

JackPot ATM

OOPT Stage 1000

<Plan and Elaboration>

Team 5

Date

2018-03-28

201312259 백만일

201112052 방민석

201211383 조영래

Activity 1001. Define Draft Plan

Activity 1002. Create Preliminary Investigation Report

Activity 1003. Define Requirements

Activity 1004. Record Terms in Glossary

Activity 1005. Implement Prototype

Activity 1006. Define Draft System Architecture

Activity 1007. Define Business Use Case

Activity 1008. Define Business Concept Model

Activity 1009. Define System Test Case

Activity 1010. Refine Plan

Activity 1001. Define Draft Plan

1. Motivation

경제가 발전함에 따라 많은 사람들이 금융 거래를 많이 하게 되었고, 모든 금융 거래를 은행의 은행원이 모두 처리 할 수 없기에 은행원의 일부 업무를 대신 할 수 있는 시스템을 개발하게 되었다. 하지만 금융시장에서 ATM 기기는 현재 포화상태이며 특별한 기능을 하지 않는 ATM 대신 전자송금 어플리케이션 등을 통한 거래가 증가 추세에 있다. ATM기기를 통한 수수료 수입은 기업에게 있어서도 주요 수익원 중 하나이기 때문에 이미 막대한 ATM 설치에 막대한 설비투자를 마친 상황에서 고객들의 ATM 기기 사용 감소는 기업에게 위협의 요인이 될 수 있다. 이에 따라 팀 3/4 은 사람들이 직접 와서 거래를 하고싶도록 만들기 위해 특별한 ATM 기기를 제작하고자 하였으며 게이미피케이션의 원리를 ATM 에 도입하여 본 ATM 거래자 중 1일 1명 랜덤으로 5만원을 추가 지급하는 Jackpot ATM 을 제작하게 되었다.

2. Project Objective

본 프로젝트의 목표는 기존 포화상태인 ATM 기기 시장에서 새로운 형태의 독특한 ATM 기기를 도입함으로써 해당 ATM 소프트웨어를 사용하는 금융기업의 이윤을 극대화시키는 것에

있다. Jackpot ATM 은 기존 ATM 이 가지고 있는 '신속하고 간편한 금융거래' 뿐만아니라 1일 1 Jackpot 기능을 현실성 있게 구현하여 ATM 기기로서의 차별성을 확보하고자 한다.

- 신속 정확한 금융 서비스
- 차별성을 가지는 Jackpot 기능 구현
- Jackpot 기능의 악용 가능성(ex, 한사람이 만원씩 지속적으로 출금하는 등)을 제거

3. Functional Requirements

- 조회,인출,송금,입금 메뉴 선택
- 카드번호 혹은 계좌번호(통장) 입력
- 계좌 비밀번호 입력
- 인출/입금/송금 금액 입력
- 명세서 출력 기능 제공
- 수수료 계산
- ATM 잔고 확인 및 추가
- Jackpot 계산

4. Non Functional Requirements

- 사용자가 보기 편한 화면을 제공해야 한다.
- 금융 거래 프로그램이므로 보안이 잘 되어야 한다.
- 신속한 금융 거래가 가능해야 한다.

5. Resource Estimation

- Human Efforts (M/M) : 3M / 3M
- Human Resource : 1 컴퓨터 공학 / 2 다전공
- Project Duration : 3개월
- Cost : 식비 (5,000) * 주당 미팅 횟수(2) * 학업주(12) * 전체인원(3) =360,000

6. Other Information

- None

Activity 1002. Create Preliminary Investigation Report

1. Alternative Solutions

- 이미 존재하는 ATM S/W 구매한다.
- 개발 전문 업체에 프로그램 제작을 요청한다.

2. Project Justification(Business Demands)

- Cost : 5000(인당 식비) * 2(주당 미팅 횟수) * 3(전체인원) * 12주 = 360,000원
- Duration : 3 months
- Risk : Plan management , OOPT skill, Java skill, UML skill, 시험 및 타 과목 공부
- Effect :

3. Risk Management

Risk	Probability	Significance	Weight
Plan management	5	5	25
OOPT skill 부족	4	5	20
Java sill 부족	3	4	12
UML skill 부족	3	3	9
시험 및 타 과목 공부	5	5	25

4. Risk Reduction Plan

Risk	Reduction plan
Plan Management	관련 서적이거나 검색을 통해 도움을 받는다.
OOPT skill 부족	수업자료와 사이트를 참고해서 보완한다.
Java sill 부족	관련 서적이거나 검색을 통해 도움을 받는다.
UML skill 부족	수업자료와 관련사이트, 질문을 통해 보완한다.
시험 및 타 과목 공부	서로 도와가며 시험에 대비하고, 타 과목 공부로 프로젝트에 영향을 미치는 일이 없도록 한다.

5. Market Analysis

- 현재 각 은행별 ATM은 물론 편의점에도 설치되어 있는 상황이다.
- 핸드폰을 이용해서 간단히 일을 처리할 수 있는 Application들로 인해 이용률이 낮아지고 있다.

6. Other Managerial issues

- 2018년 6월전까지 제작이 완료되어야 한다.

Activity 1003. Define Requirements

1. Functional Requirements

1) 조회

- 조회버튼을 터치
- 카드 혹은 통장을 넣으라는 메시지를 출력
- 카드 혹은 통장을 입력받으면 비밀번호를 입력하라는 메시지를 출력
- 비밀번호를 입력 받으면 현재 잔고와 거래내역을 보여줌

2) 인출

- 인출 버튼 터치
- 카드 혹은 통장을 넣으라는 메시지를 출력
- 카드 혹은 통장을 넣으면 인출 금액을 입력하라는 메시지를 출력
- 인출금액을 입력받으면 수수료를 더한 금액을 통장잔고와 비교
- 잔고보다 많은 금액이면 오류
- 인출금액을 ATM 잔고와 비교
- 잔고보다 많은 금액이면 오류
- 통장 잔고와 ATM 잔고보다 적은금액이면 인출을 위한 비밀번호 입력 메시지를 보여줌
- 비밀번호가 틀리면 오류
- 비밀번호가 맞으면 명세서 출력 확인 맞으면 출력 아니면 생략
- 이후 인출금액과, 인출 후 잔고를 보여줌
- 인출금액을 ATM 잔고에서 빼줌

3) 입금

- 입금 버튼을 터치
- 카드 혹은 통장을 넣으라는 메시지를 출력
- 카드 혹은 통장을 넣으면 입금 금액을 입력하라는 메시지를 출력
- 입금금액과 잔고금액을 더함
- 이후 입금금액과 입금 후 잔고를 보여줌
- 입금 금액을 ATM 잔고에 더해줌
- 명세서 출력 확인 맞으면 출력 아니면 생략

4) 송금

- 송금 버튼을 누르면 카드 혹은 통장을 넣으라는 메시지를 보여줌

- 카드 혹은 통장을 넣으면 계좌번호를 입력하도록 함
- 계좌번호가 없으면 오류
- 계좌번호가 맞으면 송금 금액을 입력하라는 메시지를 보여줌
- 송금 금액을 입력받으면 수수료를 더해서 계좌 잔고와 비교
- 잔고가 부족하면 에러 출력
- 잔고가 충분하면 비밀번호 입력
- 비밀번호가 다르면 오류
- 비밀번호가 같으면 카드 잔고에서 송금금액만큼을 빼줌
- 송금금액 만큼을 입력 계좌에 더해줌
- 성공한 거래의 송금금액/입금계좌/이름을 보여줌
- 명세서 출력 확인 맞으면 출력 아니면 생략

5) Jackpot

- 랜덤으로 N번째 출금을 한 고객에게 5만원을 추가적으로 지급한다

6) Other

- ATM 잔고를 추가하거나 뺄 수 있다.
- ATM 잔고가 일정수준(50만원) 이하가 되면 잔고부족 메시지를 출력한다

Function	Description
withdraw	출금 프로세스로 진입한다
deposit	입금 프로세스로 진입한다
remittance	송금 프로세스로 진입한다
view_account_detail	조회 프로세스로 진입한다
get_account	카드번호 혹은 계좌번호(통장)을 입력받음
check_account	입력받은 계좌번호의 은행과 이름을 확인
account_error	계좌번호 오류를 출력
get_password	계좌 비밀번호를 입력받음
check_password	계좌 비밀번호를 확인
password_error	비밀번호가 틀리면 에러메시지를 출력함
show_transaction_info	계좌 잔고와 거래내역을 출력함

get_amount	인출/입금/송금 금액을 입력받음
compare_balance	인출/송금 금액 + 수수료와 통장 잔고를 비교함
balance_error	통장 잔고가 부족하면 에러 메시지를 출력함
compare_ATM	인출 금액을 ATM 잔고와 비교함
amount_error	ATM 잔고가 부족하면 에러 메시지를 출력함
ask_specification	명세서 출력 여부를 묻음
print_specification	명세서를 출력함
show_transaction_result	입금/출금/송금 금액과 인출 후 잔고를 화면에 표시해줌
sub_ATM	인출 금액을 ATM 잔고에서 빼줌
sub_balance	인출/송금 금액을 통장 잔고에서 빼줌
add_balance	입금/송금 금액을 통장 잔고에 더해줌
add_ATM	입금 금액을 ATM 잔고에 더해줌
fee_calculation	시간에 따른 수수료를 계산해줌
sub_fee	거래에 따른 수수료를 잔고에서 빼줌
manage_ATM	ATM 기기 잔고를 추가하거나 빼줌
ATM_charge_request	ATM 잔고가 일정수준(50만원) 이하가 되면 관리자에게 메시지를 출력한다.
Random_Jackpot	1일 1명 랜덤 N 번째 고객에게 5만원을 추가 출금

2. Performance Requirements

- 1) 모든 입력에 대한 기기의 반응은 1초 이내로 이루어진다.

3. Operating Environments

- 1) OS : Windows 7, Windows 10
- 2) 개발언어: JAVA

4. Develop Environments

- 1) OS: Windows7, Windows 10, Mac
- 2) IDE : Eclipse
- 3) 개발언어: JAVA

5. Interface Requirements

- 1) 입금/출금/송금/조회 버튼
- 2) 숫자 입력 버튼
- 3) 거래 결과 화면

6. Other Requirements

- 1) 공동 문서 편집을 위한 구글 계정

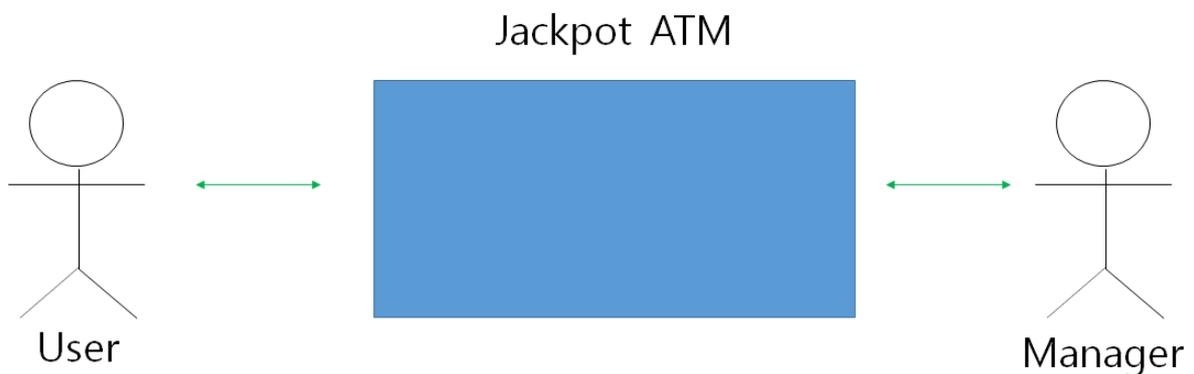
Activity 1004. Record Terms in Glossary

Term	Description	Remarks
withdraw	출금	
deposit	입금	
remittance	송금	
view_account detail	조회	
get	입력	
check	확인	
account	계좌	
password	비밀번호	
error	에러	
show	화면 출력	
transaction	거래	
info	정보	

amount	금액	
compare	비교	
balance	잔액	
ATM	ATM 기기	
ask	요청	
specification	영수증	
print	하드카피 출력	
result	결과	
sub	뺄셈	
add	더하기	
fee	수수료	
calculation	계산	
manage	관리	

Activity 1006. Define Draft System Architecture

1. Define System Boundary

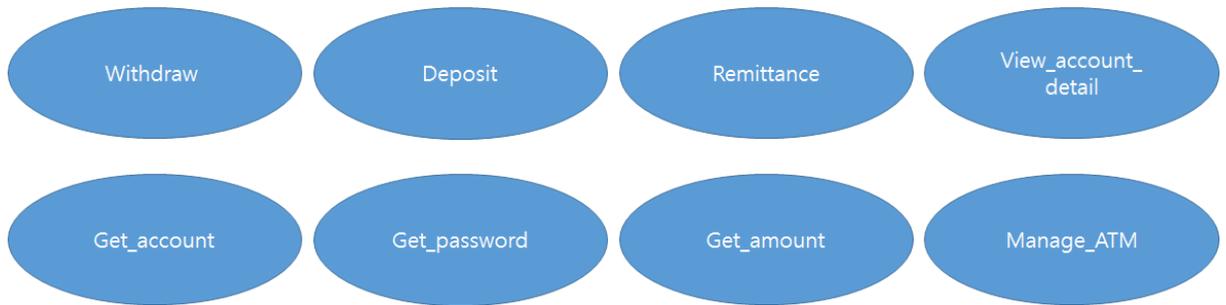


2. Identify and Describe Actors

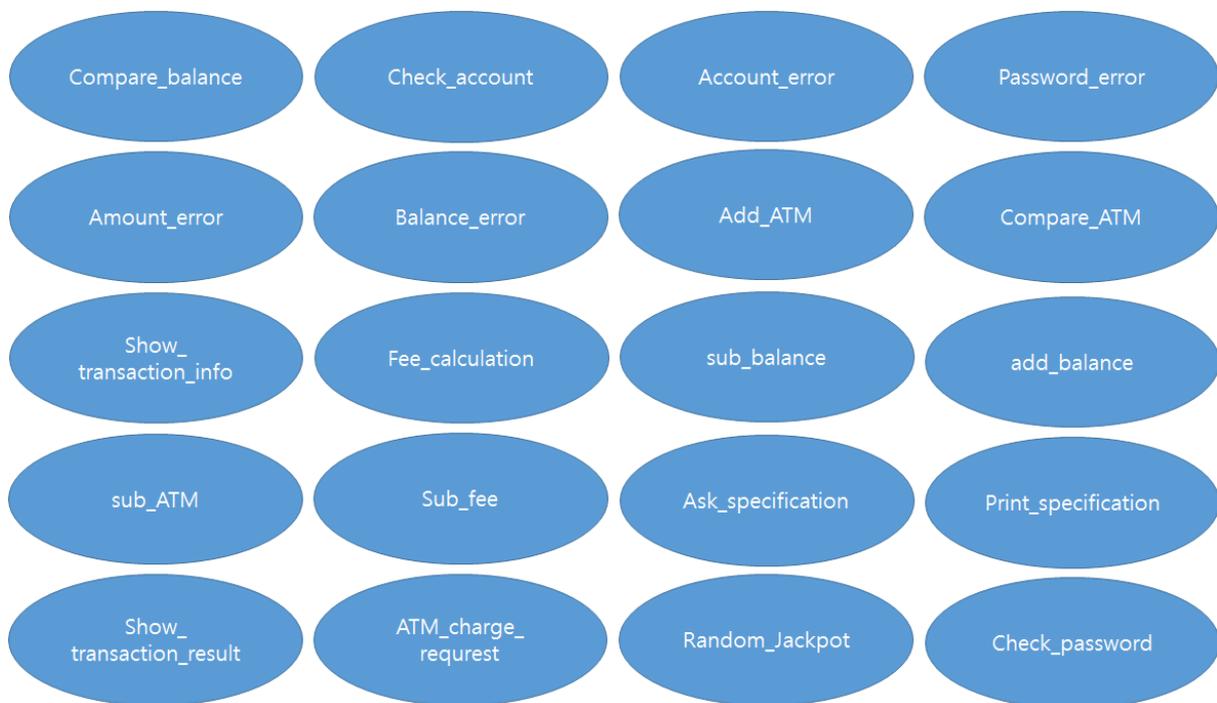
- a. User : ATM을 이용하는 사람
- b. Manager : ATM의 잔고를 확인하고 관리하는 사람

3. Identify Use-Case

a. Use-Cases by Actor-Based



b. Use-Cases by Event-Based



4. Allocate System Functions into Related Use-Cases

Ref.	Function	Use-Case Number & Name	Remarks
R.1.1	withdraw	1.withdraw	

R.1.2	deposit	2.deposit	
R.1.3	remittance	3.remittance	
R.1.4	view_account_detail	4.view_account_detail	
R.2.1.1	get_account	5.get_account	
R.2.1.2	check_account	6.check_account	
R.2.1.3	account_error	7.account_error	
R.2.2.1	get_password	8.get_password	
R.2.2.2	check_password	9.check_password	
R.2.2.3	password_error	10.password_error	
R.3	show_transaction_info	11.show_transaction_info	
R.4.1	get_amount	12.get_amount	
R.4.2.1	compare_balance	13.compare_balance	
R.4.2.2	balance_error	14.balance_error	
R.5.1	compare_ATM	15.compare_ATM	
R.5.2	amount_error	16.amount_error	
R.6.1	ask_specification	17.ask_specification	
R.6.2	print_specification	18.print_specification	
R.7	show_transaction_result	19.show_transaction_result	
R.8	sub_ATM	20.sub_ATM	
R.9	sub_balance	21.sub_balance	
R.10	add_balance	22.add_balance	
R.11	add_ATM	23.add_ATM	
R.12.1	fee_calculation	24.fee_calculation	
R.12.2	sub_fee	25.sub_fee	

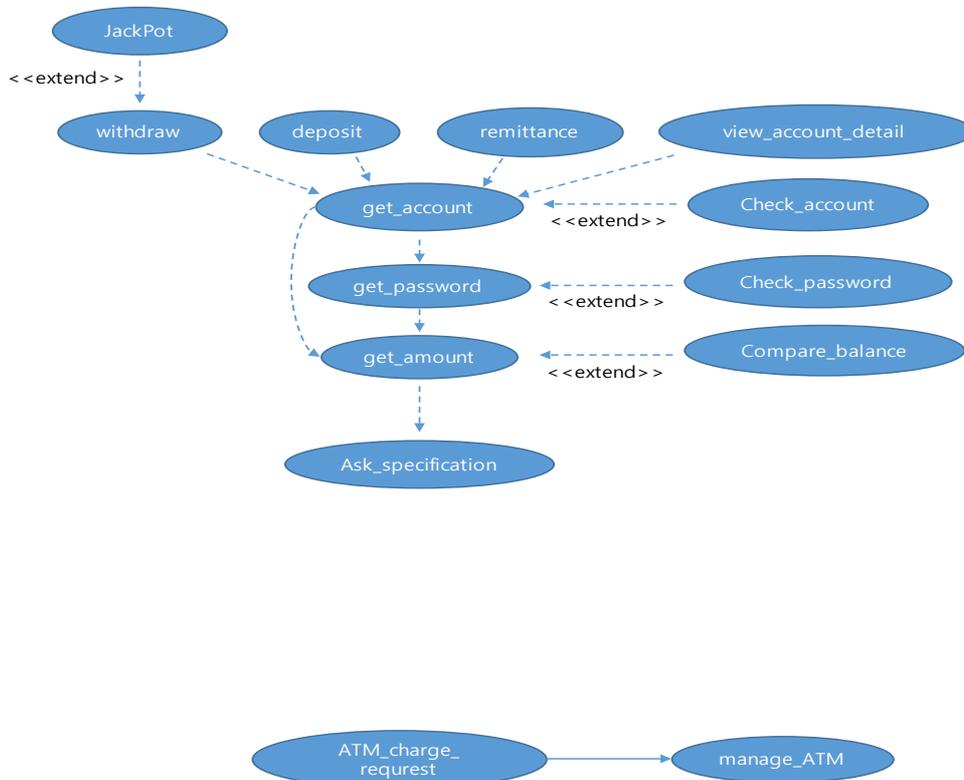
R.13	manage_ATM	26.manage_ATM	
R.14	ATM_charge_request	27.ATM_charge_request	
R.15	Random_Jackpot	28.Random_Jackpot	

5. Categorize Use-Cases

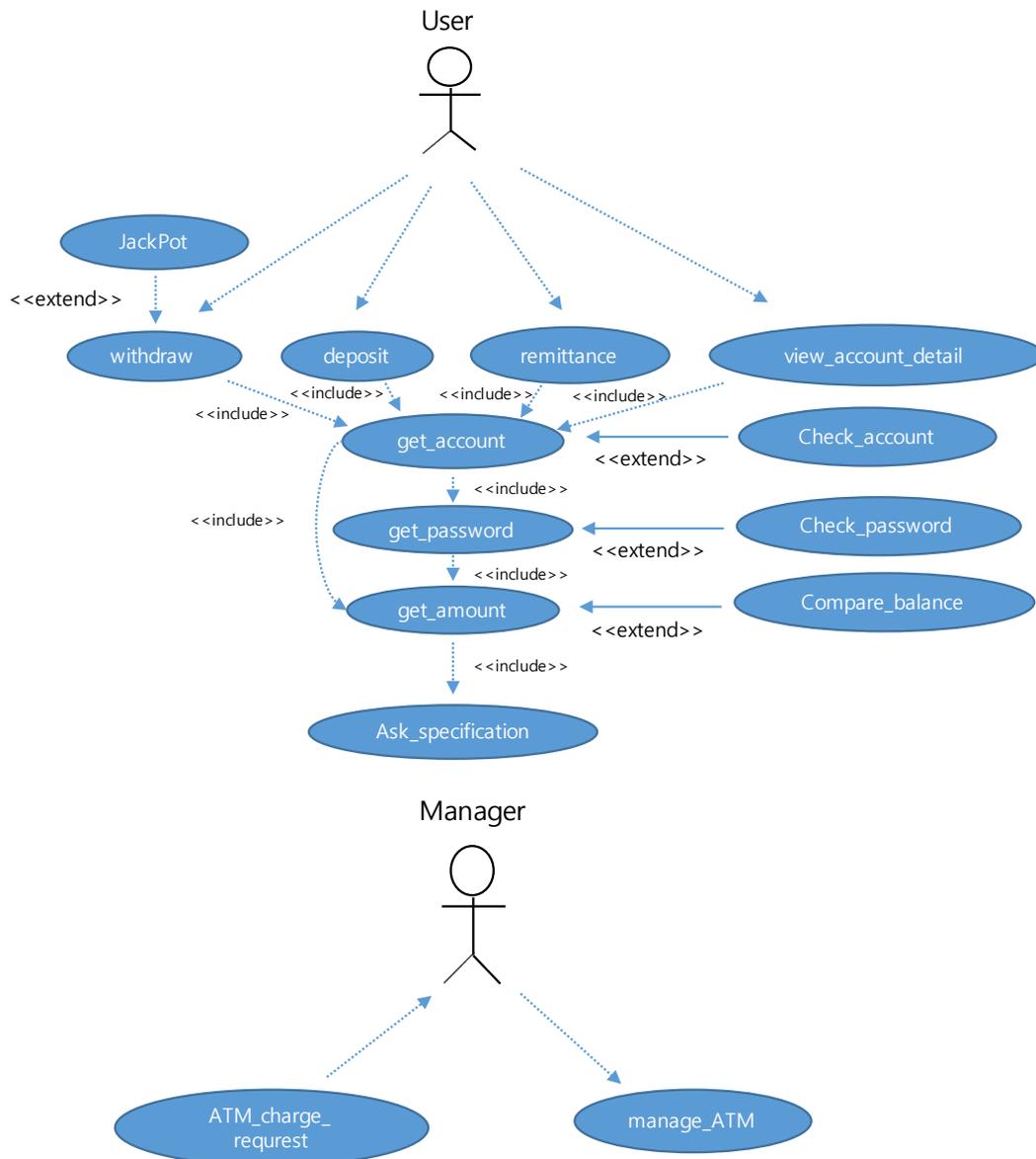
Ref.	Function	Use-Case Number & Name	Category
R.1.1	withdraw	1.withdraw	Primary
R.1.2	deposit	2.deposit	Primary
R.1.3	remittance	3.remittance	Primary
R.1.4	view_account_detail	4.view_account_detail	Primary
R.2.1.1	get_account	5.get_account	Primary
R.2.1.2	check_account	6.check_account	Primary
R.2.1.3	account_error	7.account_error	Primary
R.2.2.1	get_password	8.get_password	Primary
R.2.2.2	check_password	9.check_password	Primary
R.2.2.3	password_error	10.password_error	Primary
R.3	show_transaction_info	11.show_transaction_info	Primary
R.4.1	get_amount	12.get_amount	Primary
R.4.2.1	compare_balance	13.compare_balance	Primary
R.4.2.2	balance_error	14.balance_error	Primary
R.5.1	compare_ATM	15.compare_ATM	Primary
R.5.2	amount_error	16.amount_error	Primary

R.6.1	ask_specification	17.ask_specification	Primary
R.6.2	print_specification	18.print_specification	Primary
R.7	show_transaction_result	19.show_transaction_result	Primary
R.8	sub_ATM	20.sub_ATM	Primary
R.9	sub_balance	21.sub_balance	Primary
R.10	add_balance	22.add_balance	Primary
R.11	add_ATM	23.add_ATM	Primary
R.12.1	fee_calculation	24.fee_calculation	Primary
R.12.2	sub_fee	25.sub_fee	Primary
R.13	manage_ATM	26.manage_ATM	Primary
R.14	ATM_charge_request	27.ATM_charge_request	Primary
R.15	Random_Jackpot	28.Random_Jackpot	Primary

7. Identify Relationships between Use-Cases



8. Draw a Use-Case Diagram



9. Describe Use-Cases

Use-Case Name	Description
1.withdraw	출금 프로세스로 진입한다.
Actor	

User	
------	--

Use-Case Name	Description
2.deposit	입금 프로세스로 진입한다.
Actor	
User	

Use-Case Name	Description
3.remittance	송금 프로세스로 진입한다
Actor	
User	

Use-Case Name	Description
4.view_account_detail	조회 프로세스로 진입한다.
Actor	
User	

Use-Case Name	Description
---------------	-------------

5.get_account	카드번호 혹은 계좌번호(통장)을 입력받는다.
Actor	
User	

Use-Case Name	Description
6.check_account	입력받은 계좌번호의 은행과 이름을 확인한다.
Actor	
None	

Use-Case Name	Description
7.account_error	계좌번호 오류를 출력한다.
Actor	
None	

Use-Case Name	Description
8.get_password	계좌 비밀번호를 입력받는다.
Actor	

User	
------	--

Use-Case Name	Description
9.check_password	입력받은 비밀번호를 확인한다.

Use-Case Name	Description
10.password_error	비밀번호가 틀리면 에러메시지를 출력한다.
Actor	
None	

Use-Case Name	Description
11.show_transaction_info	계좌 잔고와 거래내역을 출력한다.
Actor	
None	

Use-Case Name	Description
12.get_amount	인출/입금/송금 금액을 입력받는다.

Actor	
User	

Use-Case Name	Description
13.compare_balance	인출/송금 금액 + 수수료와 통장 잔고를 비교한다.
Actor	
None	

Use-Case Name	Description
14.balance_error	통장 잔고가 부족하면 에러 메시지를 출력한다.
Actor	
None	

Use-Case Name	Description
15.compare_ATM	인출 금액을 ATM 잔고와 비교한다.
Actor	
None	

Use-Case Name	Description
16.amount_error	ATM 잔고가 부족하면 에러 메시지를 출력한다.
Actor	
None	

Use-Case Name	Description
17.ask_specification	명세서 출력 여부를 물어본다.
Actor	
None	

Use-Case Name	Description
18.print_specification	명세서를 출력한다.
Actor	
None	

Use-Case Name	Description
19.show_transaction_result	입금/출금/송금 금액과 인출 후 잔고를 화면에 표시해준다.
Actor	
None	

Use-Case Name	Description
20.sub_ATM	인출 금액을 ATM 잔고에서 빼준다.
Actor	
None	

Use-Case Name	Description
21.sub_balance	인출/송금 금액을 통장 잔고에서 빼준다.
Actor	
None	

Use-Case Name	Description
22.add_balance	입금/송금 금액을 통장 잔고에 더해준다.
Actor	
None	

Use-Case Name	Description
23.add_ATM	입금 금액을 ATM 잔고에 더해준다.
Actor	
None	

Use-Case Name	Description
24.fee_calculation	시간에 따른 수수료를 계산해준다.
Actor	
None	

Use-Case Name	Description
25.sub_fee	거래에 따른 수수료를 잔고에서 빼준다.
Actor	
None	

Use-Case Name	Description
26.manage_ATM	ATM 기기 잔고를 추가하거나 빼준다.
Actor	
manager	

Use-Case Name	Description
27.ATM_charge_request	ATM 잔고가 일정수준(50만원) 이하가 되면 관리자에게 메시지를 출력한다.
Actor	

None	
------	--

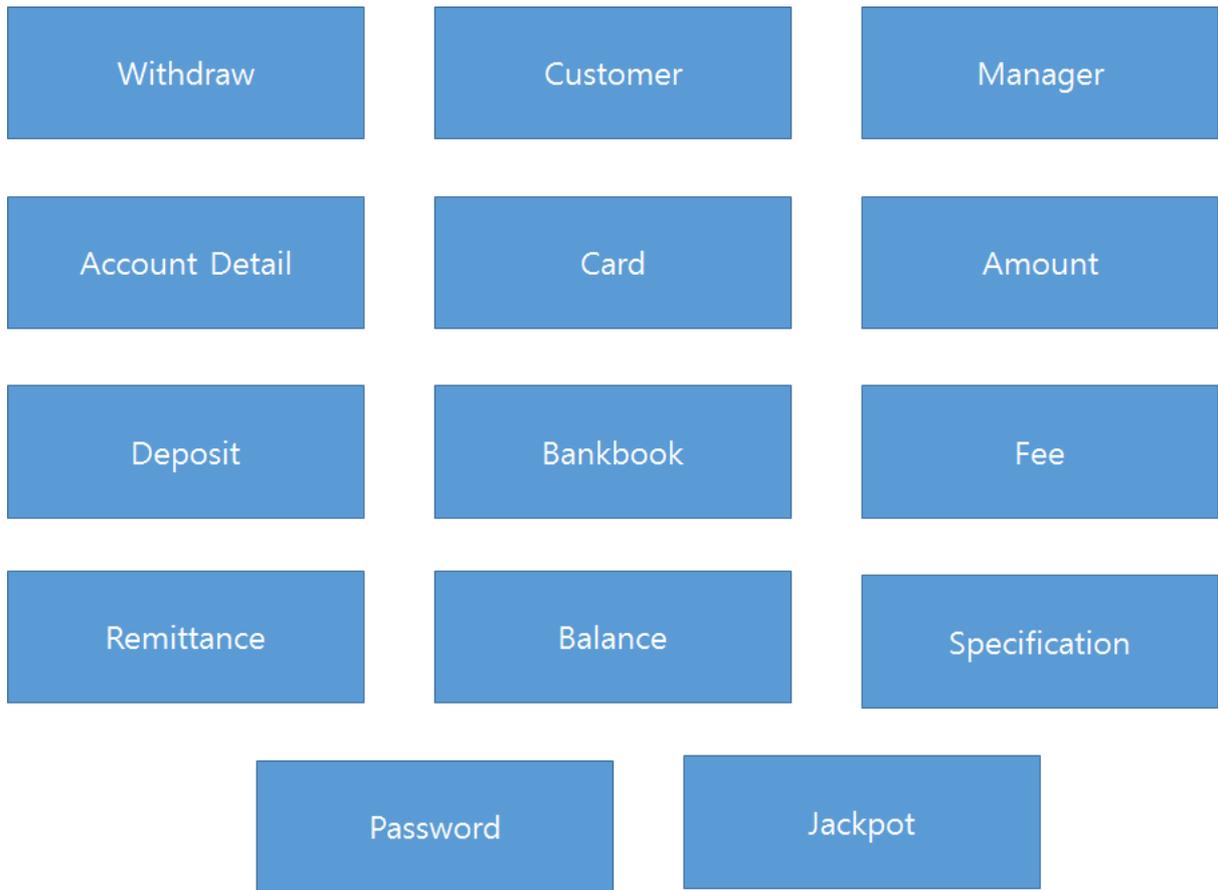
Use-Case Name	Description
28.Random_Jackpot	1일 1명 랜덤 N 번째 고객에게 5만원을 추가 출금한다.
Actor	
None	

10. Rank Use-Cases

Use-Case Name	Rank
withdraw	High
deposit	High
remittance	High
view_account detail	High
get_password	High
check_password	High
password_error	High
show_transaction_info	High
get_amount	High
compare_balance	High
balance_error	High

compare_ATM	High
amount_error	High
ask_specification	Medium
print_specification	Medium
show_transaction_result	High
sub_ATM	High
sub_balance	High
add_balance	High
add_ATM	High
fee_calculation	High
sub_fee	High
manage_ATM	High
ATM_charge_request	Medium
Random_Jackpot	High

Activity 1007. Define Business Use Case



Activity 1008. Define Business Concept Model



Activity 1009. Define System Test Case

Test Number	Test 항목	Description	Use Case	System Function
1	출금 프로세스 진입 시험	출금 프로세스로 제대로 진입하는지 시험한다	withdraw	
2	입금 프로세스 진입 시험	입금 프로세스로 제대로 진입하는지 시험한다	deposit	
3	송금 프로세스 진입 시험	송금 프로세스로 제대로 진입하는지 시험한다	remittance	
4	조회 프로세스 진입 시험	조회 프로세스로 제대로 진입하는지 시험한다	view_account_detail	
5	계좌 입력 시험	계좌번호를 입력하였을때 그 값을 정확히 받아오는지 시험한다.	get_account	
6	계좌 확인 시험	계좌번호에 따른 이름과 은행을 받아오는지 시험	check_account	
7	계좌 에러 시험	잘못된 계좌를 입력하였을 때 에러메시지를 출력하는지 시험한다.	account_error	
8	패스워드 받기 시험	패스워드를 제대로 입력받는지 확인한다	get_password	
9	패스워드 확인 시험	맞는 패스워드를 입력받았는지 확인한다	check_password	
10	패드워드 오류 시험	잘못된 패스워드를 입력하였을 때 오류를 출력하는지 시험	password_error	
11	거래 내역 출력 시험	거래 내역을 제대로 출력하는지 시험한다	show_transaction_info	
12	금액 입력 시험	사용자로부터 입력된 금액을 제대로 받아오는지 시험한다	get_amount	

13	잔고비교 시험	통장 잔고와 입력된 출금금액을 제대로 비교하는지 시험한다	compare_balance	
14	잔고 에러 시험	잔고보다 큰 금액을 입력하였을 때, 에러를 제대로 출력하는지 시험한다	balance_error	
15	ATM잔고비교 시험	ATM에 잔고와 입력된 출금 금액을 제대로 비교 하는지 시험한다	compare_ATM	
16	ATM잔고 에러 시험	ATM 잔고보다 큰 금액을 입력하였을 때, 에러를 제대로 출력하는지 시험한다	amount_error	
17	명세서 요청 시험	명세서 요청 화면을 제대로 출력하는지 시험한다	ask_specification	
18	명세서 프린트 시험	거래 후 명세서를 제대로 출력하는지 시험한다	print_specification	
19	거래 결과 출력 시험	입금/출금/송금 후 그 결과를 제대로 출력하는지 시험한다	show_transaction_result	
20	ATM 잔고 감소 시험	인출시 ATM 기기의 잔고를 제대로 감소시키는지 시험한다	sub_ATM	
21	잔고 감소 시험	인출시 계좌의 잔고를 제대로 감소시키는지 시험한다	sub_balance	
22	잔고 증가 시험	입금시 계좌의 잔고를 제대로 증가시키는지 시험한다	add_balance	
23	ATM 잔고 증가 시험	입금시 ATM 잔고를 제대로 증가시키는지 시험한다	add_ATM	
24	수수료 계산 시험	시간에 따른 수수료 계산을 제대로 하는지 시험한다	fee_calculation	
25	수수료 계산	거래에 따른 수수료를	sub_fee	

	시험	계좌에서 제대로 감소시키는지 시험한다		
26	ATM 잔고 증가 시험	ATM 잔고에 입력한 금액만큼을 제대로 증가시키는지 시험한다	manage_ATM	
27	ATM 잔고 충전요청 시험	ATM잔고가 일정수준(50만원) 이하가 되면 충전요청	ATM_charge_request	
28	Jackpot 출금 시험	랜덤 N번째 고객에게 추가 5만원이 출금되는지 확인한다.	Random_Jackpot	

Activity 1010. Refine Plan

1. Project Scope

팀 3/4은 기존 ATM 기기 시장의 포화라는 위협요인에서 게이미피케이션의 요소를 활용한 Jackpot ATM 을 제작하여 차별화된 형태의 ATM 기기 시장의 문을 열고 이를 통하여 고객들이 같은 조건이라면 Jackpot ATM 기기를 이용하도록 하여 사용자를 늘려 궁극적으로는 수수료 수익을 극대화 하는것에 있다.

2. Project Objectives

Jackpot ATM 기기를 사용하는 고객으로 하여금 기본적으로 ATM 기기가 가지고 있는 신속하고 정확한 금융거래뿐만 아니라 추가적인 보상을 제공하여 고객의 이용률을 증대시킴에 있어 오남용을 막을수 있도록 소프트웨어를 구축하고, Jackpot 이라는 보상을 제공할 때에도 그 투명성을 지키기 위해 랜덤하게 당첨자가 도출되도록 한다.

- 사용자가 금융거래를 신속하고 정확하게 할 수 있도록 한다.
- Jackpot 이라는 추가적인 보상을 제공한다.
- Jackpot 의 오남용이 일어나지 않을 수 있도록 제어한다.

3. Functional Requirements

- withdraw
- deposit
- remittance
- view_account_detail
- get_account
- check_account

- get_password
- check_password
- password_error
- show_transaction_info
- get_amount
- compare_balance
- balance_error
- compare_ATM
- amount_error
- ask_specification
- printf_specification
- show_withdraw_result
- sub_ATM
- sub_balance
- add_balance
- add_ATM
- fee_calculation
- sub_fee
- manage_ATM
- ATM_charge_request
- Random_Jackpot

4. Performance Requirements

- 기기조작에 대한 반응은 1초 이내로 이루어 져야한다.
- 금융 거래는 1분 이내로 이루어 져야한다

5. Operating Environment

- OS : Windows 7, Windows 10
- IDE : Eclipse
- 개발언어: JAVA

6. User Interface Requirements

- 메뉴 선택을 통한 접근방식
- 터치스크린에 기반을 둔 개발

7. Resource Estimation

- Human Efforts (M/M) : 3M / 3M

- Human Resource : 1 컴퓨터공학 / 2 다전공
- Project Duration : 3개월
- Cost : 식비 (5,000) * 주당 미팅 횟수(2) * 학업주(12) * 전체인원(3) =360,000

8. Scheduling

Stage	Phase(00X0)/Activity(000X)	Schedule(Week)											
		1	2	3	4	5	6	7	8	9	10	11	12
1000. Plan & Elaboration	1001. Define Draft Plan	█											
	1002. Create Preliminary Investigation	█											
	1003. Define Requirements	█											
	1004. Record Terms in Glossary	█											
	1005. Implement Prototype	█	█										
	1006. Define Business Use Case		█										
	1007. Define Business Concept Model		█										
	1008. 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 Cases				█								
	2032. Refine Use Case Diagrams				█								
	2033. Define Domain Model				█								
	2034. Refine Glossary				█								

