

# A New Reliable ATM

OOPT Stage 1000

Plan and Elaboration

---

## Project Team T6

201411140 권성완

201511247 김선정

201510436 허윤아

201510285 조수빈

## Date

2018-05-30

## Activity 1001. Define Draft Plan

## 1. Motivation

기술의 발전으로 인하여 현금보다 카드를 많이 사용하게 되어 사람들이 현금을 많이 갖고 다니지 않게 되었다. 하지만 급하게 현금이 필요한 경우가 여전히 종종 남아있어 ATM 기기의 필요성은 더욱 증대되어 이 프로그램을 개발하게 되었다. 이 시스템을 통하여 고객들은 현금 입출금 서비스 뿐만 아니라 대출, 공과금 납부, 환전 등 다양한 업무들을 은행을 가지 않고도 이용할 수 있게 될 것이다.

## 2. Project Objectives &amp; Scope

- 은행에 가지않고 간단한 은행 업무를 볼 수 있다.
- 현금서비스를 이용할 수 있다.

## 3. Functional Requirements

- Deposit
- Deposit without Bankbook
- Withdraw
- Transfer
- Exchange
- Loan
- Pay Utility Bill
- Check Balance
- Take charge
- Insert
- Check Validation
- Check Password
- Update Database
- Print Error
- Print Transaction Receipt

## 4. Non-functional Requirements

- 실제 프로그램을 ATM에 도입할 경우, 24시간 동안 구동될 시스템이므로 안정적으로 설계한다.
- 돈과 관련된 정보를 다루므로 정보가 정확해야 한다.
- 환전 시 환율은 국가마다 다르다.
- 환전 가능한 외화는 엔화, 위안화, 달러, 유로로 총 네 종류이다.
- 시스템에 대한 컴퓨터 상에서의 시뮬레이션이므로 모든 입력은 키보드와 마우스를 통해 이루어진다.
- ATM에 들어있는 현금은 무한대라고 가정한다.

#### 5. Resource Estimation

- Human Efforts(Man-Month) : 4 – 3months
- Project Duration : 12 weeks
- Other Efforts : 좋은 학점을 향한 집념, 조원들의 협동심

#### 6. Other Information

N/A

## Activity 1002. Preliminary Investigation Report

## 1. Alternative Solutions

- 기존의 잘 짜여진 시스템을 구매한다
- 다른 소프트웨어 개발 회사에 의뢰한다.

## 2. Project Justification (Business Demands)

- Cost : 시중의 제품보다 저렴하게 개발할 수 있다(인력비 감소).
- Duration : 회사에 의뢰했을 때보다 짧은 시간 내에 개발이 가능하다.
- Risk : 숙달되지 않은 개발자들의 개발로 인해 에러가 발생할 수 있고, 이로 인해 requirement를 충족시키지 못할 수 있다.
- Effect : requirement를 직접 짜 구현할 수 있고, 노력으로 비용을 감소시킬 수 있다.

## 3. Risk Management

Risk	Probability	Significance	Weight
Lack of JAVA Experience	3	4	12
Lack of UML Experience	3	4	12
Lack of domain knowledge	3	4	12
Lack of GUI Experience	3	4	12
Lack of Time	2	2	4
Lack of Sleep	2	2	4

## 4. Risk Reduction plan

- Lack of Java Experience (12) : JAVA 언어에 관한 책을 통해 공부한다.
- Lack of UML Experience (12) : 강의시간에 배운 내용을 복습하고 팀원끼리 서로 알려준다.
- Lack of domain knowledge (12) : domain diagram을 통해 domain에 대한 이해도를 늘린다. 평소 ATM기를 사용했던 경험을 떠올리며 작업한다.
- Lack of GUI Experience (12) : 책과 우리의 친구 구글에게 도움을 요청한다.
- Lack of Time (4) : 고학년들의 고질적인 문제로 부족한 시간은 여가시간 등을 줄여서 채운다.
- Lack of sleep (4) : 어느정도 감수할 부분으로 개인이 체력관리에 힘쓴다.

## 5. Market Analysis

- 이미 상용화되어 있는 ATM들이 많이 있으므로 이를 참조할 수 있으나, 개발의 필요성이 줄어든다.
- 상용화되어 있는 ATM들의 기능이 다양하므로, 이를 차용한다.

## 6. Other Managerial Issues

- 홈페이지에 기재된 개발 일정에 맞추어 작업한다.

## Activity 1003. Define Requirements

## 1. Functional Requirements (Version 3.0)

## - Deposit

고객은 체크카드 또는 통장을 이용해 자신의 계좌에 현금을 입금한다. 신용카드를 통해 입금할 경우 이전에 대출(loan) 기록이 있을 때 해당 신용카드의 대출 금액에서 입금한 금액만큼 자동으로 갚아진다. ATM이 처리할 수 있는 금액의 단위는 만원, 5만원 단위이다.

## - Deposit without bankbook

고객은 입금하려는 계좌번호를 정확히 알고 있을 경우 무통장 입금을 할 수 있다. 만약 계좌 번호가 정확하지 않을 경우, 거래는 진행되지 않는다. ATM이 처리할 수 있는 금액의 단위는 만원, 5만원 단위이다.

## - Withdraw

고객은 카드 또는 통장을 이용해 자신의 계좌에서 현금을 인출한다. 출금 시 고객은 계좌의 비밀번호를 알고 있어야 한다. 비밀번호가 정확하지 않을 경우 거래는 진행되지 않는다. ATM이 처리할 수 있는 금액의 단위는 만원, 5만원 단위이다.

## - Transfer

고객은 카드 또는 통장을 이용해 자신의 계좌에서 돈을 송금할 수 있다. 송금 시에는 정확한 상대방의 계좌번호와, 자신의 계좌의 비밀번호를 알고 있어야 한다. 계좌번호와 비밀번호가 정확하지 않을 경우 거래는 진행되지 않는다.

## - Exchange

환율에 따라 고객이 원하는 원화에서 국가의 현금으로 환전해준다. 환율은 국가마다 상이하다. 환전 시 현금이 아닌, 고객 본인의 카드 또는 통장이 있어야 한다. ATM이 처리할 수 있는 금액의 단위는 만원, 5만원 단위이다.

## - Loan

고객의 신용카드 한도 내에서 현금을 인출 할 수 있는 기능으로, 고객은 신용카드의 비밀번호를 알고 있어야 한다. ATM이 처리할 수 있는 금액의 단위는 만원 단위이다.

## - Pay Utility Bill

고객은 지로 고지서를 입력해, 체크카드 혹은 통장을 통해 공과금을 납부한다. 고객은 계좌의 비밀번호를 알고 있어야 한다.

- Check Balance

고객은 카드 또는 통장을 이용해 해당 계좌의 최근 거래 내역과 각각의 거래 내역에 따른 잔고를 조회할 수 있다. 고객은 계좌의 비밀번호를 알고 있어야 한다. 조회 후 고객이 확인 버튼을 누르면 초기 화면으로 돌아간다.

- Take Charge

환전, 타행 송금, 대출 과정에서 수수료를 부과한다. 등급에 따라 면제될 수 있다.

- Print Transaction Receipt

모든 process를 종료한 이후 DB를 업데이트하기 전에 거래 명세서를 출력한다.

- Insert

고객은 카드 또는 통장을 투입하고, 거래를 시작한다.

- Check Validation

ATM은 offer에게 고객의 정보를 요청하여 투입된 카드 또는 통장의 정보가 유효한지 확인한다.

- Check Password

ATM은 Offer 서버에 정보를 요청해, 고객이 입력한 계좌 또는 신용카드의 비밀번호가 기존에 등록된 것과 일치하는지 확인한다. 비밀번호 오류가 3회 이상 누적될 경우 고객은 해당 계좌 혹은 카드로 ATM을 이용할 수 없다.

- Update Database

ATM은 잔고 확인을 제외한 모든 process를 종료한 이후, 추가 혹은 변경된 정보를 갱신할 것을 Offer 서버에 요청한다.

- Print Error

ATM은 Error가 발생한 상황에서 Error 메시지를 고객에게 알린다. Error가 발생하는 상황은 고객의 정보가 DB에 저장된 정보와 일치하지 않을 때이다.

- Do Forced Termination

오류가 3번 발생할 경우 강제로 process를 종료하고 기본 화면으로 돌아간다.

- Transaction Lock

비밀번호 오류가 3회 누적된 고객의 경우, 부정한 거래 시도로 간주되어 ATM 이용이 금지된다.

## 2. System Functions (Table)

Ref. #	Function	Category
R1.1	Deposit	Evident
R1.2	Deposit Without bankbook	Evident
R1.3	Withdraw	Evident
R1.4	Transfer	Evident
R1.5	Exchange	Evident
R1.6	Loan	Evident
R1.7	Pay Utility Bill	Evident
R1.8	Check Balance	Evident
R2.1	Insert	Evident
R2.2	Print Transaction Receipt	Evident
R2.3	Print Error	Evident
R2.4	Do Forced Termination	Evident
R3.1	Take Charge	Hidden
R4.1	Check Password	Hidden
R5.1	Transaction Lock	Hidden
R6.1	Check Validation	Hidden
R6.2	Update Database	Hidden

## 3. Performance Requirements

- 현금 거래에 관한 정보를 신속하고 정확하게 처리하여 서비스를 제공한다.

## 4. Operating Environments

- OS : Microsoft Windows 10 (64bit)
- CPU : i5-7200U
- Memory : 8GB

## 5. Development Environments

- OS : Microsoft Windows 10 pro (64bit)
- CPU : i5-7200U
- Memory : 8GB
- Program Language : JAVA
- Editor : eclipse



6. Interface Requirements

- 사용자의 메뉴 선택에 의해 각 기능들을 수행할 수 있는 화면으로 전환된다.
- 다음 버전의 업데이트를 통해 좀 더 정교한 프로그램을 설계한다.

7. Other Requirements

(N/A)

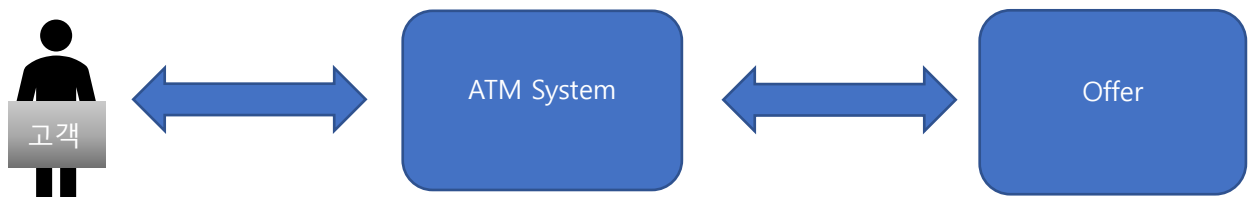
Activity 1004. Record terms in Glossary

Term	Description	Remarks
ATM	Automated Teller Machine의 준말로 현금 자동 입출금기를 의미한다	
Offer	ATM 서비스를 제공하는 회사로 은행과 카드사가 이에 해당한다.	
Bankbook	고객의 계좌 정보를 담고있는 통장	
DB	Database의 약자로, account 정보를 저장해둔다.	
KRW	Korean Won, 원화 단위를 의미한다	
Giro bill	지로 고지서를 의미한다	
Credit rating	신용 등급을 의미한다	

Activity 1005. Implement Prototype  
(생략)

Activity 1006. Define Business Use-Cases

1. Define System Boundary



2. Identify and Describe Actors

- Customer : ATM을 이용하는 고객
- ATM : Offer와 고객의 중간다리 역할을 해주는 시스템. 고객이 더 편하게 금융거래를 할 수 있도록 돕는다.
- Offer(Secondary Actor) : 고객의 금융거래 정보를 저장하고 제공하는 secondary actor

3. Identify Use-Case

Use Cases by Actor – Customer

Deposit	Deposit without bankbook	Withdraw	Transfer	Exchange
Loan	Pay Utility Bill	Check Balance	Insert	

Use Cases by Actor – Offer

Update Database	Check Validation
-----------------	------------------

## Use Cases by Event

Print Transaction Receipt	Print Error	Do Forced Termination
Take Charge	Check Password	Transaction Lock

## 4. Allocate system functions into Related Use Cases

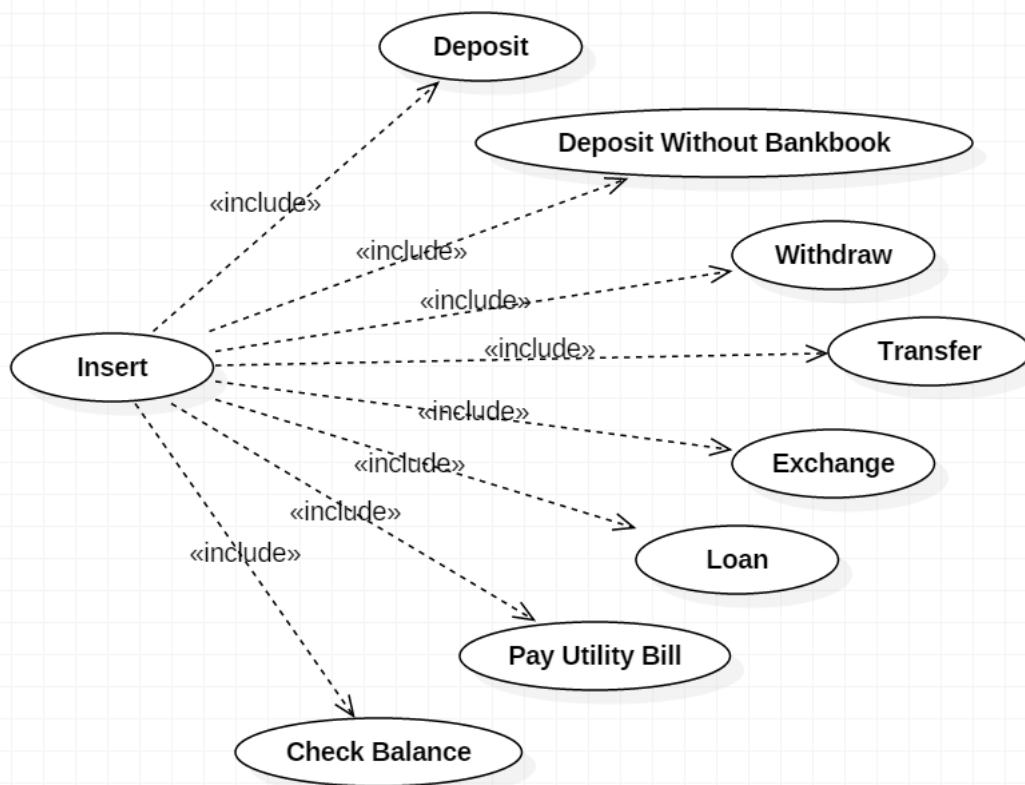
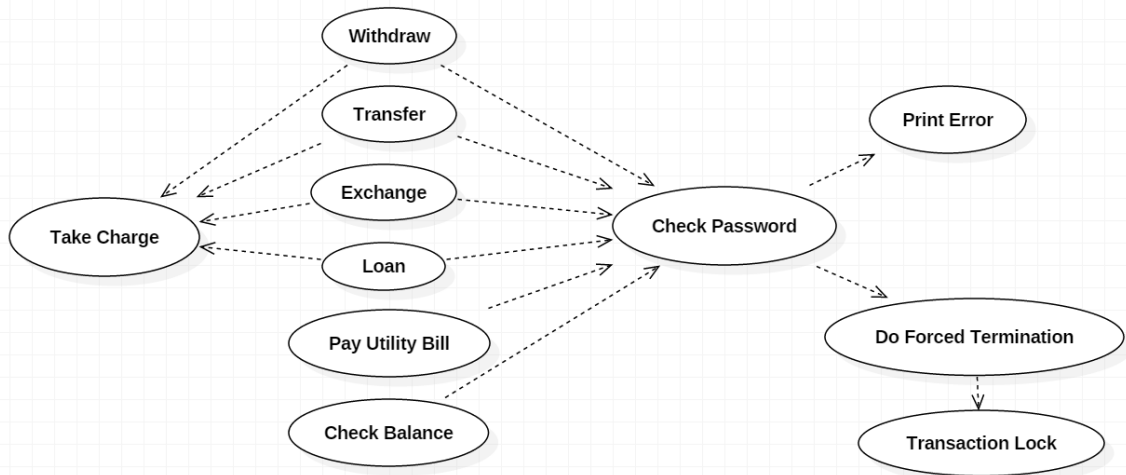
Ref. #	Function	Use Case Number & Name	Category
R1.1	Deposit	1. Deposit	
R1.2	Deposit Without bankbook	2. Deposit Without Bankbook	
R1.3	Withdraw	3. Withdraw	
R1.4	Transfer	4. Transfer	
R1.5	Exchange	5. Exchange	
R1.6	Loan	6. Loan	
R1.7	Pay Utility Bill	7. Pay Utility Bill	
R1.8	Check Balance	8. Check Balance	
R2.1	Insert	9. Insert	
R2.2	Print Transaction Receipt	10. Print Transaction Receipt	
R2.3	Print Error	11. Print Error	
R2.4	Do Forced Termination	12. Do Forced Termination	
R3.1	Take Charge	13. Take Charge	
R4.1	Check Password	14. Check Password	
R5.1	Transaction Lock	15. Transaction Lock	
R6.1	Check Validation	16. Check Validation	
R6.2	Update Database	17. Update Database	

## 5. Categorize Use Cases

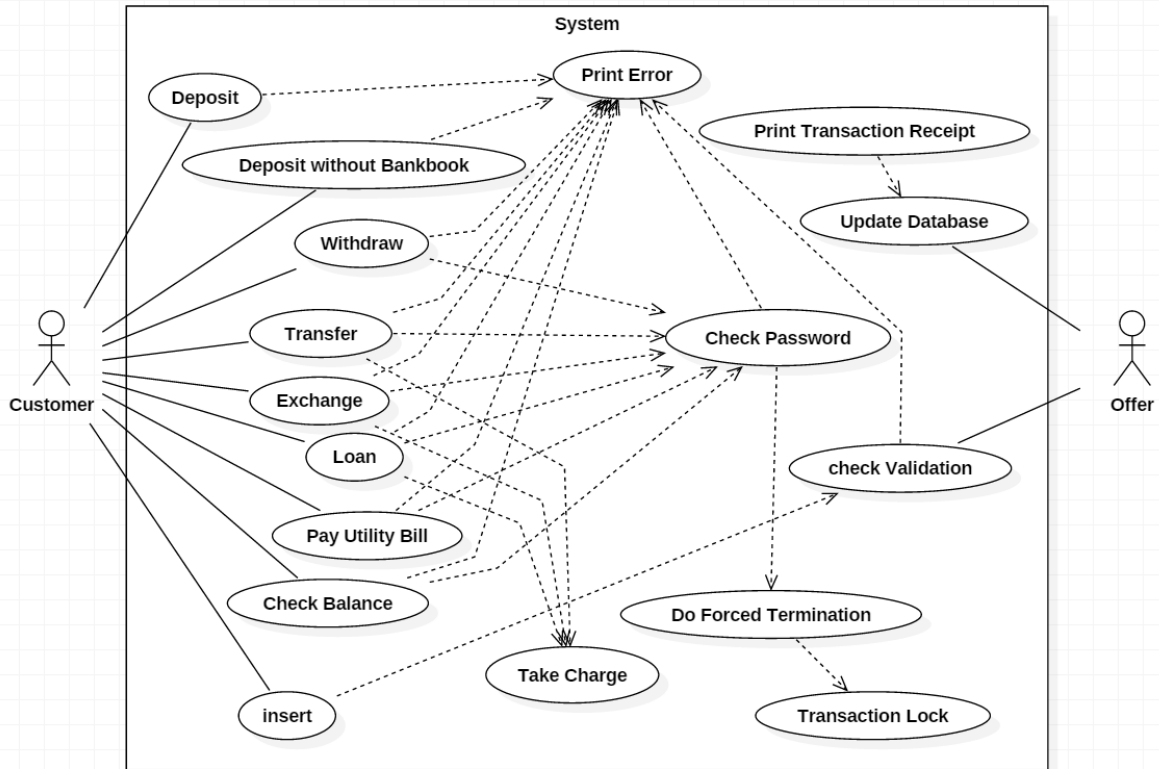
Ref. #	Function	Use Case Number & Name	Category
R1.1	Deposit	1. Deposit	Primary
R1.2	Deposit Without bankbook	2. Deposit Without Bankbook	Primary
R1.3	Withdraw	3. Withdraw	Primary
R1.4	Transfer	4. Transfer	Primary
R1.5	Exchange	5. Exchange	Primary

R1.6	Loan	6. Loan	Primary
R1.7	Pay Utility Bill	7. Pay Utility Bill	Primary
R1.8	Check Balance	8. Check Balance	Primary
R2.1	Insert	9. Insert	Primary
R2.2	Print Transaction Receipt	10. Print Transaction Receipt	Primary
R2.3	Print Error	11. Print Error	Primary
R2.4	Do Forced Termination	12. Do Forced Termination	Primary
R3.1	Take Charge	13. Take Charge	Primary
R4.1	Check Password	14. Check Password	Primary
R5.1	Transaction Lock	15. Transaction Lock	Primary
R6.1	Check Validation	16. Check Validation	Primary
R6.2	Update Database	17. Update Database	Primary

6. Identify the relationships between Use-Case



7. Draw a Use-Case diagram



8. Describe Use-Case

<b>Use Case</b>	1. Deposit
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can deposit cash into account by using check card or bankbook.</li> <li>-Customer can deposit money into credit card.</li> <li>-Customer can only deposit money in unit of 10000₩, 50000₩.</li> <li>-If customer successfully deposits, ATM requests Offer to update database.(Hidden)</li> <li>-If customer deposits cash into credit card, ATM requests Offer to update server information. Amount of money customer inputted is automatically used to repay from the amount customer loaned. (Hidden)</li> </ul>

<b>Use Case</b>	2. Deposit Without bankbook
<b>Actors</b>	Customer
<b>Description</b>	-Customer can deposit money to other customer's account by paying

	<p>cash.</p> <ul style="list-style-type: none"> <li>-Customer needs to input receiver's account number correctly.</li> <li>-If customer did it wrong, ATM prints out the error message.</li> </ul>
--	--

<b>Use Case</b>	3. Withdraw
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can withdraw cash from account by using check card or bankbook.</li> <li>-Customer can only withdraw money in unit of 10000₩ and 50000₩ under balance.</li> <li>-Customer needs to choose the number of 50000₩.</li> <li>-Customer needs to know password of an account.</li> <li>-If customer successfully withdraws, ATM requests Offer to update database.(Hidden)</li> </ul>

<b>Use Case</b>	4. Transfer
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can transfer money to other customer's account by using check card or bankbook.</li> <li>-If customer inserts check card or bankbook, ATM requests account information to Offer.(Hidden)</li> <li>-If customer enters wrong password, ATM prints out error message.</li> <li>-Customer needs to input receiver's account number correctly.</li> <li>-If customer did it wrong, ATM prints out error message.</li> <li>-If customer is transferring money to other bank's account, ATM takes charge of certain percentage.</li> <li>-If customer successfully transfers money from account, ATM requests Offer to update database.(Hidden)</li> </ul>

<b>Use Case</b>	5. Exchange
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can exchange KRW into foreign currency.</li> <li>-Customer can only use check card or bankbook to exchange money.</li> <li>-If customer enters wrong password, ATM prints out error message.</li> <li>-If customer successfully exchanges money, ATM requests Offer to update database.(Hidden)</li> </ul>

<b>Use Case</b>	6. Loan
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can withdraw cash in unit of 10000₩ and 50000₩ with credit card, and we call it Loan.</li> <li>-Each customer has loan limits.</li> <li>-If customer enters wrong password, ATM prints out error message.</li> <li>-Credit Card Company(included in Secondary Actor called Offer) lends money, and requests ATM to take charge.</li> <li>-If customer successfully loans money, ATM requests Offer to update database.(Hidden)</li> </ul>

<b>Use Case</b>	7. Pay Utility Bill
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can only use check card or bankbook to pay utility bill.</li> <li>-Customer needs to input giro bill before paying bill.</li> <li>-Customer can only deposit money into account of Korea bank.</li> <li>-If customer enters wrong password, ATM prints out error message.</li> </ul>

<b>Use Case</b>	8. Check balance
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can check balance of own account.</li> <li>-If customer enters wrong password, ATM prints out error message.</li> <li>-ATM prints out customer's recent transaction history(called log).</li> <li>-If customer inputs 'OK' button, ATM returns to basic screen.(GUI level)</li> </ul>

<b>Use Case</b>	9. Insert
<b>Actors</b>	Customer
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer inserts check card, bankbook, credit card, or giro bill into ATM.</li> <li>-If inserted, ATM proceeds 'Check Validation' use case.(Hidden)</li> </ul>

<b>Use Case</b>	10. Print Transaction Receipt
<b>Actors</b>	(None)
<b>Description</b>	<ul style="list-style-type: none"> <li>-ATM prints out transaction receipt after end of every process.</li> <li>-Transaction Receipt has record of current transaction such as amount</li> </ul>



	of money used in transaction, account balance or etc.
--	---

<b>Use Case</b>	11. Print Error
<b>Actors</b>	(None)
<b>Description</b>	-This use case begins when error occurred(customer inputted wrong information such as account number, card, bankbook, giro bill, password, or when customer wants to withdraw or loan or deposit and something went wrong).

<b>Use Case</b>	12. Do Forced Termination
<b>Actors</b>	(None)
<b>Description</b>	-This use case begins when customer inputted wrong information 3 times. -Every process is immediately stopped and returns to basic screen.(GUI level)

<b>Use Case</b>	13. Take Charge
<b>Actors</b>	(None)
<b>Description</b>	-ATM takes charge from customer in process of transfer, loan, and exchange. -If customer's credit rating is high, ATM doesn't take charge.

<b>Use Case</b>	14. Check password
<b>Actors</b>	(None)
<b>Description</b>	-This use case begins when customer inputs password of an account or credit card during transaction. -ATM compares inputted password and data from DB. If customer inputted wrong password, use case 'Print Error' occurs.

<b>Use Case</b>	15. Transaction Lock
<b>Actors</b>	(None)
<b>Description</b>	-This use case begins when customer inputted wrong password for account or credit card 3 times. -Customer cannot use account or credit card immediately, and cannot unlock with ATM.

<b>Use Case</b>	16. Check Validation
<b>Actors</b>	Offer
<b>Description</b>	-This use case begins after use case 'Insert'. -When customer inserts method for transaction, ATM gives information to Offer to compare with data in DB.

<b>Use Case</b>	17. Update Database
<b>Actors</b>	Offer
<b>Description</b>	-This use case begins when every transaction ends. -ATM requests Offer to update DB by sending updated data(recent transaction records). -Offer update DB with information provided by ATM.

#### 9. Rank Use-Cases

Ref. #	Function	Use Case Number & Name	Category	Rank
R1.1	Deposit	1. Deposit	Primary	High
R1.2	Deposit Without bankbook	2. Deposit Without Bankbook	Primary	High
R1.3	Withdraw	3. Withdraw	Primary	High
R1.4	Transfer	4. Transfer	Primary	High
R1.5	Exchange	5. Exchange	Primary	High
R1.6	Loan	6. Loan	Primary	High
R1.7	Pay Utility Bill	7. Pay Utility Bill	Primary	High
R1.8	Check Balance	8. Check Balance	Primary	High
R2.1	Insert	9. Insert	Primary	High
R2.2	Print Transaction Receipt	10. Print Transaction Receipt	Primary	High
R2.3	Print Error	11. Print Error	Primary	High
R2.4	Do Forced Termination	12. Do Forced Termination	Primary	High
R3.1	Take Charge	13. Take Charge	Primary	High
R4.1	Check Password	14. Check Password	Primary	High
R5.1	Transaction Lock	15. Transaction Lock	Primary	High
R6.1	Check Validation	16. Check Validation	Primary	High
R6.2	Update Database	17. Update Database	Primary	High

## Activity 1007. Define Business Concept Model

Customer(User)	ATM	Credit Card Company	Bank	Credit Card	Check Card	Cash
Name	Password	Utility Bill	Exchange rate(USD)	Exchange rate(JPY)	Exchange rate(EUR)	Exchange rate(CNY)
Deposit	Withdraw	Transfer	Check Balance	Exchange	Loan	Input
Commission	Store	Check Error	Print Error	Bank book	Limit	Dept
Account Number	Account Owner	10000₩	50000₩	Grade	Customer's Bank Data	Customer's Grade Data

## Activity 1008. Define System Test Case

Ref. #	Function	Use Case Number & Name	Test	Test description
R1.1	Deposit	1. Deposit	입금 TEST	- 고객이 올바른 계좌를 입력했을 때만 입금이 가능한지 TEST - 고객이 입력한 계좌에 입금한 금액만큼 입금이 되는지 TEST - 만원 단위로 입금되는지 TEST - 신용카드를 이용할 시, 대출금이 상환되는지 TEST
R1.2	Deposit Without bankbook	2. Deposit Without Bankbook	무통장입금 TEST	- 고객이 올바른 계좌번호를 입력했을 때만 입금이 가능한지 TEST - ATM을 관리하는 은행과 동일한 은행의 계좌에만 입금이 되는지 TEST - 만원 단위로 입금되는지 TEST - 해당 계좌에 돈이 입금되었는지 TEST
R1.3	Withdraw	3. Withdraw	출금 TEST	- 고객이 올바른 계좌를 입력했을 때만 인출이 가능한지 TEST - 비밀번호가 맞아야 인출되는지 TEST - 만원 단위로만 인출되는지 TEST - 고객이 선택한 오만원권과 만원권의 장수대로 출금되는지 TEST - 계좌에 남아있는 잔고보다 더 많은 금액을 출금하려고 할 경우 출금을 할 수 없는지 TEST

R1.4	Transfer	4. Transfer	송금 TEST	<ul style="list-style-type: none"> <li>- 고객이 본인의 계좌에 있는 돈을 보낼 수 있는지 TEST</li> <li>- 고객의 계좌에서 송금하려는 금액 + (타행의 경우)수수료가 인출되는지 TEST</li> <li>- 수신자의 계좌에 돈이 입금되었는지 TEST</li> <li>- 계좌의 비밀번호가 맞는 경우만 돈을 보내는지 TEST</li> <li>- 수신인의 정보가 정확해야 송금이 되는지 TEST</li> </ul>
R1.5	Exchange	5. Exchange	한화에서 외화로 환전 TEST	<ul style="list-style-type: none"> <li>- 고객이 올바른 계좌를 입력했을 때만 환전이 가능한지 TEST</li> <li>- 비밀번호가 맞아야 환전이 되는지 TEST</li> <li>- 나라마다 환율이 알맞게 적용되는지 TEST</li> <li>- 고객의 계좌에서 환전하는 금액 + 수수료만큼 인출되는지 TEST</li> </ul>
R1.6	Loan	6. Loan	대출 TEST	<ul style="list-style-type: none"> <li>- 고객이 올바른 신용카드 번호를 입력했을 때만 대출이 되는지 TEST</li> <li>- 만원 단위로만 인출되는지 TEST</li> <li>- 한도를 넘지 않았을 경우에만 대출이 되는지 TEST</li> <li>- 비밀번호가 맞을 경우에만 대출이 되는지 TEST</li> <li>- 카드사에서 대출 금액 + 수수료만큼 대출했다는 정보가 제대로 처리되는지 TEST</li> </ul>
R1.7	Pay Utility Bill	7. Pay Utility Bill	공과금 납부 TEST	<ul style="list-style-type: none"> <li>- 고객이 올바른 계좌를 입력했을 때만 공과금을 납부 할 수 있는지 TEST</li> <li>- 지로고지서를 입력했을 때, 잔고가 충분한 경우 지로고지서에 입력되어 있는 금액만큼 납부가 되는지 TEST</li> <li>- 국가 계좌로 납부할 수 있는지 TEST</li> <li>- 비밀번호가 맞아야 납부할 수 있는지 TEST</li> </ul>
R1.8	Check Balance	8. Check Balance	계좌 조회 TEST	<ul style="list-style-type: none"> <li>- 고객이 올바른 계좌를 입력했을 때만 계좌 조회를 할 수 있는지 TEST</li> <li>- 비밀번호가 맞을 경우에만 계좌 조회가 되는지 TEST</li> <li>- 계좌의 최근 거래 내역(계좌의 경우 일시, 거래 종류, 거래 금액, 잔고, 신용카드의 경우 일시, 거래 종류, 대출 금액, 남</li> </ul>

				은 한도)을 제대로 출력하는지 TEST
R2.1	Insert	9. Insert	매체 삽입 TEST	- 매체(계좌번호/신용카드/지로그지서)가 제대로 삽입되는지 TEST
R2.2	Print Transaction Receipt	10.Print Transaction Receipt	거래명세서 출력 TEST	- 각각의 프로세스가 끝난 후 거래명세서가 올바르게 출력되는지 TEST - 입금 : (계좌번호 이용 시) 거래 일시/입금 금액/잔고, 혹은 (신용카드 이용 시) 거래 일시/입금 금액/남은 대출 금액 - 무통장입금: 거래 일시/입금액 - 출금 : 거래 일시/출금 금액/남은 잔고 - 송금 : 거래 일시/송금 금액 + (타행일 경우) 수수료/송금 계좌/잔고 - 환전 : 거래 일시/환전 금액 + 수수료/잔고 - 대출 : 거래 일시/대출 금액/남은 한도 - 공과금 납부: 거래 일시/납부 금액/잔고
R2.3	Print Error	11. Print Error	에러 출력 TEST	- 각종 에러들이 발생했을 경우 그에 알맞은 에러 메시지를 출력하는지 TEST (존재하지 않는 계좌/카드번호를 입력한 경우, 비밀번호를 잘못 입력한 경우, 한도/잔고 이상의 금액을 거래한 경우 등)
R2.4	Do Forced Termination	12. Do Forced Termination	강제 종료 TEST	- 오류가 3번 발생했을 경우 강제 종료가 되는지(처음 화면으로 돌아가는지) TEST
R3.1	Take Charge	13.Take Charge	수수료 부과 TEST	- 송금(타행의 경우)/대출/환전 시 수수료가 붙는지 TEST - ATM 내부에서 등급에 따라 수수료를 계산해 부과하는지 TEST
R4.1	Check Password	14.Check Password	비밀번호 확인 TEST	- 각 프로세스에서 올바른 비밀번호를 입력했을 때만 프로세스가 진행되는지 TEST - 필요한 프로세스 : 출금/송금/환전/대출/공과금 납부/계좌 조회
R5.1	Transaction Lock	15. Transaction Lock	거래 잠금 TEST	- 비밀번호를 3번 틀렸을 시 해당 계좌/카드를 거래할 수 없도록 계좌/카드를 잠그는지 TEST
R6.1	Check Validation	16.Check Validation	삽입된 매체의 유효성 TEST	- 삽입된 매체(계좌번호/신용카드/지로그지서)가 유효할 때만 프로세스가 진행되는지 TEST
R6.2	Update Database	17.Update Database	DB 업데이트 TEST	- 계좌 조회를 제외한 각 프로세스에서, 프로세스가 끝난 후 DB가 제대로 업데이트 되는지 TEST - 입금 : (계좌번호 이용 시) 거래 일시/입

				금 금액/잔고, 혹은 (신용카드 이용 시) 거래 일시/입금 금액/남은 대출 금액 - 무통장입금: 거래 일시/입금 금액/잔고 - 출금 : 거래 일시/출금 금액/남은 잔고 - 송금 : (송금한 계좌의 경우)거래 일시/ 송금 금액 + (타행일 경우) 수수료/송금 계좌/잔고, (송금 받은 계좌의 경우)거래 일시/입금 금액/송금한 계좌/잔고 - 환전 : 거래 일시/환전 금액 + 수수료/ 잔고 - 대출 : 거래 일시/대출 금액/남은 한도 - 공과금 납부: 거래 일시/납부 금액/잔고
--	--	--	--	--

Activity 1009. Refine Plan

1. Project Scope & Objectives

- 은행에 가지않고 간단한 은행 업무를 볼 수 있다.
- 현금서비스를 이용할 수 있다.

2. Functional Requirements

Ref. #	Function	Category
R1.1	Deposit	Evident
R1.2	Deposit Without bankbook	Evident
R1.3	Withdraw	Evident
R1.4	Transfer	Evident
R1.5	Exchange	Evident
R1.6	Loan	Evident
R1.7	Pay Utility Bill	Evident
R1.8	Check Balance	Evident
R2.1	Insert	Evident
R2.2	Print Transaction Receipt	Evident
R2.3	Print Error	Evident
R2.4	Do Forced Termination	Evident
R3.1	Take Charge	Hidden
R4.1	Check Password	Hidden
R5.1	Transaction Lock	Hidden
R6.1	Check Validation	Hidden
R6.2	Update Database	Hidden

### 3. Performance Requirements

- 현금 거래에 관한 정보를 신속하고 정확하게 처리하여 서비스를 제공한다.

### 4. Operating Environments

- OS : Microsoft Windows 10 (64bit)
- CPU : i5-7200U
- Memory : 8GB

### 5. User Interface Requirements

- 사용자의 메뉴 선택에 의해 각 기능들을 수행할 수 있는 화면으로 전환된다.
- 다음 버전의 업데이트를 통해 좀 더 정교한 프로그램을 설계한다.

### 6. Other Requirements (N/A)

### 7. Resources

- Human Efforts(Man-Month) : 4 – 3months
- Project Duration : 12 weeks
- Other Efforts : 좋은 학점을 향한 집념, 조원들의 협동심

### 8. Scheduling

Stage	Phase(00x0)/Activity(000x)	Schedule(Day)				
		1	2	3	4	5
1000. Plan & Elaborate	1001. Define Draft Plan	█				
	1002. Create Preliminary Investigation Report	█				
	1003. Define Requirements	█				
	1004. Record Terms in Glossary	█				
	1005. Implement Prototype	█				
	1006. Define Use Cases	█				
	1007. Define Draft Conceptual Model	█				
	1008. Define Draft System Architecture	█				
	1009. Refine Plan	█				
	2000. Build	2010. Revise Plan				
2020. Synchronize Artifacts						
2030. Analyze						
2031. Define Essential Use Case		█				
2032. Refine Use Case Diagrams		█				
2033. Refine Conceptual Model		█				
2034. Refine Glossary		█				
2035. Define System Sequence Diagrams		█				
2036. Define Operation Contracts		█				
2037. Define State Diagrams		█				
2040. Design						
2041. Define Real Use Cases						
2042. Define Reports, UI and Storyboards						
2043. Refine System Architecture						
2044. Define Interaction Diagrams						
2045. Define Design Class Diagrams						
2046. Define Database Schema						
2050. Construct						
2051. Implement Class & Interface Definition						
2052. Implement Methods.						
2053. Implement Windows						
2054. Implement Reports						
2055. Implement DB Schema						
2056. Write Test Code						
2060. Test						
2061. Unit Testing						
2062. Integration Testing						
2063. System Testing						
2064. Performance Testing						
2065. Acceptance Testing						
2066. Documentation Testing						
3000. Deploy-ment	3001. Complete Technical Documents					
	3002. Complete User Documents					
	3003. System Testing					
	3004. Acceptance Testing					
	3005. Documentation Testing					
	3006. Train					
	3007. Establish Parallel Runs and Crossover					
	3008. Establish Support					
	3009. Install					

(SMA Case Study : Plan 마지막 도표 참조)