

---

# Software Modeling & Analysis

[OOPT stage 1000]

---

No.	T7
<b>Subject</b>	Software Modeling & Analysis
<b>Professor</b>	JUNBEOM YOO
<b>Team Member</b>	201410621 한상민
	201615007 문기태

# Chart.

## Activity 1001. Define Draft Plan

- 1) Motivation
- 2) Project Objectives
- 3) Functional Requirements
- 4) Non-Functional Requirements
- 5) Resource Estimation

## Activity 1002. Create Preliminary Investigation Report

- 1) Alternative Solutions
- 2) Project Justification (Business Needs)
- 3) Risk Management
- 4) Risk Reduction Plan
- 5) Market Analysis
- 6) Managerial Issue

## Activity 1003. Define Requirements

- 1) Functional Requirements
- 2) Performance Requirements
- 3) Operating Environments

## Activity 1004. Record Terms in Glossary

Activity 1006. Define Business Use Case

- 1) Define System Boundary
- 2) Identify and Describe Actors
- 3) Identify Use-Case
  - A. Actor-Based
  - B. Event-Based
- 4) Allocate system functions into related use cases
- 5) Categorize use cases
- 6) Draw a Use-Case Diagram
- 7) Describe Use-Case
- 8) Rank Use-Case

Activity 1007. Define Business Concept Model

Activity 1008. Define Draft System Architecture

Activity 1009. Define System Test Case

Activity 1010. Refine Plan

- 1) Project Scope
- 2) Project Objectives
- 3) Functional Requirements
- 4) Performance Requirements
- 5) Operating Environments
- 6) Resources
- 7) Plan Scheduling

## Activity 1001. Define Draft Plan

### 1. Motivation

통합 ATM의 부담은 수수료이다. 그렇기 때문에 수수료를 부과하는 것이 부담되지 않고 오히려 통합 ATM을 더 많이 사용할 수 있도록 기본적인 기능을 제공하되 ATM 거래횟수에 따라 혜택을 제공하는 ATM으로 통합 ATM의 사용을 적극 권장하는 것이 목표이다.

### 2. Project Objectives

통합 ATM기기에 혜택기능을 추가하여 업무를 하기 위해 Payback ATM을 적극 사용한다.

### 3. Functional Requirements

#### -Send Money

- ATM을 통하여 **고객의 계좌에서 다른 계좌로 돈을 보낼 수 있다.**
- 송금을 수행할 경우 **수수료**가 부과된다.
- 송금을 하려면 해당 계좌의 **비밀번호**를 입력해야 하고 **다른 계좌의 은행과 계좌번호**를 입력 해야 한다.
- 고객이 설정한 **송금 한도**보다 많은 금액을 보내려 할 경우 거래가 취소된다.
- 거래 도중에 언제든지 취소를 할 수 있다.
- 거래가 끝나면 거래 내용을 업데이트한다.
- 거래가 끝나고 송금에 대한 **명세서**를 출력할 수 있다.

#### -Withdraw Money

- 고객은 ATM을 통하여 **계좌에서 돈을 뺄 수 있다.**
- 인출을 수행할 경우 **수수료**가 부과된다.
- 인출을 하려면 해당 계좌의 **비밀번호**를 입력해야 한다.
- 고객이 설정한 **인출 한도**보다 더 많은 금액을 보내려 할 경우 거래가 취소된다.
- 거래 도중에 언제든지 취소를 할 수 있다.

-거래가 끝나면 거래 내용을 업데이트한다.

-거래가 끝나고 송금에 대한 명세서를 출력할 수 있다.

-Deposit Money

- ATM을 통하여 계좌에 돈을 입금 할 수 있다.

- 입금을 하려면 고객이 돈을 직접 넣어야 한다.

- 거래 도중에 언제든지 취소를 할 수 있다.

- 거래가 끝나면 거래 내용을 업데이트한다.

- 거래가 끝나고 송금에 대한 명세서를 출력할 수 있다.

-Check Remain Money

- ATM을 통하여 고객의 계좌에 있는 잔액을 확인할 수 있다.

- 잔액 조회를 수행하려면 비밀번호를 입력 받아야 한다.

- 거래 도중에 언제든지 취소를 할 수 있다.

- 거래가 끝나고 송금에 대한 명세서를 출력할 수 있다.

-Check Password

-Limit Amount

-Print Statement

-Count Commission

-Find Info

-Payback

#### 4. Non-Functional Requirement

-Effective UI

-Comfortable Control

-Fast Process Work (within 1min.)

## 5. Resource Estimation

-Human Efforts: 2man, 4month

-Human Resources: 2 programmers

-Duration: 4 month

-Budget

(단위:만원)

	부분 계	비고
인건비	0	
활동비	150	
계	150	

### Activity 1002. Create Preliminary Investigation Report

#### 1. Alternative Solutions

-시중에 있는 ATM기의 소프트웨어를 사용한다

-타 업체의 외주를 부탁한다.

#### 2. Project Justification (Business Needs)

-cost: 인건비가 들지 않는다는 장점이 있다

-duration: 4month

-Risk: OOAD JAVA UML 등에 대한 숙달 부족, 타 실제 은행과의 협약 필요

-Effect: 유지보수에 용이하다. 관리자 권한으로 쉽게 접근할 수 있다.

#### 3. Risk Management

Risk	Probability	Significance	Weight
OOPT Skill	3	5	15
Programming Skill	3	5	15
UML Skill	1	3	3
Mid & Final Exam	10	10	100

#### 4. Risk Reduction Plan

Risk	Reduction Plan
OOPT Skill	교수님 및 조교님의 수업을 열심히 따라간다
Programming Skill	관련 서적을 참고한다
UML Skill	관련 서적과 조교님의 도움을 받는다
Mid&Final Exam	프로젝트와 수업을 골고루 병행하며 열심히 한다

#### 5. Market Analysis

-현재 이미 많이 분포 되어 있고 독점적으로 운영하고 있는 기업이 있으므로 진입이 다소 어려울 수도 있다.

-여러 ATM의 소프트웨어의 디자인이 모두 비슷하기 때문에 디자인의 구성요소에 대해 생각할 필요는 없다.

#### 6. Managerial Issue

-제한된 시간 안에 개발을 완료 해야 된다.

-개발언어에 대한 자문을 구하기 위한 계획을 추가적으로 세워야 한다.

### Activity 1003. Define Requirements

#### 1. Functional Requirements (Rev. Activity1001)

Function	Description
Send Money	송금을 실시한다
Withdraw Money	인출을 실시한다
Deposit Money	입금을 실시한다
Check Remain Money	잔액조회를 실시한다
Find Info	입력된 계좌번호의 계좌정보를 시스템 내에 적재한다.
Check Password	입력한 비밀번호가 맞는지 확인하다
Limited Amount	출금 송금 한도를 나타낸다
Count Commission	거래를 수행하기 위한 수수료를 계산한다.
Print statement	명세서를 출력한다
Payback	거래량을 조회하여 환급여부를 결정한다.

Reference #	Function	Category
1-1	Find Info	Hidden
1-2	Check Password	Hidden
1-3	Limited Amount	Hidden
1-4	Count Commission	Hidden
2-1	Send Money	Event
2-2	Withdraw Money	Event
2-3	Deposit Money	Event
2-4	Check Remain Money	Event
3-1	Print statement	Event
3-2	Payback	Hidden

2. performance requirement

-송금 과정이 1분 이내로 되어야 한다

-계좌정보를 형식에 맞게 텍스트 파일로 저장해야 한다

-명시되어 있는 은행만을 사용하여야 한다

3. operating Environments

-OS: window7 & window10

-IDE: Eclipse

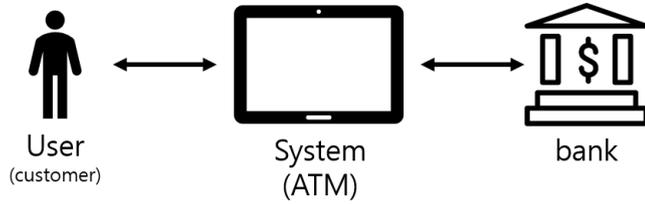
-programming language: Java

**Activity 1004. Record term in Glossary**

Glossary	Description
withdraw	(계좌에서 돈을) 인출하다
account	계좌
password	비밀번호
check	확인하다, 점검하다
limit	한계, 한도
deposit	예금, 입금
ID	신분증명서, 신분증 (identity 또는 identification의 약어)
remain	남아있는, 잔여의
statement	입출금 명세서
Payback	환급

## Activity 1006. Define Business Use Case

### 1. Define System Boundary



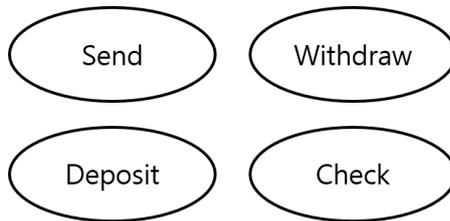
### 2. Identify and Describe Actors

-Customer: 은행의 일반 업무(입출금, 송금, 잔액확인 등)를 수행할 수 있다.

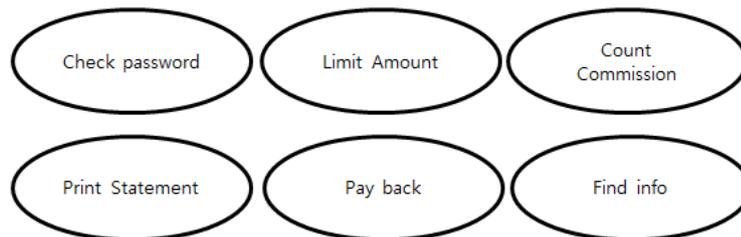
-Admin: ATM을 관리할 수 있는 권한(ATM 총액확인)을 가지고 있다.

### 3. Identify Use-Case

#### A. Actor-Based



#### B. Event-Based



### 4. Allocate system functions into related use cases

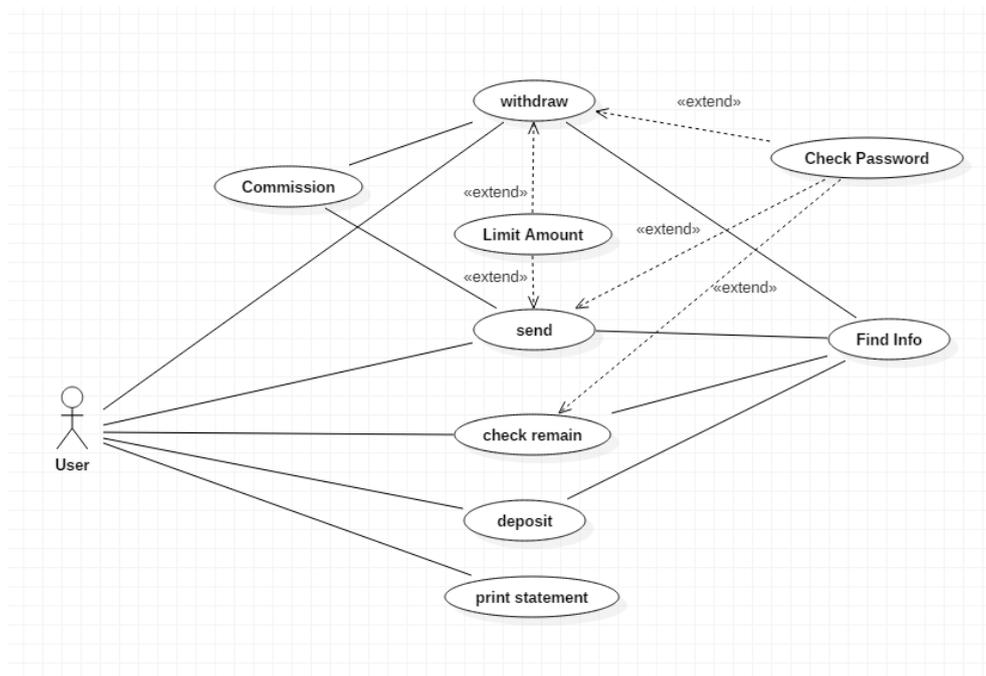
Reference #	Function	Use case Number &Name	Category
1-1	Find Info	1. Find Info	
1-2	Check Password	2. Check Password	
1-3	Limited Amount	3. Limited Amount	

1-4	Count Commission	4. Count Commission	
2-1	Send Money	5. Send	
2-2	Withdraw Money	6. Withdraw	
2-3	Deposit Money	7. Deposit	
2-4	Check Remain Money	8. Check Remain	
3-1	Print statement	9. Print statement	
3-2	Payback	10. Payback	

5. Categorize use cases

Reference #	Function	Use case Number &Name	Category
1-1	Find Info	1. Find Info	Primary
1-2	Check Password	2. Check Password	Primary
1-3	Limited Amount	3. Limited Amount	Primary
1-4	Count Commission	4. Count Commission	Primary
2-1	Send Money	5. Send	Primary
2-2	Withdraw Money	6. Withdraw	Primary
2-3	Deposit Money	7. Deposit	Primary
2-4	Check Remain Money	8. Check Remain	Primary
3-1	Print statement	9. Print statement	Primary
3-2	Payback	10. Payback	Primary

6. Draw a Use-Case Diagram



## 7. Describe Use-Case

Use Case	1. Find Info
Actors	System
Descriptions	-사용자가 거래하기 위한 계좌ID(통장, 카드, 계좌번호 등)를 입력하면 시스템은 해당 은행에서 계좌정보를 가져와 적재한다.

Use Case	2. Check Password
Actors	System
Descriptions	-고객이 카드번호나 계좌번호를 입력하고 비밀번호 까지 치게 되면 해당 비밀번호를 검증한다. -이때 해당 비밀번호는 계좌가 소속되어 있는 TXT파일안에 들어 있으므로 해당 TXT파일을 오픈하여 비교하고 확인한다. -비밀번호가 틀리면 다시 계좌번호나 카드번호를 받기 전의 초기 상태로 돌아간다

Use Case	3. Limited Amount
Actors	System
Descriptions	-송금을 하거나 출금을 할 때 자신이 정해 놓은 한도나 ATM기기 자체에 걸려 있는 한도를 알려준다 -만약 한도가 넘어가면 거래가 즉시 종료되며 RETURN TO MAIN 함수로 인하여 다시 거래를 시작할 수 있는 메인 화면으로 넘어간다

Use Case	4. Count Commission
Actors	System
Descriptions	-시스템에 Find Info로 적재한 계좌정보에서 은행의 영업시간을 조회하여 현재 시간과 비교하여 수수료를 책정한다.

Use Case	5. Send
Actors	User
Descriptions	-고객은 ATM을 통하여 송금을 실시한다 -송금을 위하여 상대 계좌번호를 입력한다 -수수료와 송금액이 합쳐진 금액이 한도와 잔액을 초과하는지 확

	<p>인한다.</p> <p>-송금이 끝나면 계좌내용을 업데이트 한다</p> <p>-거래가 끝나면 명세서를 출력한다</p>
--	---

Use Case	6. Withdraw
Actors	User
Descriptions	<p>-고객은 ATM을 통하여 계좌에서 돈을 뺐는다.</p> <p>-인출을 하려면 해당 계좌의 비밀번호를 입력해야 한다.</p> <p>-수수료와 출금액이 합쳐진 금액이 한도와 잔액을 초과하는지 확인한다.</p> <p>-출금이 끝나면 거래 내용을 업데이트한다.</p> <p>-거래가 끝나고 송금에 대한 명세서를 출력한다.</p>

Use Case	7. Deposit
Actors	User
Descriptions	<p>-ATM을 통하여 계좌에 돈을 입금한다.</p> <p>-고객이 돈을 직접 넣는다.</p> <p>입금액을 확인한 후 입금을 진행한다.</p> <p>-거래가 끝나면 거래 내용을 업데이트한다.</p> <p>-거래가 끝나고 송금에 대한 명세서를 출력한다.</p>

Use Case	8. Check Remain
Actors	User
Descriptions	-고객은 ATM을 통하여 계좌에 있는 잔액을 확인한다

Use Case	9. Print Statement
Actors	System
Descriptions	<p>-고객이 ATM기기에서 이뤄지는 거래를 마치고 종료 되면 명세서를 뽑을지 말지 선택지를 내놓는다.</p> <p>-명세서를 출력하든 출력하지 않든 다시 기본 메뉴들이 있는 화면으로 돌아가 거래를 계속 할 수 있게 한다</p>

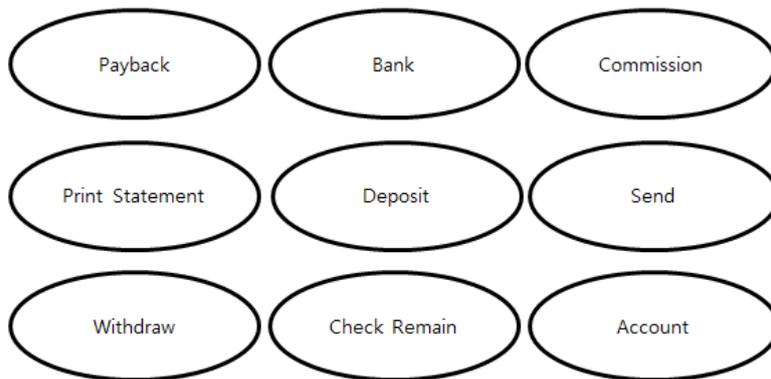
Use Case	10. Payback
Actors	System

Descriptions	-계좌를 갱신하는 동시에 계좌의 ATM 거래 횟수를 조회한다. -ATM 내에 정해 놓은 혜택 정책에 해당되면 상당 상품의 기프트콘을 영수증을 통해 제공한다.
--------------	--

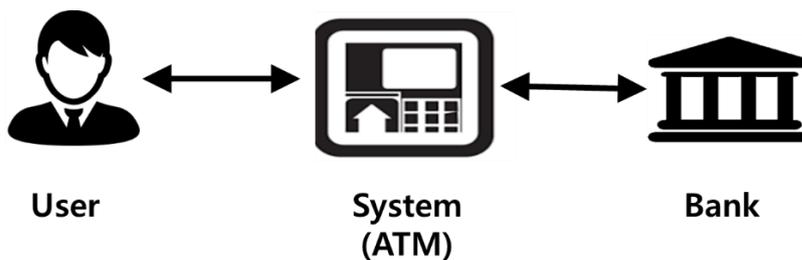
### 8. Rank Use-Case

Use-Case Number & Name	Description
1.Find Info	High
2.Check Password	High
3.Limited Amount	High
4.Count Commission	High
5.Send	High
6.Withdraw	High
7.Deposit	High
8.Check Remain	High
9.Print statement	High
10.Payback	High

### Activity 1007. Business Concept Model



### Activity 1008. Define Draft System Architecture



## Activity 1009. Define System Test Case

Test	Function	Description
클래스 간 변수가 제대로 넘어가는지 확인한다	Send	송금을 실시한다
	Withdraw	인출을 실시한다
	Deposit	입금을 실시한다
갱신된 계좌 정보가 잘 반영되는지 확인한다	Check Remain	잔액조회를 실시한다
비밀번호를 잘 판별하는지 확인한다	Check Password	입력 비밀번호를 확인하다
2가지의 경우에 맞게 출금한도가 나오는지 확인한다	Limited Amount	출금 송금 한도를 나타낸다
거래에 필요한 수수료를 계산한다.	Count Commission	출금/송금에 필요한 수수료를 계산한다.
모든 작업 종료 후 계좌 정보를 갱신했는지 확인한다	Update Account	은행 업무를 본 후 계좌의 정보를 갱신하다
명세서를 출력하는지 확인한다	Print statement	명세서를 출력한다
ATM 거래 횟수를 조회하여 환급 혜택을 제공한다.	Payback	환급 혜택을 제공하는 기능

## Activity 1010. Refine Plan

### 1. Project Scope

통합 ATM의 부담은 수수료이다. 그렇기 때문에 수수료를 부과하는 것이 부담되지 않고 오히려 통합 ATM을 더 많이 사용할 수 있도록 기본적인 기능을 제공하되 ATM 거래횟수에 따라 혜택을 제공하는 ATM으로 통합 ATM의 사용을 적극 권장하는 것이 목표이다.

### 2. Project Objectives

통합 ATM기기에 혜택기능을 추가하여 업무를 하기 위해 Payback ATM을 적극 사용한다.

### 3. Functional Requirements

A. -Send Money

B. -Withdraw Money

- C. -Deposit Money
- D. -Check Remain Amount
- E. -Find Info
- F. -Check Password
- G. -Count Commission
- H. -Limit Amount
- I. -Print Statement
- J. -Payback

-Performance Requirements

직관적인 UI와 빠른 작업 능력처리(1분 이내)를 갖고 있어야 한다.

4. Operating Environment

OS: Window7 & Window 10

5. Resources

-Human Efforts: 2man, 4month

-Human Resources: -2 programmers

-Duration: -4 month

-Budget

(단위: 만원)

	부분 계	비고
인건비	0	
활동비	150	
계	150	

## 6. Plan Scheduling

Stage	Phase(00X0)/Activity(001X)	Schedule(week)													
		1	2	3	4	5	6	7	8	9	10				
1000. Plan & Elaboration	1001. Define Draft plan	█													
	1002. Create Preliminary Investigation Report	█													
	1003. Define Requirement	█													
	1004. Record Terms in Glossary	█													
	1005. Implement Prototype	█	█												
	1006. Define Business Use-Case		█												
	1007. Define Draft System Architecture		█												
	1009. Define System Test Case		█	█											
	1010. Refine Plan		█	█	█										
	2000. Build	2010. Revise Plan			█	█	█								
	2020. Synchronize Artifacts			█	█	█									
	2030. Analyze				█	█									
	2031. Define Essential Use Case				█	█									
	2032. Refine Use Case diagram				█	█									
	2033. Define Domain Model				█	█									
	2034. Refine Glossary					█	█								
	2035. Define System Sequence Diagrams					█	█								
	2036. Define Operation Contracts					█	█								
	2038. Refine System Test Case					█	█								
	2039. Analyze (2030) Traceability Analysis					█	█	█							
	2040. Design					█	█	█							
	2041. Design Real Use Case					█	█	█							
	2042. Define Reports UI and Storyboards					█	█	█							
	2043. Refine System Architecture					█	█	█							
	2044. Define Interaction Diagrams					█	█	█							
	2045. Define Design Class Diagrams					█	█	█							
	2046. Design Traceability Analysis					█	█	█							
	2050. Construct					█	█	█							
	2051. Implement Class & Methods					█	█	█							
	2052. Implement Windows					█	█	█							
	2053. Implement Reports					█	█	█							
	2054. Write Unit Test Code					█	█	█							
	2060 Test					█	█	█							
	2061 Unit Testing					█	█	█							
	2062. Integration Testing					█	█	█							
	2063. System Testing					█	█	█							
	2064. Performance Testing					█	█	█							
	2065. Acceptance Testing					█	█	█							
	2066. Documentation Testing					█	█	█							
	2067. Testing Traceability Analysis					█	█	█							