

OOPT STAGE 1000

PLAN AND ELABORATION

SUPER-SAFE ATM

Project Team

Team T4

Date

2018 / 04 / 29

Team Members

1. 201611269 신문기
2. 201610401 손하영
3. 201510283 임진웅

ACTIVITY 1001 - DEFINE DRAFT PLAN

1. Motivation

ATM 은 은행과 달리 영업시간의 제약을 받지 않을 수 있지만, 본인 확인 절차가 간소화 되어 있기에 이용가능한 서비스의 영역이 제한되어 있다. 서비스 영역에 제한을 두어도 보이스 피싱, 중고 거래 사기 등의 범죄로부터 ATM 이 이용자를 보호할 방법은 없다. 이용자로 하여금 범죄 이력이 있는 계좌를 조회할 수 있게 하고, 해당 계좌로의 송금을 할 수 없게 하여 보다 안전한 거래를 돕는 ATM 을 개발 하고자 한다.

2. Project Objectives

1. 안전한 거래를 할 수 있는 시스템을 개발한다.

3. Functional Requirements

1. Deposit
2. Withdraw
3. Transfer
4. Check Criminal Account
5. Check Transaction History

4. Non-functional Requirements

1. Safe Transaction
2. High Performance
3. OS-Independent

5. Resource Estimation

1. Human Efforts : 3 People / 3 Months
2. Human Resources : 3 (컴퓨터공학 * 1 + 물리학 * 1 + 미디어 커뮤니케이션 * 1)
3. Duration : 12 weeks
4. Budget : 10000(하루) * 5(일) * 4(주) * 3(달) * 4(사람 수) = 1,800,000(₩)

ACTIVITY 1002 - CREATE PRELIMINARY INVESTIGATION REPORT

1. Alternative Solutions

1. 현존 ATM 시스템을 구매한다.
2. 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 OOP 에 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 OOP 에 대한 이해력 부족	예전 년도 자료 혹은 책을 참고하거나, 교수님 및 조교님께 자문을 구한다.
개발 능력 부족	소속 팀원 중 2 명의 개발자가 나머지 팀원을 잘 뒷받침하여 개발을 이끌어 나간다.
디자인 능력 부족	소속 팀원 중 1 명의 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' BATTLEGROUNDS	과제를 모두 마무리한 다음에 게임에 접속한다.

5. Other Managerial Issues

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

ACTIVITY 1003 - DEFINE REQUIREMENTS

1. Functional Requirements

1.1 Functional Requirements

Function	Description
Deposit	<ol style="list-style-type: none"> 메뉴 선택 화면에서 유저가 입금 메뉴를 선택했을 때 진행된다. 통장이나 카드를 받아 계좌 정보를 BankDB 에서 받아 계좌가 유효한지 확인한다. 계좌의 비밀번호를 받아 비밀번호가 맞는지 확인한다.

	<ol style="list-style-type: none"> 4. 비밀번호가 맞는 경우 유저에게서 돈을 받을 준비를 하고 유저에게서 돈을 받고 센다. 5. 유저의 계좌에 입금을 진행한다. 이 때, 유저가 입금을 진행하고 있는 ATM 과 계좌의 은행이 다를 경우 수수료를 계산한다. 6. 명세표를 출력할지 유저에게서 확인 받고 명세표를 출력할 경우 명세표를 출력한다. 7. 유저에게 통장 또는 카드를 돌려준다. 8. 메뉴 선택 화면으로 돌아간다.
Withdraw	<ol style="list-style-type: none"> 1. 메뉴 선택 화면에서 유저가 출금 메뉴를 선택했을 때 진행된다. 2. 통장이나 카드를 받아 계좌 정보를 BankDB 에서 받아 계좌가 유효한지 확인한다. 3. 계좌의 비밀번호를 받아 비밀번호가 맞는지 확인한다. 4. 비밀번호가 맞는 경우 유저에게서 내보낼 돈의 양을 입력 받는다. 5. 유저의 계좌에서 출금을 진행한다. 이 때, 유저가 출금을 진행하고 있는 ATM 과 계좌의 은행이 다를 경우 수수료를 계산하고, ATM 에 충분한 돈이 있는지 확인한다. 6. 명세표를 출력할지 유저에게서 확인 받고 명세표를 출력할 경우 명세표를 출력한다. 7. 유저에게 통장 또는 카드를 돌려준다. 8. 메뉴 선택 화면으로 돌아간다.
Transfer	<ol style="list-style-type: none"> 1. 메뉴 선택 화면에서 유저가 거래 메뉴를 선택했을 때 진행된다. 2. 통장이나 카드를 받아 계좌 정보를 BankDB 에서 받아 계좌가 유효한지 확인한다. 3. 계좌의 비밀번호를 받아 비밀번호가 맞는지 확인한다. 4. 비밀번호가 맞는 경우 유저에게서 송금할 계좌를 입력 받고 송금할 계좌가 유효한지 확인한다.

	<ol style="list-style-type: none"> 5. 유저에게서 송금할 금액을 받는다. 6. 유저의 계좌에서 송금을 진행한다. 이 때, 유저가 송금을 진행하고 있는 ATM 과 송금할 계좌의 은행이 다를 경우 수수료를 계산한다. 7. 명세표를 출력할지 유저에게서 확인 받고 명세표를 출력할 경우 명세표를 출력한다. 8. 유저에게 통장 또는 카드를 돌려준다. 9. 메뉴 선택 화면으로 돌아간다.
<p>Check Transaction History</p>	<ol style="list-style-type: none"> 1. 메뉴 선택 화면에서 유저가 계좌 조회 메뉴를 선택했을 때 진행된다. 2. 통장이나 카드를 받아 계좌 정보를 BankDB 에서 받아 계좌가 유효한지 확인한다. 3. 계좌의 비밀번호를 받아 비밀번호가 맞는지 확인한다. 4. 비밀번호가 맞는 경우 계좌 잔액을 표시하고 거래내역 조회 버튼을 표시한다. 5. 거래내역을 조회해서 화면에 표시한다. 6. 매체가 통장일 경우 통장 정리 버튼을 표시하고, 거래 내역을 통장에 출력한다. 7. 유저에게 통장 또는 카드를 돌려준다. 8. 메뉴 선택 화면으로 돌아간다.
<p>Check Criminal History</p>	<ol style="list-style-type: none"> 1. 메뉴 선택 화면에서 유저가 범죄 이력 조회 메뉴를 선택했을 때 진행된다. 2. 유저에게서 조회할 계좌번호를 받은 뒤 계좌 정보를 BankDB 에서 받아 계좌가 유효한지 확인한다. 3. 해당 계좌가 범죄에 사용된 이력이 있는지 확인한다. 4. 이력이 있을 경우 누적 횟수와 세부 내용을 출력하고, 없다면 '범죄 이력이 없습니다'를 출력한다. 5. 메뉴 선택 화면으로 돌아간다.

1.2 Reference Number & Categorized and Identified Functions

Reference No.	Function	Category
---------------	----------	----------

R 1	Deposit	Evident
R 2	Withdraw	Evident
R 3	Transfer	Evident
R 4	Check Transaction History	Evident
R 5	Check Criminal History	Evident

2. Performance Requirements

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

3. Operating Environments

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

4. Development Environments

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

5. Other Requirements

1. Simple 한 UI/UX 구조를 가진다.

ACTIVITY 1004 - RECORD TERMS IN GLOSSARY

Term	Description
User	ATM 을 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 n 개 생성할 수 있는 카드. 신용카드와 체크카드로 나뉘어져 있다.

Bank	ATM 과 거래하는 은행. ATM 에 지정된 당행 외에 은행은 3 개 존재한다.
Receipt	ATM 에서 거래한 뒤 거래 내역을 명시한 명세표
Bank DB	거래하는 은행의 데이터베이스
Money Limit	ATM 거래 최대 한도
CrimeNum	범죄에 사용된 횟수

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

예금을 선택하셨습니다.

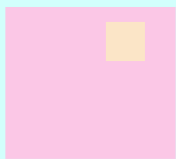
왼쪽에서 무통장 입금을 하실지, 카드나 통장을 이용하여 입금을 하실지 선택해주십시오.

1.무통장 입금 : 계좌번호를 이용하여 현금을 입금합니다.

2.카드 / 통장 : 카드나 통장을 넣어 현금을 입금합니다.

취소

카드 / 통장

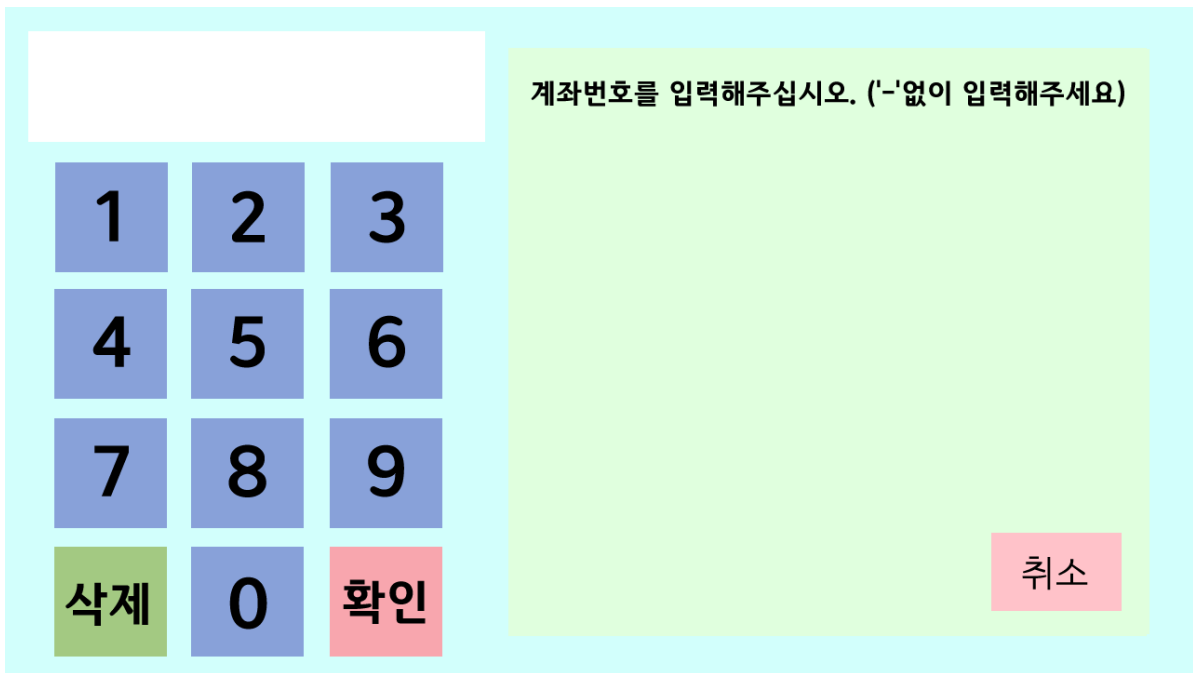


KUKMIN

출금을 선택하셨습니다.

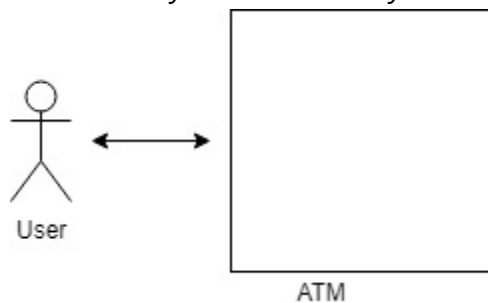
먼저 확인하실 계좌의 카드나 통장을 넣으신 다음, 저희 ATM에서 확인할 때까지 기다려 주십시오.

취소



ACTIVITY 1006 - DEFINE BUSINESS USE CASE

1. Define System Boundary



2. Identify Actors

1. User : ATM 시스템을 사용하는 사람, 개발하는 사람 및 테스트 하는 사람

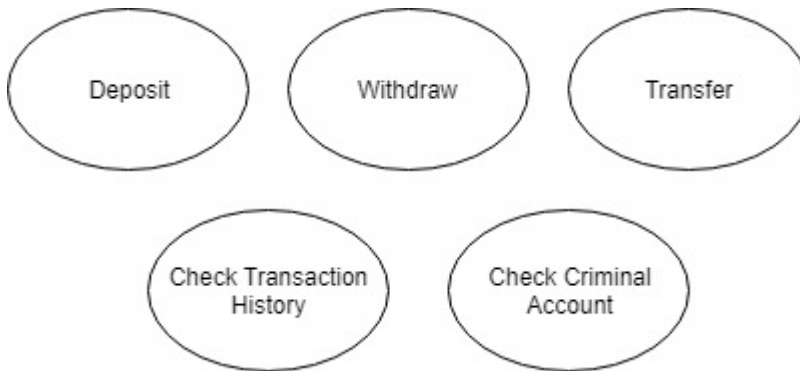
3. Actor - Goal List

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

4. Identify Use-Case

Reference No.	Function	Category
R 1	Deposit	Actor
R 2	Withdraw	Actor
R 3	Transfer	Actor
R 4	Check Transaction History	Actor
R 5	Check Criminal History	Actor

4.1. Actor-based



5. Allocate System Functions

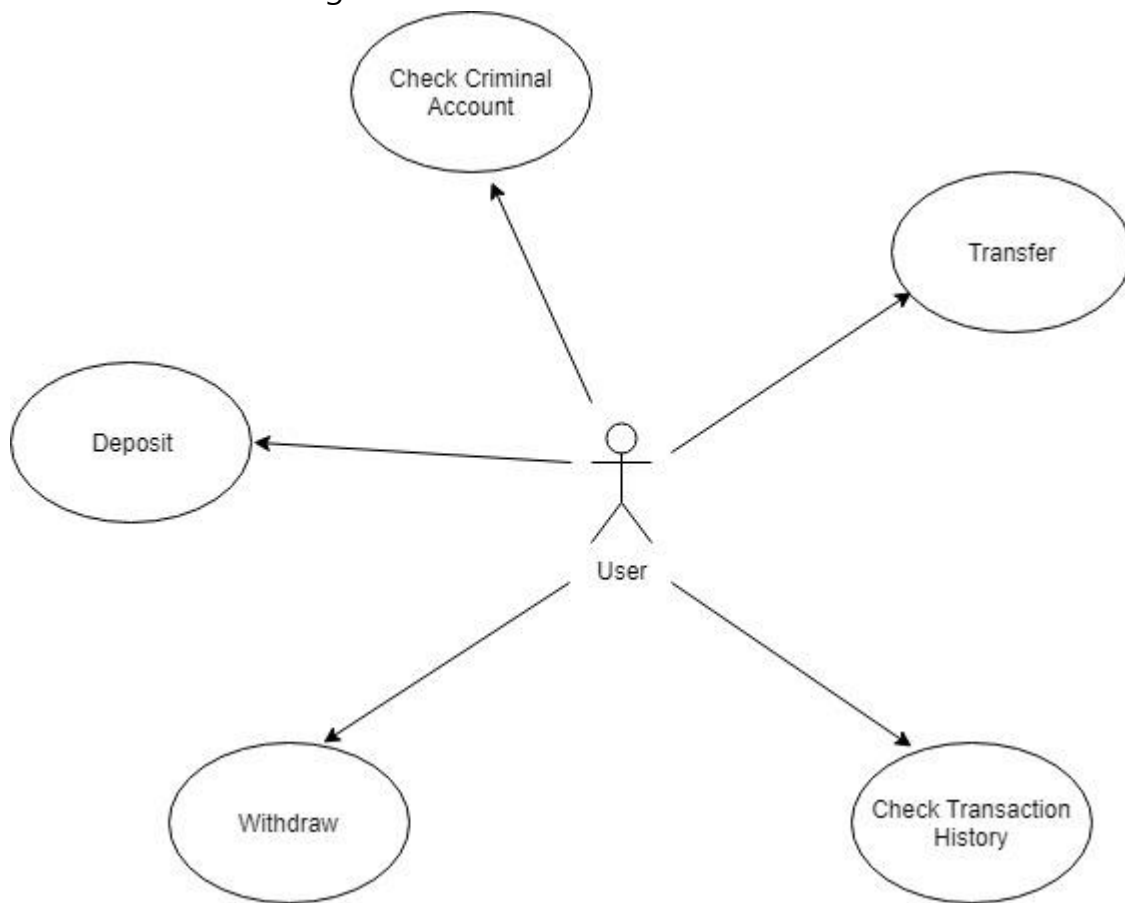
Reference No.	Function	Use-Case
R 1	Deposit	Deposit
R 2	Withdraw	Withdraw
R 3	Transfer	Transfer
R 4	Check Transaction History	Check Transaction History
R 5	Check Criminal History	Check Criminal History

6. Categorize Use-Cases

Reference No.	Function	Use-Case	Category
R 1	Deposit	Deposit	Primary

R 2	Withdraw	Withdraw	Primary
R 3	Transfer	Transfer	Primary
R 4	Check Transaction History	Check Transaction History	Primary
R 5	Check Criminal History	Check Criminal History	Primary

7. Draw Use-Case Diagram



8. Describe Use-Cases

Use Case	1.Deposit
Actor	User
Description	1. 유저가 입금 메뉴를 선택한다. 2. 통장이나 카드를 넣는다.

	<ol style="list-style-type: none"> 3. 통장이나 카드에 연결된 계좌의 비밀번호를 넣는다. 4. 비밀번호가 맞을 경우 입금할 돈을 ATM에 넣는다. 5. 입금을 진행한다. 6. 명세표를 출력할 지 결정하고, 출력할 경우 명세표를 받는다. 7. ATM에 삽입한 통장이나 카드를 되돌려 받는다.
--	--

Use Case	2. Withdraw
Actor	User
Description	<ol style="list-style-type: none"> 1. 유저가 출금 메뉴를 선택한다. 2. 통장이나 카드를 넣는다. 3. 통장이나 카드에 연결된 계좌의 비밀번호를 넣는다. 4. 비밀번호가 맞을 경우 출금할 돈을 ATM에 입력한다. 5. 출금을 진행하고 ATM에게서 현금을 받는다. 6. 명세표를 출력할 지 결정하고, 출력할 경우 명세표를 받는다. 7. ATM에 삽입한 통장이나 카드를 되돌려 받는다.

Use Case	3. Transfer
Actor	User
Description	<ol style="list-style-type: none"> 1. 유저가 거래 메뉴를 선택한다. 2. 통장이나 카드를 넣는다. 3. 통장이나 카드에 연결된 계좌의 비밀번호를 넣는다. 4. 비밀번호가 맞을 경우 송금할 계좌를 입력한다. 5. 송금할 돈의 양을 ATM에 넣고 송금을 진행한다. 6. 명세표를 출력할 지 결정하고, 출력할 경우 명세표를 받는다. 7. ATM에 삽입한 통장이나 카드를 되돌려 받는다.

Use Case	4. Check Transaction History
Actor	User
Description	<ol style="list-style-type: none"> 1. 유저가 계좌 조회 메뉴를 선택한다. 2. 통장이나 카드를 넣는다. 3. 통장이나 카드에 연결된 계좌의 비밀번호를 넣는다. 4. 비밀번호가 맞을 경우 거래 내역 조회를 진행한다. 5. 넣은 매체가 통장일 경우, 통장 정리를 진행한다. 6. ATM 에 삽입한 통장이나 카드를 되돌려 받는다.

Use Case	5. Check Criminal History
Actor	User
Description	<ol style="list-style-type: none"> 1. 유저가 범죄이력조회 메뉴를 선택한다. 2. 이력을 확인하고 싶은 계좌번호를 입력한다. 3. 해당 계좌가 범죄에 사용된 이력이 있는지 확인한다.

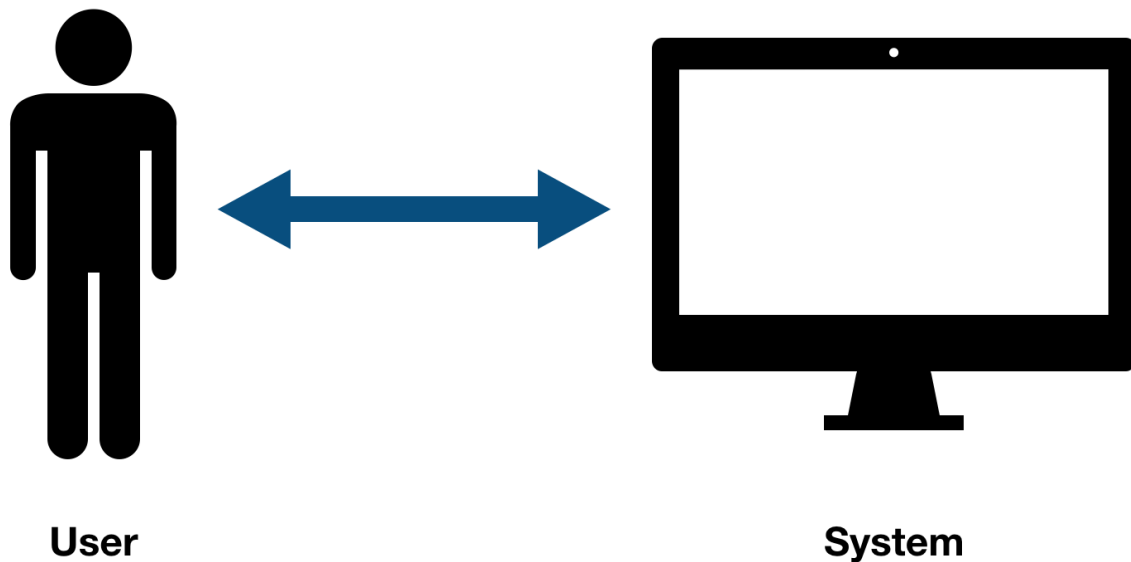
9. Rank Use-Cases

Reference No.	Function	Use-Case	Category
R 1	Deposit	Deposit	High
R 2	Withdraw	Withdraw	High
R 3	Transfer	Transfer	High
R 4	Check Transaction History	Check Transaction History	High
R 5	Check Criminal History	Check Criminal History	High

ACTIVITY 1007 - DEFINE BUSINESS CONCEPT MODEL

Deposit	Withdraw	Transfer
Check	Transaction History	Criminal History
Account	Bank	User
Card	Passbook	ATM

ACTIVITY 1008 - DEFINE DRAFT SYSTEM ARCHITECTURE



ACTIVITY 1009 - DEFINE SYSTEM TEST CASE

Identifier	Functional Requirements	Description
ATM_STC_001	Deposit	사용자가 Deposit 버튼을 누르고 입금이 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
ATM_STC_002	WithDraw	사용자가 WithDraw 버튼을 누르고 출금이 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
ATM_STC_003	Transfer	사용자가 Transfer 버튼을 누르고 송금이 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
ATM_STC_004	Check Transaction History	사용자가 거래 내역 조회 버튼을 누르고 거래 내역 조회가 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
ATM_STC_005	Check Criminal History	사용자가 범죄 이력 조회 버튼을 누르고 범죄 이력 조회가 정상적인 상황에서 정상적으로 이루어지는지 확인한다.
Identifier	Non-Functional Requirements	Description
ATM_STC_006	Safe Transaction	범죄 이력이 있는 계좌의 조회 및 해당 계좌로의 거래를 중지할 수 있는지 확인한다.
ATM_STC_007	Performance	모든 버튼 입력 및 이벤트에 대하여 1 초 이내로 반응하는지 확인한다.

ATM_STC_008	OS - Independent	프로그램이 OS 에 무관하게 정상적으로 동작하는지 확인한다.
-------------	------------------	-----------------------------------

ACTIVITY 1010 - REFINE PLAN

1. Project Scope

ATM 은 은행과 달리 영업시간의 제약을 받지 않을 수 있지만, 본인 확인 절차가 간소화 되어 있기에 이용가능한 서비스의 영역이 제한되어 있다. 서비스 영역에 제한을 두어도 보이스 피싱, 중고 거래 사기 등의 범죄로부터 ATM 이 이용자를 보호할 방법은 없다. 이용자로 하여금 범죄 이력이 있는 계좌를 조회할 수 있게 하고, 해당 계좌로의 송금을 할 수 없게 하여 보다 안전한 거래를 돕는 ATM 을 개발 하고자 한다.

2. Project Objectives

안전한 거래를 할 수 있는 시스템을 개발한다.

3. Functional Requirements

1. Deposit
2. Withdraw
3. Transfer
4. Check Criminal History
5. Check Transaction History

4. Performance Requirement

1. 모든 버튼 입력에 대한 반응 시간은 1 초 이하이다.

5. Operating Environment

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

3. **IDE** : IntelliJ IDEA Ultimate / Community 2018.1

6. Resources

1. Human Efforts : 3 People / 3 Months
2. Human Resources : 3 (컴퓨터공학 * 1 + 물리학 * 1 + 미디어 커뮤니케이션 * 1)
3. Duration : 12 weeks
4. Budget : 10000(하루) * 5(일) * 4(주) * 3(달) * 3(사람 수) = 1,800,000(₩)