

OOPT STAGE 1000

PLAN AND ELABORATION

SUPER-SAFE ATM

Project Team

Team T4

Date

2018 / 05 / 30

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

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
selectMenu	<ol style="list-style-type: none"> 입금, 출금, 송금, 거래내역 확인, 범죄이력조회 중 이용할 메뉴를 선택한다. 유저가 선택한 메뉴에 따라 입금, 출금, 송금, 거래내역 확인, 범죄이력조회가 진행된다.
MediumCheck	<ol style="list-style-type: none"> 카드 번호를 입력할지 계좌 번호를 입력할지 선택한다. 선택에 따라 카드 번호나 계좌 번호를 입력한다. 입력된 계좌 번호나 입력된 카드 번호에 연결된 계좌 번호가 DB에 존재하는 계좌인지 확인한다. 유효하지 않을 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.
Deposit	<ol style="list-style-type: none"> 입금할 금액을 입력받는다. 계좌 번호에 해당되는 비밀번호를 입력받고 입력받은 비밀번호가

	<p>해당 계좌에 맞는 비밀번호인지 확인한다. 유효하지 않을 경우 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</p> <p>3. 계좌에 타행 거래일 경우 입력받은 돈에서 수수료를 제외하고 입금한다.</p> <p>4. 계좌에 돈을 입금한 후 실행한 작업과 입금한 금액, 계좌의 잔액을 출력한다.</p>
Withdraw	<p>1. 출금할 금액을 입력받는다.</p> <p>2. 타행 거래일 경우 수수료를 계산하고 수수료를 포함하여 출금할 계좌의 잔액이 충분한지 확인한다.</p> <p>3. 계좌 번호에 해당되는 비밀번호를 입력받고 입력받은 비밀번호가 해당 계좌에 맞는 비밀번호인지 확인한다. 유효하지 않을 경우 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</p> <p>4. 계좌에 타행 거래일 경우 입력받은 돈에서 수수료를 제외하고 출금한다.</p> <p>5. 계좌에서 돈을 출금한 후 실행한 작업과 출금한 금액, 계좌의 잔액을 출력한다.</p>
Transfer	<p>1. 송금할 금액을 입력받는다.</p> <p>2. 돈을 보낼 계좌를 입력하고 입력된 계좌 번호가 DB에 존재하는 계좌인지 확인한다. 유효하지 않을 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</p> <p>3. 타행 거래일 경우 수수료를 계산하고 수수료를 포함하여 송금할 계좌의 잔액이 충분한지 확인한다.</p> <p>4. 계좌 번호에 해당되는 비밀번호를 입력받고 입력받은 비밀번호가 해당 계좌에 맞는 비밀번호인지 확인한다. 유효하지 않을 경우 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</p> <p>5. 계좌에 타행 거래일 경우 입력받은 돈에서 수수료를 제외하여 송금한다.</p> <p>6. 계좌에서 돈을 송금한 후 실행한 작업과 송금한 금액, 계좌의 잔액을 출력한다.</p>
Check TransactionHistory	<p>1. 입력받은 계좌의 거래내역을 출력한다.</p>
CheckCriminal History	<p>1. 범죄이력을 조회하고자 하는 계좌를 입력하고 입력된 계좌번호가 DB에 존재하는 계좌인지 확인한다. 유효하지 않을 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</p> <p>2. 입력된 계좌에 대한 범죄이력을 출력한다.</p>

1.2 Reference Number & Categorized and Identified Functions

Reference No.	Function	Category
R 1	selectMenu	Evident
R 2	MediumCheck	Evident
R 3	Deposit	Evident
R 4	Withdraw	Evident
R 5	Transfer	Evident
R 6	CheckTransactionHistory	Evident
R 7	CheckCriminalHistory	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.

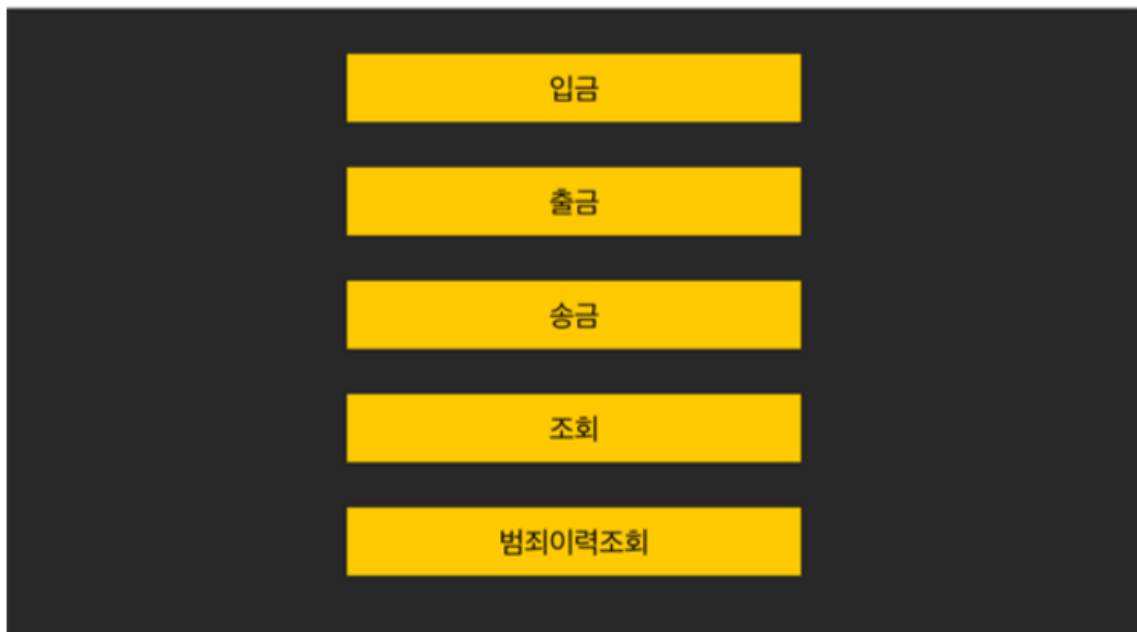
2. Performance Requirements

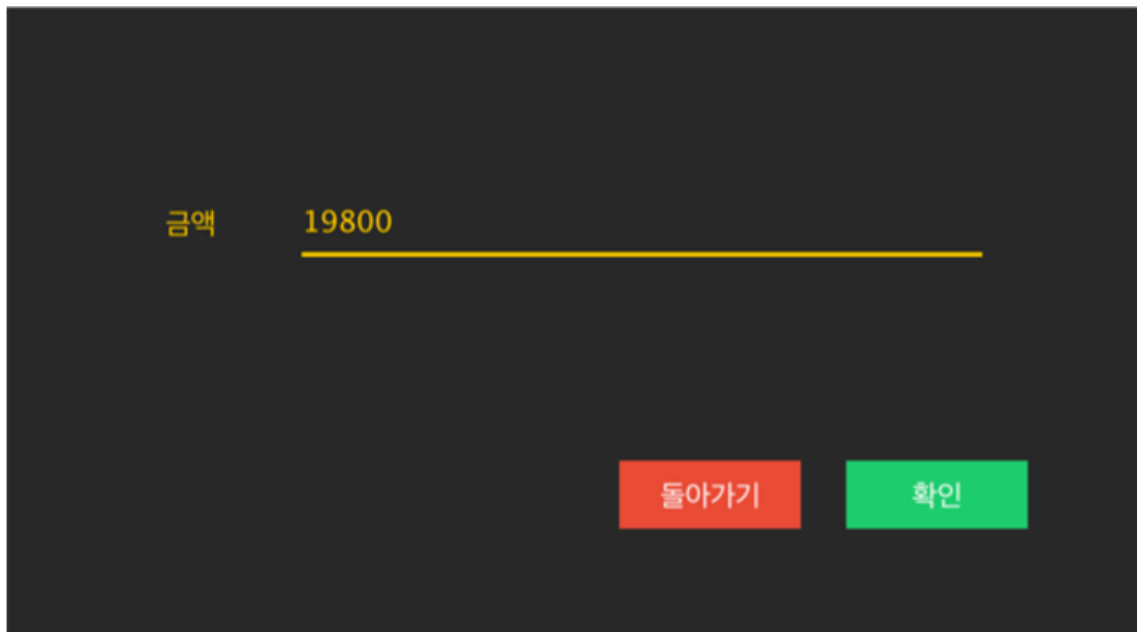
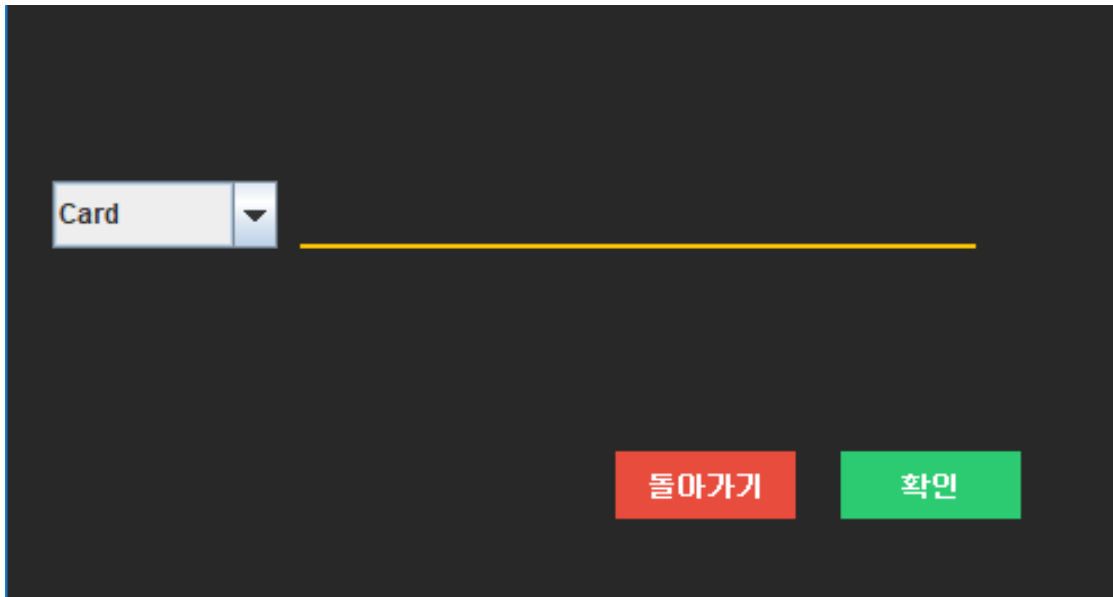
1. 버튼을 눌렀을 때 1초 안에 동작이 완료되어 다음 화면으로 넘어가야 한다.

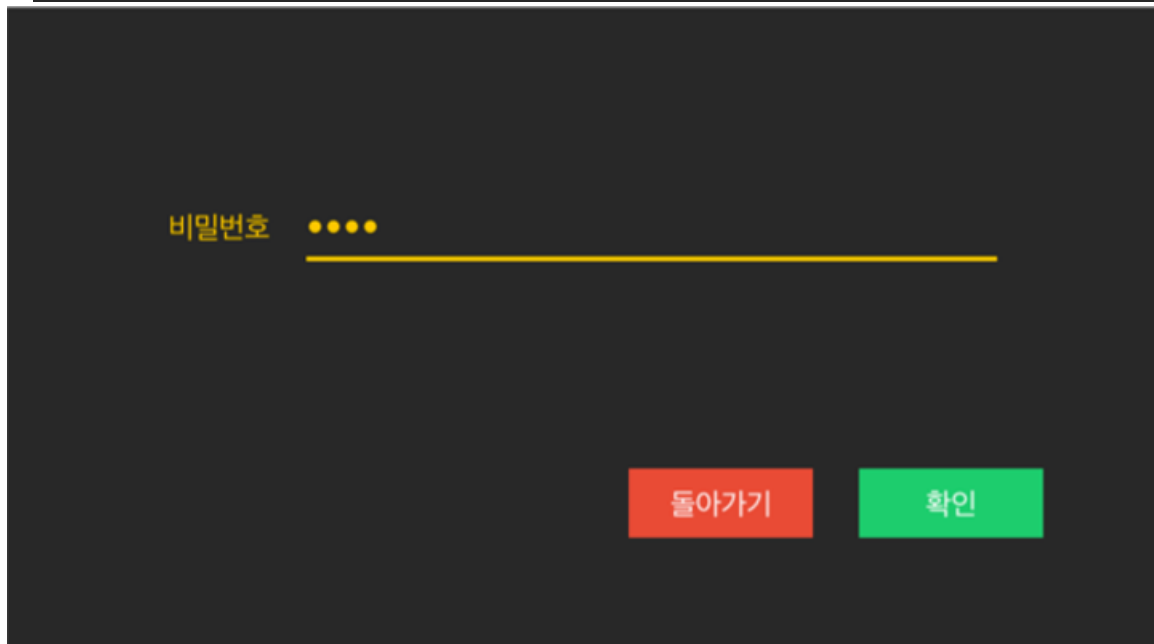
ACTIVITY 1004 - RECORD TERMS IN GLOSSARY

Term	Description
User	ATM 을 사용하는 객체
fee	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 생성할 수 있는 카드.
Bank	ATM 과 거래하는 은행. ATM 에 지정된 당행 외에 은행은 3 개 존재한다.
Result	ATM 에서 거래한 뒤 거래 내역을 명시한 명세표
DB	거래하는 은행의 데이터베이스
criminalLog	범죄에 사용된 이력

ACTIVITY 1005 - IMPLEMENT PROTOTYPE

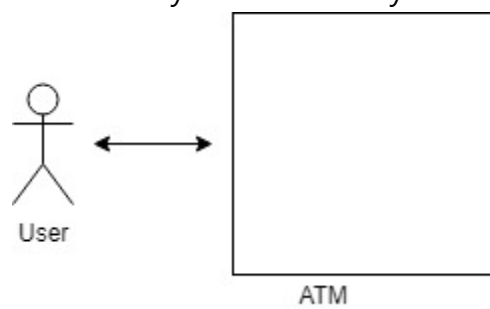






ACTIVITY 1006 - DEFINE BUSINESS USE CASE

1. Define System Boundary



2. Identify Actors

1. User : ATM 시스템을 사용하는 사람

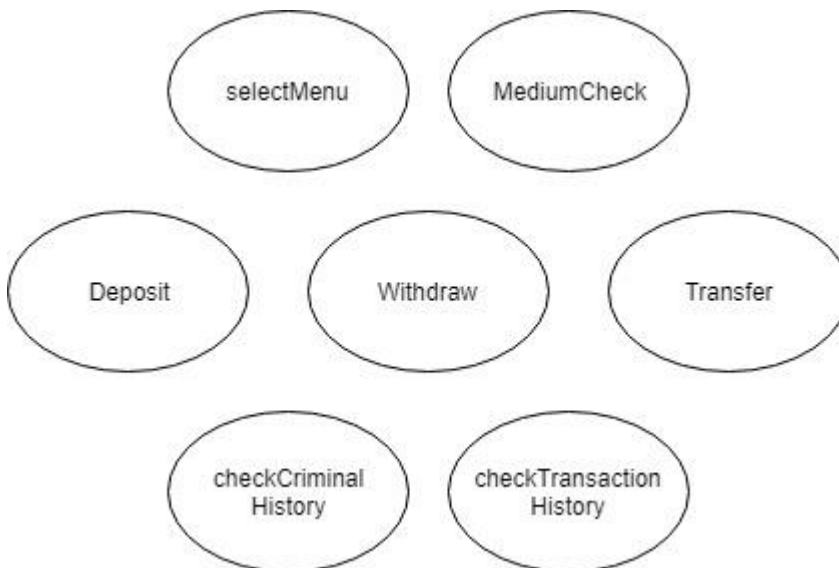
3. Actor - Goal List

Actor	Goal
User	<ul style="list-style-type: none"> - 입금/ 출금/ 송금/ 거래내역 조회/ 범죄이력 조회 중 하나를 선택하여 진행할 수 있다. - 범죄이력 조회가 가능하여 보다 안전한 거래를 할 수 있다.

4. Identify Use-Case

Reference No.	Function	Category
R 1	selectMenu	Actor
R 2	MediumCheck	Actor
R 3	Deposit	Actor
R 4	Withdraw	Actor
R 5	Transfer	Actor
R 6	CheckTransactionHistory	Actor
R 7	CheckCriminalHistory	Actor

4.1. Actor-based



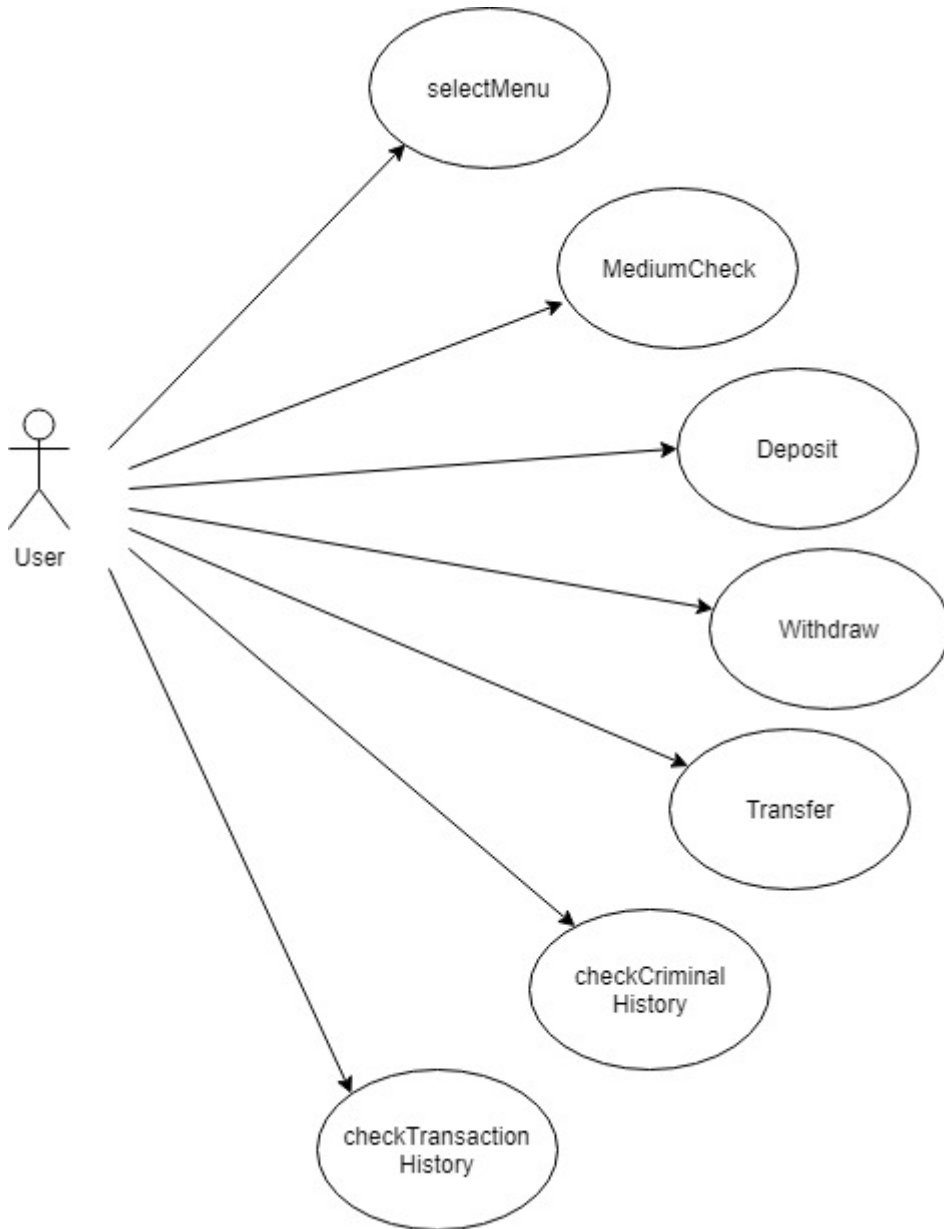
5. Allocate System Functions

Reference No.	Function	Use-Case
R 1	selectMenu	selectMenu
R 2	MediumCheck	MediumCheck
R 3	Deposit	Deposit
R 4	Withdraw	Withdraw
R 5	Transfer	Transfer
R 6	CheckTransactionHistory	CheckTransactionHistory
R 7	CheckCriminalHistory	CheckCriminalHistory

6. Categorize Use-Cases

Reference No.	Function	Use-Case	Category
R 1	selectMenu	selectMenu	Primary
R 2	MediumCheck	MediumCheck	Primary
R 3	Deposit	Deposit	Primary
R 4	Withdraw	Withdraw	Primary
R 5	Transfer	Transfer	Primary
R 6	CheckTransactionHistory	CheckTransactionHistory	Primary
R 7	CheckCriminalHistory	CheckCriminalHistory	Primary

7. Draw Use-Case Diagram



8. Describe Use-Cases

Use Case	1.selectMenu
Actor	User
Description	1. 유저가 화면에 출력되는 입금, 출금, 송금, 거래내역 확인, 범죄이력조회 5 개의 메뉴 선택 버튼 중 하나를 눌러 이용할 메뉴를 선택한다.

Use Case	2. MediumCheck
Actor	User
Description	<ol style="list-style-type: none"> 1. 유저가 카드 번호를 입력할지 계좌 번호를 입력할지 선택한다. 2. 선택에 따라 카드 번호나 계좌 번호를 입력한다. 3. 계좌 번호가 유효하지 않다고 알림을 받을 경우 다시 입력한다.

Use Case	3. Deposit
Actor	User
Description	<ol style="list-style-type: none"> 1. 입금할 금액을 입력한다. 2. 계좌 번호에 해당되는 비밀번호를 입력한다. 입력한 비밀번호가 유효하지 않을 경우 알림을 받고 다시 입력한다. 3. 계좌에 돈을 입금하고 화면에 실행된 작업과 입금한 금액, 계좌의 잔액이 출력된 것을 확인한다. 4. 끝내기 버튼을 눌러 메인 화면으로 돌아간다.

Use Case	4. Withdraw
Actor	User
Description	<ol style="list-style-type: none"> 1. 출금할 금액을 입력한다. 2. 출금할 계좌의 잔액이 부족하다고 경고가 뜰 경우 메뉴 선택 화면으로 돌아간다. 3. 잔액이 부족하지 않을 경우 계좌 번호에 해당되는 비밀번호를 입력한다. 입력한 비밀번호가 유효하지 않을 경우 알림을 받고 다시 입력한다. 4. 계좌에서 돈을 출금하고 화면에 실행된 작업과 출금한 금액, 계좌의 잔액이 출력된 것을 확인한다.

	5. 끝내기 버튼을 눌러 메인 화면으로 돌아간다.
--	-----------------------------

Use Case	5. Transfer
Actor	User
Description	<ol style="list-style-type: none"> 1. 송금할 금액을 입력한다. 2. 돈을 보낼 계좌를 입력하고 입력된 계좌번호가 유효하지 않다고 경고가 뜰 경우 다시 입력한다. 3. 송금할 계좌의 잔액이 부족하다고 경고가 뜰 경우 메뉴 선택 화면으로 돌아간다. 4. 잔액이 부족하지 않을 경우 계좌 번호에 해당되는 비밀번호를 입력한다. 입력한 비밀번호가 유효하지 않을 경우 알림을 받고 다시 입력한다. 5. 계좌에서 돈을 송금하고 화면에 실행된 작업과 송금한 금액, 계좌의 잔액이 출력된 것을 확인한다. 6. 끝내기 버튼을 눌러 메인 화면으로 돌아간다.

Use Case	6. CheckTransactionHistory
Actor	User
Description	<ol style="list-style-type: none"> 1. 입력한 계좌의 거래내역이 화면에 출력된 것을 확인한다. 2. 끝내기 버튼을 눌러 메인 화면으로 돌아간다.

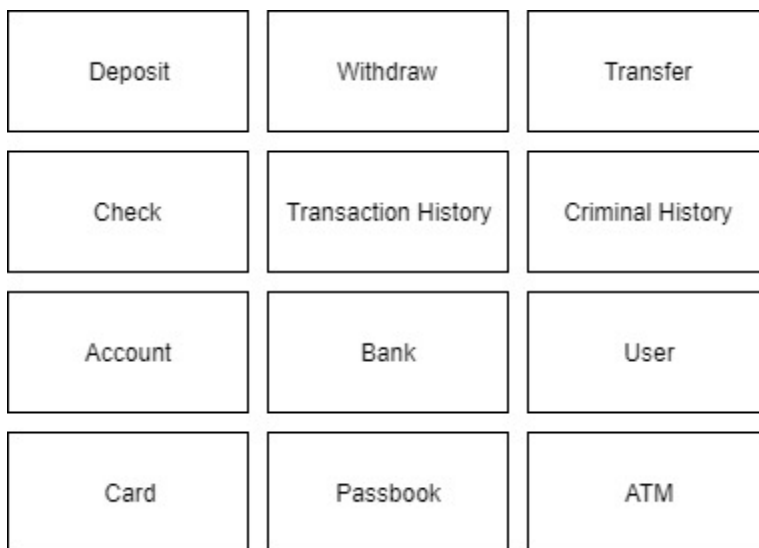
Use Case	7. CheckTransactionHistory
Actor	User
Description	<ol style="list-style-type: none"> 1. 범죄 이력을 조회하고자 하는 계좌를 입력하고 입력된 계좌번호가 유효하지 않다고 경고가 뜰 경우 다시 입력한다.

	<p>2. 입력한 계좌의 범죠크내역이 화면에 출력된 것을 확인한다.</p> <p>3. 끝내기 버튼을 눌러 메인 화면으로 돌아간다.</p>
--	--

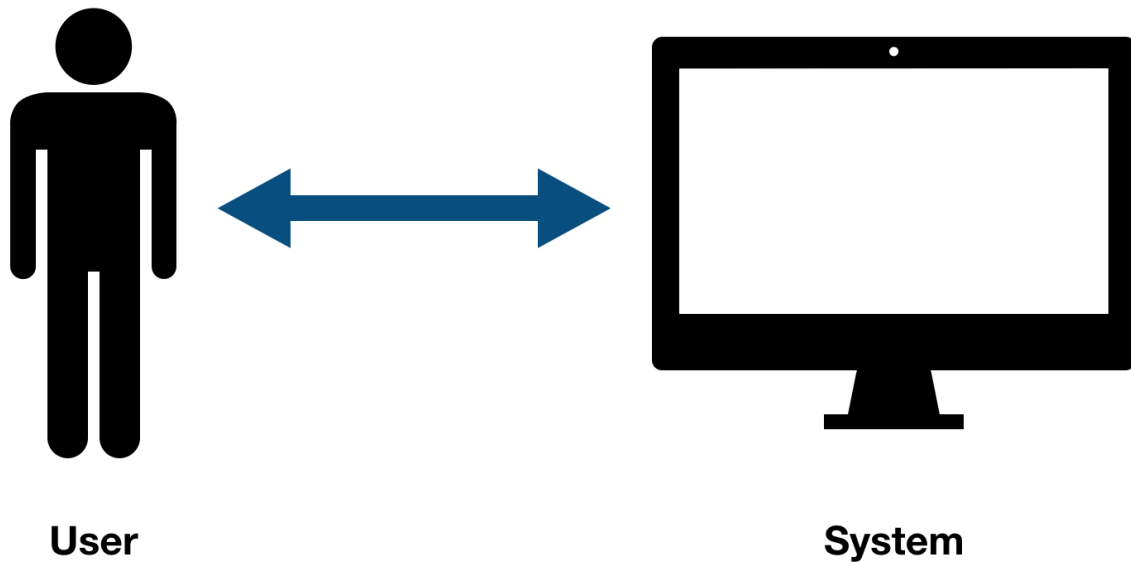
9. Rank Use-Cases

Reference No.	Function	Use-Case	Category
R 1	selectMenu	selectMenu	High
R 2	MediumCheck	MediumCheck	High
R 3	Deposit	Deposit	High
R 4	Withdraw	Withdraw	High
R 5	Transfer	Transfer	High
R 6	CheckTransactionHistory	CheckTransactionHistory	High
R 7	CheckCriminalHistory	CheckCriminalHistory	Primary

ACTIVITY 1007 - DEFINE BUSINESS CONCEPT MODEL



ACTIVITY 1008 - DEFINE DRAFT SYSTEM ARCHITECTURE



ACTIVITY 1009 - DEFINE SYSTEM TEST CASE

Identifier	Functional Requirements	Description
ATM_STC_001	selectMenu	사용자가 입금, 출금, 송금, 거래내역 확인, 범죄이력조회 중 사용할 메뉴를 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.
ATM_STC_002	MediumCheck	카드 번호나 계좌 번호를 입력할지 선택하고 이에 따라 카드 번호나 계좌 번호를 입력했을 때 계좌 확인이 정상적으로 진행되며 유효하지 않은 번호를 넣었을 때 다시 입력받을 수 있는지 확인한다.
ATM_STC_003	Deposit	입금할 금액을 잘 입력받을 수 있는지 확인한다. 계좌 번호에 해당되는 비밀번호가 입력되었을 때 수수료를

		제외하고 입금되는지 확인하고 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다. 입금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.
ATM_STC_004	Withdraw	출금할 금액을 잘 입력받을 수 있는지 확인한다. 수수료를 포함하여 출금할 계좌의 잔액이 충분하지 않을 경우 출금 진행이 멈추는지 확인한다. 계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 제외하고 출금되는지 확인하고 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다. 출금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.
ATM_STC_005	Transfer	송금할 금액을 잘 입력받을 수 있는지 확인한다. 돈을 보낼 계좌를 입력하고 유효하지 않은 계좌를 입력했을 때 경고를 받고 다시 입력받는지 확인한다. 수수료를 포함하여 송금할 계좌의 잔액이 충분하지 않을 경우 송금 진행이 멈추는지 확인한다. 계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 제외하고 송금되는지 확인하고 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다. 송금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.
ATM_STC_006	CheckTransaction History	입력한 계좌의 거래내역이 화면에 정상적으로 출력되는지 확인한다.

ATM_STC_007	CheckCriminal History	범죄이력을 조회하고 싶은 계좌를 입력하고 유효하지 않은 계좌를 입력했을 때 경고를 받고 다시 입력받는지 확인한다. 입력한 계좌의 범죄이력내역이 화면에 정상적으로 출력되는지 확인한다.
Identifier	Non-Functional Requirements	Description
ATM_STC_008	Safe Transaction	범죄 이력이 있는 계좌의 조회가 가능한지 확인한다.
ATM_STC_009	Performance	모든 버튼 입력 및 이벤트에 대하여 1 초 이내로 반응하는지 확인한다.
ATM_STC_010	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(₩)