A New Reliable ATM

OOPT Phase 2030

Analyze

Project Team T6

201411140 권성완
201511247 김선정
201510436 허윤아
201510285 조수빈

Date

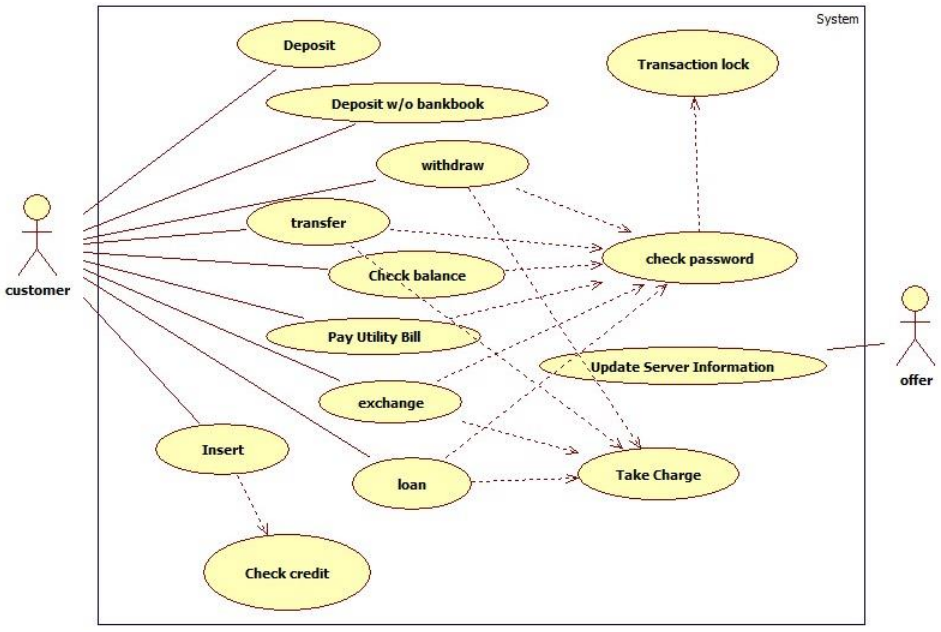
2018-05-13

Activity 2010. Revise Plan

1. OOPT 1000 version 2 중 Activity 1003의 functional requirements

Ref. #	Function
R1.1	Deposit
R1.2	Withdraw
R1.3	Deposit Without bankbook
R1.4	Transfer
R1.5	Exchange
R1.6	Loan
R1.7	Check Balance
R1.8	Pay Utility Bill
R2.1	Print Transaction Receipt
R2.2	Take Charge
R2.3	Print Error
R2.4	Proceed Forced Termination
R2.5	Transaction Lock
R3.1	Request Customer's Data
R3.2	Check Validation
R3.3	Check Password
R3.4	Check Credit
R4.1	Update Server Information

2. OOPT 1000 version 2 중 Activity 1006의 use case diagram



Activity 2020. Synchronize Artifacts

- OOPT Stage 1000이 version 2로 수정되면서 Functional requirement와 use case가 추가 및 수정되고 use case diagram이 이에 맞춰 수정되었다. Test case plan 또한 추가되었다.

Activity 2031. Define Essential 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
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.
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.
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.
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".

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.
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.

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.
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”.

Use Case	7. Check balance
Actor	Customer
Purpose	Check balance of account
Overview	Customer checks balance of account.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.7, R2.3, R2.4, R3.1, R3.2, R3.3 Use Case : Print Error, Forced Termination, Request Customer’s Data, Insert, Check Password
Pre-Requisites	Customer should know password for the account.
Typical Courses of Events	(A) : Actor, (S) : System 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
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”.

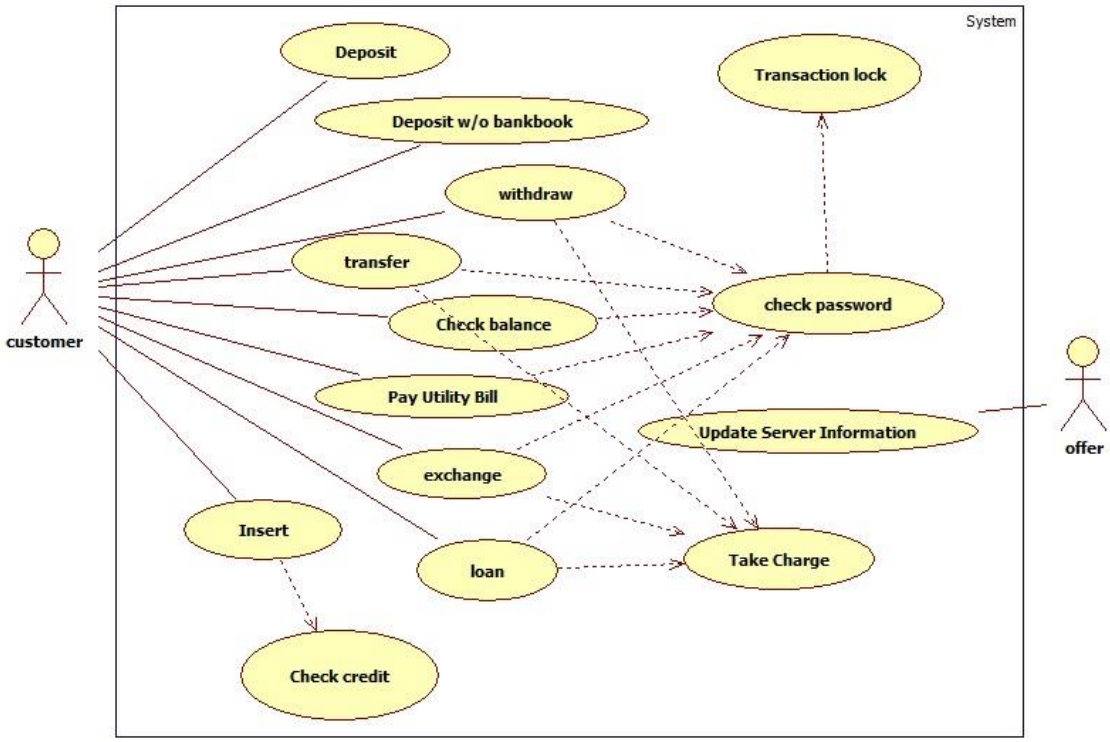
Use Case	16. Check Password
Actor	(None)
Purpose	Check password of account or credit card
Overview	System checks password of account or credit card inputted by Customer.
Type	Primary and Essential
Cross Reference	Functional Requirements : R1.2, R1.4, R1.5, R1.6, R1.7, R1.8, R2.4, R2.5, R3.3, R4.1 Use 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 2032. Refine Use Case Diagrams



Activity 2033. Define Domain Model

1. List concepts(domain class) from use-cases

- Guideline 1

Concept Category	Examples
Physical or tangible objects	Credit Card, Check Card, Bankbook ,Cash, Utility Bill, Exchange Rate, Balance, Receipt
Customer's Data	Name, Bank, Rating, Account number, Password, Dept(loan), Limit
Exchange Rate	USD, JPY, EUR, CNY, KRW
Basic function	Deposit, Withdraw, Deposit Without Bankbook, Transfer, Exchange, Loan, Check balance, Pay utility bill
Dealing Error	Print Error, Forced Termination, Transaction Lock
Dealing Validation & Update	Check Password, Check Credit, Insert, Update Server Information
Account State	Lock, Normal
Actor	Customer, Offer
Need to take charge	Transfer, Exchange, Loan

- Guideline 2 ; Using Noun Phrases

Deposit	Withdraw	Deposit Without Bankbook	Transfer	Loan
Transaction Receipt	Charge	Utility Bill	Balance	Exchange
Cash	Credit Card	Check Card	Bank	Customer
Offer	Customer Information	Validation	Limit	Password
Account Number	Giro	receiver(transfer)	sender(transfer)	Error
Charge	ATM	Lock	Update Server	payment

2. Assign class name into a concept

Offer	Customer	Basic function	Error
Update Server	Account	Validation	

3. Draw a conceptual class diagram

Offer	Customer	Basic function	Error
Update Server	Account	Validation	

4. Identify associations according to association categories

A selects B	Customer – Basic Function
A requests to B	Update Server – Offer
A refers to B	Basic function – Error Basic function – Validation Error – Validation
A occurs after B	Update server - Basic function Update Server - Error
A handles B	Offer – Account
A has B	Customer - Account

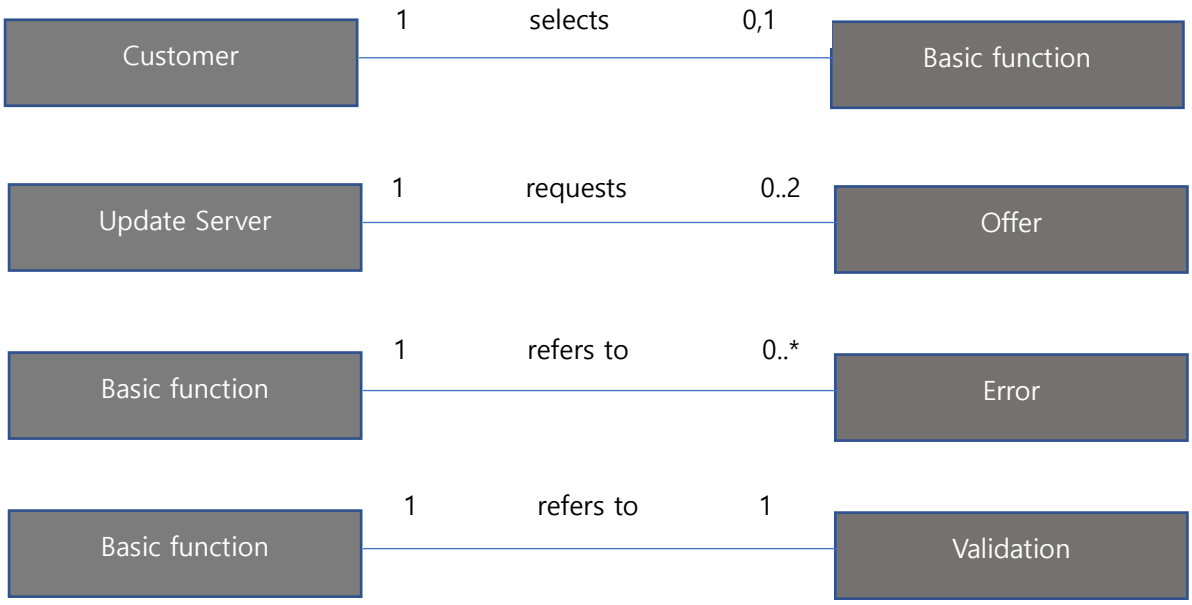
5. Assign priority into associations

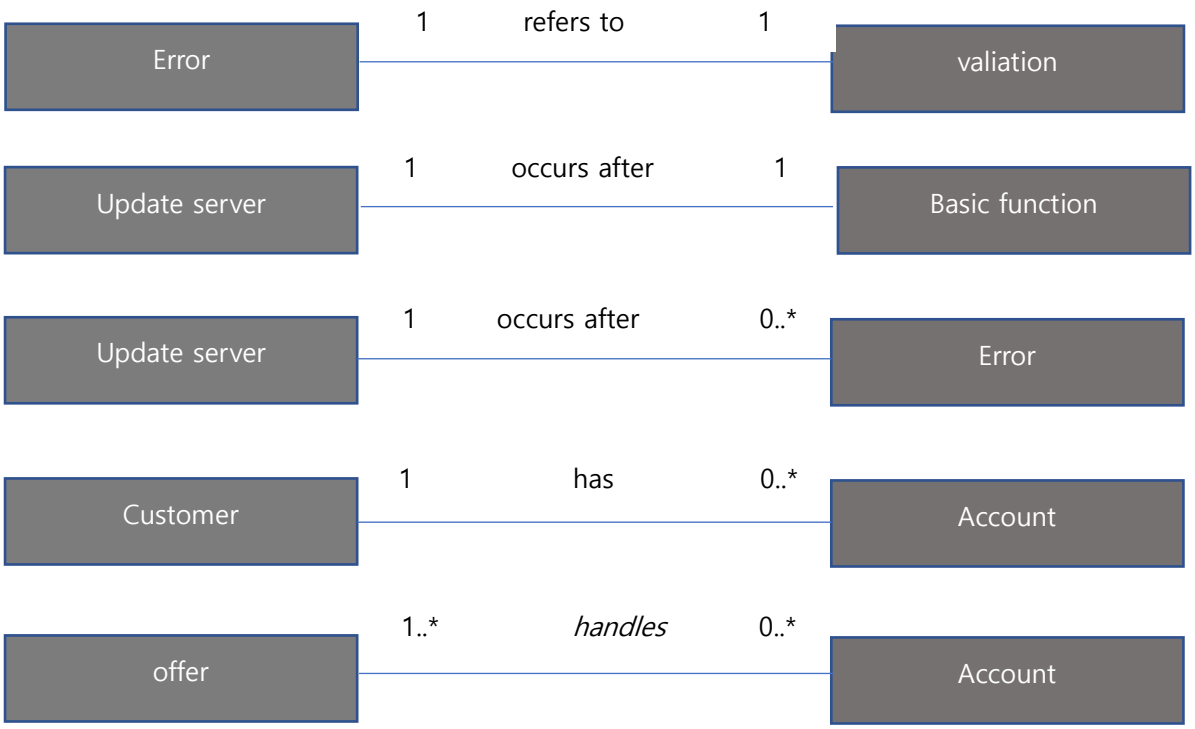
Customer – Basic Function	High
Basic Function – Error	High
Basic Function – Validation	High

6. Assign names into associations

- Customer *selects* Basic Function
- Update Server *requests* to Offer
- Basic function *refers to* Error
- Basic function *refers to* Validation
- Error *refers to* Validation
- Update server *occurs after* Basic function
- Update Server *occurs after* Error
- Offer *handles* Account

7. Add roles and Multiplicity





8. Add Attributes

Customer
Name: String Account: Account Rating: Enum Dept: int Limit: int credit card num: int check card num: int

Basic function
giro: int payments: enum Exchange Rate: int

Update server
Bank: enum Current Account: Account

Validation
payment: enum available: boolean

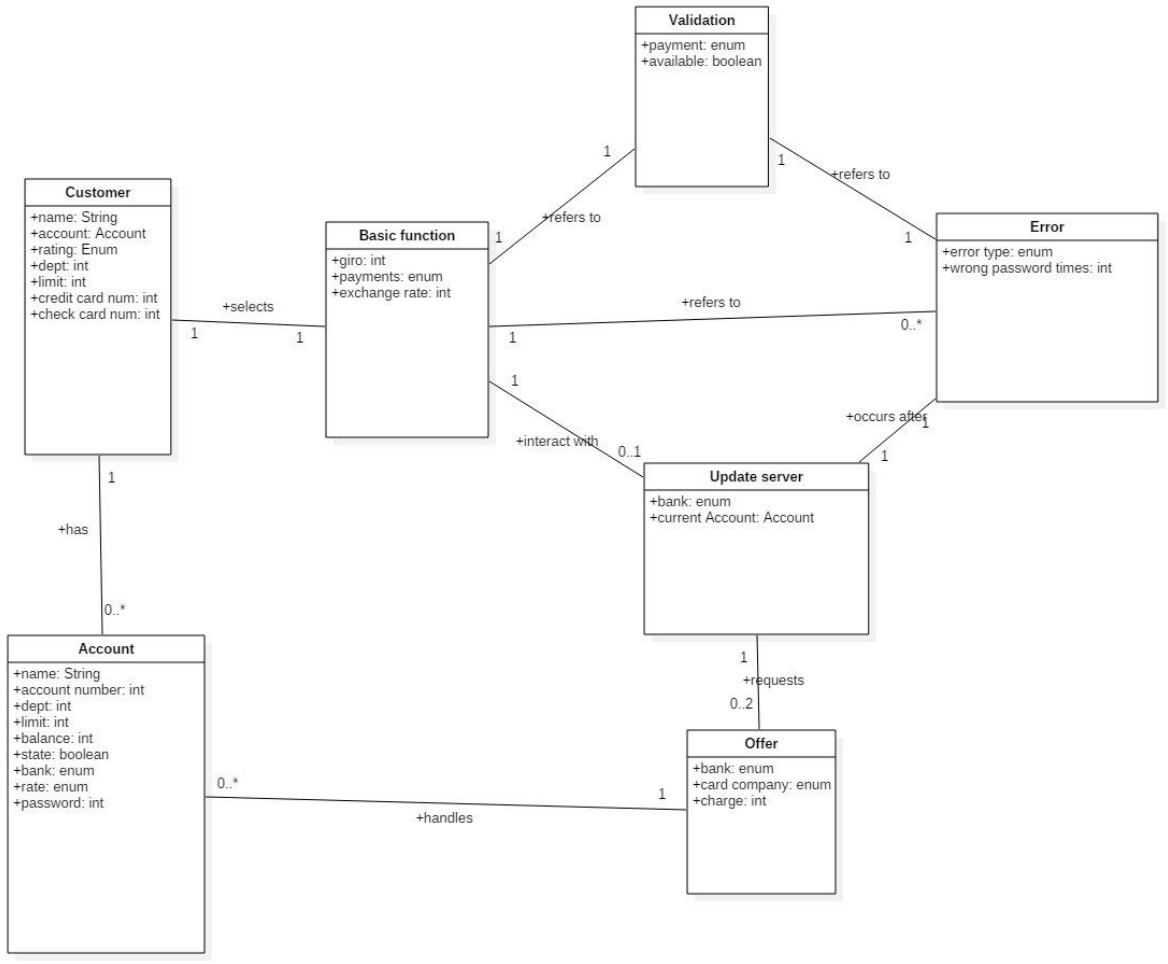
Offer
Bank: enum Card company: enum charge: int

Error
Error type: enum

Account
name: String Account number: int

	dept: int limit: int balance: int state: boolean bank: enum rate: enum password: int

9. Define Domain Model

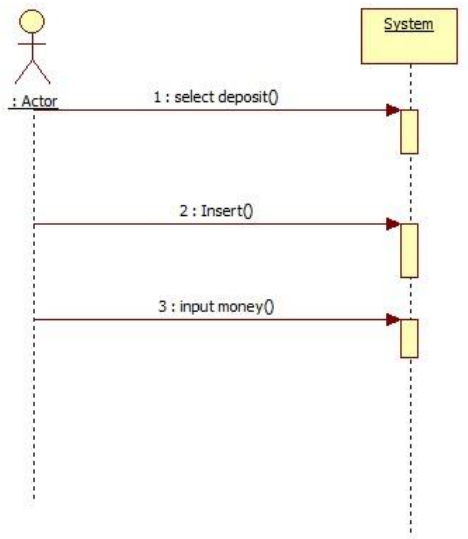


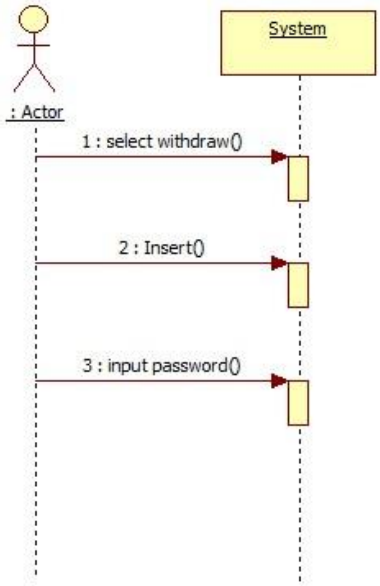
Activity 2034. Refine Glossary

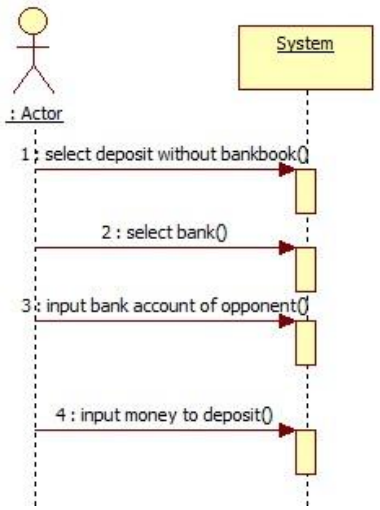
Term	Category	Remarks
Customer	Class	The person who has accounts
Basic function	Class	functions that customer can uses(select)

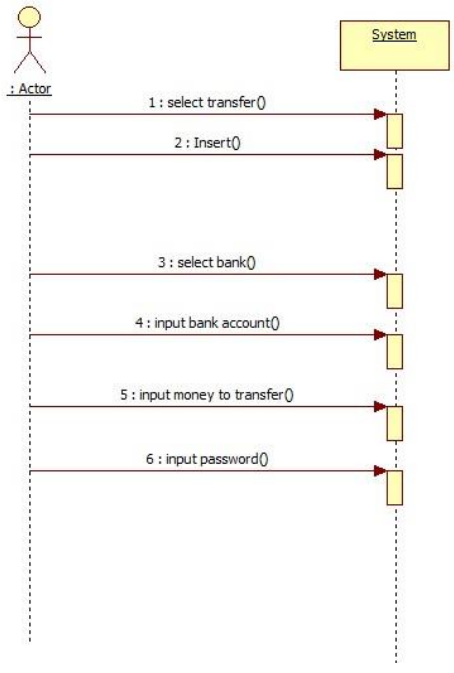
Account	Class	Account information
Offer	Class	Deal with all customer's accounts and credit card information
Update server	Class	ATM requests Offer to update customer's information which has changed by basic function
Error	Class	Deal with errors
Validation	Class	Deal with validations
account num	Attribute	account number(bankbook number)
check card num	Attribute	check card number
credit card num	Attribute	credit card number
rating	Attribute	credit rating(VIP, Gold, Silver)
Dept	Attribute	amount of dept(occurs by loan)
Limit	Attribute	loan limit
giro	Attribute	has virtual account number
payments	Attribute	credit card/ check card/ cash
exchange rate	Attribute	USD, JPY, EUR, CNY, KRW
bank	Attribute	bank company name
card company	Attribute	card company name
charge	Attribute	commission
error type	Attribute	password inconsistency, transaction lock, over limit, not enough balance, not available transfer receiver…
available	Attribute	check account is available
name	Attribute	customer or account owner's name
balance	Attribute	amount of money that account has
password	Attribute	password
state	Attribute	Account lock/ normal

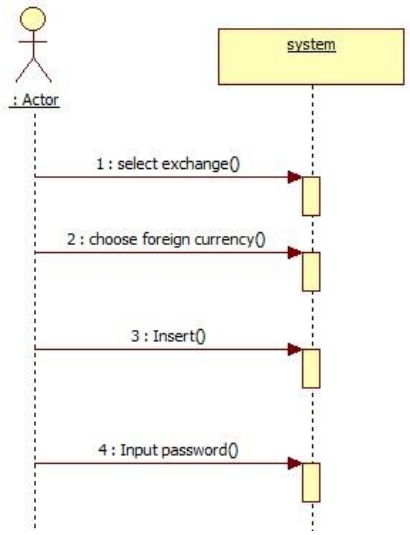
Activity 2035. Define System Sequence Diagrams

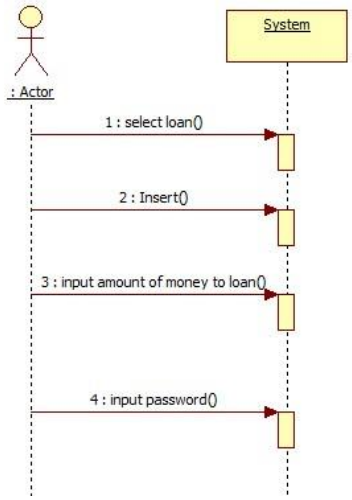
Use Case : Deposit	
1. Customer selects deposit menu from the basic screen. 2. System prints out "Input check card or bankbook or credit card". 3. Customer inputs credit card, check card or bankbook. 4. If user information is valid(from "Insert" and "Request Customer's Data"), ask customer to input cash. 5. Customer Inputs cash in unit of 10000₩ and 50000₩. 6. System checks amount of cash inputted. 7. If total amount of money is correct, system invokes "Print transaction receipt". 8. (S) If Transaction Receipt is successfully printed, ask Offer to update server information.	

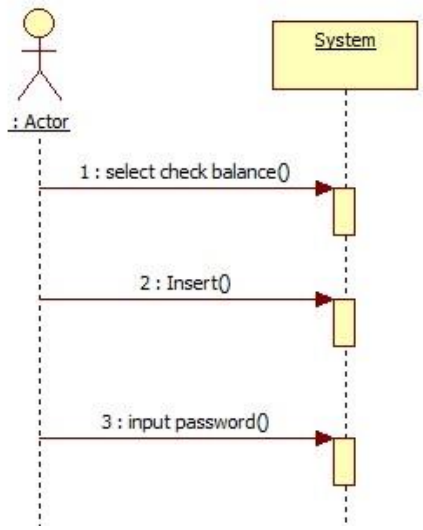
Use Case : Withdraw	
1. Customer selects withdraw menu from the basic screen. 2. System prints out "Input check card or bankbook". 3. Customer inputs check card or bankbook. 4. If user information is valid(from "Insert"), System asks customer to input amount of money to withdraw. 5. Customer inputs amount of money in unit of 10000₩ and 50000₩ to withdraw from account. 6. System asks for password for the account. 7. Customer inputs password for the account. 8. Invoke "Check Password". If password is correct, System counts numbers of bills. 9. Invoke "Request Customer's Data". If balance is enough, System withdraws cash and invokes "Print Transaction Receipt". 10. If Transaction Receipt is successfully printed, System asks Offer to update server information.	

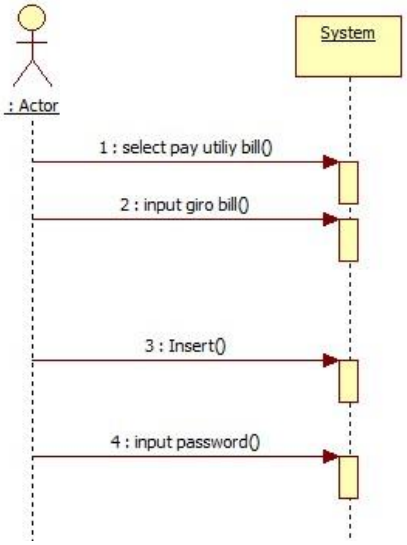
Use Case : Deposit without Bankbook	
1. Customer chooses Deposit without Bankbook menu from screen. 2. System asks customer to input bank account number to deposit. 3. Customer inputs bank account number. 4. If bank account number is valid, System asks customer to input cash in unit of 10000₩ and 50000₩. 5. System checks total amount of cash. 6. If counted right, invoke "Print Transaction Receipt". 7. If Transaction Receipt is successfully printed, System asks Offer to update server information.	

Use Case : Transfer	
1. Customer selects transfer menu from the basic screen. 2. System prints out "Input check card or bankbook". 3. Customer inputs check card or bankbook. Invoke "Insert". 4. If user information is valid, System asks customer to input bank and account number to transfer money. 5. Customer inputs bank and account number to transfer money. 6. If inputted information is valid, System asks customer to input amount of money to transfer. 7. Customer inputs amount of money to transfer. 8. Invoke "Request Customer's Data". If balance is enough, System asks for password for the account. 9. Customer inputs password for the account. 10. Invoke "Check Password". If password is correct, invoke "Print Transaction Receipt". 11. If Transaction Receipt is successfully printed, System asks Offer to update server information.	

Use Case : Exchange	
1. Customer selects exchange menu from basic screen. 2. System prints out "input check card or bankbook". 3. Customer inputs check card or bankbook. 4. Invoke "Insert". If user information is valid, System prints out list of countries available. 5. Customer selects country to exchange money. 6. System prints out "input amount of money to exchange". 7. Customer inputs amount of money to exchange. 8. System calculates total amount of money based on exchange rate and asks for password. 9. Customer inputs password. 10. Invoke "Check Password". If password is correct, withdraw cash in foreign currency. 11. If withdrew correctly, invoke "Print Transaction Receipt". 12. If Transaction Receipt is successfully printed, System asks Offer to update server information.	

Use Case : Loan	
1. Customer chooses loan menu from basic screen. 2. System prints out "Input credit card". 3. Customer inputs credit card. 4. Invoke "Insert". If user information is valid, Invoke "Check Credit". Print out "input amount of money to loan". 5. Customer inputs amount of money to loan in unit of 10000₩ and 50000₩. 6. System prints out "input password for the credit card". 7. Customer inputs password for the credit card. 8. Invoke "Check Password". If password is correct, count number of bills. 9. 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. If Transaction Receipt is successfully printed, System asks Offer to update server information.	

Use Case : Check Balance	
1. Customer chooses check balance menu from basic screen. 2. System prints out "input check card or bankbook". 3. Customer inputs check card or bankbook. 4. Invoke "Insert". If user information is valid, System prints out "input password for the account". 5. Customer inputs password for the account. 6. Invoke "Check Password". If password is correct, System prints out recent transactional information(under 100) and balance after each transactional process. 7. Customer input 'OK' button. 8. Return to basic screen.	

Use Case : Pay Utility Bill	
1. Customer chooses pay utility bill menu from basic screen. 2. System prints out "input giro bill". 3. Customer inputs giro bill. 4. Invoke "Insert". If giro bill is valid, System prints out "input check card or bankbook". 5. Customer inputs check card or bankbook. 6. Invoke "Insert". If user information is valid, System prints out "input password for the account". 7. Customer inputs password for the account. 8. Invoke "Check Password". If password is correct, invoke "Request Customer's Data". 9. If balance is enough to pay utility bill, invoke "Print Transaction Receipt". 10. If Transaction Receipt is successfully printed, System asks Offer to update server information.	

Activity 2036. Define Operation Contracts

Use Case	System Operation
1.deposit	1.selectMenu 2.insert 3.inputMoney 4.printTransationReceipt
2.withdraw	1.selectMenu 2.insert 3.inputPassword 4.checkPassword 5.withdrawMoney 6.PrintTransactionReceipt
3.transfer	1.selectMenu 2.insert 3.selectBank 4.inputAccount 5.inputMoney 6.inputPassword 7.checkPassword 8.takeCharge 9.printTransactionReceipt
4.exchange	1.selectMenu 2.insert 3.chooseCurrency 4.inputPassword 5.checkPassword 6.TakeCharge 7.giveCurrency 8.printTransactionReceipt
5.Loan	1.selectMenu 2.insert 3.checkCredit 4.inputMoneyToLoan 5.inputPassword 6.checkPassword 7.TakeCharge 8.prinTransactionReceipt
6.checkbalance	1.selectMenu

	2.insert 3.inputPassword 4.checkPassword 5.printRecentTransaction
7.payutilitybill	1.selectMenu 2.inputBill 3.checkValidation 4.insert 5.inputPassword 6.checkPassword 7.TransactionReceipt
8.deposit without bank	1.seletMenu 2.selectBank 3.inputAccount 4.inputMoney 7.printTransactionReceipt

Name	insert()
Responsibilities	이용자가 매체를 삽입한다(체크카드, 신용카드, 통장)
Type	System
Cross References	R1.2, R2.2, R3.2, R4.2, R5.2, R6.2, R7.4 Usecase: deposit, withdraw, transfer, exchange, loan, check balance, payutilitybill
Notes	
Exceptions	N/A
Output	매체를 삽입한 후의 단계
Pre-conditions	이용자가 기능을 선택해야 한다.
Post-conditions	시스템은 이용자의 정보를 받아서 저장한다. 이용자의 정보: 이름, 계좌, 카드번호(체크카드나 통장을 삽입했을 경우), 신용카드번호(신용카드를 삽입했을 경우)

Name	inputMoney()
Responsibilities	이용자는 돈을 넣고 시스템은 금액이 얼마인지 확인한다.
Type	System

Cross References	R1.3, Usecase: Deposit
Notes	
Exceptions	만원 단위가 아닐 경우.
Output	넣은 금액이 얼마인지 확인하고 계좌에 추가하거나 신용카드 빚을 갚는다.
Pre-conditions	고객이 Deposit을 선택했어야 한다. 고객이 삽입한 매체가 유효해야 한다.
Post-conditions	고객의 통장에 해당 금액만큼 더해서 저장한다. 고객의 신용카드 정보에 해당 금액만큼 빚을 제해서 저장한다.

Name	print TransactionReceipt
Responsibilities	선택한 기능이 끝나고 거래명세표를 출력한다.
Type	System
Cross References	R1.4, R2.6, R3.9, R4.8, R5.8, R6.5, R7.7, R8.7 Usecase: deposit, withdraw, transfer, exchange, loan, check balance, payutilitybill, deposit without bank
Notes	
Exceptions	N/A
Output	거래명세표를 출력한다.
Pre-conditions	고객이 선택한 기능이 마무리 되어야 한다.
Post-conditions	Offer에 변경사항을 저장 요청해야 한다. ATM에 저장된 정보는 리셋해야 한다.

Name	checkPassword()
Responsibilities	고객이 입력한 비밀번호가 정확한지 확인한다.
Type	System
Cross References	R2.4, R3.7, R4.5, R5.6, R6.4, R7.6 Usecase: withdraw, transfer, exchange, loan, check balance, payutilitybill
Notes	
Exceptions	패스워드가 정확하지 않을 경우.
Output	비밀번호가 정확한지 아닌지 알아낸다.
Pre-conditions	고객이 비밀번호를 입력해야 한다.

	시스템은 offer에서 고객에 대한 정보를 받아와야 한다.
Post-conditions	고객이 입력한 번호와 시스템에 저장한 비밀번호가 일치한다면 다음 단계를 수행한다. 만약 일치하지 않는다면, 에러 메시지를 띄우고 고객에게 다시 비밀번호를 입력받아야 한다. 만약 3번을 틀린다면 해당 계좌에 transition lock을 걸고, 초기 화면으로 돌아가야 한다.

Name	withdrawMoney()
Responsibilities	고객은 입력한 금액만큼 ATM이 고객에게 돈을 인출해준다.
Type	System
Cross References	R2.5
Notes	
Exceptions	계좌에 돈이 충분하지 않을 때 인출금액이 10000원 단위가 아닐 때
Output	돈을 인출하고 해당 계좌에서 금액만큼을 제한다.
Pre-conditions	비밀번호가 맞아야 한다. ATM은 해당 계좌에 대한 정보를 가지고 있어야 한다.
Post-conditions	ATM이 해당 계좌에서 인출한 금액만큼 돈을 뺀다. 인출한 후의 계좌에 남은 금액이 -가 아니어야 한다.

Name	selectBank()
Responsibilities	고객이 송금할 은행을 입력받는다.
Type	System.
Cross References	R1.3, R8.2 Usecase: Transfer, deposit without bank
Notes	
Exceptions	N/A
Output	해당 은행 정보를 저장하고 계좌번호를 입력받는다.

Pre-conditions	고객이 transfer 메뉴를 누르거나 deposit without bank를 눌러야 한다.
Post-conditions	송금받을 은행의 정보를 저장해야 한다.

Name	takeCharge()
Responsibilities	수수료를 부과한다.
Type	System.
Cross References	R3.8, R4.6, R5.7
Notes	
Exceptions	고객 등급이 높을 경우
Output	고객의 등급에 따라 수수료를 부과하여 추가 요금을 받는다.
Pre-conditions	고객의 신용등급을 알고 있어야 한다. 고객이 타행이체를 선택해야 한다(송금의 경우) 고객이 돈을 빌려야 한다(대출의 경우) 고객이 환전을 선택해야 한다(환전의 경우)
Post-conditions	고객이 입력한 금액 외에 붙은 수수료만큼 더 인출해야 한다. 대출의 경우 신용카드사에서 정한 금액만큼 더 부과해야 한다. 통장거래의 경우 각 은행에서 정한 금액만큼 더 부과해야 한다.

Name	chooseCurrency()
Responsibilities	어느 나라의 화폐로 바꿀 것인지 고른다.
Type	System
Cross References	R 4.3 Usecase: exchange
Notes	
Exceptions	계좌에 충분하지 않은 경우
Output	원하는 나라의 통화로 환전한다.
Pre-conditions	고객의 비밀번호가 정확해야 한다. 계좌에 충분한 양의 돈이 있어야 한다. (바꾸기 원하는 금액 + 수수료) 고객이 원하는 외화의 금액을 입력해야 한다.
Post-conditions	원화에서 외화로 환전하는 경우, 해당 계좌에

	서 환율을 적용한 원화+수수료만큼을 빼서 저장해야 한다. 외화를 현금으로 고객에게 제공해야 한다.

Name	inputPassword
Responsibiliies	Password를 입력받는다.
Type	System
Cross Reference	R1.2,R1.4,R1.5,R1.6,R1.7,R1.8
Exception	checkPassword에서 에러가 검출됐을 경우, 다시 input 받는다.
Output	N/A
Pre-Conditions	카드나 통장을 투입한 이후여야 한다.
Post-Conditions	N/A

Name	inputAccount
Responsibiliies	계좌번호를 입력받고 계좌 정보의 유효성을 검증한다.
Type	System
Cross Reference	R1.3
Exception	계좌 정보가 존재하지 않을 경우, 다시 입력받는다.
Output	N/A
Pre-Conditions	Deposit without bankbook을 선택한 이후여야 한다.
Post-Conditions	N/A

Name	giveCurrency
Responsibiliies	화폐를 인출한다.
Type	System
Cross Reference	R1.5
Exception	N/A
Output	Foreign currency
Pre-Conditions	원하는 국가 화폐를 선택한 이후여야 한다.
Post-Conditions	N/A

Name	checkCredit

Responsibiliies	User의 신용 등급을 체크한다.
Type	System
Cross Reference	R1.6
Exception	N/A
Output	신용 등급에 따른 limit을 출력한다.
Pre-Conditions	카드를 투입한 이후여야 한다.
Post-Conditions	N/A

Name	inputMoneyToLoan
Responsibiliies	대출할 금액을 입력받는다.
Type	System
Cross Reference	R1.6
Exception	N/A
Output	N/A
Pre-Conditions	카드를 투입한 이후여야 한다.
Post-Conditions	Credit rating의 limit을 넘지 않아야 한다.

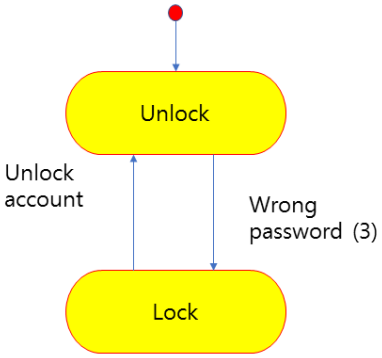
Name	inputBill
Responsibiliies	지로용지를 투입한다.
Type	System
Cross Reference	R1.8
Exception	N/A
Output	지로용지에 담긴 정보를 출력한다.
Pre-Conditions	Pay utility bill을 선택한 이후여야 한다.
Post-Conditions	카드 또는 통장을 투입해야 한다.

Name	printRecentTransaction
Responsibiliies	User의 최근 거래내역을 보여준다.
Type	System
Cross Reference	R1.7
Exception	거래 내력이 존재하지 않을 경우 내역을 출력하지 않는다.
Output	N/A
Pre-Conditions	Password를 입력받은 이후여야 한다.
Post-Conditions	N/A

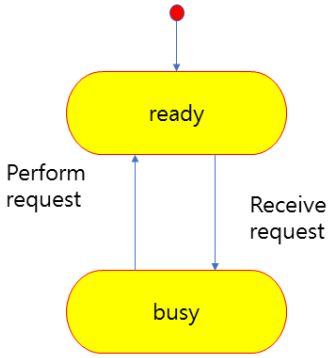
Activity 2037. Define State Diagrams

<Class State Diagram>

< State Diagram for "Account" >

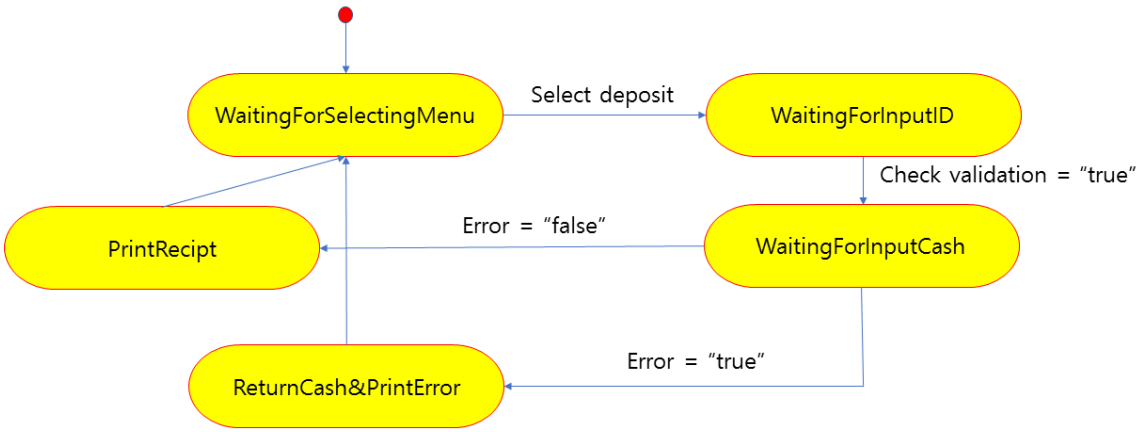


< State Diagram for "offer" >

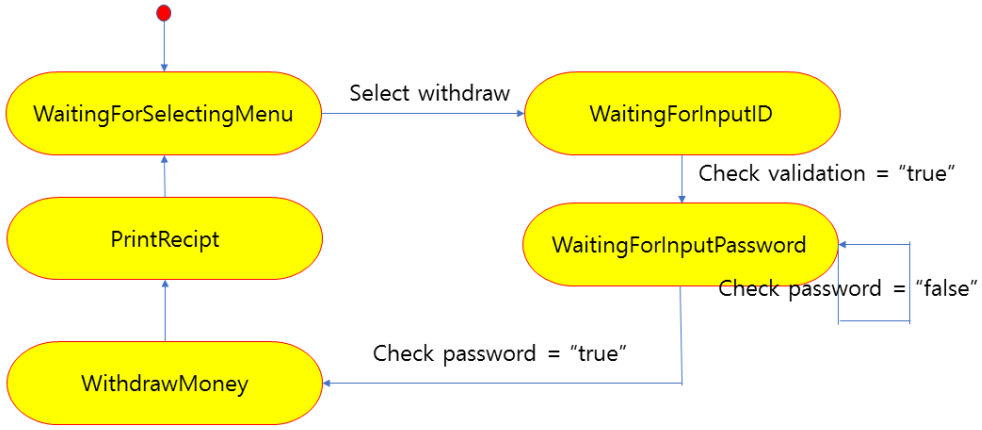


<Use Case State Diagram>

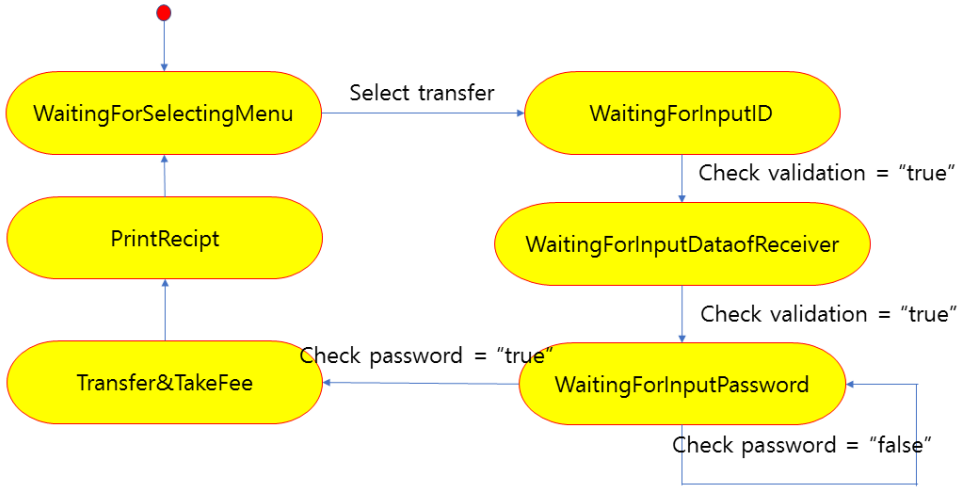
< Use Case : Deposit >



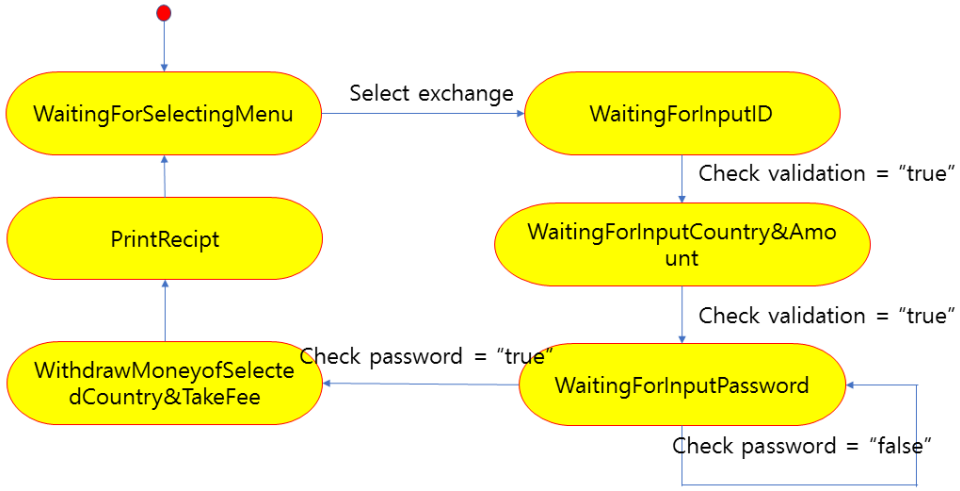
< Use Case : withdraw >



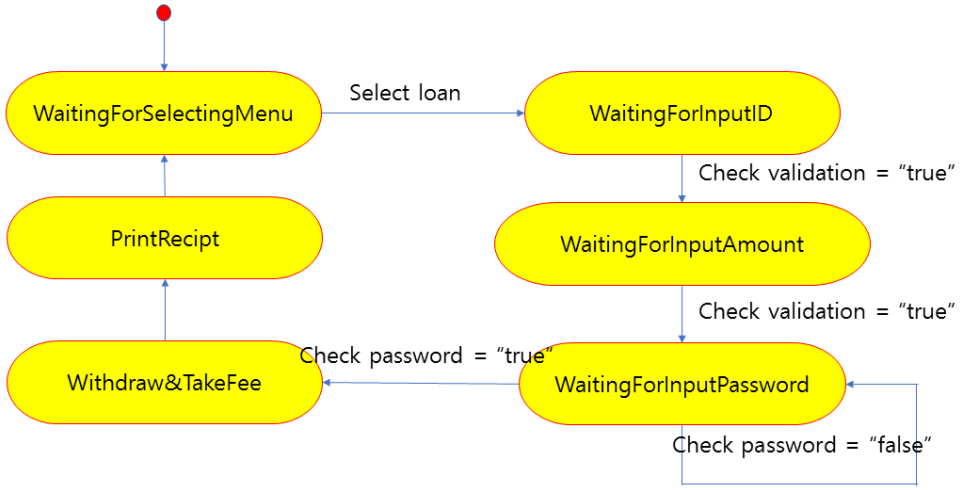
< Use Case : transfer >



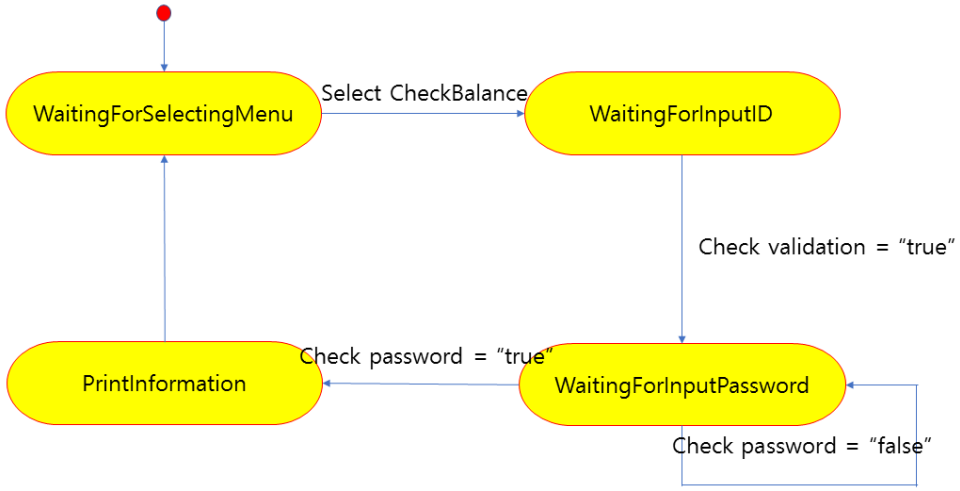
< Use Case : exchange >



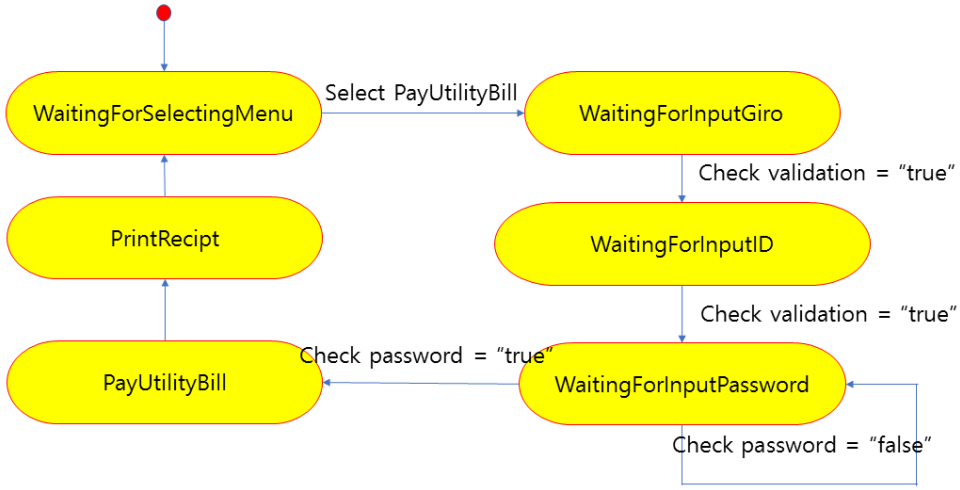
< Use Case : loan >



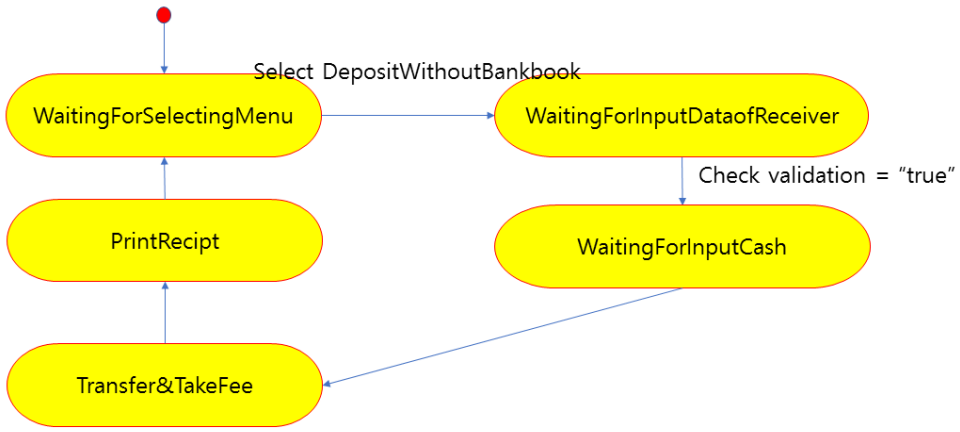
< Use Case : CheckBalance >



< Use Case : PayUtilityBill >



< Use Case : DepositWithoutBankbook >



Activity 2038. Refine System Test Case

<OOPT Stage 1000 Activity 1008 참고>

Ref. #	Function	Use Case Name	Test	Test description
R1.1	Deposit	1. Deposit	계좌에 맞게 돈이 입금되는지 TEST.	- 통장/체크카드 이용 시 해당 계좌에 맞게 입금되는지 TEST - 만원 단위로 입금되는지 TEST - 신용카드를 이용할 시, 대출금이 상환 되는지 TEST
R1.2	Withdraw	2. Withdraw	고객의 계좌에서 인출이 되는지 TEST	- 비밀번호가 맞아야 인출되는지 TEST - 만원 단위로만 인출되는지 TEST - 고객이 선택한 오만원권과 만원권의 장 수대로 출금되는지 TEST - 인출 제한 금액을 넘지 않았는지 TEST
R1.3	Deposit Without bankbook	3. Deposit without bankbook	무통장입금이 되는지 TEST	- 통장/체크카드 없이도 다른 사람에게 송금할 수 있는지 TEST - ATM과 같은 은행에만 보낼 수 있는지 TEST - 수신인의 정보가 정확해야 송금이 되는지 TEST - 만원 단위로 입금하는지 TEST - 수신자의 계좌에 돈이 입금되었

				는지 TEST
R1.4	Transfer	4. Transfer	송금이 되는지 TEST	- 고객이 통장이나 체크카드를 이용해서 돈을 보낼 수 있는지 TEST - 고객의 계좌에서 송금하는 금액 + (타행의 경우)수수료가 인출되는지 TEST - 수신자의 계좌에 돈이 입금되었는지 TEST - 통장/체크카드 이용 시 비밀번호가 맞는 경우만 돈을 보내는지 TEST - 수신인의 정보가 정확해야 송금이 되는지 TEST
R1.5	Exchange	5. Exchange	한화를 외화로 환전이 되는지 TEST	- 고객이 통장이나 체크카드를 입금했을 때만 환전이 가능한지 TEST - 비밀번호가 맞아야 환전이 되는지 TEST - 나라마다 환율이 맞게 적용되는지 TEST - 고객의 계좌에서 환전하는 금액 + 수수료만큼 인출되는지 TEST - 환전한 금액만큼 현금으로 나오는지 TEST
R1.6	Loanq	6. loan	대출(카드론)이 되는지 TEST	- 신용카드를 입금했을때만 대출이 되는지 TEST - 만원 단위로만 인출되는지 TEST - 한도를 넘지 않았을 경우에만 대출이 되는지 TEST - 비밀번호가 맞을 경우에만 대출이 되는지 TEST - offer(카드사)에 수수료+ 빌린 금액만큼 정보 업데이트 요청이 되는지 TEST
R1.7	Check Balance	7. Check balance	계좌조회가 되는지 TEST	- 통장이나 체크카드를 넣었을 때 계좌 조회를 할 수 있는지 TEST - 비밀번호가 맞을 경우에만 계좌조회가 되는지 TEST - 계좌의 최근 거래내역을 출력하는지 TEST
R1.8	Pay Utility Bill	8. Pay utility bill	공과금을 납부할 수 있는지 TEST	- 통장이나 체크카드를 넣었을 때 공과금을 납부 할 수 있는지 TEST - 지로를 통해 납부할 수 있는지 TEST

				- 가상계좌를 이용해 납부할 수 있는지 TEST - 비밀번호가 맞아야 납부할 수 있는지 TEST
R2.1	Print Transaction Receipt	9. Print Transaction Receipt	거래명세서 출력이 되는지 TEST	- 각각의 프로세스가 끝난 후 거래명세서를 출력할 수 있는지 TEST - 입금; 거래명세서에 계좌정보/ 입금액/ 입금 후 금액 - 출금: 거래명세서에 계좌정보/ 출금액/ 출금 후 금액 - 송금: 거래명세서에 계좌정보/ 송금액/ 송금 후 금액 - 무통장입금: 거래명세서에 계좌정보/ 송금액 - 공과금 납부: 거래명세서에 계좌정보/ 납부액/ 납부 후 금액 - 환전: 거래명세서에 계좌정보/ 환전액/환전 후 금액 - 대출: 거래명세서에 카드정보/ 대출한 금액/ 갚아야 하는 금액(대출금+수수료) - 각각 맞게 출력되는지 TEST
R2.2	Take Charge	10. Take Charge	수수료가 부과되는지 TEST	- 송금/대출/환전 시 수수료가 붙는지 TEST - atm이 작업할 때, offer가 요청한 수수료만큼 더 붙여서 작업하는지 TEST - 고객의 신용등급이 높으면 atm이 수수료를 받지 않는지 TEST
R2.3	Print Error	11. Print Error	에러를 프린트하는지 TEST	- 비밀번호가 틀린 경우를 제외한 에러들을 프린트하는지 TEST - 각 과정에서 계좌/카드/체크카드/지로의 정보가 틀린 경우 에러출력 TEST - 출금/대출시 한도를 넘은 경우 에러 출력 TEST
R2.4	Proceed Forced Termination	12. Forced Termination	강제종료 TEST	- 비밀번호를 3번 틀렸을 경우 강제종료가 되는지(처음 화면으로 돌아가는지) TEST
R2.5	Transaction Lock	13. Transaction Lock	거래 잠금 TEST	- 비밀번호가 3번 틀렸을 시 해당 계좌/카드로 거래를 못 하게 잠기는지 TEST
R3.1	Request	14. Request	고객정보 얻어	- ATM이 OFFER에게서 고객정보를

	Customer's Data	customer's data	오기 TEST	얻어올 수 있는지 TEST - 각 과정에서 필요한 계좌/체크카드/카드/지로 등의 정보를 얻어오는지 TEST - 필요한 과정: 입금/ 출금/ 송금/ 대출/ 무통장입금/ 공과금 납부/ 환전
R3.2	Check Validation	15. Insert	통장/카드/체크카드 삽입 TEST	- 삽입된 매체(통장/카드/체크카드)가 유효한지 확인하는 기능 TEST
R3.3	Check Password	16. Check Password	비밀번호 확인 TEST	- 각 프로세스에서 입력한 비밀번호가 맞는지 확인하는 기능 TEST - 필요한 프로세스; 출금/ 송금/ 대출/ 계좌조회/ 공과금 납부/ 환전
R3.4	Check Credit	17. Check Credit	신용등급 확인 TEST	- ATM이 수수료를 부과할 때 신용등급을 확인 할 수 있는지 TEST
R4.1	Update Server Information	18. Update Server Information	서버 정보 업데이트 TEST	- 계좌조회를 제외한 각 프로세스에서, 프로세스가 끝난 후 서버가 정보를 업데이트 하는지 TEST - ATM이 서버에 정보 업데이트를 요청하면, 서버가 그대로 수행하는지 TEST - 입금; 입금 계좌/ 입금액/ 입금 후 금액 - 입금(카드사용); 카드 정보/ 상환액/ 상환 후 남은 대출금 - 출금; 출금 계좌/ 출금액/ 출금 후 금액 - 송금; 송금 계좌/ 송금액/ 송금 후 금액 - 송금; 송금 받는 계좌/ 송금액/ 송금 받은 후 금액 - 무통장입금; 송금 받는 계좌/송금액/ 송금 받은 후 금액 - 공과금 납부; 공과금이 인출된 계좌/ 인출액/ 인출 후 금액 - 환전; 환전 계좌/ 환전액/ 환전 후 금액(환전액+수수료) - 대출/ 카드 정보/ 대출 후 한도/ 대출금/ 상환해야 하는 금액(대출금+수수료)

Activity 2039. Analyze (2030) Traceability Analysis

Ref. #	Function
R1.1	Deposit
R1.2	Withdraw
R1.3	Deposit Without bankbook
R1.4	Transfer
R1.5	Exchange
R1.6	Loan
R1.7	Check Balance
R1.8	Pay Utility Bill
R2.1	Print Transaction Receipt
R2.2	Take Charge
R2.3	Print Error
R2.4	Proceed Forced Termination
R2.5	Transaction Lock
R3.1	Request Customer's Data
R3.2	Check Validation
R3.3	Check Password
R3.4	Check Credit
R4.1	Update Server Information

Use case
Deposit
Withdraw
Deposit Without bankbook
Transfer
Exchange
Loan
Check Balance
Pay Utility Bill
Print Transaction Receipt
Take Charge
Print Error
Forced Termination
Transaction Lock
Request Customer's Data
Insert
Check Password
Check Credit
Update Server Information

Operation
1.selectMenu
2.insert
3.inputMoney
4. inputAccount
5.inputPassword
6.inputMoneyToLoan
7.withdrawMoney
8.selectBank
9.printTransationReceipt
10.inputMoney
11.takeCharge
12.chooseCurrency
13.giveCurrency
14.inputBill
15.checkValidation
16.checkPassword
17.checkCredit
18.printRecentTranscation