, Do Forced Termination, Transaction Lock
Pre-Requisites	Customer inserts check card, credit card, bankbook, or giro bill after selecting menu from basic screen.
UI-Widget	Window 2-1
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Requests to check validation of inserted method to Offer. 2. (A) Offer compares information of inserted method between information stored in Database. 3. (A) If inserted method is valid, Offer gives "valid" to System, and system goes on a process.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 3 : If invalid or already has "Transaction Lock", invoke "Print Error". If error occurs 3 times, invoke "Do Forced Termination".

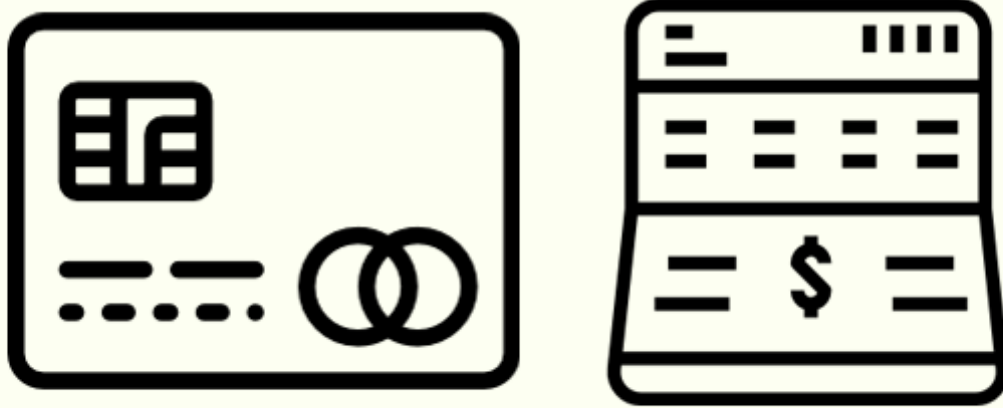
Use Case	17. Update Database
Actor	Offer
Purpose	Update Database after transaction
Overview	System requests Offer to update database after every transaction.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R2.2, R5.1, R6.2 Use Case : Deposit, Deposit Without Bankbook, Withdraw, Transfer, Exchange, Loan, Pay Utility Bill, Check Balance, Print Transaction Receipt, Transaction Lock
Pre-Requisites	Transaction ended successfully and use case "Print Transaction Receipt" occurred successfully
UI-Widget	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) Request Offer to update database after every transaction. 2. (A) Offer updates database.

Alternative Courses of Events	Line 1 : If account already has "Transaction Lock", this use case is ignored.
Exceptional Courses of Events	N/A

Activity 2042. Define Reports, UI, and Storyboards

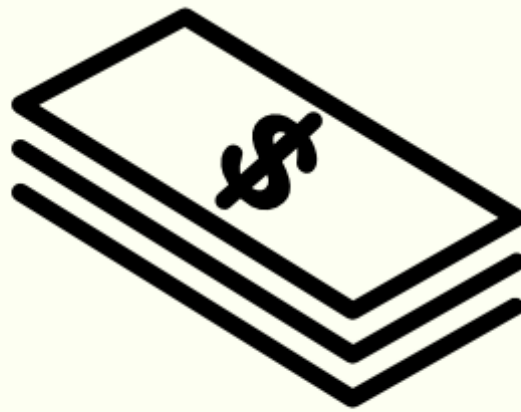


Window 1



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

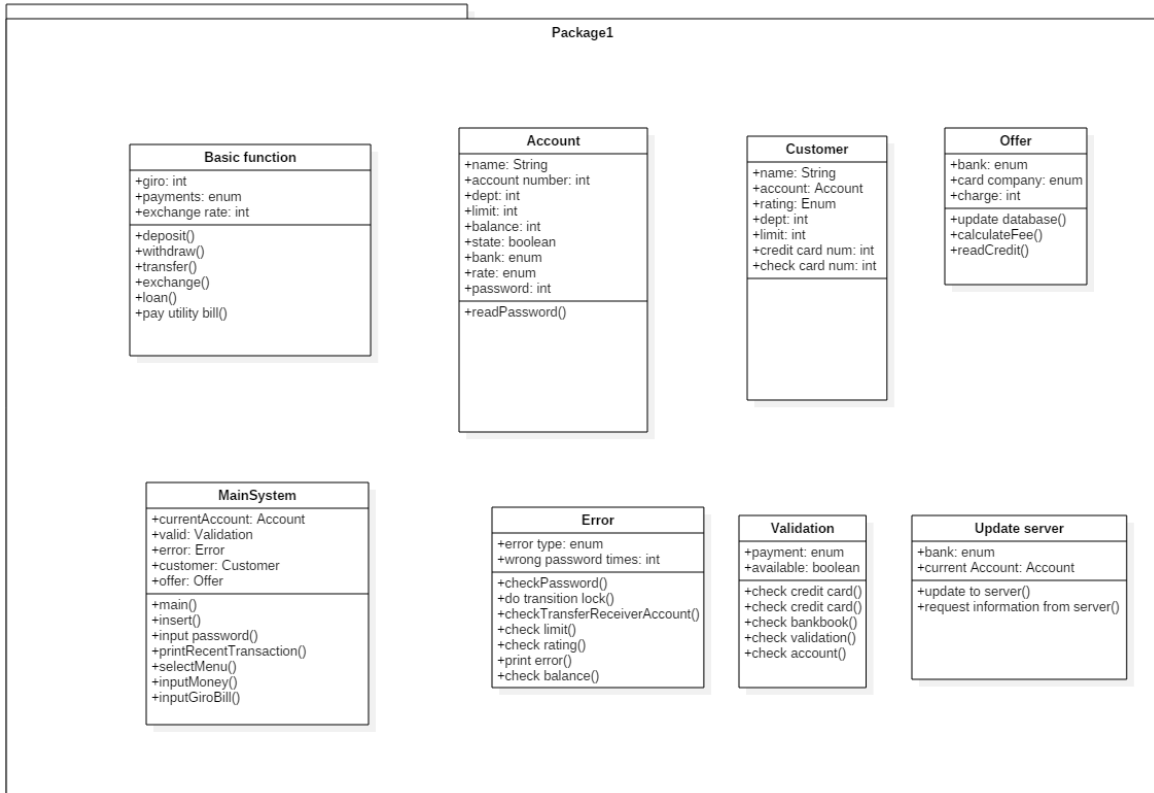
Input amount of money

₩

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

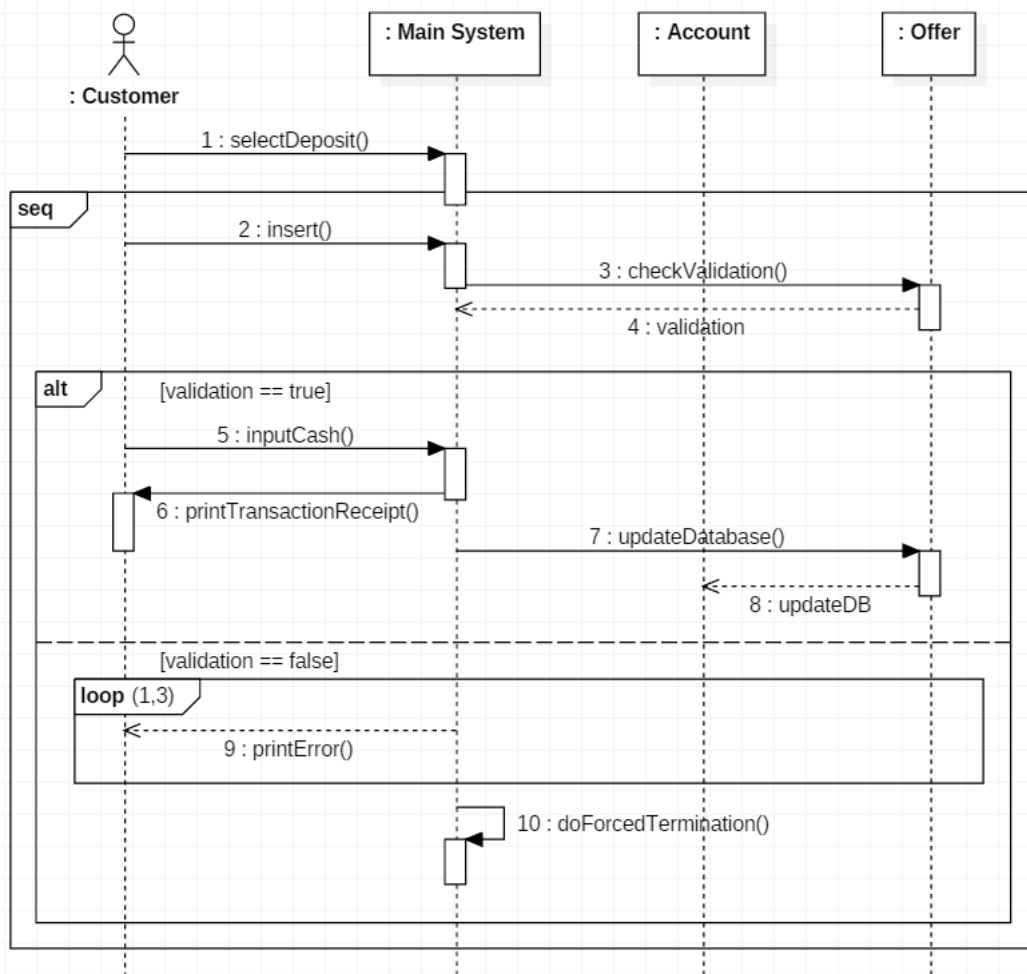
Window 6

Activity 2043. Refine System Architecture

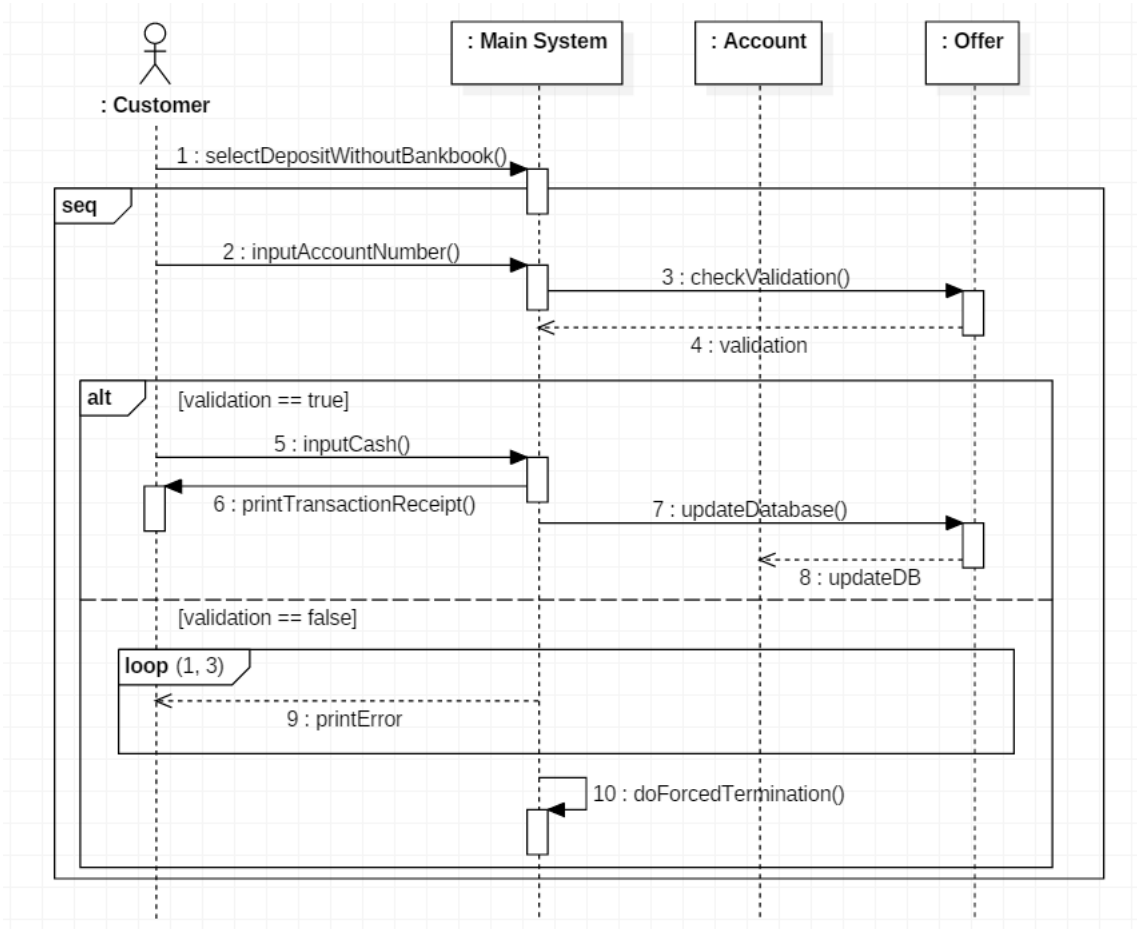


Activity 2044. Define Interaction Diagrams

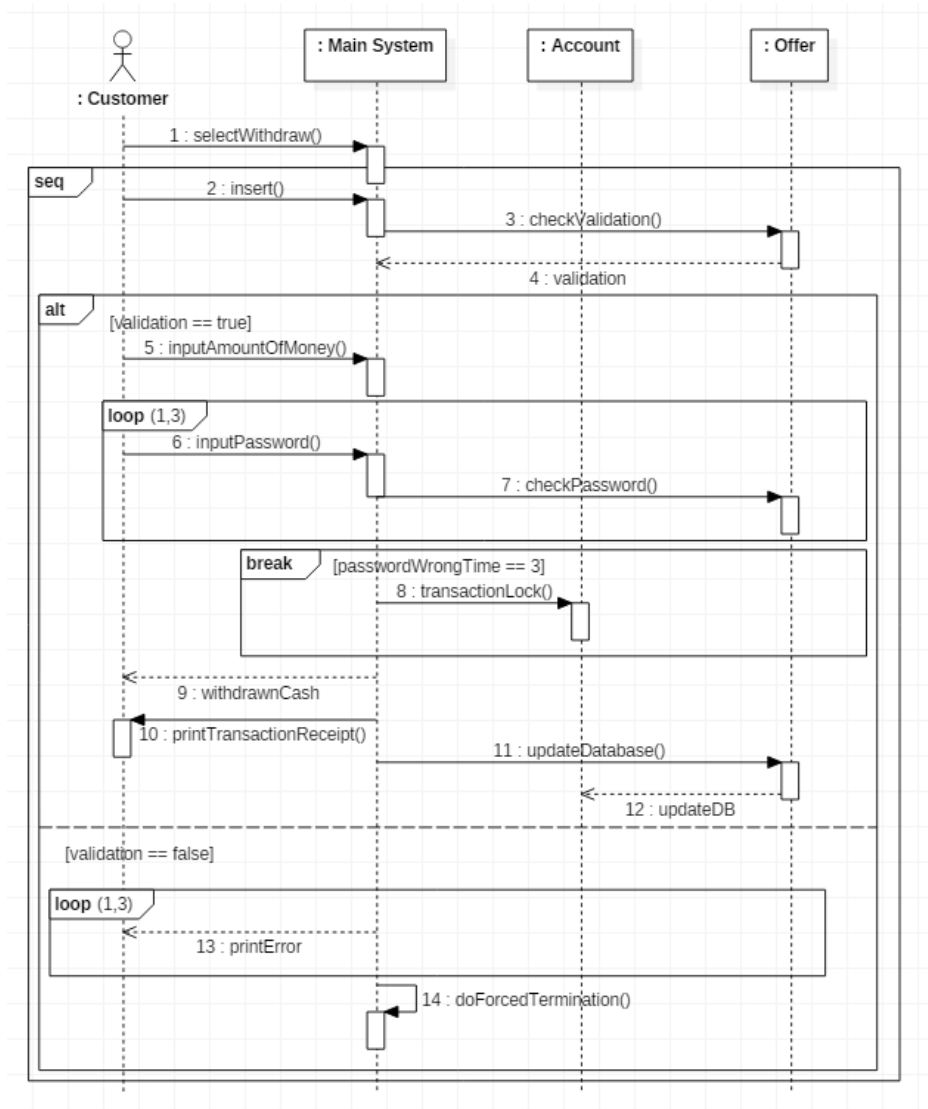
1. Deposit



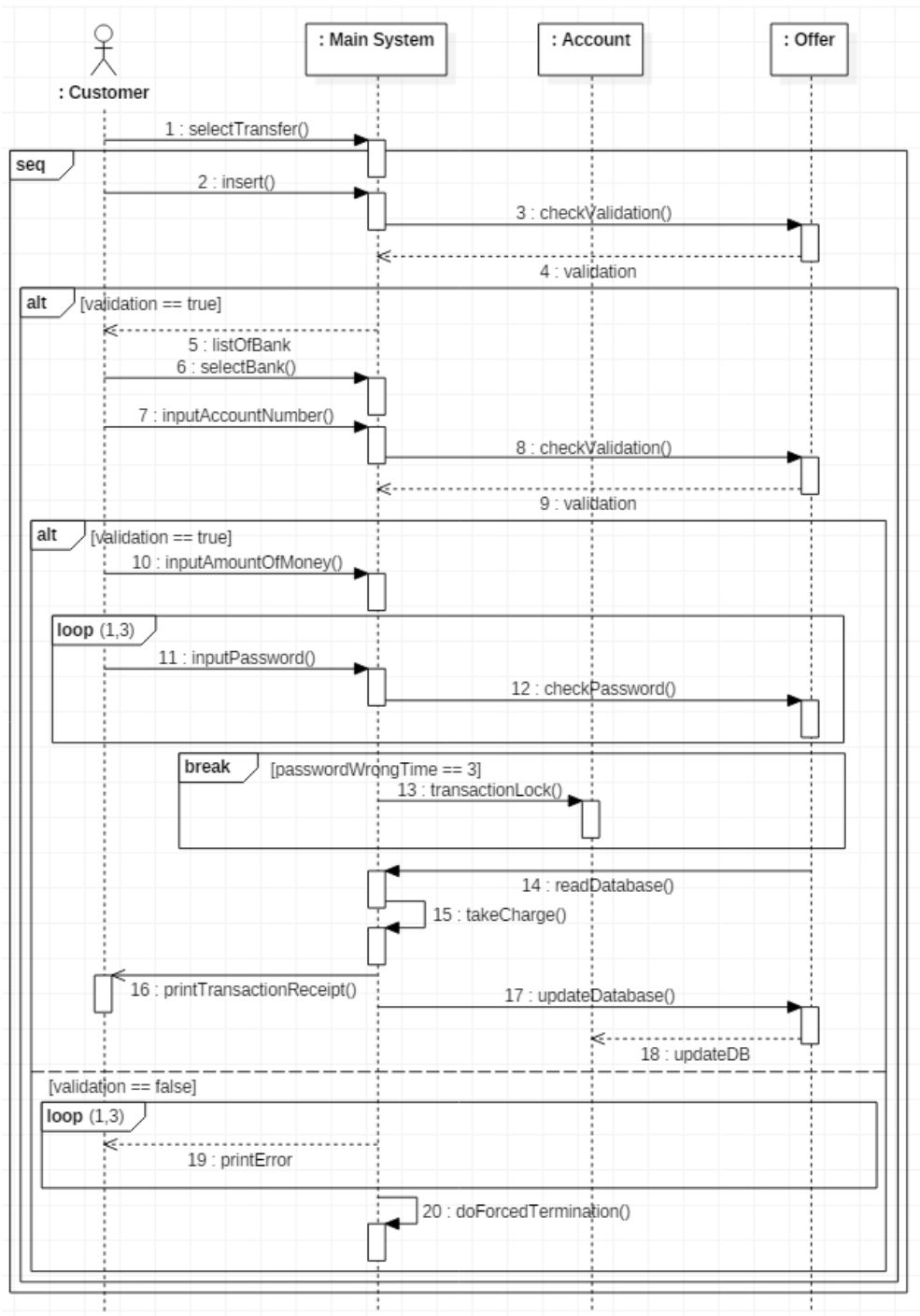
2. Deposit Without Bankbook



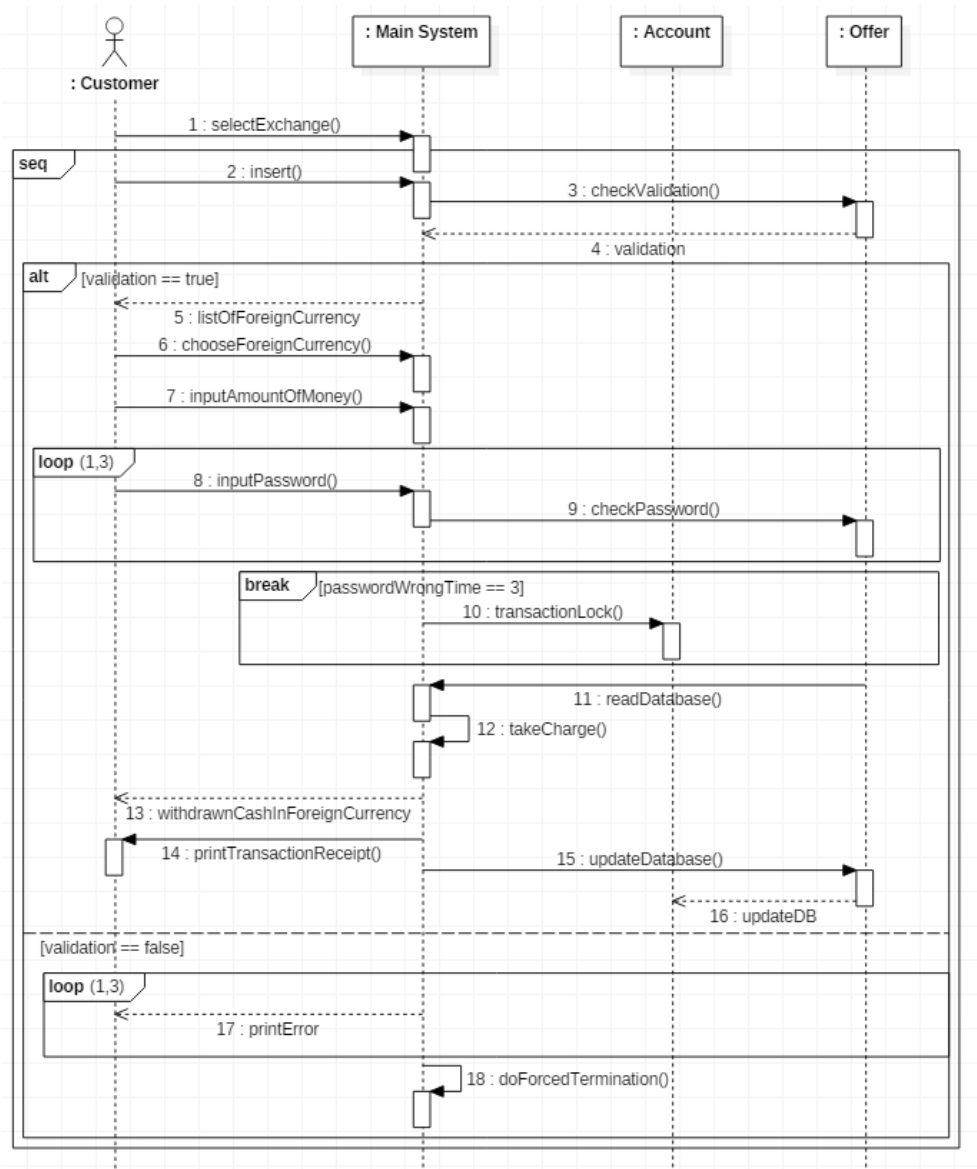
3. Withdraw



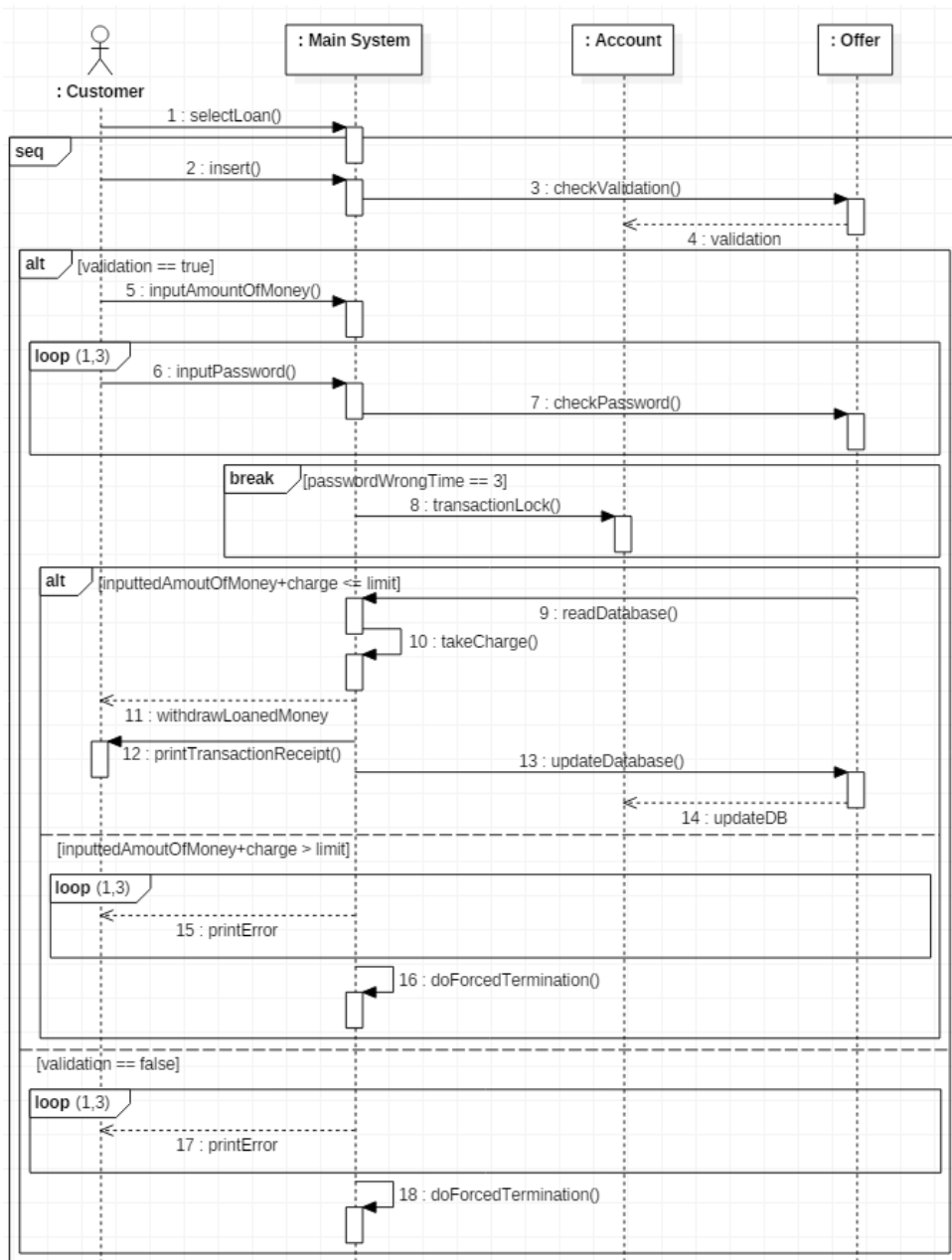
4. Transfer



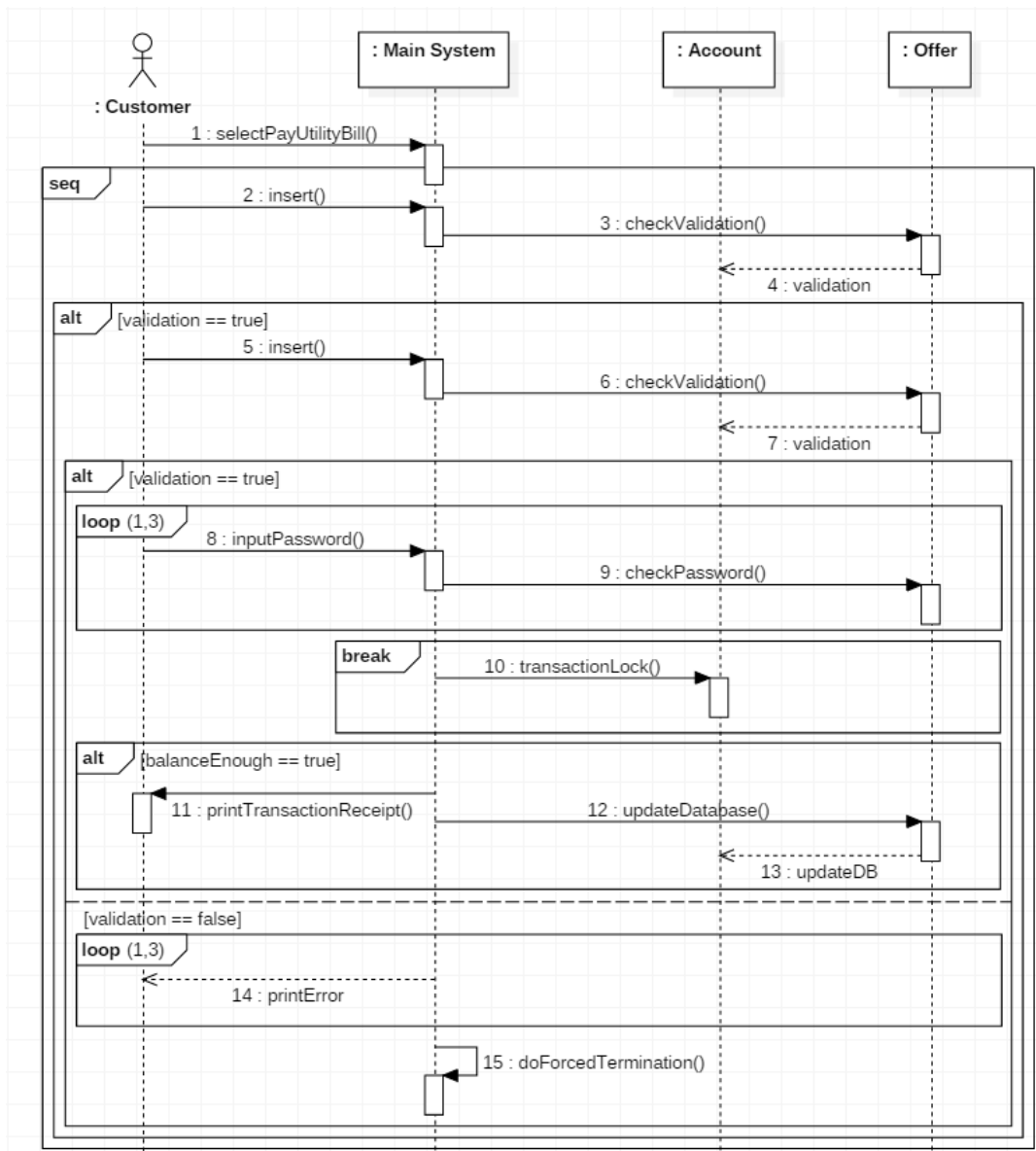
5. Exchange



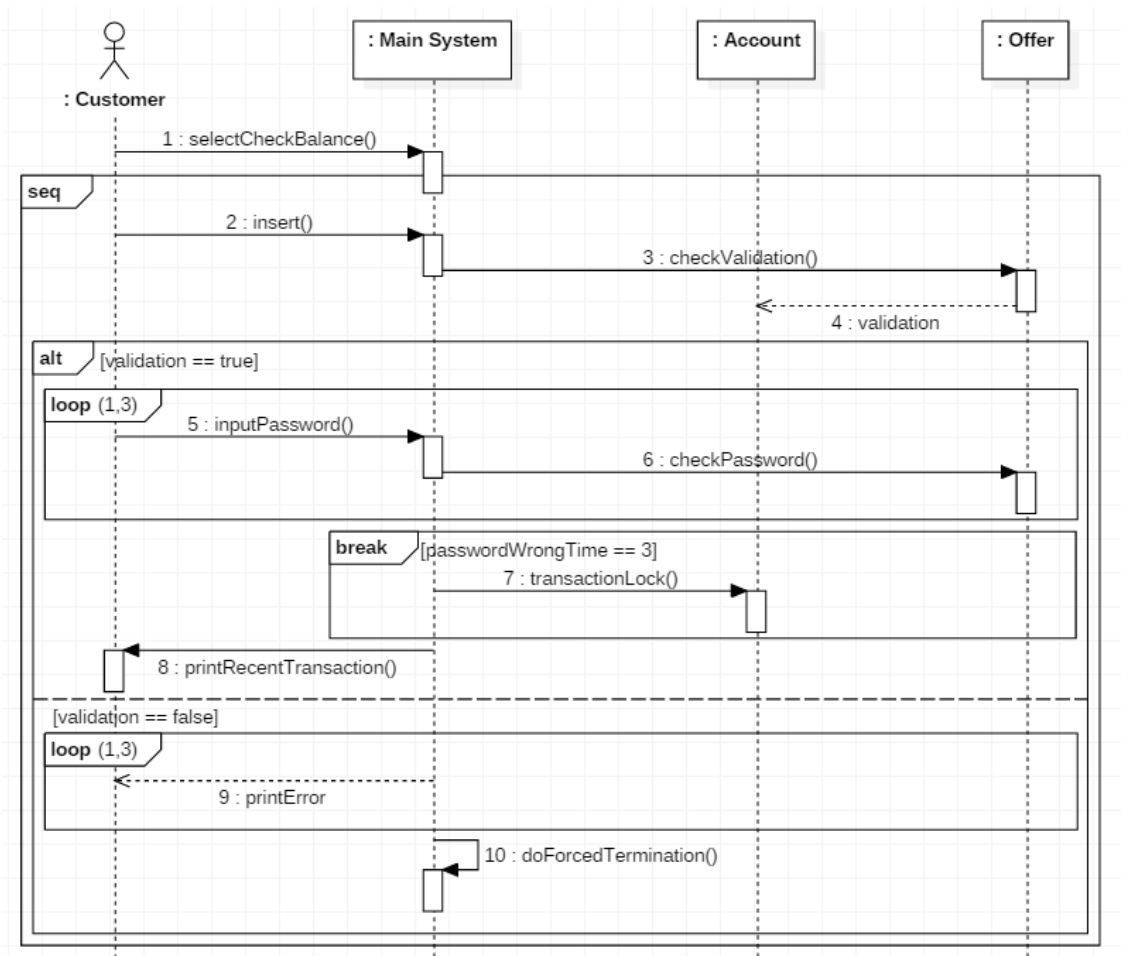
6. Loan



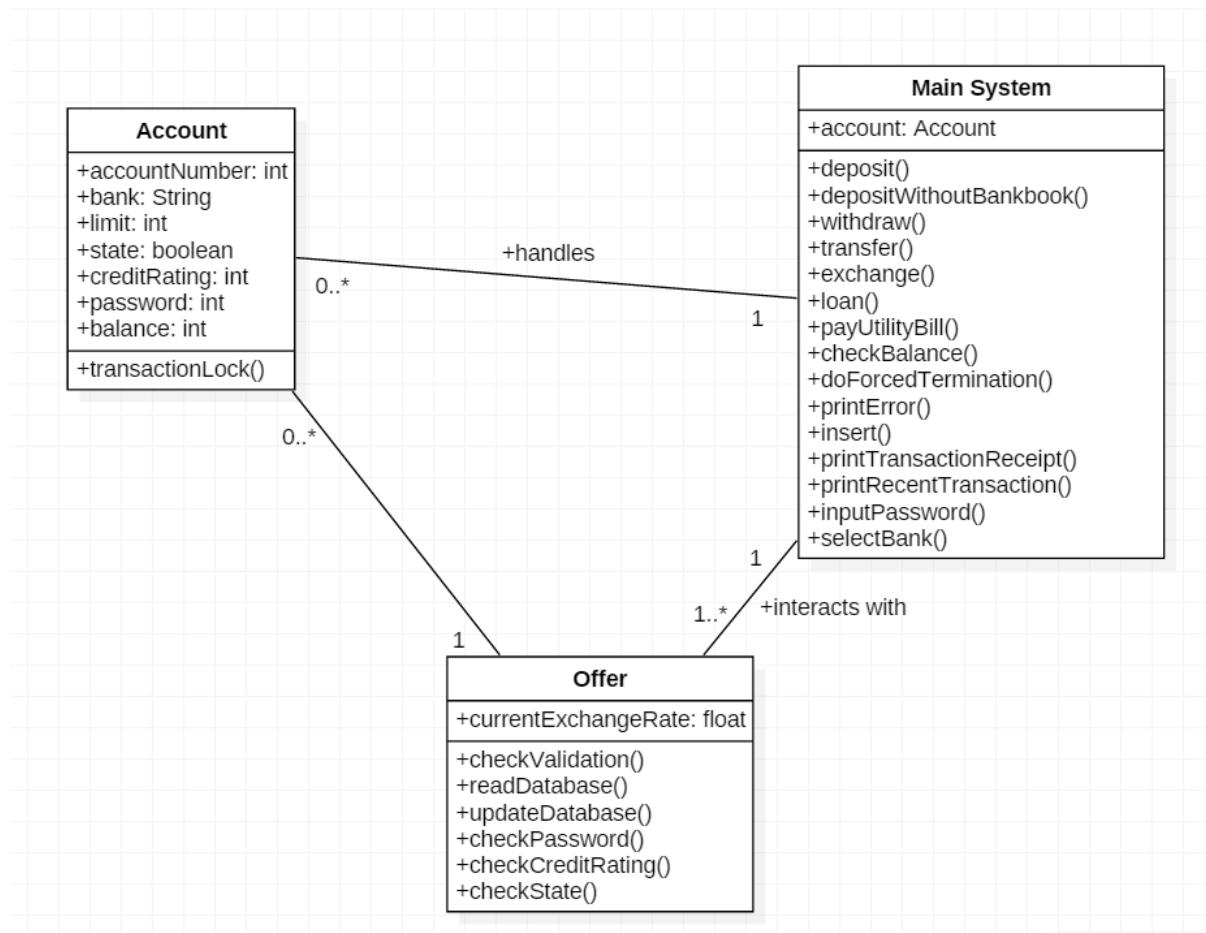
7. Pay Utility Bill



8. Check Balance



Activity 2045. Define Design Class Diagrams



Activity 2046. Design Traceability Analysis

Operation in sequence diagram	Operation in interaction diagram	Method	Class
selectDeposit()	selectDeposit()	deposit()	
selectDepositWithoutBankbook()	selectDepositWithoutBankbook()	depositWithoutBankbook()	
selectWithdraw()	selectWithdraw()	withdraw()	
selectTransfer()	selectTransfer()	transfer()	
selectExchange()	selectExchange()	exchange()	
selectLoan()	selectLoan()	loan()	
selectPayUtilityBill()	selectPayUtilityBill()	payUtilityBill()	MainSystem
selectCheckBalance()	selectCheckBalance()	checkBalance()	
selectForeignCurrency()	chooseForeignCurrency()	doForcedTermination()	
selectBank()	selectBank()	printError()	
insert()	insert()	insert()	
inputCash()	checkValidation()	printTransactionReceipt()	
inputAccountNumber()	updateDatabase()	printRecentTransaction()	
inputAmountOfMoney()	printError()	inputPassword()	
inputPassword()	doForcedTermination()	selectBank()	
validation()	printTransactionReceipt()	transactionLock()	Account
	inputCash()	checkValidation()	
	inputAccountNumber()	readDatabase()	
	transactionLock()	updateDatabase()	
	checkPassword()	checkPassword()	Offer
	readDatabase()	checkCreditRating()	
	takeCharge()	checkState()	
	inputPassword()		
	printRecentTransaction()		
	inputAmountOfMoney()		

Activity 2047. Define Database Schema

(생략)