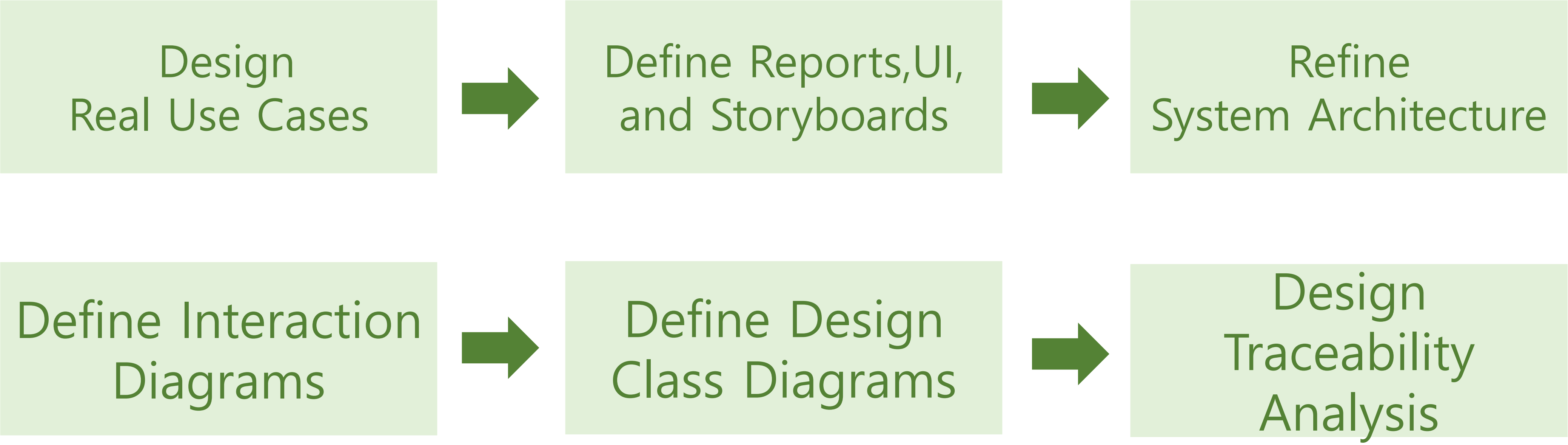
Software Modeling

A NEW RELIABLE ATM

OOPT STAGE 2040

201411140 권성완
201511247 김선정
201510436 허윤아
201510285 조수빈

CONTENTS



Design Real Use Cases

Define Reports,UI, and Storyboards

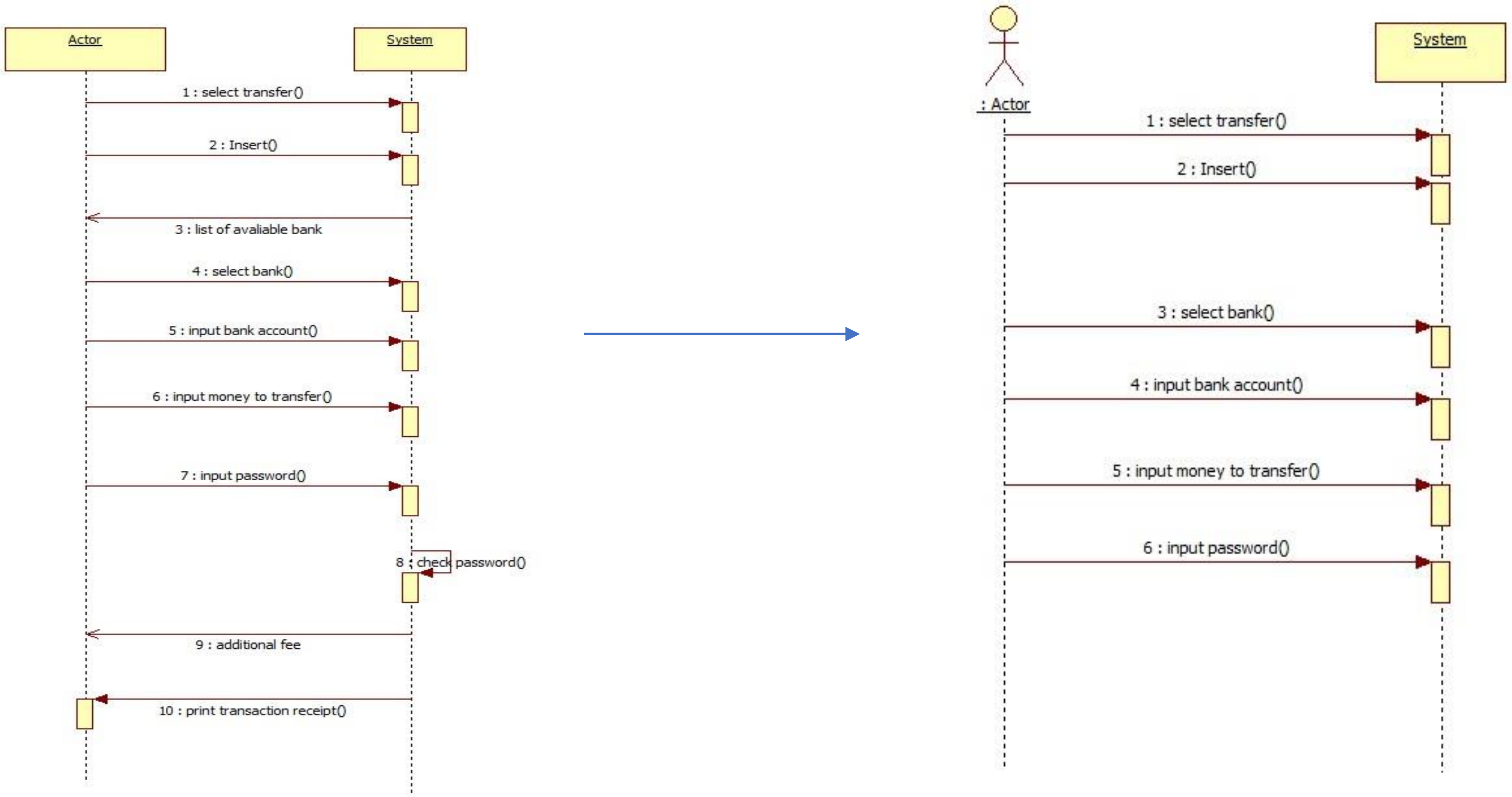
Refine System Architecture

Define Interaction Diagrams

Define Design Class Diagrams

Design Traceability Analysis

Revise


Actor
System
1 : select transfer()
2 : Insert()
3 : list of avaliable bank
4 : select bank()
5 : input bank account()
6 : input money to transfer()
7 : input password()
8 : check password()
9 : additional fee
10 : print transaction receipt()
: Actor
System
1 : select transfer()
2 : Insert()
3 : select bank()
4 : input bank account()
5 : input money to transfer()
6 : input password()

OOPT STAGE 01. Define 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.

OOPT STAGE 01. Define Real Use Cases

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.

OOPT STAGE 01. Define Real Use Cases

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"

OOPT STAGE 01. Define Real Use Cases

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

OOPT STAGE 01. Define Real Use Cases

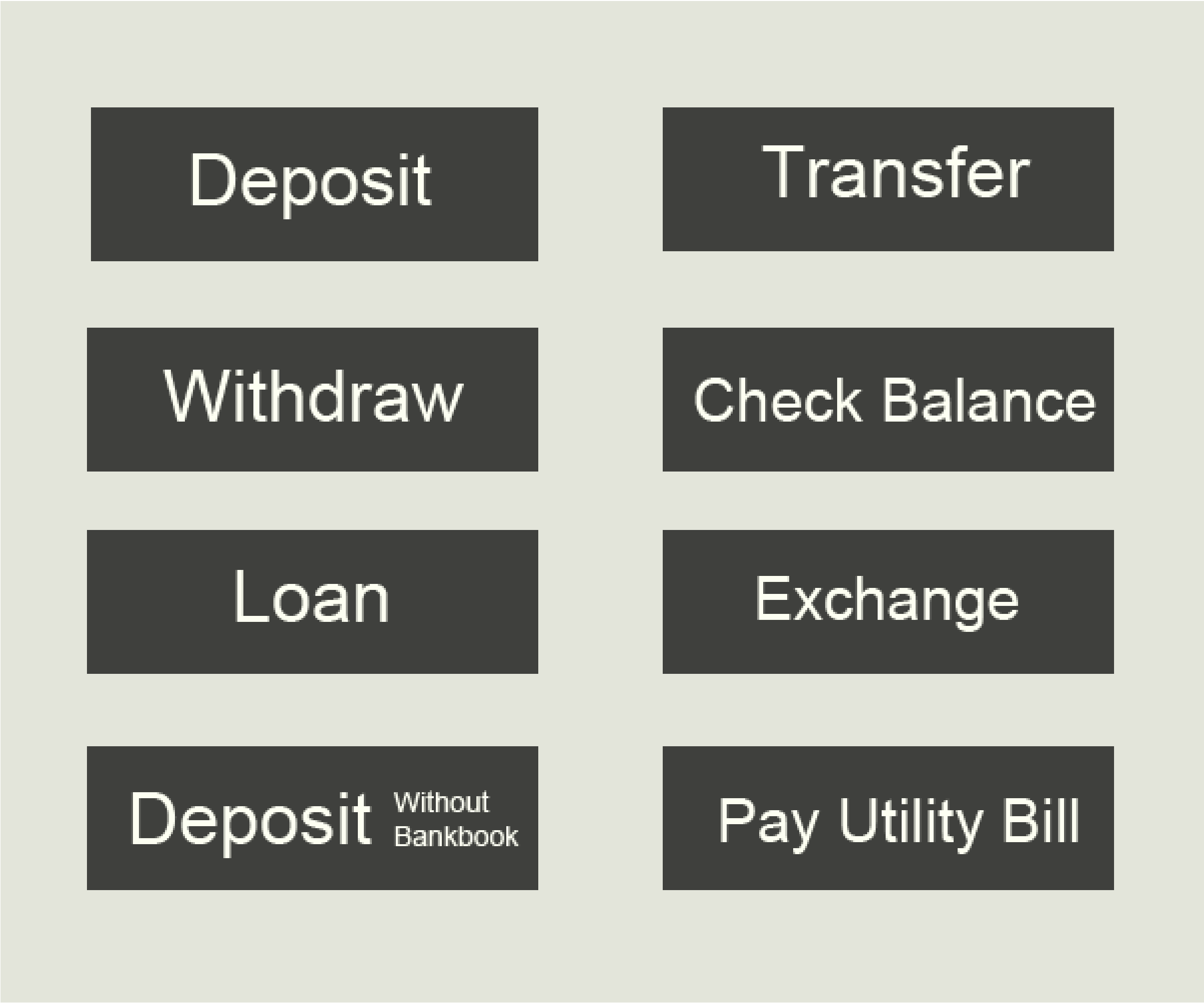
Use Case	5. Exchange
Actor	Customer
Purpose	Exchange KRW into foreign currency
Overview	Customer exchanges KRW in account into foreign currency.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.5, R2.1, R2.2, R2.3, R2.4, R3.1, R3.2, R3.3, R4.1 Use Case : Print Transaction Receipt, 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. ATM only handles unit of Yen, RMB, Dollar, Euro.
UI Widget	Window 1, 2-1, 4, 5, 6
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer selects exchange 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 list of countries available. 5. (A) Customer selects country to exchange money. 6. (S) Print out “input amount of money to exchange”. 7. (A) Customer inputs amount of money to exchange. 8. (S) Calculate total amount of money based on exchange rate and ask for password. 9. (A) Customer inputs password. 10. (S) Invoke “Check Password”. If password is correct, withdraw cash in foreign currency. 11. (S) If withdrew correctly, invoke “Print Transaction Receipt”. 12. (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, 9 : If information or password is incorrect over 3 times, invoke “Forced Termination”. Line 10 : Invoke “Take Charge”. Charge is deducted from balance.

OOPT STAGE 01. Define Real Use Cases

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	Functional Requirements : R1.6, R2.1, R2.2, R2.3, R2.4, R3.1, R3.2, R3.3, R3.4, R4.1 Use Case : Print Transaction Receipt, Take Charge, Print Error, Forced Termination, Request Customer's Data, Insert, Check Password, Check Credit, Update Server Information
Pre-Requisites	Customer should know password for the credit card.
UI Widget	Window 1,4,6
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) Customer chooses loan menu from basic screen. 2. (S) Print out "Input credit card". 3. (A) Customer inputs credit card. 4. (S) Invoke "Insert". If user information is valid, Invoke "Check Credit". Print out "input amount of money to loan". 5. (A) Customer inputs amount of money to loan in unit of 10000₩ and 50000₩. 6. (S) print out "input password for the credit card". 7. (A) Customer inputs password for the credit card. 8. (S) Invoke "Check Password". If password is correct, count number of bills. 9. (S) Invoke "Request Customer's Data". If total amount of money to loan is under credit card limit, 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 5 : If total amount of money to withdraw is over 50000₩, customer inputs number of 50000₩ bill. Line 4, 8 : If information or password is incorrect over 3 times, invoke "Forced Termination". Line 9 : : Invoke "Take Charge".

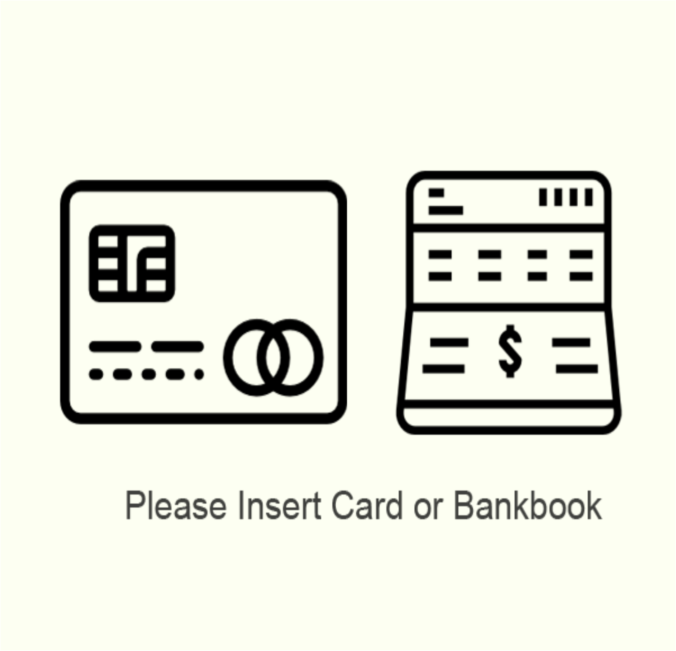
OOPT STAGE 02. Define UI

Window 1



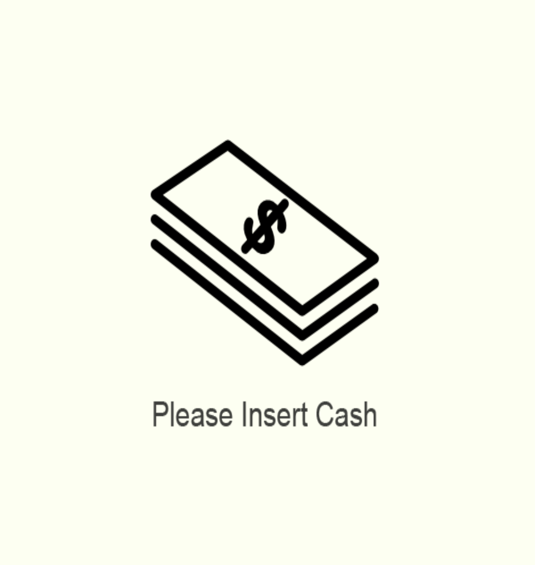
OOPT STAGE 02. Define UI

Window 2



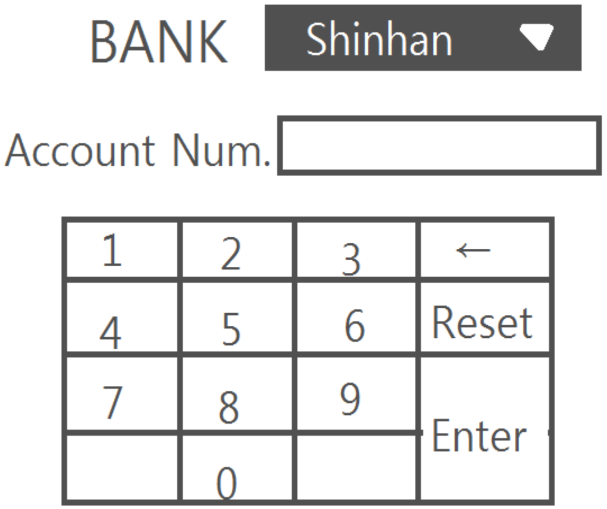
Please Insert Card or Bankbook

Window 2-2



Please Insert Cash

Window 3



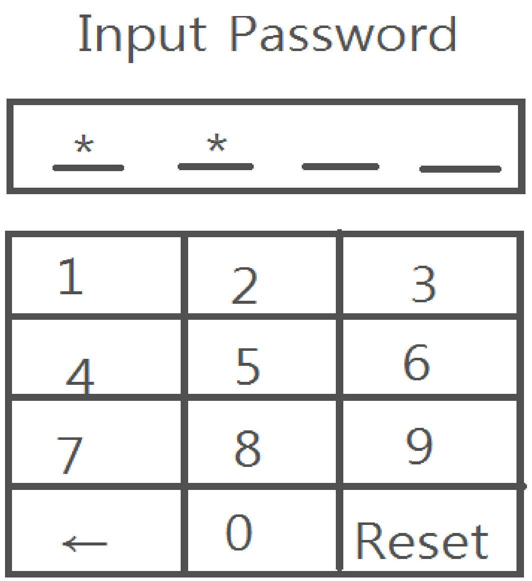
BANK Shinhan ▼

Account Num.

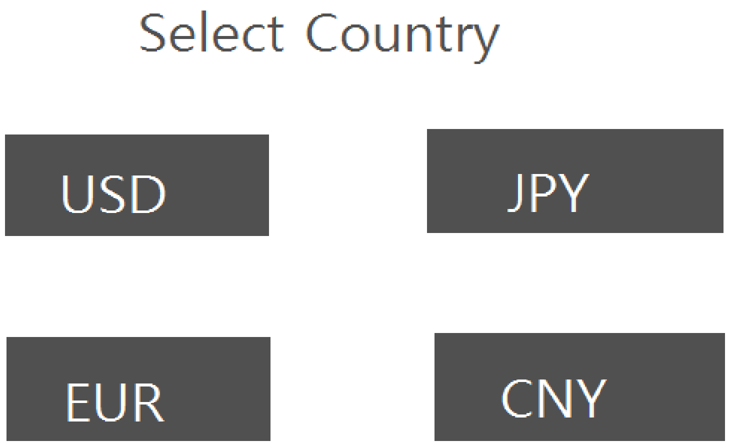
1	2	3	←
4	5	6	Reset
7	8	9	Enter
	0		

OOPT STAGE 02. Define UI

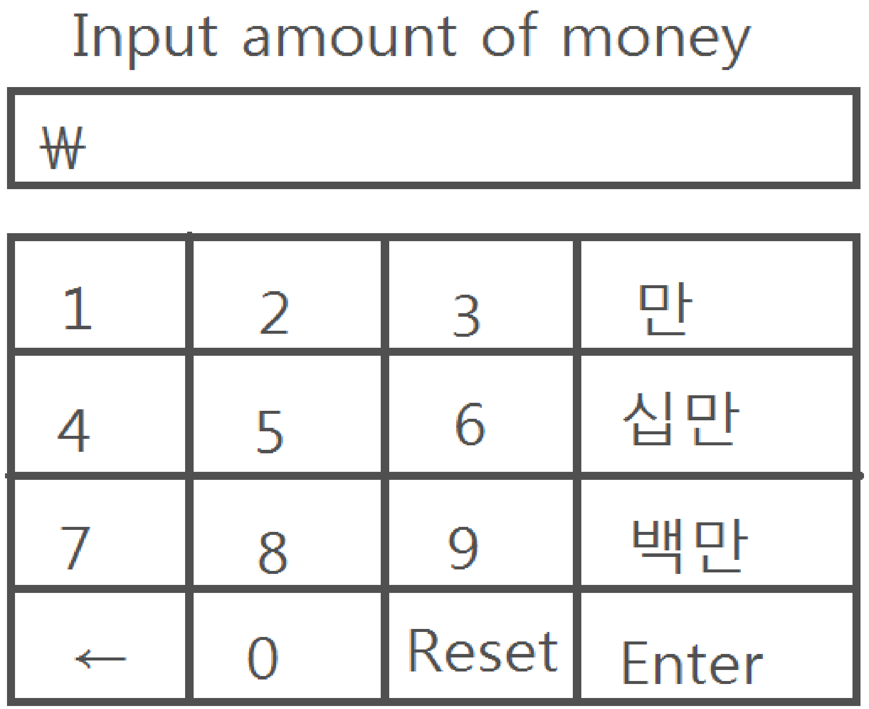
Window 4



Window 5



Window 6



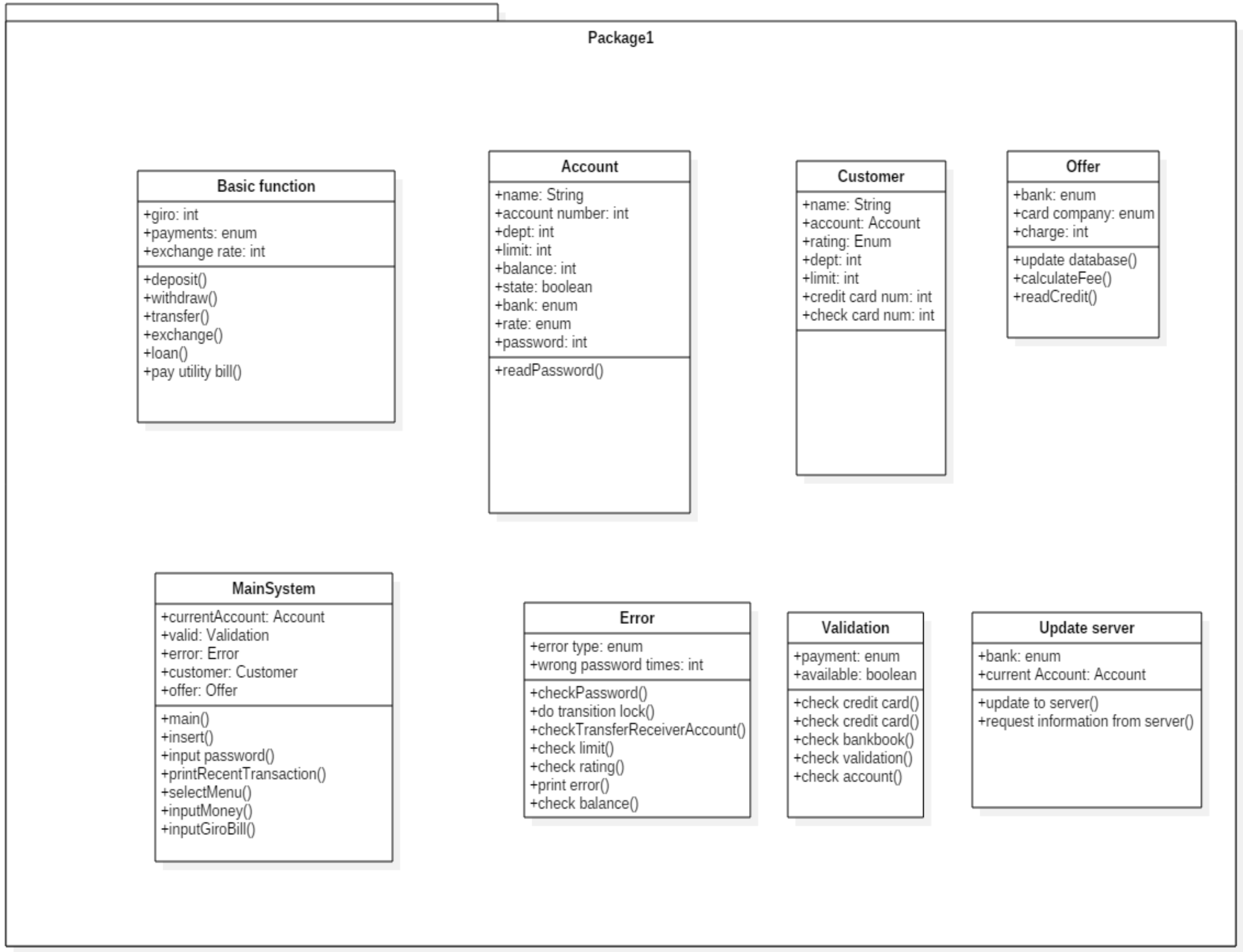
Input amount of money

₩

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

A NEW RELIABLE ATM

OOPT STAGE 03. Refine System Architecture



Package1

Basic function

+giro: int
+payments: enum
+exchange rate: int

+deposit()
+withdraw()
+transfer()
+exchange()
+loan()
+pay utility bill()

Account

+name: String
+account number: int
+dept: int
+limit: int
+balance: int
+state: boolean
+bank: enum
+rate: enum
+password: int

+readPassword()

Customer

+name: String
+account: Account
+rating: Enum
+dept: int
+limit: int
+credit card num: int
+check card num: int

Offer

+bank: enum
+card company: enum
+charge: int

+update database()
+calculateFee()
+readCredit()

MainSystem

+currentAccount: Account
+valid: Validation
+error: Error
+customer: Customer
+offer: Offer

+main()
+insert()
+input password()
+printRecentTransaction()
+selectMenu()
+inputMoney()
+inputGiroBill()

Error

+error type: enum
+wrong password times: int

+checkPassword()
+do transition lock()
+checkTransferReceiverAccount()
+check limit()
+check rating()
+print error()
+check balance()

Validation

+payment: enum
+available: boolean

+check credit card()
+check credit card()
+check bankbook()
+check validation()
+check account()

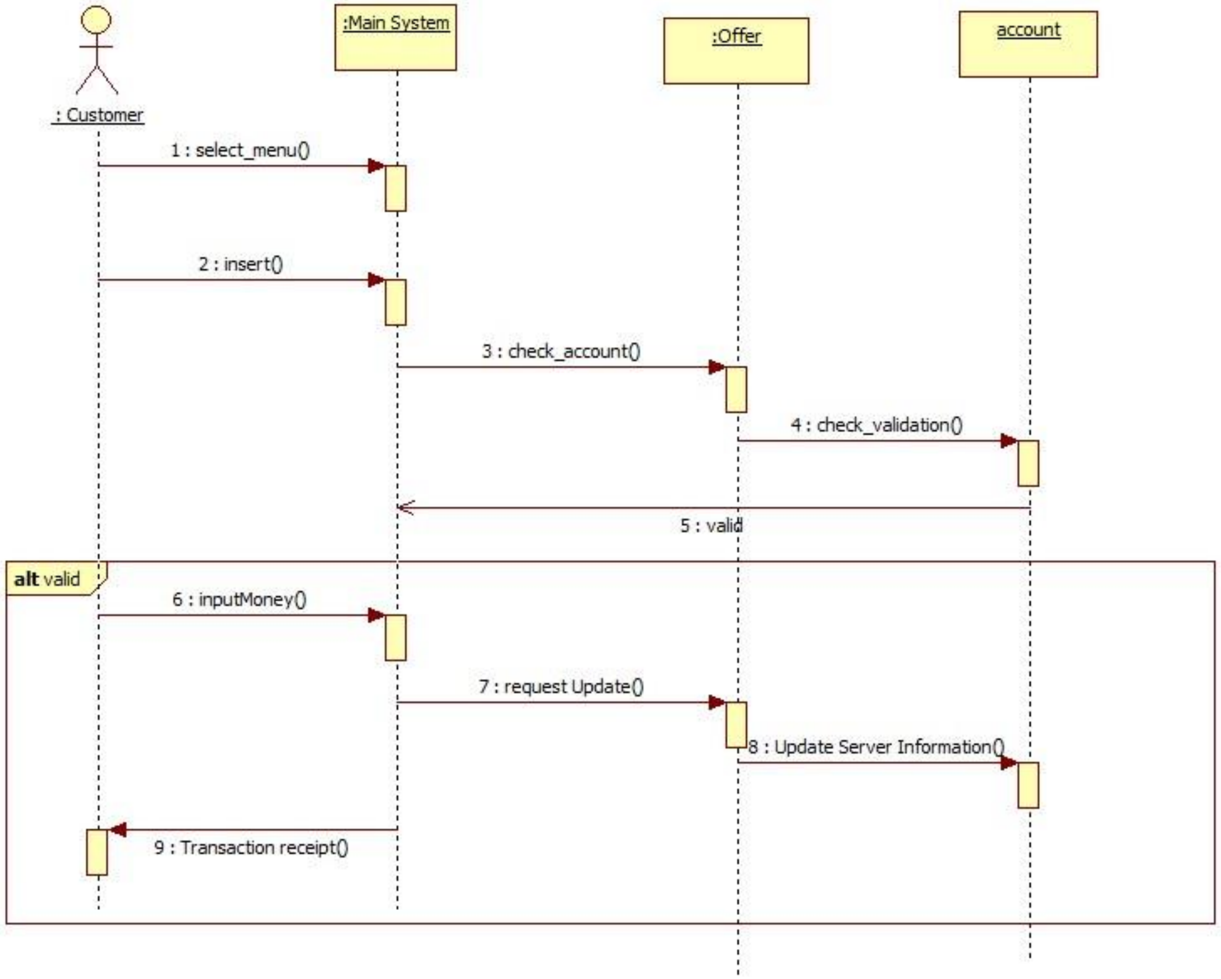
Update server

+bank: enum
+current Account: Account

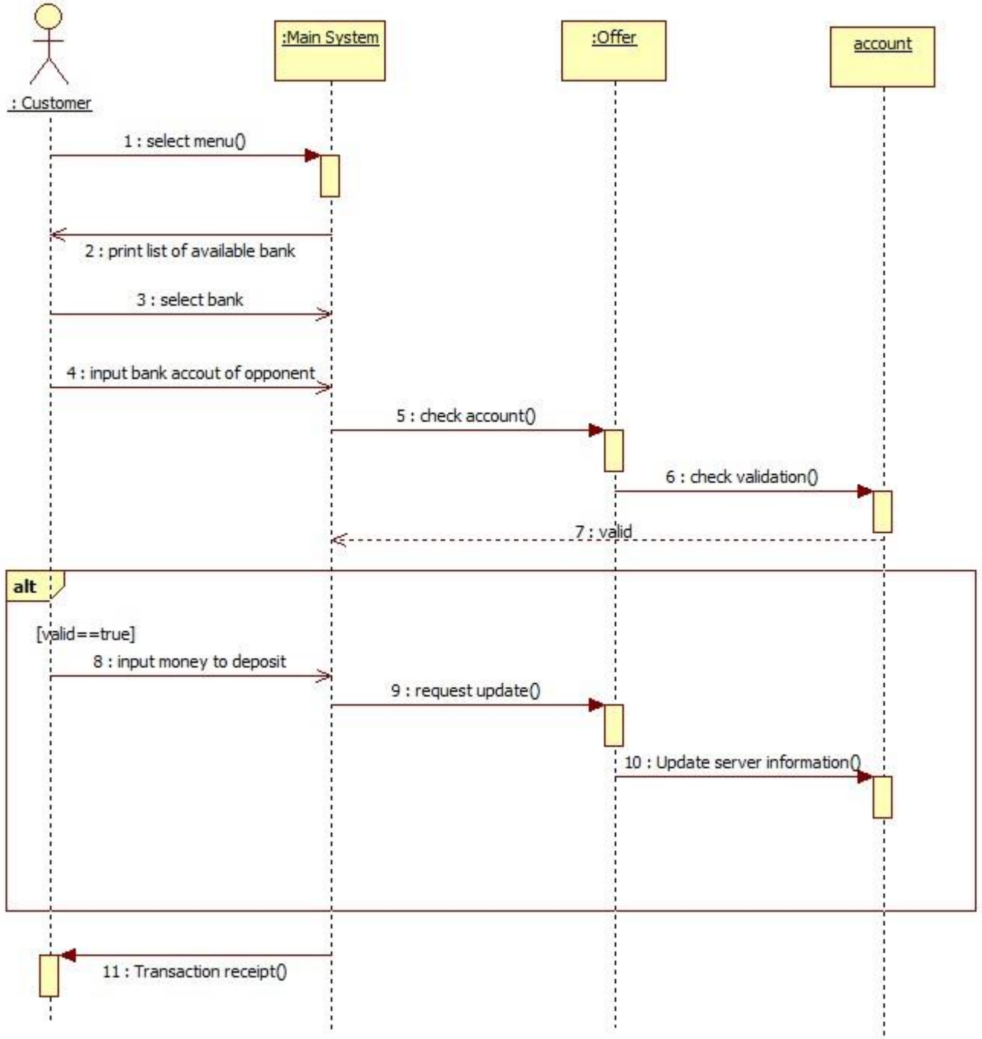
+update to server()
+request information from server()

OOPT STAGE 04. Define Interaction Diagrams

Deposit

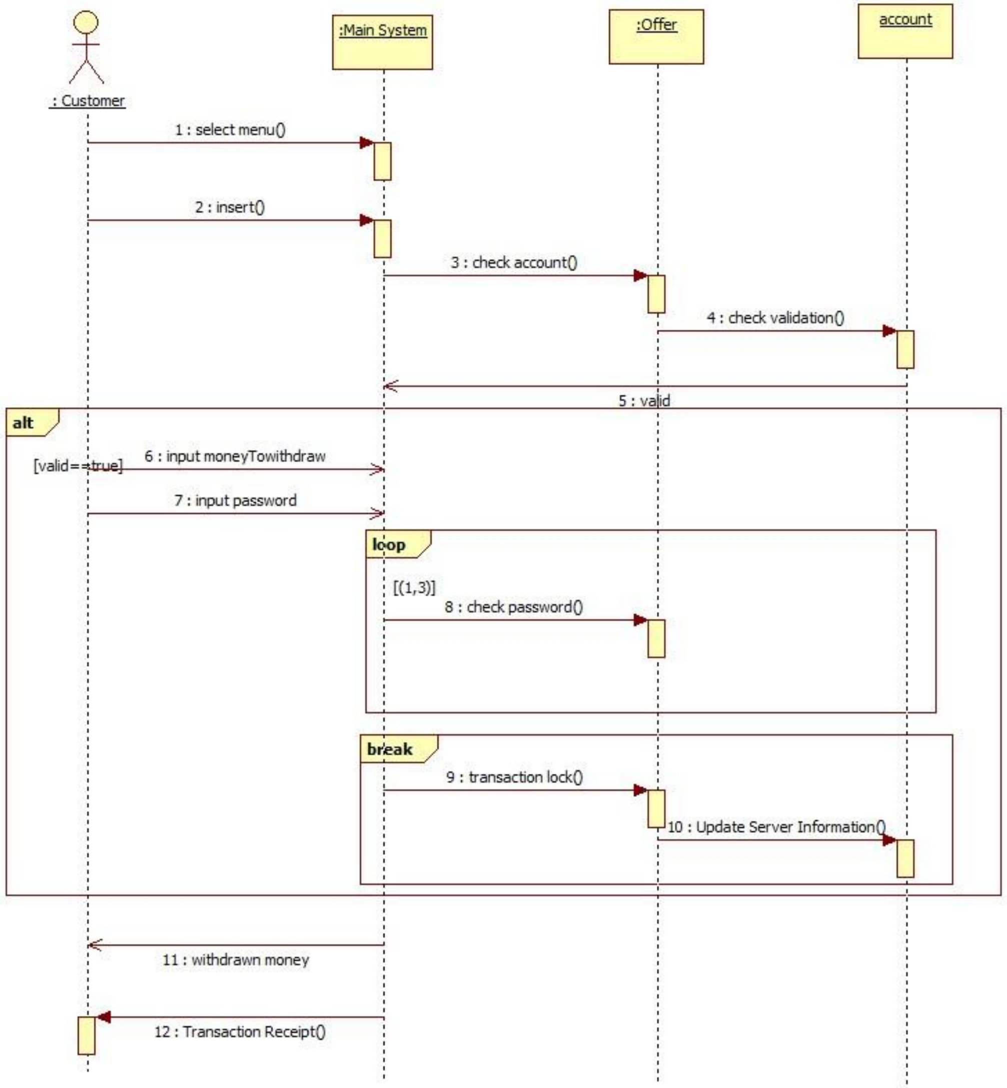


OOPT STAGE 04. Define Interaction Diagrams



Deposit without bankbook

OOPT STAGE 04. Define Interaction Diagrams

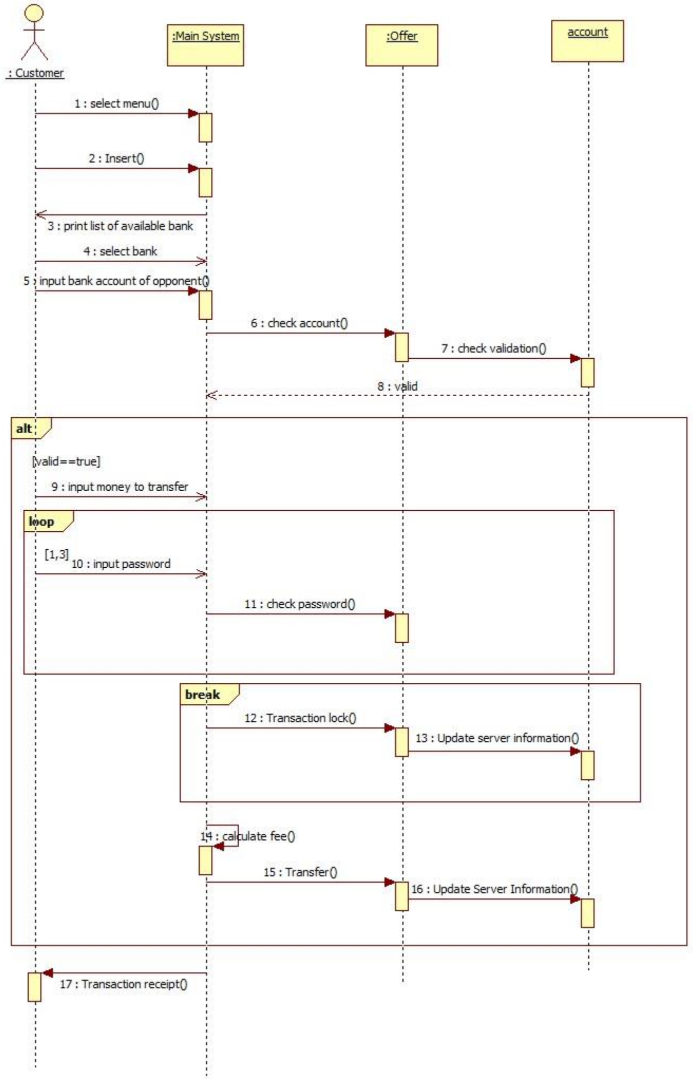


Withdraw

A NEW RELIABLE ATM

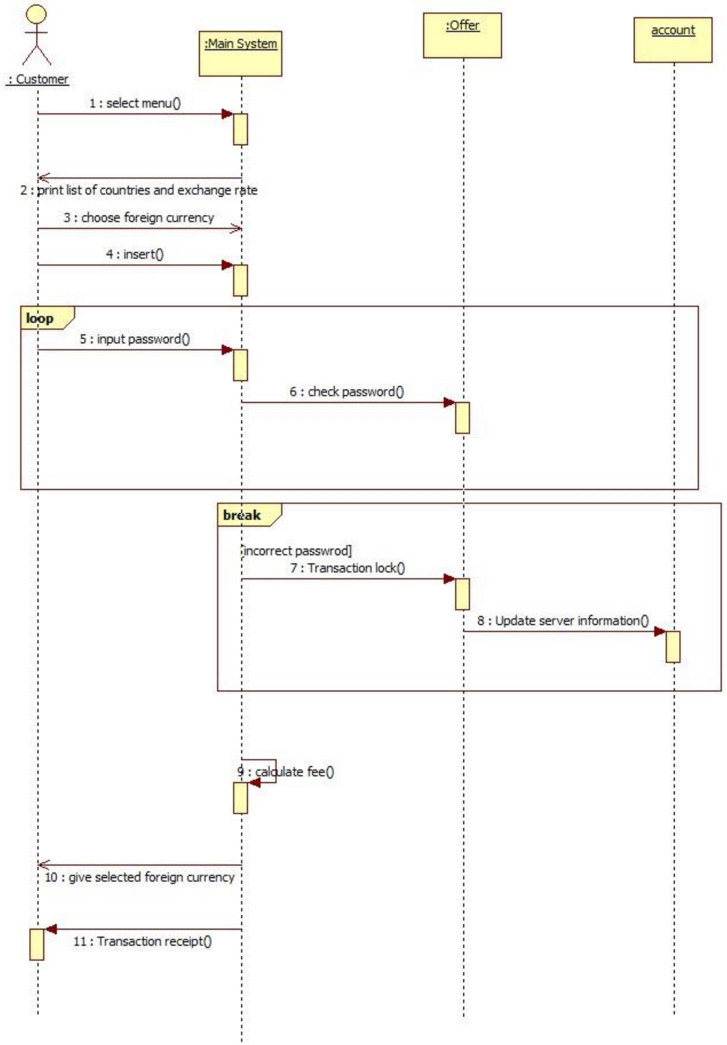
OOPT STAGE 04. Define Interaction Diagrams

Transfer

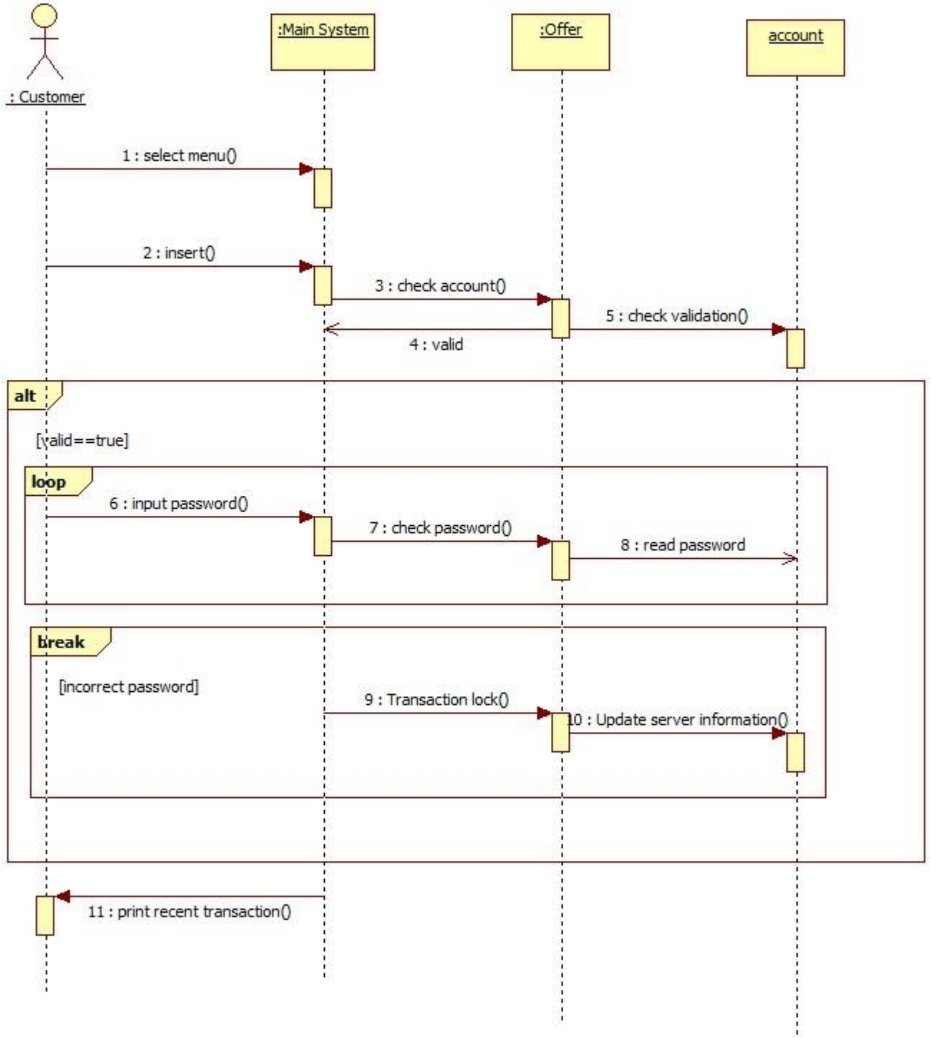


OOPT STAGE 04. Define Interaction Diagrams

Exchange



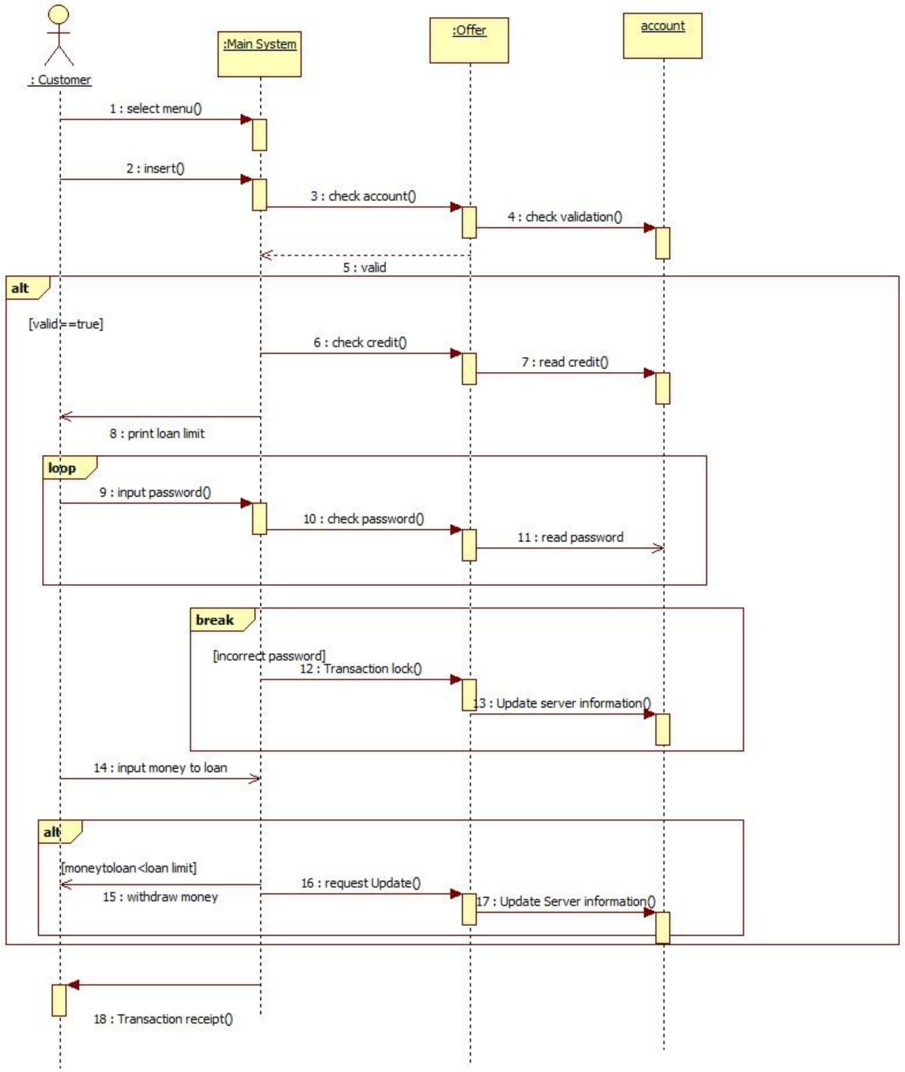
OOPT STAGE 04. Define Interaction Diagrams



Check balance

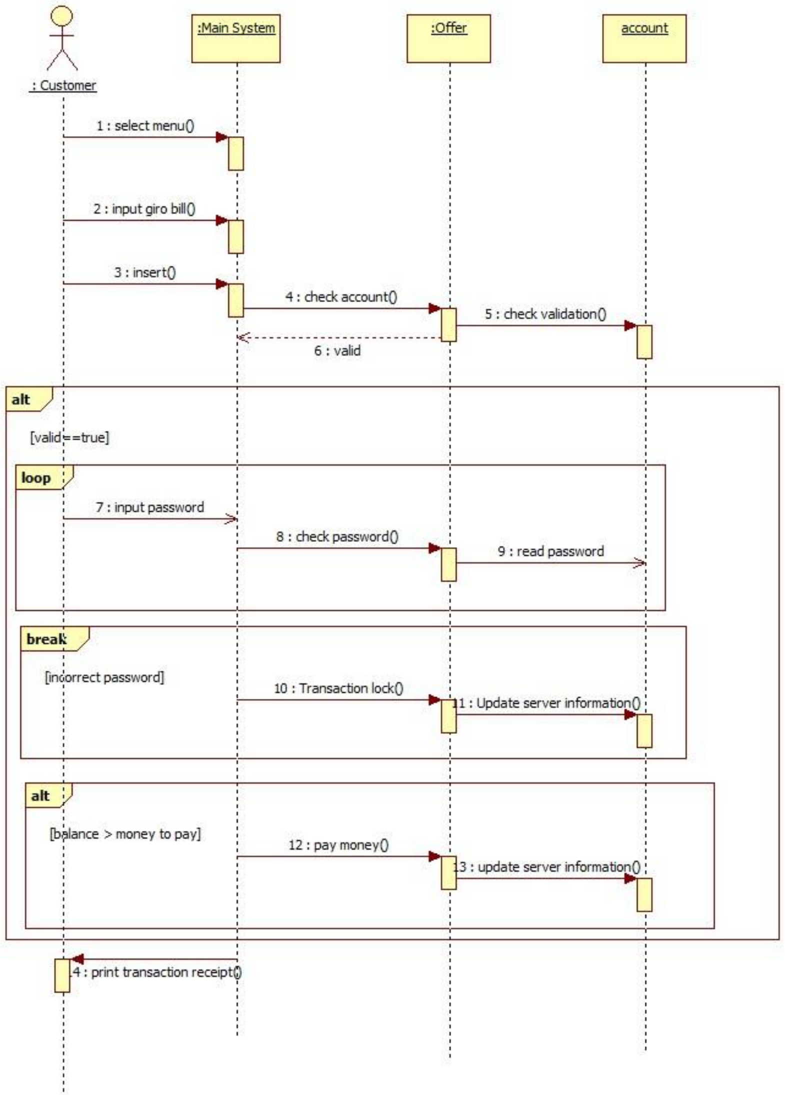
OOPT STAGE 04. Define Interaction Diagrams

Loan



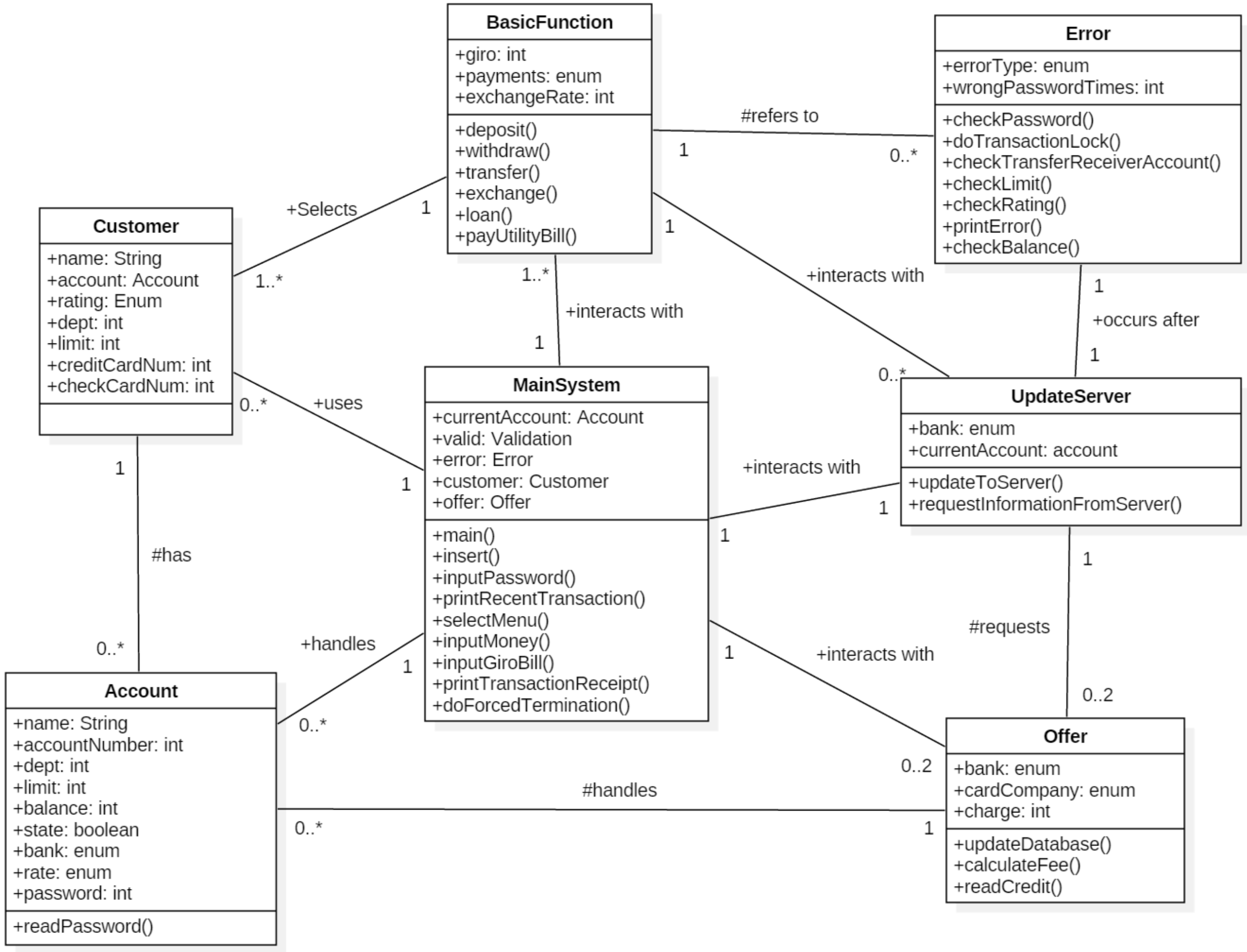
A NEW RELIABLE ATM

OOPT STAGE 04. Define Interaction Diagrams



Pay utility bill

OOPT STAGE 05. Define design class diagrams



BasicFunction
+giro: int
+payments: enum
+exchangeRate: int

+deposit()
+withdraw()
+transfer()
+exchange()
+loan()
+payUtilityBill()

Error
+errorType: enum
+wrongPasswordTimes: int

+checkPassword()
+doTransactionLock()
+checkTransferReceiverAccount()
+checkLimit()
+checkRating()
+printError()
+checkBalance()

Customer
+name: String
+account: Account
+rating: Enum
+dept: int
+limit: int
+creditCardNum: int
+checkCardNum: int

MainSystem
+currentAccount: Account
+valid: Validation
+error: Error
+customer: Customer
+offer: Offer

+main()
+insert()
+inputPassword()
+printRecentTransaction()
+selectMenu()
+inputMoney()
+inputGiroBill()
+printTransactionReceipt()
+doForcedTermination()

UpdateServer
+bank: enum
+currentAccount: account

+updateToServer()
+requestInformationFromServer()

Account
+name: String
+accountNumber: int
+dept: int
+limit: int
+balance: int
+state: boolean
+bank: enum
+rate: enum
+password: int

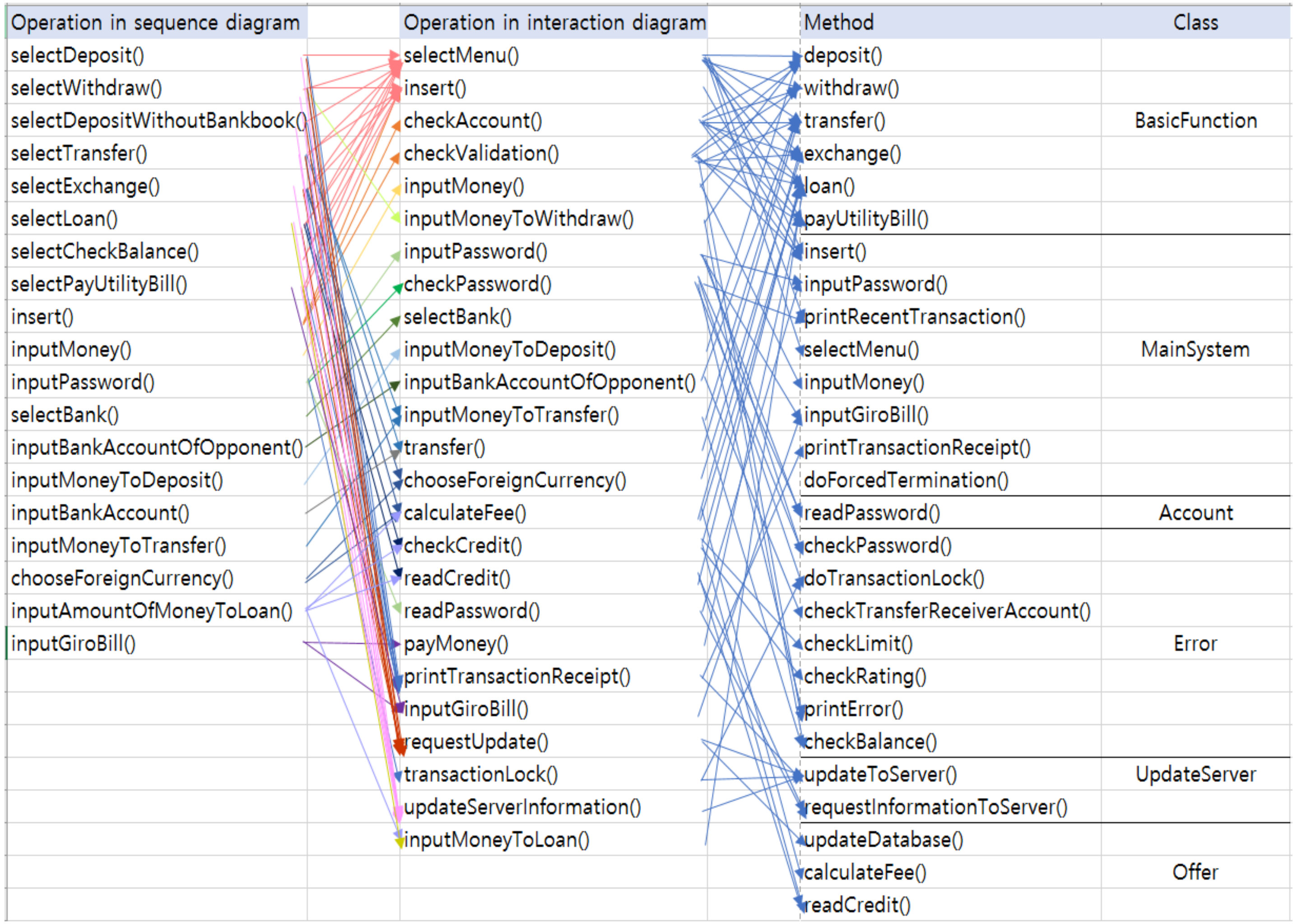
+readPassword()

Offer
+bank: enum
+cardCompany: enum
+charge: int

+updateDatabase()
+calculateFee()
+readCredit()

A NEW RELIABLE ATM

OOPT STAGE 06. 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()	