

JackPot ATM

OOPT Stage 2030

<Analyze>

Team 5

Date

2018-04-16

201312259 백만일

201112052 방민석

201211383 조영래

Contents

Activity2010. Revise Plan Activity

Activity2020. Synchronize Artifacts

Activity2031. Define Essential Use Cases

Activity2032. Refine Use Case Diagrams

Activity2033. Define Domain Model

Activity2034. Refine Glossary

Activity2035. Define System Sequence Diagrams

Activity2036. Define Operation Contracts

Activity2038. Refine System Test Case

Activity2039. Analyze Traceability Analysis

Activity2010. Revise Plan Activity

1) functional requirements는 상세하게 작성되어 있어서 따로 수정하지 않았다.

2) 이전 버전에서는 implementation 의 측면에서 고려하여 function을 너무 세분화 했다는 문제점이 있어서, function을 requirement를 이용해 작성하였다.

Ref.#	Function	Category
R.1.1	jackPot	Hidden
R.1.2	withdraw	Evident
R.2	deposit	Evident
R.3	remittance	Evident
R.4	view_account_detail	Evident
R.5	manage_ATM	Evident

3) 이전 버전에서 너무 세분화하다 보니 각 function의 description 이 간단했는데, function을 큰 단위로 생각하면서 description에 좀 더 자세한 내용이 들어가도록 수정하였다.

Use-Case Name	Description
1.withdraw	<ul style="list-style-type: none"> - 유저가 인출 버튼을 누르면 시작되는 Use Case - 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
Actor	
User	

	<ul style="list-style-type: none"> - 통장 잔고와 ATM 잔고보다 적은금액 이면 인출을 위한 비밀번호 입력 메시지 출력 - 비밀번호가 틀리면 오류 - 비밀번호가 맞으면 명세서 출력 확인 맞으면 출력 아니면 생략 - 이후 인출금액과, 인출 후 잔고를 보여줌 - 인출금액을 ATM 잔고에서 빼줌
--	---

Use-Case Name	Description
2.deposit	<ul style="list-style-type: none"> - 유저가 입금 버튼을 터치하면 시작되는 Use Case - 카드 혹은 통장 넣으라는 메시지 출력 - 카드 혹은 통장 넣으면 입금할 금액 입력하라는 메시지 출력 - 입금금액과 잔고금액 더함 - 이후 입금금액과 입금 후 잔고 보여줌 - 입금 금액 ATM 잔고에 더함 - 명세서 출력 확인 맞으면 출력 아니면 생략
Actor	
User	

Use-Case Name	Description
3.remittance	<ul style="list-style-type: none"> - 유저가 송금 버튼을 누르면 시작하는 Use Case - 카드 혹은 통장을 넣으라는 메시지를 보여줌 - 카드나 통장을 넣으면 계좌번호를 입력하도록 함 - 계좌번호가 없으면 오류 - 계좌번호가 맞으면 송금 금액을 입력하라는 메시지를 보여줌
Actor	
User	

	<ul style="list-style-type: none"> - 송금 금액 입력받으면 수수료를 더해서 계좌 잔고와 비교 - 잔고가 부족하면 에러 출력 - 잔고가 충분하면 비밀번호 입력 - 비밀번호가 다르면 오류 - 비밀번호가 같으면 카드 잔고에서 송금액만큼을 빼줌 - 송금액 만큼을 입력 계좌에 더해줌 - 성공한 거래의 송금액/입금계좌/이름을 보여줌 - 명세서 출력 확인 맞으면 출력 아니면 생략
--	--

Use-Case Name	Description
4.view_account_detail	<ul style="list-style-type: none"> - 유저가 조회버튼을 터치하면 시작되는 Use Case - 카드 혹은 통장을 넣으라는 메시지를 출력 - 카드 혹은 통장을 입력받으면 비밀번호를 입력하라는 메시지를 출력 - 비밀번호를 입력 받으면 현재 잔고와 거래내역을 보여줌
Actor	
User	

Use-Case Name	Description
5.jackPot	<ul style="list-style-type: none"> - 랜덤으로 N번째 출금을 한 고객에게 5만원을 추가적으로 지급하는 Use case
Actor	
User	

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

6.manage_ATM	<ul style="list-style-type: none"> - 관리자가 ATM 잔고에 문제가 있음을 알았을때 관리하기 위한 Use case - ATM 잔고를 추가하거나 뺄 수 있다.
Actor	
None	

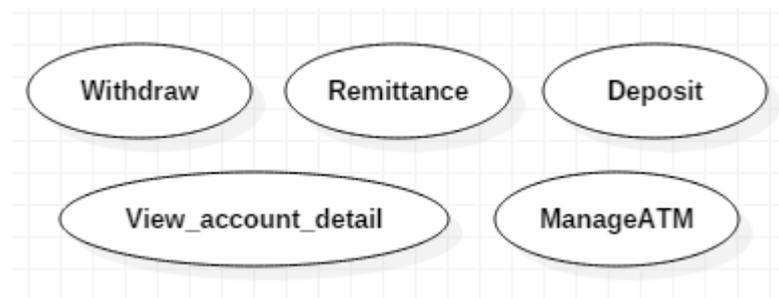
4) Record terms in Glossary 수정

Term	Description	Remarks
withdraw	이용자가 통장에서 돈을 빼기 위한 작업	
deposit	이용자가 통장에 돈을 넣기 위한 작업	
remittance	이용자가 다른 계좌에 돈을 보내기 위한 작업	
account	계좌	
manage	관리	
jackPot	임의의 이용자에게 5만원을 추가로 지급하는 기능	

5) Use case / Use case Diagram 수정

- Use case

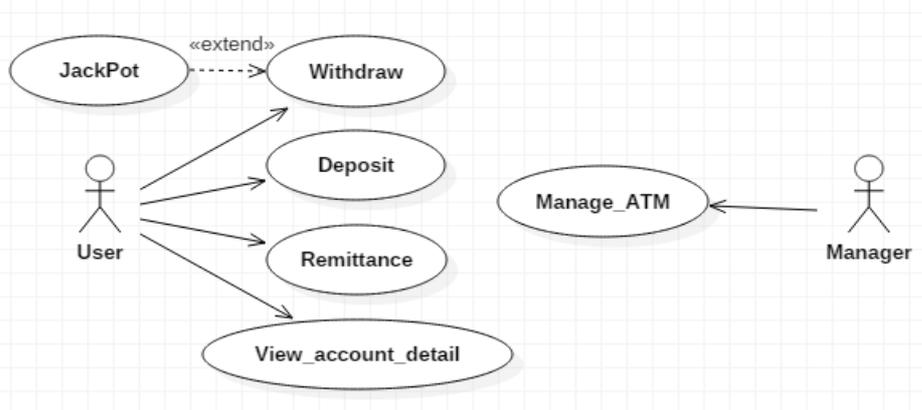
1. Actor-based



2. Event-based



- Use case Diagram



6) Allocate System Functions into Related Use-Cases 수정

Ref.	Function	Use-Case Number & Name	Remarks
R.1.1	jackPot	1.jackPot	
R.1.2	withdraw	2.withdraw	
R.2	deposit	3.deposit	
R.3	remittance	4.remittance	
R.4	view_account_detail	5.view_account_detail	
R.5	manage_ATM	6.manage_ATM	

7) Categorize Use-Cases 수정

Ref.	Function	Use-Case Number & Name	Category
R.1.1	jackPot	1.jackPot	Primary
R.1.2	withdraw	2.withdraw	Primary
R.2	deposit	3.deposit	Primary
R.3	remittance	4.remittance	Primary
R.4	view_account_detail	5.view_account_detail	Primary
R.5	manage_ATM	6.manage_ATM	Primary

8) Identify Relationships between Use-Cases 수정



Activity2020. Synchronize Artifacts

- 1) 기존 Jackpot system 의 Stage1000 ver 1.0 에서 Stage1000 ver 2.0으로 수정하였다.

Activity2031. Define Essential Use Cases

Use Case	withdraw
Actor	User
Purpose	사용자 계좌에 있는 돈을 출금한다.
Overview	사용자가 ATM기의 출금 버튼을 누르고 출금할 계좌를 입력한다. 계좌가 확인되면 금액을 입력받아 잔고보다 적을 경우 출금을 진행한다. 그 후 사용자가 돈을 갖고 떠난다.
Type	Primary and Essential
Cross Reference	System Function R.1.2
Pre-Requisites	ATM에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 출금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User는 계좌를 입력한다. 4.(S) 유효한 계좌인지 확인한다. 5.(S) 출금 화면을 출력한다. 6.(A) 출금 금액을 입력한다. 7.(S) 비밀번호 입력 화면을 출력한다. 8.(A) 비밀번호를 입력한다. 9.(S) 비밀번호가 일치하는지 확인한다. 10.(S) User의 잔고와 출금금액을 비교한다. 11.(S) User의 잔고를 출금금액만큼 감소시킨다. 12.(S) ATM 잔액을 감소시킨다.
Alternative Courses of Events	N/A

Exceptional Courses of Events	E1: 잘못된 계좌를 입력받으면 에러 메시지를 출력한다 E2: 비밀번호가 틀릴 경우 에러 메시지를 출력한다 E3: 계좌 잔고가 출금액보다 적을 경우 에러 메시지를 출력한다
Use Case	deposit
Actor	User
Purpose	사용자 계좌에 돈을 입금한다.
Overview	사용자가 ATM기의 입금 버튼을 누르고 입금할 계좌를 입력한다. 사용자가 입금할 금액을 기기에 넣으면 ATM이 금액을 세고, 그 액수만큼 입력된 계좌의 잔고를 증가시킨다.
Type	Primary and Essential
Cross Reference	System Function R.2
Pre-Requisites	ATM에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 입금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User는 계좌를 입력한다. 4.(S) 계좌 일치하는지 확인한다. 5.(S) 금액 입력 화면을 출력한다. 6.(A) 입금 금액을 입력한다. 7.(S) 계좌 잔액을 증가시킨다. 8.(S) ATM 잔액을 증가시킨다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1: 잘못된 계좌를 입력받으면 에러 메시지를 출력한다

Use Case	remittance
Actor	User
Purpose	사용자 계좌의 돈을 다른 사용자 계좌로 송금한다.
Overview	사용자가 ATM기의 송금 버튼을 누르고 자신의 계좌를 입력한다. 계좌가 확인되면 송금할 계좌와 금액을

	입력받는다. 송금 계좌가 맞고 통장 잔고가 송금액보다 많으면 송금 계좌의 잔고가 액수만큼 증가한다.
Type	Primary and Essential
Cross Reference	System Function R.3
Pre-Requisites	ATM에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 송금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User는 계좌를 입력한다. 4.(S) 계좌가 유효한지 확인한다. 5.(S) 송금받을 계좌 입력 화면을 출력한다. 6.(A) User는 송금받을 계좌를 입력한다. 7.(S) 계좌가 유효한지 확인한다. 8.(A) User는 송금 금액을 입력한다. 9.(A) User는 비밀번호를 입력한다. 10.(S) 비밀번호가 일치하는지 확인한다. 11.(S) User의 잔고와 송금금액을 비교한다. 12.(S) 받는사람의 통장잔고를 증가시킨다. 13.(S) 보내는 사람의 통장잔고를 감소시킨다. 14.(S) ATM 잔고를 증가시킨다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1: 잘못된 계좌를 입력받으면 에러 메시지를 출력한다 E2: 비밀번호가 틀릴 경우 에러 메시지를 출력한다 E3: 송금 계좌가 잘못된 경우 에러 메시지를 출력한다 E4: 계좌 잔고가 송금액보다 적을 경우 에러 메시지를 출력한다

Use Case	view_account_detail
Actor	User
Purpose	사용자가 입력한 계좌의 잔액을 보여준다.
Overview	사용자가 계좌조회 버튼을 누르고 자신의 계좌를 입력한다.

	계좌가 확인되면 입력된 계좌의 잔고를 표시한다.
Type	Primary and Essential
Cross Reference	System Function R.4
Pre-Requisites	ATM에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 계좌 조회 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User는 계좌를 입력한다 4.(S) 유효한 계좌인지 확인한다. 5.(S) 비밀번호 입력 화면을 출력한다. 6.(A) 비밀번호를 입력한다. 4.(S) 비밀번호가 일치하는지 확인한다. 5.(S) 확인이 되면 사용자 계좌의 잔고를 표시한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1: 잘못된 계좌를 입력받으면 에러 메시지를 출력한다 E2: 비밀번호가 틀릴 경우 에러 메시지를 출력한다

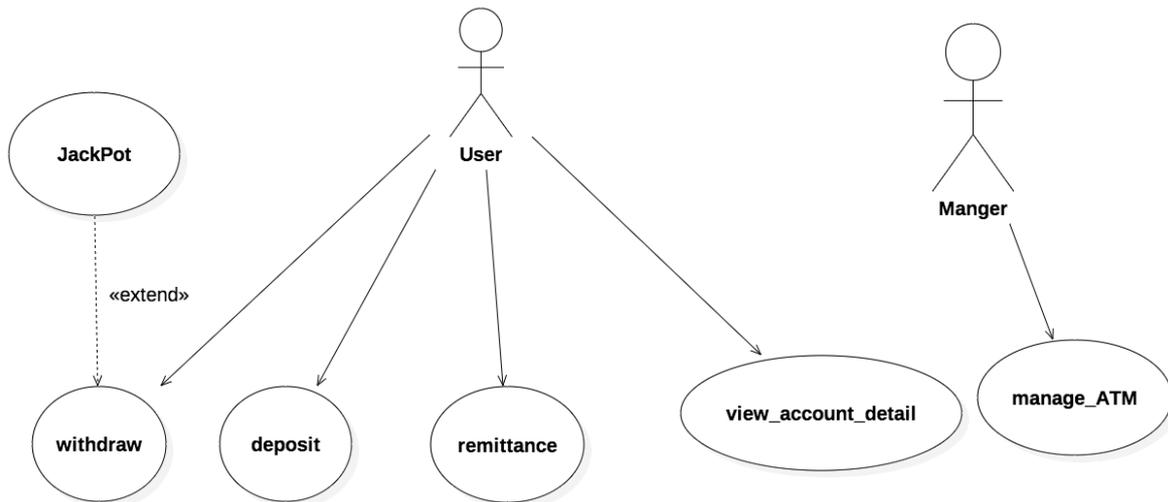
Use Case	Random_Jackpot
Actor	User
Purpose	인출 기능을 이용하는 임의의 사용자에게 추가로 5만원을 지급한다.
Overview	00시가 되면 ATM기의 몇번째 인출 고객에게 추가로 5만원이 인출될지 정해진다. 해당 순서의 사용자가 인출시에 5만원이 추가로 인출된다.
Type	Primary and Essential
Cross Reference	System Function R.1.1
Pre-Requisites	N/A
Typical Courses of Events	(A):Actor, (S):System 1.(S) 00시가 되면 당첨되는 순서를 랜덤하게 선택한다. 2.(S) User가 돈을 인출했을때 해당 User가 해당 순서의

	<p>사용자인지 확인한다.</p> <p>3.(S) 해당 User에게 돈을 5만원 더 추가로 인출해준다.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

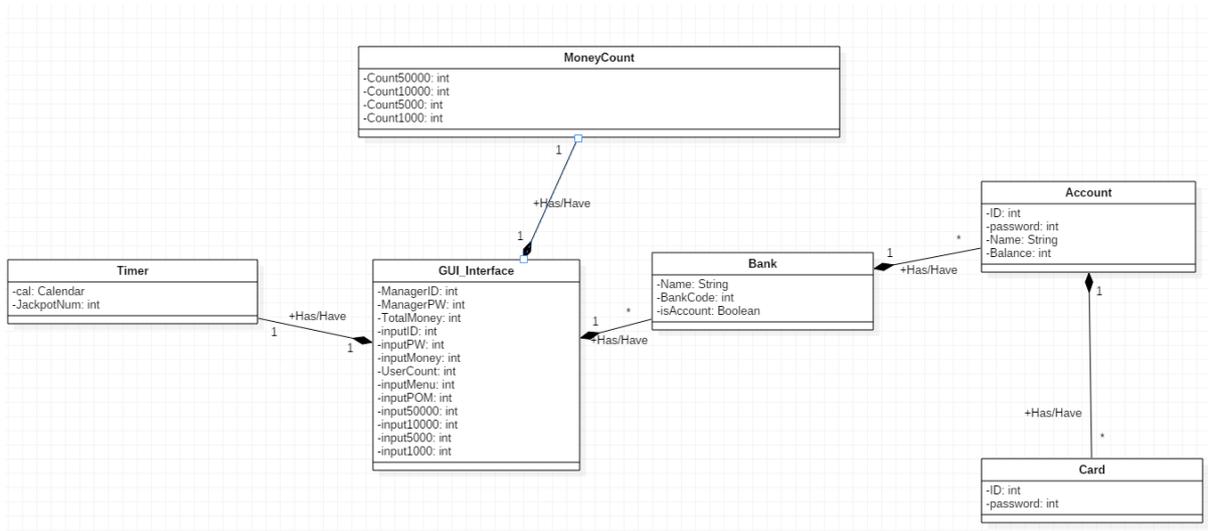
Use Case	manage_ATM
Actor	Manager
Purpose	관리자가 ATM의 잔고를 추가하거나 뺄 수 있다
Overview	ATM의 잔고가 일정 수준 이하로 낮아지면 관리자가 ATM의 잔고를 추가하고 잔고가 일정 수준 이상으로 많아지면 ATM의 잔고를 뺀다
Type	Primary and Essential
Cross Reference	System Function R.5
Pre-Requisites	N/A
Typical Courses of Events	<p>(A):Actor, (S):System</p> <ol style="list-style-type: none"> 1.(A) 관리자가 관리 메뉴 버튼을 누른다. 2.(S) ID와 입력 화면을 출력한다 3.(A) 관리자가 ID를 입력한다 4.(S) 유효한 관리자 아이디인지 확인한다. 5.(A) PW 입력화면을 출력한다. 6.(A) 관리자의 PW를 입력한다 7.(S) 관리자의 PW를 확인한다 8.(S) 잔고를 추가할 것인지 뺄것인지 선택하는 메뉴를 출력한다. 9.(A) 관리자가 잔고를 추가할 것인지 뺄 것인지 메뉴를 선택한다. 10.(S) 금액을 입력하는 메뉴를 출력한다. 11.(A)변화시킬 잔고 금액을 입력한다. 12.(S) ATM에 변화된 잔고를 갱신한다
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 관리자의 ID와 PW가 잘못된 경우 에러 메시지를

	<p>출력한다 E2. 빼낼 잔고가 ATM 잔고보다 많을 경우 에러 메시지를 출력한다</p>
--	--

Activity2032. Refine Use Case Diagrams



Activity2033. Define Domain Model

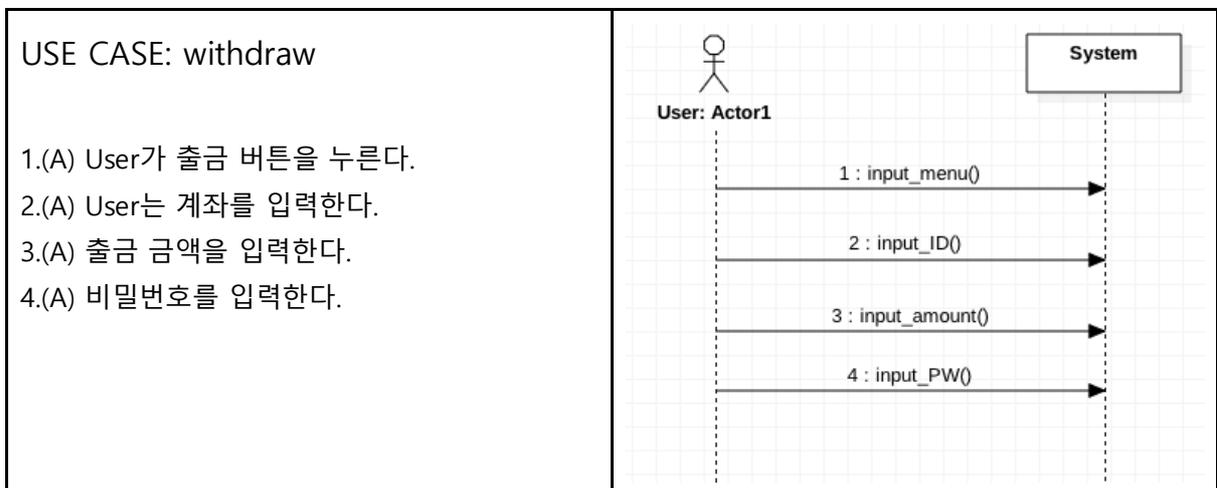


Activity2034. Refine Glossary

Glossary	Category	Description
Timer	Class	ATM 시간을 체크하는 클래스
MoneyCount	Class	기기에 있는 돈을 카운트 하는 클래스
Bank	Class	은행에 대한 정보를 가지고 있는 클래스
Account	Class	계좌(통장) 정보를 가지고 있는 클래스
Card	Class	카드 정보를 가지고 있는 클래스
GUI_Interface	Class	사용자로부터 메뉴를 입력 받아 처리하고 컨트롤 하는 클래스
GUI_Interface.ManagerID	attribute	ATM을 관리하는 관리자 아이디
GUI_Interface.ManagerPW	attribute	ATM을 관리하는 관리자 패스워드
GUI_Interface.TotalMoney	attribute	ATM 이 가지고있는 현금 금액
GUI_Interface.inputID	attribute	ATM에 입력하는 계좌 혹은 카드 번호
GUI_Interface.inputPW	attribute	ATM에 입력하는 비밀번호
GUI_Interface.inputMenu	attribute	ATM에 입력받은 메뉴 번호
GUI_Interface.inputMoney	attribute	ATM에 입력하는 출금,입금,송금 금액
GUI_Interface.UserCount	attribute	인출한 고객의 순서
GUI_Interface.inputPOM	attribute	지폐를 입금 혹은 출금
GUI_Interface.input50000	attribute	입력받은 5만원권의 숫자
GUI_Interface.input10000	attribute	입력받은 만원권의 숫자
GUI_Interface.input5000	attribute	입력받은 5천원권의 숫자
GUI_Interface.input1000	attribute	입력받은 천원권의 숫자

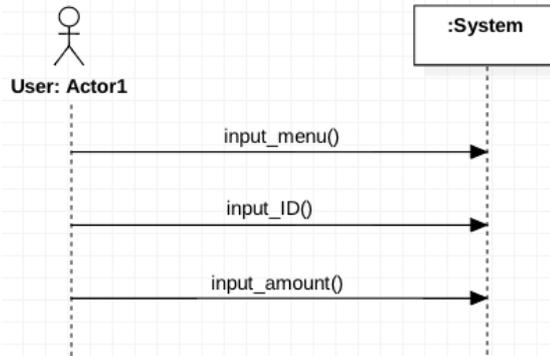
MoneyCount.Count50000	attribute	ATM 에 있는 5만원권의 숫자
MoneyCount.Count10000	attribute	ATM 에 있는 1만원권의 숫자
MoneyCount.Count5000	attribute	ATM 에 있는 5천원권의 숫자
MoneyCount.Count1000	attribute	ATM 에 있는 1천원권의 숫자
Bank.Name	attribute	은행 이름
Bank.BankCode	attribute	각 은행에 대한 코드번호
Bank.isAccount	attribute	찾은 ID가 계좌 혹은 카드
Account.ID	attribute	계좌번호
Account.password	attribute	계좌 비밀번호
Account.name	attribute	계좌 소유자의 이름
Account.Balance	attribute	계좌의 잔고
Card.ID	attribute	카드 번호
Card.password	attribute	카드의 비밀번호
Timer.cal	attribute	ATM의 현재 시간
Timer.JackpotNum	attribute	당첨 번호

Activity2035. Define System Sequence Diagrams



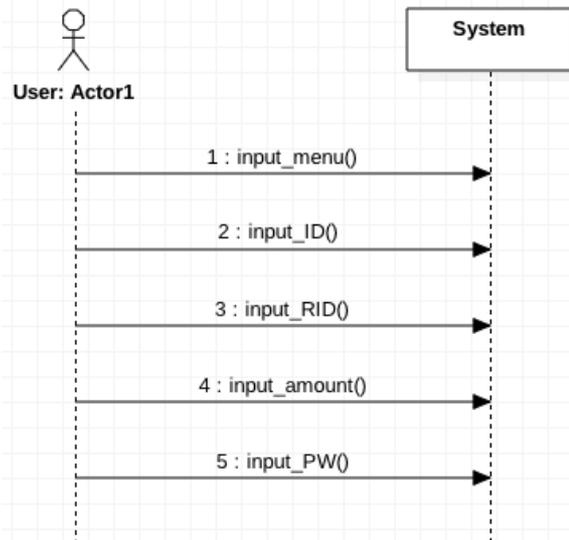
USE CASE: deposit

- 1.(A) User가 입금 버튼을 누른다.
- 2.(A) User는 계좌를 입력한다.
- 3.(A) 입금 금액을 입력한다.



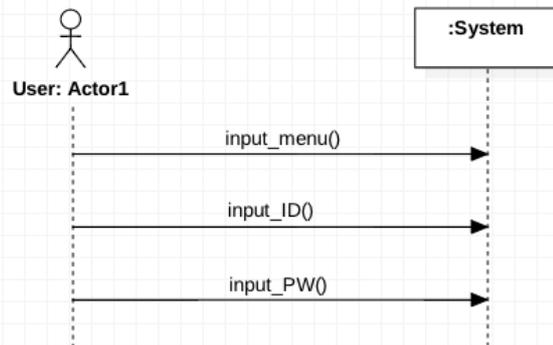
USE CASE: remittance

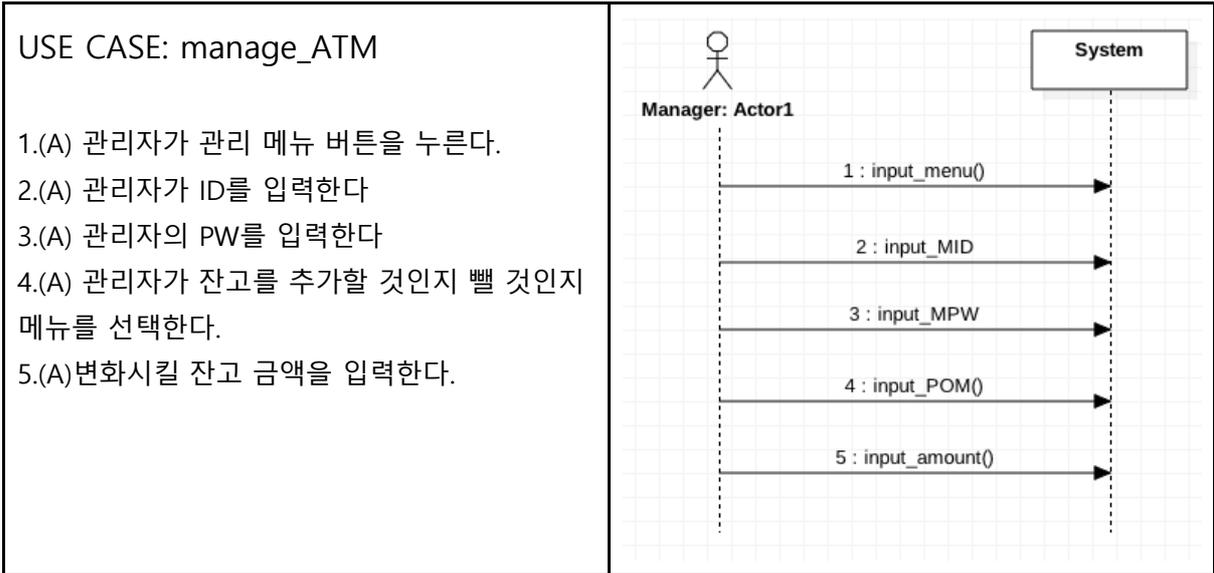
- 1.(A) User가 송금 버튼을 누른다.
- 2.(A) User는 계좌를 입력한다.
- 3.(A) User는 송금받을 계좌를 입력한다.
- 4.(A) User는 송금 금액을 입력한다.
- 5.(A) User는 비밀번호를 입력한다.



USE CASE: view_account_detail

- 1.(A) User가 계좌 조회 버튼을 누른다.
- 2.(A) User는 계좌를 입력한다
- 3.(A) 비밀번호를 입력한다.





Activity2036. Define Operation Contracts

Name	input_menu()
Responsibilities	사용자가 원하는 메뉴의 번호를 입력한다
Type	System
Cross Reference	R1.2 ,R.2, R.3, R.4, R.5
Exception	N/A
Output	입금,출금,송금,조회,매니지 메뉴에 해당하는 메뉴로 진입한다.
Pre-Condition	N/A
Post-Condition	N/A

Name	input_ID()
Responsibilities	사용자의 어카운트번호 또는 카드번호를 입력받는다

Type	System
Cross Reference	R1.2, R.2,R.3,R.4
Exception	N/A
Output	사용자의 어카운트 아이디가 유효한지를 확인한다
Pre-Condition	N/A
Post-Condition	N/A

Name	input_PW()
Responsibilities	사용자에 어카운트 혹은 카드번호에 해당하는 비밀번호를 입력한다
Type	System
Cross Reference	R1.2, R.3, R.4
Exception	N/A
Output	사용자 어카운트 아이디에 해당하는 비밀번호가 맞는지 확인한다
Pre-Condition	사용자 어카운트의 유효성이 확인된 상태여야 한다
Post-Condition	N/A

Name	input_amount()
Responsibilities	사용자가 출금/입금/송금 하고자 하는 금액을 입력한다
Type	System
Cross Reference	R1.2, R.2, R.3
Exception	N/A

Output	출금/송금의 경우 사용자의 입력금액이 사용자가 이용 가능한 금액인지 확인한다. 출금의 경우 ATM 기기 잔고와 비교하여 출금 가능한지 확인한다.
Pre-Condition	사용자 어카운트와 비밀번호가 확인된 상태여야 한다
Post-Condition	N/A

Name	input_RID()
Responsibilities	사용자가 송금받기를 원하는 사람의 계좌번호를 입력한다.
Type	System
Cross Reference	R.3
Exception	N/A
Output	입력된 받는사람 계좌가 유효한지 확인한다.
Pre-Condition	사용자가 보내고자 하는 송금 금액 유효성이 완료된 상태여야 한다
Post-Condition	N/A

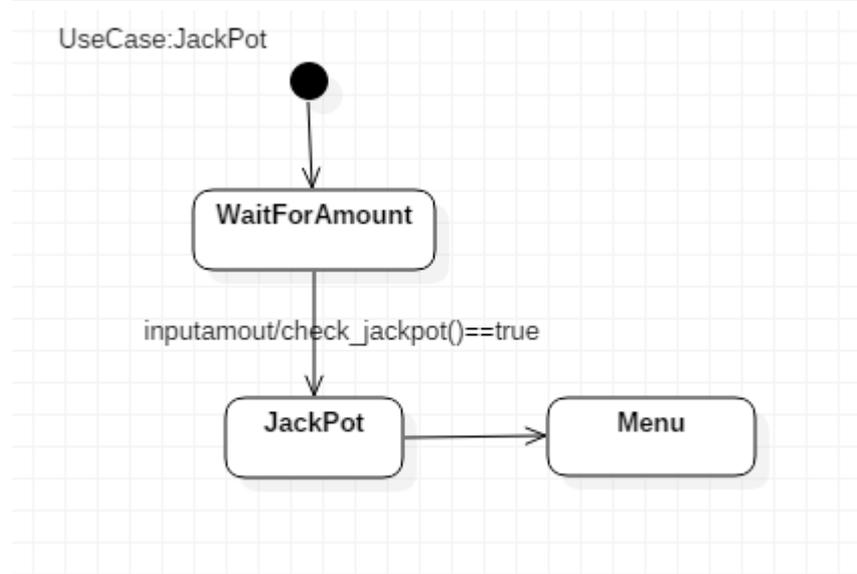
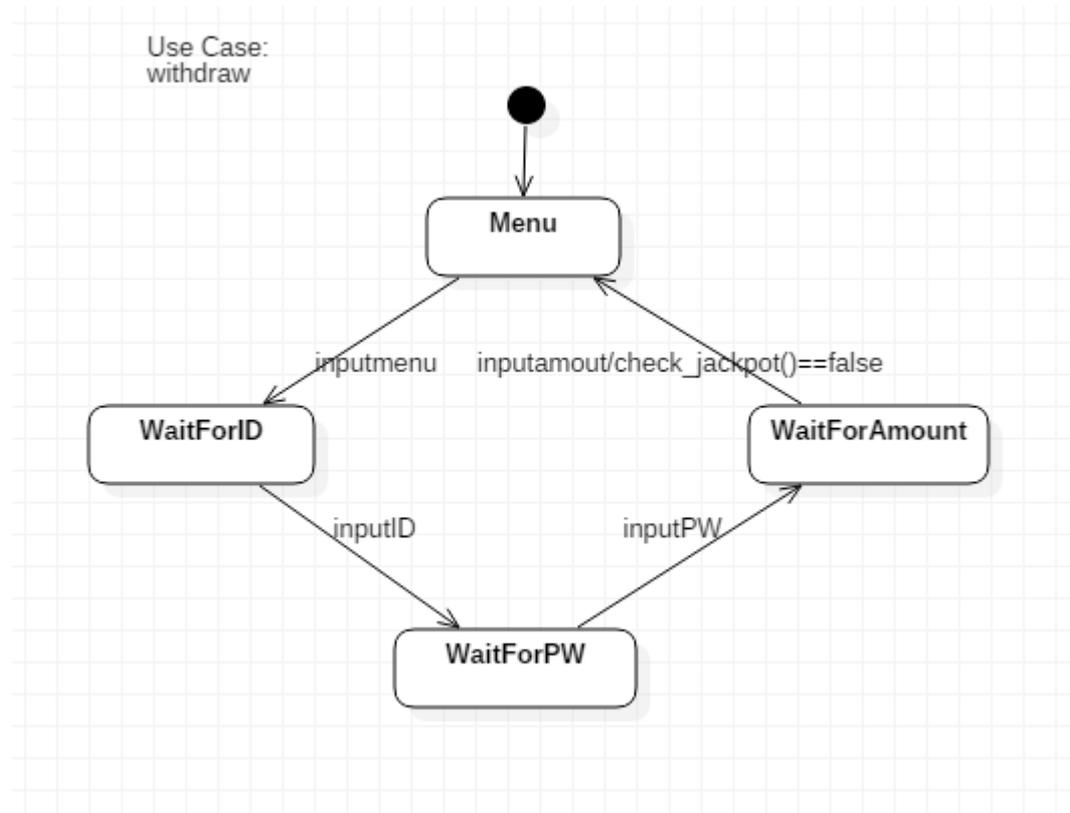
Name	input_MID()
Responsibilities	사용자로부터 매니저의 ID번호를 입력받는다.
Type	System
Cross Reference	R.5
Exception	N/A
Output	매니저의 ID 가 시스템 내부에 저장된 매니저 ID 와 같은지 확인한다.

Pre-Condition	N/A
Post-Condition	N/A

Name	input_MPW()
Responsibilities	사용자로부터 매니저의 패스워드를 입력받는다.
Type	System
Cross Reference	R.5
Exception	N/A
Output	입력된 PW 가 시스템 내부에 저장된 매니저 PW 와 같은지 확인한다.
Pre-Condition	매니저 ID 의 유효성이 확인된 상태여야 한다.
Post-Condition	N/A

Name	input_POM()
Responsibilities	매니저가 입력한 금액을 추가할 것인지 빼줄 것인지를 선택한다.
Type	System
Cross Reference	R.5
Exception	N/A
Output	매니저의 입력에 따라 입력된 금액을 ATM 기기에서 추가하거나 빼준다.
Pre-Condition	매니저의 ID 와 PW 유효성이 확인된 상태여야 한다.
Post-Condition	N/A

Activity3037. Define State Diagram



Activity2038. Refine System Test Case

Test Num	Test 항목	Description	Use case
1-1	input_menu Test	사용자가 입력한 해당 메뉴 화면으로 올바르게 진입하는지 확인한다.	R1.2 ,R.2, R.3, R.4, R.5
2-1	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하지 않으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2,R.3,R.4
2-2	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하면 다음 단계로 넘어가는지 확인한다.	R1.2, R.2,R.3,R.4
3-1	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 아니면 오류메시지를 출력하는지 확인한다	R1.2, R.3, R.4
3-2	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 맞으면 다음단계로 넘어가는지 확인한다.	R1.2, R.3, R.4
4-1	input_amount	출금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-2	input_amount	출금의 경우 사용자가 입력한 금액이 ATM 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-3	input_amount	송금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-4	input_amount	사용자의 입력금액이 계좌 잔고보다 작고 ATM 잔고보다 작을때 다음 단계로 넘어가는지 확인한다.	R1.2, R.2, R.3

5-1	input_RID	입력받은 받는사람 계좌가 유효한지 않을 때 오류메시지를 출력하는지 확인한다	R.3
5-2	input_RID	입력받은 받는사람 계좌가 유효할때 다음 단계로 넘어가는지 확인한다.	R.3
6-1	input_MID	입력받은 매니저ID 가 정확하지 않을 때 오류메시지를 출력하는지 확인한다	R.5
6-2	input_MID	입력받은 매니저ID 가 정확할때 다음단계로 넘어가는지 확인한다	R.5
7-1	input_MPW	입력받은 매니저PW 가 정확하지 않을때 오류 메시지를 출력하는지 확인한다.	R.5
7-2	input_MPW	입력받은 매니저 PW 가 정확할때 다음단계로 넘어가는지 확인한다.	R.5
8-1	input_POM	plus를 입력하였을때, 해당 금액만큼을 ATM 잔고에 더해주는지 확인한다.	R.5
8-2	input_POM	minus를 입력하였을때, 해당 금액만큼을 ATM 잔고에서 빼주는지를 확인한다.	R.5
9-1	Random_Jackpot	난수로 생성되는 숫자를 임의의 숫자로 지정하고 해당 숫자번째 출금 거래에서 추가적으로 출금을 해주는지 확인한다.	R.1.1

Activity2039. Analyze Traceability Analysis

