

OOPT STAGE 1000 - PLAN AND ELABORATION

SO USER-FRIENDLY ATM

Project Team

Team T4

Date

2018 / 03 / 29

Team Members

- 201611269 신문기
- 201611306 최병규
- 201610401 손하영
- 201510283 임진웅

ACTIVITY 1001 - DEFINE DRAFT PLAN

1. Motivation

은행 영업시간에 은행을 직접 방문하기 힘든 고객들이 언제든지 은행 서비스를 이용할 수 있는 서비스를 구축하고자 한다. ATM은 은행과 달리 영업시간의 제약을 받지 않을 수 있지만, 본인 확인 절차가 간소화 되어 있기에 이용가능한 서비스의 영역이 제한되어 있다. 24시간 이용가능한 또 다른 은행 서비스에는 인터넷뱅킹과 텔레뱅킹이 있다. ATM은 현금거래가 가능하다는 점에서 이들과 차이가 있다. 영업점, 인터넷뱅킹, 텔레뱅킹과 ATM의 차이점을 고려하면서 다양한 사용자들이 이용하기 편리한 은행 ATM 서비스를 만들고자 한다.

2. Project Objectives

- 다양한 이용자를 고려하는 시스템을 구축한다.
- 사용법이 간단한 시스템을 구축한다.
- 직관적인 인터페이스를 가진 시스템을 구축한다.

3. Functional Requirements

- Select Menu
- Insert Card
- Insert Passbook
- Deposit
- Withdraw
- Transfer
- Check Transaction History
- Print Receipt
- Return Card
- Return Passbook

4. Non-functional Requirements

- Comfortable UI
- High Performance
- Derivative Interface

5. Resource Estimation

- Human Efforts : 4 People / 3 Months
- Human Resources : 4 (컴퓨터공학 * 2 + 물리학 * 1 + 미디어 커뮤니케이션 * 1)
- Duration : 12 weeks
- Budget : 10000(하루) * 5(일) * 4(주) * 3(달) * 4(사람 수) = 2,400,000(₩)

ACTIVITY 1002 - CREATE PRELIMINARY INVESTIGATION REPORT

1. Alternative Solutions

- 현존 ATM 시스템을 구매한다.
- 타 SW 개발 업체에 외주를 의뢰한다

2. Project's Justification (Business Needs)

1. **Cost** : ATM 시스템 구매 및 외주 의뢰보다 싸다.
2. **Duration** : 2018 / 06 / 10 까지 개발이 완료되어야 한다.
3. **Risk**:
 1. OOAD 및 OOP 에 대한 이해력 부족
 2. 개발 능력 부족
 3. 디자인 능력 부족
 4. 문서 작성 능력 부족
 5. 타 과목의 과제
 6. 고양이의 아픔
 7. PLAYERUNKNOWN'S BATTLEGROUNDS

3. Risk Management

Risk	Probability(0 ~ 5)	Significance(0 ~ 10)	Weight(Prob. * Sig.)
OOAD 및 OOP에 대한 이해력 부족	5	10	50
개발 능력 부족	3	10	30
디자인 능력 부족	4	8	32
문서 작성 능력 부족	3	9	27
타 과목의 과제	5	5	25
고양이의 간섭	2	2	4
PLAYERUNKNOWN'S BATTLEGROUNDS	5	8	40

4. Risk Reduction Plan

Risk	Way of Reduction
OOAD 및 OOP에 대한 이해력 부족	예전 년도 자료 혹은 책을 참고하거나, 교수님 및 조교님께 자문을 구한다.
개발 능력 부족	소속 팀원 중 2명의 개발자가 나머지 팀원을 잘 뒷받침하여 개발을 이끌어 나간다.
디자인 능력 부족	소속 팀원 중 1명의 디자이너가 UI 설계를 맡고, 개발자는 설계대로 구현한다.
문서 작성 능력 부족	팀에서 가장 문장력이 뛰어난 사람한테 검증을 받는다.
타 과목의 과제	마감일이 임박한 과제를 먼저 처리하거나, 소프트웨어 모델링 및 분석 과제를 처리한 후에 처리한다.
고양이의 간섭	맛있는 것을 물려주고 나오고, 아프면 그 즉시 동물병원에 데려간다.
PLAYERUNKONWS' BATTLEGROUND	과제를 모두 마무리한 다음에 게임에 접속한다.

5. Other Managerial Issues

2018년 6월까지 개발을 완료하여, 100%의 코드 커버리지로 A+를 받아야 한다.

ACTIVITY 1003 - DEFINE REQUIREMENTS

1. Functional Requirements

1.1 Functional Requirements

Function	Description
Select Menu	처음 화면에서 메뉴(입금 / 출금 / 송금 / 잔액 확인)를 선택한다.
Insert Card	카드를 받는다.
Insert Passbook	통장을 받는다.
Deposit	입금을 진행한다.
Withdraw	출금을 진행한다.
Transfer	송금을 진행한다.

Check Transaction History	거래 내역을 조회한다.
Print Receipt	거래 명세표를 출력한다.
Return Card	거래가 종료된 후 카드를 반환한다.
Return Passbook	거래가 종료된 후 통장을 반환한다.
Display Manual	사용 설명서를 보여준다.

1.2 Reference Number & Categorized and Identified Functions

Reference No.	Function	Category
R 1	Select Menu	Evident
R 2	Insert Card	Evident
R 3	Insert Passbook	Evident
R 4	Deposit	Hidden
R 5	Withdraw	Hidden
R 6	Transfer	Hidden
R 7	Check Transaction History	Hidden
R 8	Print Receipt	Evident
R 9	Return Card	Evident
R 10	Return Passbook	Evident

2. Performance Requirements

- 버튼을 눌렀을 때 1초 안에 동작이 완료되어 다음 화면으로 넘어가야 한다.

3. Operating Environments

- **OS** : OS-Independent - Windows 7 / 10, MacOS

4. Development Environments

- **OS** : OS-Independent - Windows 7 / 10, MacOS
- **Language** : Java

- IDE : IntelliJ IDEA Ultimate / Community 2018.1

5. Other Requirements

- Simple 한 UI/UX 구조를 가진다.

ACTIVITY 1004 - RECORD TERMS IN GLOSSARY

Term	Description
User	ATM을 사용하는 객체
Commission	타행 거래 시 붙는 수수료
Password	각 계좌에 접근할 때 고객이 설정했던 비밀번호
Account	고객의 계좌번호
Bankbook	각 계좌 당 생성되는 통장
Card	각 계좌 당 n개 생성할 수 있는 카드. 신용카드와 체크카드로 나뉘어져 있다.
Bank	ATM과 거래하는 은행. ATM에 지정된 당행 외에 은행은 3개 존재한다.
Receipt	ATM에서 거래한 뒤 거래 내역을 명시한 명세표
Bank DB	거래하는 은행의 데이터베이스
Money Limit	ATM 거래 최대 한도

ACTIVITY 1005 - IMPLEMENT PROTOTYPE

입금

출금

거래

내역 확인

저희 국민 은행의 ATM에 오신 것을 환영합니다.
먼저 왼쪽에서 원하시는 작업을 선택해주세요.
만약 모르시는 것이 있으시다면 아래의 메뉴얼을
참고해주시기 바랍니다.

- 1.입금 : 현금을 통장이나 카드에 입금하거나,
무통장 입금이 가능합니다.
- 2.출금 : 통장이나 카드를 이용하여 계좌에서 현금을
가져오실 수 있습니다.
- 3.거래 : 타행이나 당행의 계좌로 돈을 송금하실 수 있습니다.
- 4.내역 확인 : 계좌의 내역과 잔액을 확인하실 수 있습니다.

무통장 입금

카드 / 통장

예금을 선택하셨습니다.
왼쪽에서 무통장 입금을 하실지, 카드나 통장을
이용하여 입금을 하실지 선택해주시시오.

- 1.무통장 입금 : 계좌번호를 이용하여 현금을 입금합니다.
- 2.카드 / 통장 : 카드나 통장을 넣어 현금을 입금합니다.

취소



출금을 선택하셨습니다.

먼저 확인하실 계좌의 카드나 통장을 넣으신 다음,
저희 ATM에서 확인할 때까지 기다려 주십시오.

취소

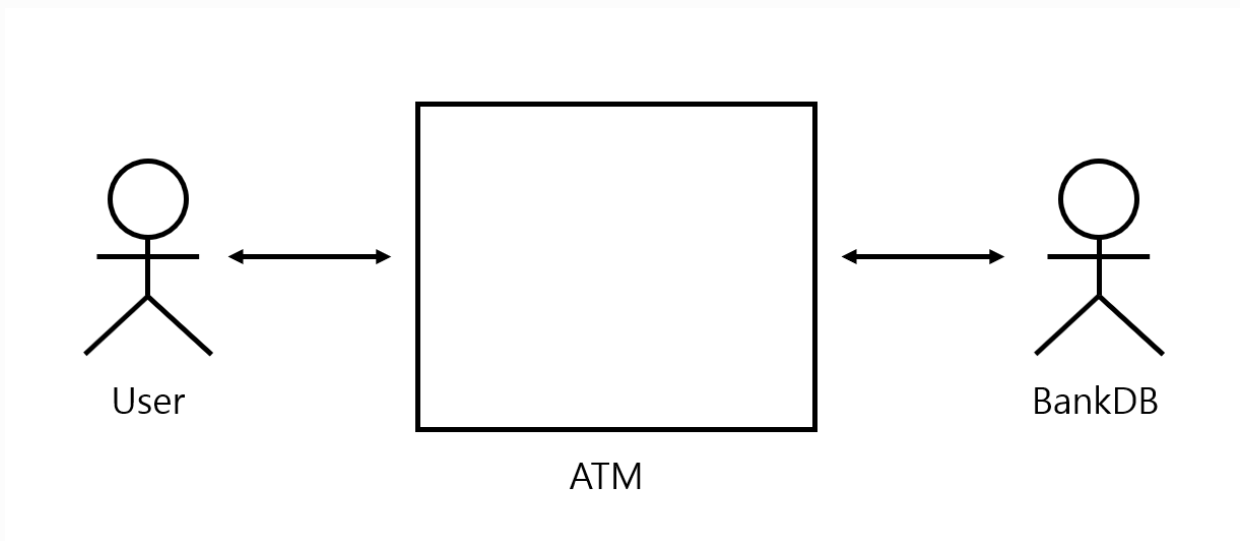


계좌번호를 입력해 주십시오. ('-'없이 입력해 주세요)

취소

ACTIVITY 1006 - DEFINE BUSINESS USE CASE

1. Define System Boundary



2. Identify Actors

- **User** : ATM 시스템을 사용하는 사람, 개발하는 사람 및 테스트 하는 사람

3. Actor - Goal List

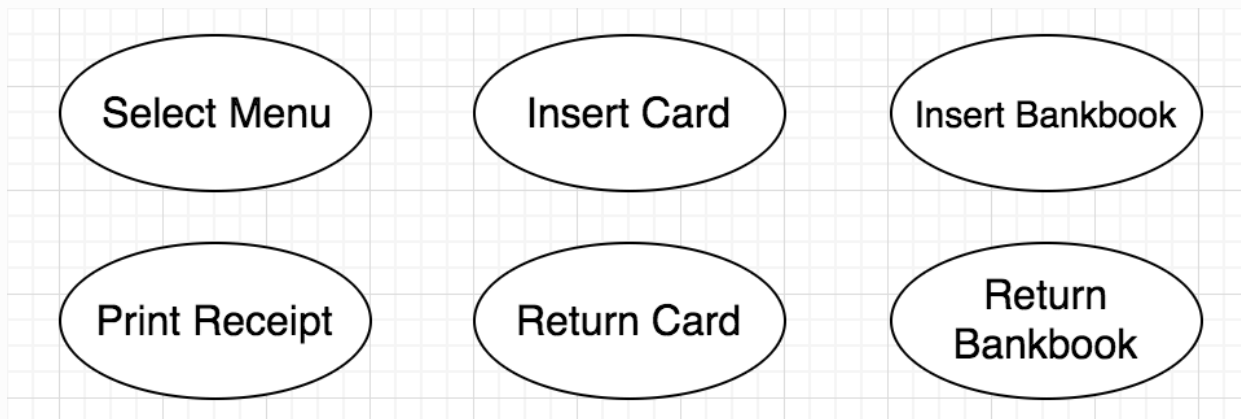
Actor	Goal
User	- 입금 / 출금 / 송금 / 거래내역 조회 - 보다 완벽한 프로그램 개발

4. Identify Use-Case

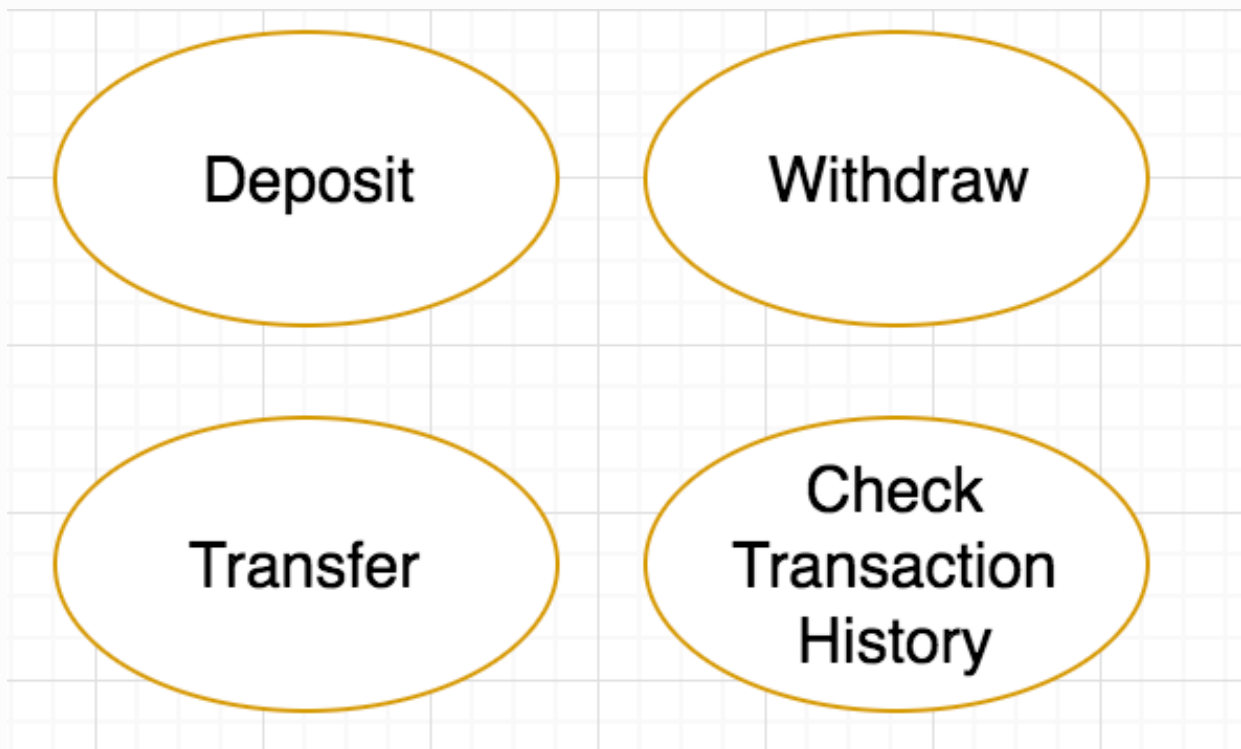
Reference No.	Function	Category
R 1	Select Menu	Actor
R 2	Insert Card	Actor
R 3	Insert Passbook	Actor
R 4	Deposit	Event
R 5	Withdraw	Event
R 6	Transfer	Event

R 7	Check Transaction History	Event
R 8	Print Receipt	Actor
R 9	Return Card	Actor
R 10	Return Passbook	Actor

4.1. Actor-based



4.2. Event-based



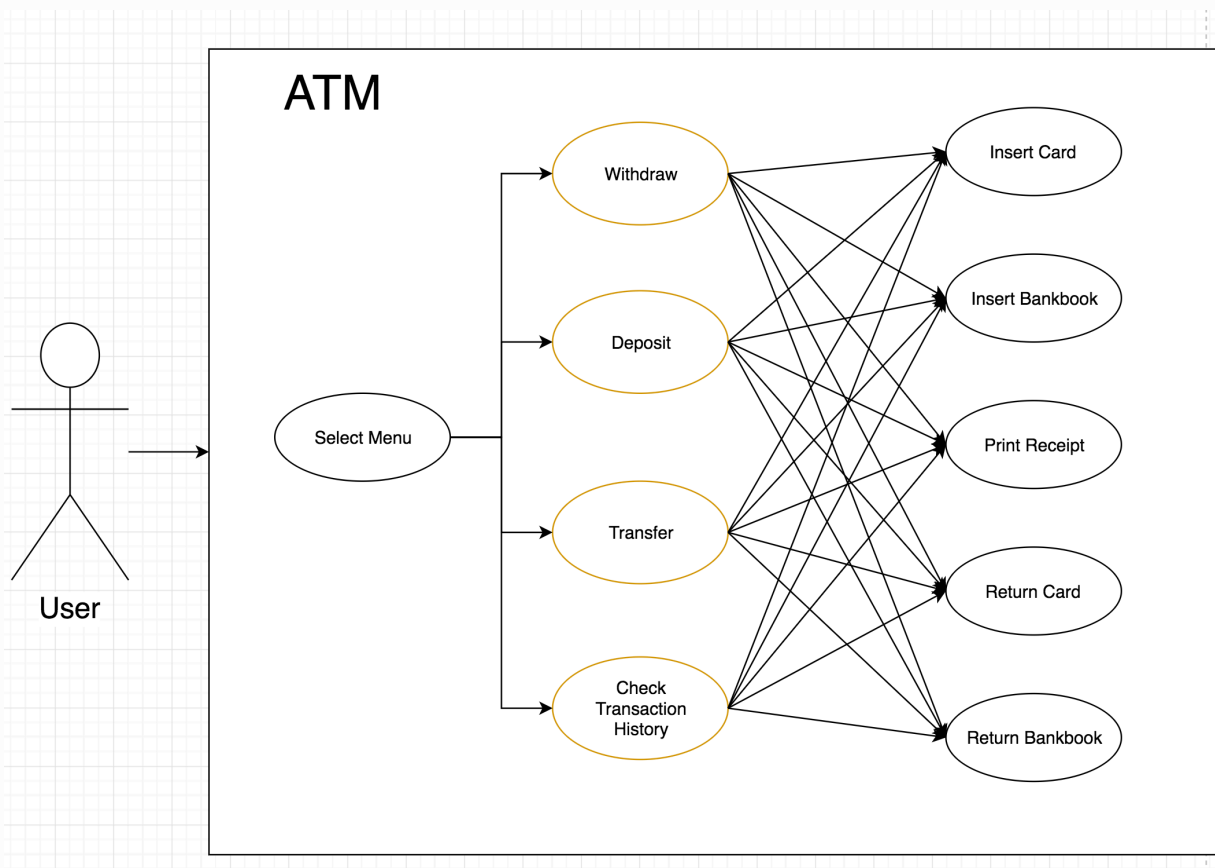
5. Allocate System Functions

Reference No.	Function	Use-Case
R 1	Select Menu	Select Menu
R 2	Insert Card	Insert Card
R 3	Insert Passbook	Insert Passbook
R 4	Deposit	Deposit
R 5	Withdraw	Withdraw
R 6	Transfer	Transfer
R 7	Check Transaction History	Check Transaction History
R 8	Print Receipt	Print Receipt
R 9	Return Card	Return Card
R 10	Return Passbook	Return Passbook

6. Categorize Use-Cases

Reference No.	Function	Use-Case	Category
R 1	Select Menu	Select Menu	Primary
R 2	Insert Card	Insert Card	Primary
R 3	Insert Passbook	Insert Passbook	Primary
R 4	Deposit	Deposit	Primary
R 5	Withdraw	Withdraw	Primary
R 6	Transfer	Transfer	Primary
R 7	Check Transaction History	Check Transaction History	Primary
R 8	Print Receipt	Print Receipt	Primary
R 9	Return Card	Return Card	Primary
R 10	Return Passbook	Return Passbook	Primary

7. Draw Use-Case Diagram



8. Describe Use-Cases

Use Case	Actor	Description
Select Menu	User	입금/출금/송금/계좌 확인 중 이용할 메뉴를 선택한다
Deposit	System	입금을 진행한다
Withdraw	System	출금을 진행한다
Transfer	System	송금을 진행한다
Check Transaction History	System	거래 내역 조회를 진행한다
Insert Card	User	카드를 투입한다
Insert Bankbook	User	통장을 투입한다
Print Receipt	User	거래 완료 후 명세표를 출력한다
Return Card	User	거래 완료 후 카드를 반환한다
Return Bankbook	User	거래 완료 후 통장을 반환한다

9. Rank Use-Cases

Reference No.	Function	Use-Case	Category
R 1	Select Menu	Select Menu	High
R 2	Insert Card	Insert Card	High
R 3	Insert Passbook	Insert Passbook	High
R 4	Deposit	Deposit	High
R 5	Withdraw	Withdraw	High
R 6	Transfer	Transfer	High
R 7	Check Transaction History	Check Transaction History	High
R 8	Print Receipt	Print Receipt	High
R 9	Return Card	Return Card	High
R 10	Return Passbook	Return Passbook	High

ACTIVITY 1007 - DEFINE BUSINESS CONCEPT MODEL

Deposit

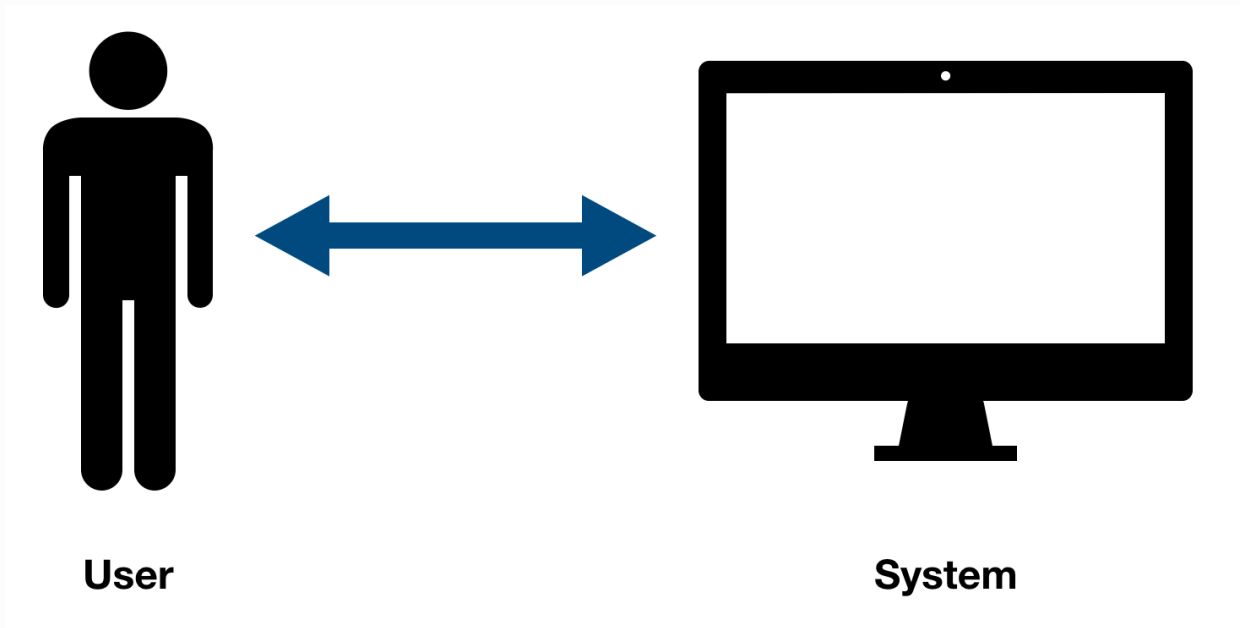
Withdraw

Transfer

Check

Receipt

ACTIVITY 1008 - DEFINE DRAFT SYSTEM ARCHITECTURE



ACTIVITY 1009 - DEFINE SYSTEM TEST CASE

Identifier	Functional Requirements	Description
ATM_STC_001	Select Menu	사용자가 메뉴 버튼을 클릭하면 다음 메뉴로 정상적으로 진행되는지 확인한다.
ATM_STC_002	Insert Card	카드가 정상적으로 투입되었는지 확인한다.
ATM_STC_003	Insert Passbook	통장이 정상적으로 투입되었는지 확인한다.
ATM_STC_004	Deposit	입금이 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
ATM_STC_005	Withdraw	출금이 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
ATM_STC_006	Transfer	송금이 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
ATM_STC_007	Check Transaction History	거래 내역 조회가 정상적으로 이루어지는지 확인한다.
ATM_STC_008	Print Receipt	명세표가 정상적으로 출력되는지 확인한다.
ATM_STC_009	Return Card	거래 종료 후 카드가 정상적으로 반환되는지 확인한다.
ATM_STC_010	Return Passbook	거래 종료 후 통장이 정상적으로 반환되는지 확인한다.

Identifier	Non-Functional Requirements	Description
ATM_STC_011	User Interface	각 버튼을 누르면 어떤 동작이 일어날지가 명확한지, 그리고 버튼 안의 글씨는 알아보기 쉬운지 확인한다.
ATM_STC_012	Performance	모든 버튼 입력 및 이벤트에 대하여 1초 이내로 반응하는지 확인한다.
ATM_STC_013	OS - Independent	프로그램이 OS에 무관하게 정상적으로 동작하는지 확인한다.

ACTIVITY 1010 - REFINE PLAN

1. Project Scope

은행 영업시간에 은행을 직접 방문하기 힘든 고객들이 언제든지 은행 서비스를 이용할 수 있는 서비스를 구축하고자 한다. ATM은 은행과 달리 영업시간의 제약을 받지 않을 수 있지만, 본인 확인 절차가 간소화 되어 있기에 이용가능한 서비스의 영역이 제한되어 있다. 24시간 이용가능한 또 다른 은행 서비스에는 인터넷뱅킹과 텔레뱅킹이 있다. ATM은 현금거래가 가능하다는 점에서 이들과 차이가 있다. 영업점, 인터넷뱅킹, 텔레뱅킹과 ATM의 차이점을 고려하면서 다양한 사용자들이 이용하기 편리한 은행 ATM 서비스를 만들고자 한다.

2. Project Objectives

- 다양한 이용자를 고려하는 시스템을 구축한다.
- 사용법이 간단한 시스템을 구축한다.
- 직관적인 인터페이스를 가진 시스템을 구축한다.

3. Functional Requirements

- Select Menu
- Insert Card
- Insert Passbook
- Deposit
- Withdraw
- Transfer
- Check Transaction History
- Print Receipt
- Return Card
- Return Passbook

4. Performance Requirement

- 모든 버튼 입력에 대한 반응 시간은 1초 이하이다.

5. Operating Environment

- **OS** : OS-Independent - Windows 7 / 10, MacOS
- **Language** : Java
- **IDE** : IntelliJ IDEA Ultimate / Community 2018.1

6. Resources

- Human Efforts : 4 People / 3 Months
- Human Resources : 4 (컴퓨터공학 * 2 + 물리학 * 1 + 미디어 커뮤니케이션 * 1)
- Duration : 12 weeks
- Budget : 10000(하루) * 5(일) * 4(주) * 3(달) * 4(사람 수) = 2,400,000(₩)