



# SUPER - SAFE ATM



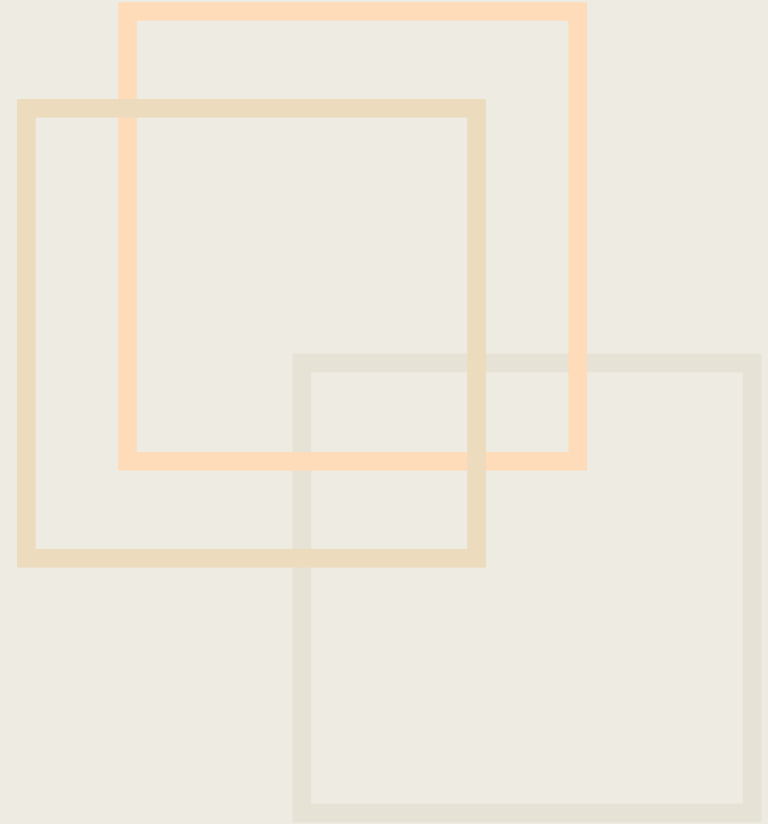
201611269 신문기  
201610401 손하영  
201510283 임진웅

# Index >>

1. Feedback

2. Revised

3. Q & A





# Feedback

---



# 1. Feedback

Test Case No.	Test Case	Result
1	1001	F
2	1101	F
3	1201	F
4	1301	F
5	4001	F
6	4002	P
7	4101	P
8	4201	F
9	4202	F
10	1000.1100.1200.2000.3000.3100.3200.4000.4100.4200.5000.5100.	F
11	1000.1100.1200.2100.3000.3100.3200.4000.4100.4200.5000.5100.	F
12	1000.1100.1200.2200.3000.3100.3200.4000.4100.4200.5000.5100.	F
13	1000.1100.1200.2001.3000.3100.3200.4000.4100.4200.5000.5100.	F
14	1000.1100.1200.2101.3000.3100.3200.4000.4100.4200.5000.5100.	F
15	1000.1100.1200.2201.3000.3100.3200.4000.4100.4200.5000.5100.	F
16	1000.1100.1200.2000.3000.3101.3200.4000.4100.4200.5000.5100.	P
17	1000.1100.1200.2100.3000.3101.3200.4000.4100.4200.5000.5100.	P
18	1000.1100.1200.2200.3000.3101.3200.4000.4100.4200.5000.5100.	P
19	1000.1100.1200.2001.3000.3101.3200.4000.4100.4200.5000.5100.	P
20	1000.1100.1200.2101.3000.3101.3200.4000.4100.4200.5000.5100.	P
21	1000.1100.1200.2201.3000.3101.3200.4000.4100.4200.5000.5100.	P
22	1000.1100.1200.2000.3000.3100.3201.4000.4100.4200.5000.5100.	F
23	1000.1100.1200.2100.3000.3100.3201.4000.4100.4200.5000.5100.	F
24	1000.1100.1200.2200.3000.3100.3201.4000.4100.4200.5000.5100.	F
25	1000.1100.1200.2001.3000.3100.3201.4000.4100.4200.5000.5100.	F
26	1000.1100.1200.2101.3000.3100.3201.4000.4100.4200.5000.5100.	P
27	1000.1100.1200.2201.3000.3100.3201.4000.4100.4200.5000.5100.	P
28	1000.1100.1200.2000.3000.3101.3201.4000.4100.4200.5000.5100.	P
29	1000.1100.1200.2100.3000.3101.3201.4000.4100.4200.5000.5100.	P
30	1000.1100.1200.2200.3000.3101.3201.4000.4100.4200.5000.5100.	P

31	1000.1100.1200.2001.3000.3100.3200.4000.4100.4200.5000.5100.	P
32	1000.1100.1200.2101.3000.3100.3200.4000.4100.4200.5000.5100.	P
33	1000.1100.1200.2201.3000.3100.3200.4000.4100.4200.5000.5100.	P
34	1000.1100.1200.2000.3000.3101.3200.4000.4100.4200.5000.5101.	P
35	1000.1100.1200.2100.3000.3101.3200.4000.4100.4200.5000.5101.	P
36	1000.1100.1200.2200.3000.3101.3200.4000.4100.4200.5000.5101.	P
37	1000.1100.1200.2001.3000.3101.3200.4000.4100.4200.5000.5101.	P
38	1000.1100.1200.2101.3000.3101.3200.4000.4100.4200.5000.5101.	P
39	1000.1100.1200.2201.3000.3101.3200.4000.4100.4200.5000.5101.	P
40	1000.1100.1200.2000.3000.3101.3201.4000.4100.4200.5000.5101.	P
41	1000.1100.1200.2100.3000.3101.3201.4000.4100.4200.5000.5101.	P
42	1000.1100.1200.2200.3000.3101.3201.4000.4100.4200.5000.5101.	F
43	1000.1100.1200.2001.3000.3101.3201.4000.4100.4200.5000.5101.	P
44	1000.1100.1200.2101.3000.3101.3201.4000.4100.4200.5000.5101.	F
45	1000.1100.1200.2201.3000.3101.3201.4000.4100.4200.5000.5101.	F
46	1000.1100.1200.2000.3001.3100.3201.4000.4100.4200.5001.5100.	P
47	1000.1100.1200.2100.3001.3100.3201.4000.4100.4200.5001.5100.	P
48	1000.1100.1200.2200.3001.3100.3201.4000.4100.4200.5001.5100.	P
49	1000.1100.1200.2001.3001.3100.3201.4000.4100.4200.5001.5100.	P
50	1000.1100.1200.2101.3001.3100.3201.4000.4100.4200.5001.5100.	P
51	1000.1100.1200.2201.3001.3100.3201.4000.4100.4200.5001.5100.	P



# 1. Feedback

Pairwise Test Case.20							
	Bank	Action	Log File	PW	Account	4	Result
1	Same	Criminal	Non	-	Not Valid		F
2	Different	Criminal	Exist both	-	Valid		F
3	Same	History	Exist History Only	Valid	Valid		F
4	Different	Deposit	Exist History Only	Not Valid	Not Valid		F
5	Same	Withdraw	Exist both	Valid	Not Valid		F
6	Different	Criminal	Exist History Only	-	Not Valid		F
7	Same	Criminal	Exist Criminal Only	-	Valid		F
8	Different	Withdraw	Exist Criminal Only	Not Valid	Valid		F
9	Different	History	Non	Not Valid	Valid		F
10	Same	Transfer	Exist Criminal Only	Valid	Not Valid		F
11	Different	Withdraw	Non	Not Valid	Valid		F
12	Same	History	Exist both	Valid	Not Valid		F
13	Same	History	Exist Criminal Only	Not Valid	Not Valid		F
14	Same	Deposit	Non	Valid	Valid		F
15	Different	Transfer	Exist both	Not Valid	Valid		F
16	Same	Withdraw	Exist History Only	Valid	Valid		F
17	Same	Deposit	Exist both	Valid	Valid		F
18	Same	Deposit	Exist Criminal Only	Valid	Valid		F
19	Same	Transfer	Exist History Only	Valid	Not Valid		F
20	Different	Transfer	Non	Not Valid	Not Valid		F

# 1. Feedback

Number	Test Case	Result
1	입금 시 입금 금액에 음수를 넣고 확인버튼을 누른다.	Fail
2	출금 시 출금 금액에 음수를 넣고 확인버튼을 누른다.	Fail
3	송금 시 송금 금액에 음수를 넣고 확인버튼을 누른다.	Fail
4	입금 시 입금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	Fail
5	출금 시 출금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	Fail
6	송금 시 송금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	Fail
7	입금 시 입금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	Fail
8	출금 시 출금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	Fail
9	송금 시 송금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	Fail
10	입금 시 입금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	Fail
11	출금 시 출금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	Fail
12	송금 시 송금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	Fail
13	입금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	Fail
14	출금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	Fail
15	송금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	Fail
16	입금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
17	출금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
18	송금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
19	거래 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
20	범죄 이력이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
21	거래 내역이 많을 경우, 거래 내역 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	Fail
22	범죄 내역이 많을 경우, 범죄 이력 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	Fail

# 1. Feedback

- 문서의 Stage별로 누락되거나 새로 생기는 부분들이 존재한다.
- 문서가 일관성 없이 작성되어 각 단계마다 글을 작성하는 방식, 내용 등 달라지는 것이 많다.
- Refine 부분을 명확히 할 필요가 있으며 이에 대한 문서 작성자와 개발자의 명확한 인식이 필요해 보인다.
- 문서에서 존재하나 코드에서 구현이 전혀 되지 않은 부분이 상당수 존재한다.



“1000부터 문서를 다시 작성”

# 1. Feedback

-입금액을 입력할 때 음수 값을 입력해주면 계좌 잔액이 차감되는 등의 기본적인 계산에서 오류가 발생하므로, Super Safe ATM이라는 Non Functional Requirement에 적합하지 않다.

Number	Test Case	Result
1	입금 시 입금 금액에 음수를 넣고 확인버튼을 누른다.	Fail
2	출금 시 출금 금액에 음수를 넣고 확인버튼을 누른다.	Fail
3	송금 시 송금 금액에 음수를 넣고 확인버튼을 누른다.	Fail
4	입금 시 입금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	Fail
5	출금 시 출금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	Fail
6	송금 시 송금 금액에 숫자 이외의 문자를 넣고 확인버튼을 누른다.	Fail
7	입금 시 입금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	Fail
8	출금 시 출금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	Fail
9	송금 시 송금 금액에 int 범위 이외의 숫자를 넣고 확인버튼을 누른다.	Fail
10	입금 시 입금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	Fail
11	출금 시 출금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	Fail
12	송금 시 송금 금액에 아무것도 넣지 않고 확인버튼을 누른다.	Fail
13	입금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	Fail
14	출금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	Fail
15	송금 시 비밀번호를 3회 이상 틀렸을 때 예외처리를 확인 한다.	Fail
16	입금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
17	출금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
18	송금 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
19	거래 내역이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
20	범죄 이력이 많을 경우 1초 이내에 처리 되는지 확인 한다.	Fail
21	거래 내역이 많을 경우, 거래 내역 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	Fail
22	범죄 내역이 많을 경우, 범죄 이력 조회 기능에서 Simple한 UI/UX를 가지는지 확인한다.	Fail



“예외 처리”



# Revised

---

2.1 Stage 1000.

2.2 Stage 2030.

2.3 Stage 2040.

2.4 Stage 2050.



# 2-1. Stage 1000 \_Act 1003

Function	Description
selectMenu	<ol style="list-style-type: none"><li>1. 입금, 출금, 송금, 거래내역 확인, 범죄이력조회 중 이용할 메뉴를 선택한다.</li><li>2. 유저가 선택한 메뉴에 따라 입금, 출금, 송금, 거래내역 확인, 범죄이력조회가 진행된다.</li></ol>
MediumCheck	<ol style="list-style-type: none"><li>1. 카드 번호를 입력할지 계좌 번호를 입력할지 선택한다.</li><li>2. 선택에 따라 카드 번호나 계좌 번호를 입력한다.</li><li>3. 입력된 계좌 번호나 입력된 카드 번호에 연결된 계좌 번호가 DB에 존재하는 계좌인지 확인한다.</li><li>4. 유효하지 않을 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</li></ol>
Deposit	<ol style="list-style-type: none"><li>1. 입금할 금액을 입력받는다.</li><li>2. 계좌 번호에 해당되는 비밀번호를 입력받고 입력받은 비밀번호가 해당 계좌에 맞는 비밀번호인지 확인한다. 유효하지 않을 경우 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</li><li>3. 계좌에 타행 거래일 경우 입력받은 돈에서 수수료를 제외하고 입금한다.</li><li>4. 계좌에 돈을 입금한 후 실행한 작업과 입금한 금액, 계좌의 잔액을 출력한다.</li></ol>
Withdraw	<ol style="list-style-type: none"><li>1. 출금할 금액을 입력받는다.</li><li>2. 타행 거래일 경우 수수료를 계산하고 수수료를 포함하여 출금할 계좌의 잔액이 충분한지 확인한다.</li><li>3. 계좌 번호에 해당되는 비밀번호를 입력받고 입력받은 비밀번호가 해당 계좌에 맞는 비밀번호인지 확인한다. 유효하지 않을 경우 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</li><li>4. 계좌에 타행 거래일 경우 입력받은 돈에서 수수료를 제외하고 출금한다.</li><li>5. 계좌에서 돈을 출금한 후 실행한 작업과 출금한 금액, 계좌의 잔액을 출력한다.</li></ol>

# 2-1. Stage 1000 \_Act 1003

Function	Description
Transfer	<ol style="list-style-type: none"><li>1. 송금할 금액을 입력받는다.</li><li>2. 돈을 보낼 계좌를 입력하고 입력된 계좌 번호가 DB에 존재하는 계좌인지 확인한다. 유효하지 않을 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</li><li>3. 타행 거래일 경우 수수료를 계산하고 수수료를 포함하여 송금할 계좌의 잔액이 충분한지 확인한다.</li><li>4. 계좌 번호에 해당되는 비밀번호를 입력받고 입력받은 비밀번호가 해당 계좌에 맞는 비밀번호인지 확인한다. 유효하지 않을 경우 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</li><li>5. 계좌에 타행 거래일 경우 입력받은 돈에서 수수료를 제외하여 송금한다.</li><li>6. 계좌에서 돈을 송금한 후 실행한 작업과 송금한 금액, 계좌의 잔액을 출력한다.</li></ol>
Check TransactionHistory	<ol style="list-style-type: none"><li>1. 입력받은 계좌의 거래내역을 출력한다.</li></ol>
CheckCriminal History	<ol style="list-style-type: none"><li>1. 범죄이력을 조회하고자 하는 계좌를 입력하고 입력된 계좌번호가 DB에 존재하는 계좌인지 확인한다. 유효하지 않을 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.</li><li>2. 입력된 계좌에 대한 범죄이력을 출력한다.</li></ol>

# 2-1. Stage 1000 \_Act 1003

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



# 2-1. Stage 1000 \_Act 1004

Term	Description
User	ATM을 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	ATM과 거래하는 은행. ATM에 지정된 당행 외에 은행은 3개 존재한다.
Result	ATM에서 거래한 뒤 거래 내역을 명시한 명세표
DB	거래하는 은행의 데이터베이스
criminalLog	범죄에 사용된 이력

# 2-1. Stage 1000 \_Act 1005

입금

출금

송금

조회

범죄이력조회

금액 19800

돌아가기 확인

Card

돌아가기 확인

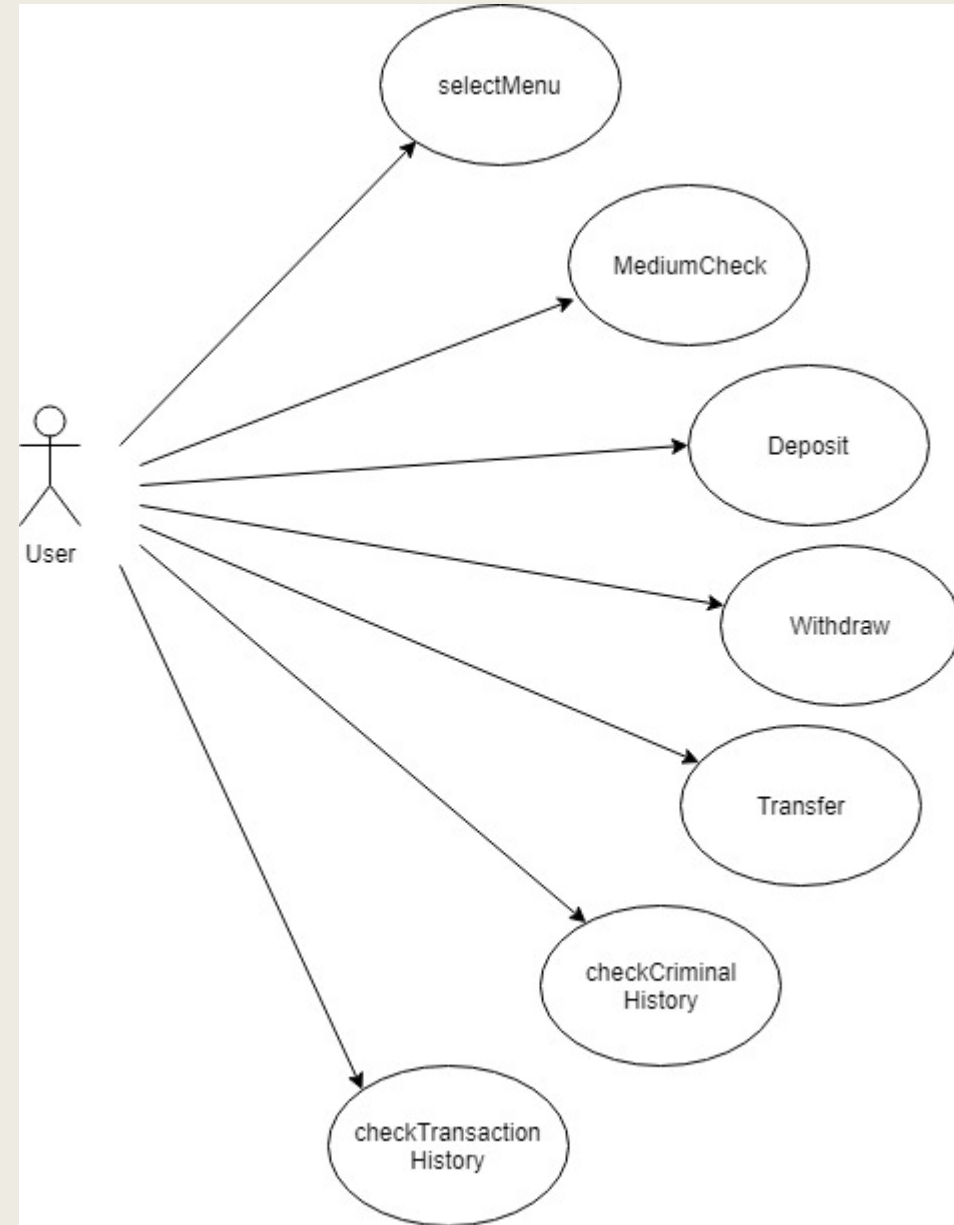
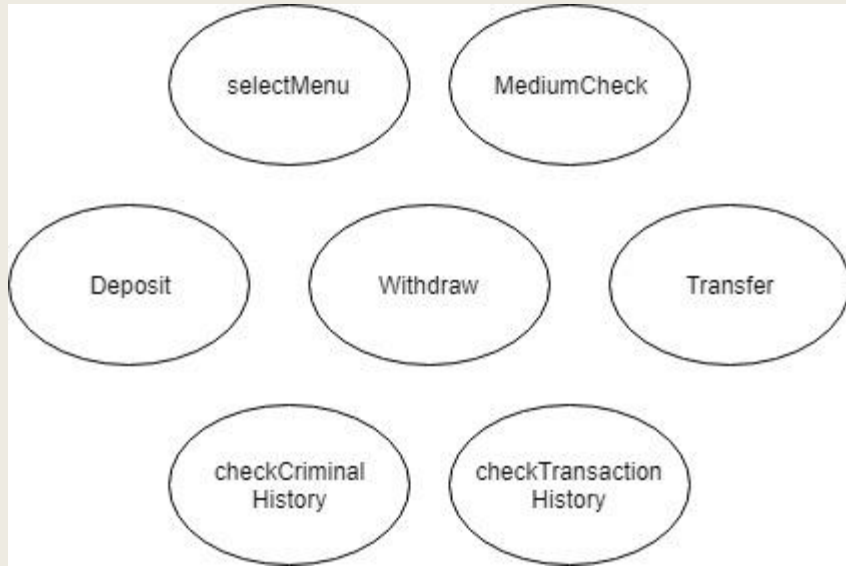
비밀번호

돌아가기 확인

# 2-1. Stage 1000 \_Act 1006

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

# 2-1. Stage 1000 \_Act 1006



# 2-1. Stage 1000 \_Act 1006

Use Case	1.selectMenu
Actor	User
Description	1. 유저가 화면에 출력되는 입금, 출금, 송금, 거래내역 확인, 범죄이력조회 5개의 메뉴 선택 버튼 중 하나를 눌러 이용할 메뉴를 선택한다.
Use Case	2. MediumCheck
Actor	User
Description	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	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.

# 2-1. Stage 1000 \_Act 1006

Use Case	4. Withdraw
Actor	User
Description	<ol style="list-style-type: none"><li>1. 출금할 금액을 입력한다.</li><li>2. 출금할 계좌의 잔액이 부족하다고 경고가 뜰 경우 메뉴 선택 화면으로 돌아간다.</li><li>3. 잔액이 부족하지 않을 경우 계좌 번호에 해당되는 비밀번호를 입력한다. 입력한 비밀번호가 유효하지 않을 경우 알림을 받고 다시 입력한다.</li><li>4. 계좌에서 돈을 출금하고 화면에 실행된 작업과 출금한 금액, 계좌의 잔액이 출력된 것을 확인한다.</li><li>5. 끝내기 버튼을 눌러 메인 화면으로 돌아간다.</li></ol>

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

# 2-1. Stage 1000 \_Act 1006

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

Use Case	7. CheckTransactionHistory
Actor	User
Description	<ol style="list-style-type: none"><li>1. 범죄 이력을 조회하고자 하는 계좌를 입력하고 입력된 계좌번호가 유효하지 않다고 경고가 뜰 경우 다시 입력한다.</li><li>2. 입력한 계좌의 범죄이력내역이 화면에 출력된 것을 확인한다.</li><li>3. 끝내기 버튼을 눌러 메인 화면으로 돌아간다.</li></ol>

# 2-1. Stage 1000 \_Act 1009

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.



# 2-1. Stage 1000 \_Act 1009

Identifier	Functional Requirements	Description
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.

# 2-1. Stage 1000 \_Act 1009

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 1초 이내로 반응하는지 확인한다.
ATM_STC_010	OS - Independent	프로그램이 OS에 무관하게 정상적으로 동작하는지 확인한다.

# 2-2. Stage 2030 \_Act 2031

Use Case	1. selectMenu
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: "selectMenu"
Pre-Requisites	ATM System Started
Typical Courses of Events	(A): Actor , (S):System (S) 화면에 메뉴(입금, 출금, 송금, 거래내역, 범죄이력확인 버튼)를 띄운다. (A) 5개의 버튼 중 하나를 누른다. (S) 입력 받은 버튼에 따라 입금, 출금, 송금, 거래내역 확인, 범죄이력 조회가 진행된다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

# 2-2. Stage 2030 \_Act 2031

Use Case	2. MediumCheck
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: "MediumCheck"
Pre-Requisites	SelectMenu
Typical Courses of Events	(A): Actor , (S):System (S) 화면에 매체 선택 버튼(카드 / 통장)을 띄운다. (A) 둘 중에 이용할 매체에 해당하는 버튼을 입력한다. (S) 번호 입력창을 띄운다. (A) 선택에 따라 카드 번호나 계좌 번호를 입력한다. (S) 입력된 계좌 번호나 입력된 카드번호에 연결된 계좌 번호가 DB에 존재하는 계좌인지 확인한다. .
Alternative Courses of Events	N/A
Exceptional Courses of Events	5. 유효하지 않은 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.(1번부터 다시 진행)

# 2-2. Stage 2030 \_Act 2031

Use Case	3. Deposit
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: "Deposit"
Pre-Requisites	MediumCheck
Typical Courses of Events	<p>(A): Actor , (S):System            (S) 화면에 금액 입력 창을 띄운다.            (A) 입금할 금액을 입력한다.            (S) 비밀번호 입력 창을 띄운다.            (A) 비밀번호를 입력한다.            (S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다. .            (S) 비밀번호가 일치할 시 ATM의 소속은행과 계좌의 소속은행을 비교하여 수수료를 계산한다.            (S) 수수료를 제외한 금액만큼 계좌의 보유현금을 증가시킨다.            (S) 계좌에 돈을 입금한 후 실행한 작업(입금)과 입금한 금액, 계좌의 잔액을 화면에 출력하고 로그에 저장한다.</p>
Alternative Courses of Events	6. 타행 계좌의 경우 수수료를 부과하고, 당행 계좌일 경우 수수료를 부과하지 않는다.
Exceptional Courses of Events	<p>2. 입금할 금액이 음수일 경우 Error            5. 비밀번호가 일치하지 않을 시 유효하지 않다고 알림을 보낸후 다시 3번 부터 진행한다.</p>

# 2-2. Stage 2030 \_Act 2031

Use Case	4. Withdraw
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: "Withdraw"
Pre-Requisites	MediumCheck
Typical Courses of Events	<p>(A): Actor , (S):System            (S) 화면에 금액 입력창을 띄운다.            (A) 출금할 금액을 입력한다.            (S) ATM의 소속은행과 계좌의 소속은행을 비교하여 수수료를 계산한다.            (S) 입력받은 금액과 수수료를 더한 만큼의 금액이 계좌의 잔액보다 크지 않은지 확인한다.            (S) 비밀번호 입력 창을 띄운다.            (A) 비밀번호를 입력한다.            (S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다. .            (S) 출금 금액과 수수료만큼 계좌의 보유현금을 감소시킨다.            (S) 계좌에 돈을 출금한 후 실행한 작업(출금)과 출금한 금액, 계좌의 잔액을 화면에 출력하고 로그에 저장한다.</p>
Alternative Courses of Events	2. 타행 계좌의 경우 수수료를 부과하고, 당행 계좌일 경우 수수료를 부과하지 않는다.
Exceptional Courses of Events	<p>2. 출금할 금액이 음수이면 Error            4. 출금하려는 금액과 수수료보다 잔액이 적을 시 Error            7. 비밀번호가 일치하지 않을 시 유효하지 않다고 알림을 보낸후 다시 5번 부터 진행한다.</p>

# 2-2. Stage 2030 \_Act 2031

Use Case	5. Transfer
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: "Transfer"
Pre-Requisites	MediumCheck
Typical Courses of Events	<p>(A): Actor , (S):System</p> <p>(S) 화면에 금액 입력 창을 띄운다.</p> <p>(A) 송금할 금액을 입력한다.</p> <p>(S) 화면에 계좌번호 입력 창을 띄운다.</p> <p>(A) 송금할 계좌번호를 입력한다.</p> <p>(S) 입력된 계좌 번호가 DB에 존재하는 계좌인지 확인한다.</p> <p>(S) 송금할 계좌와 송금 받을 계좌의 소속은행을 비교하여 수수료를 계산한다.</p> <p>(S) 입력받은 금액과 수수료를 더한 만큼의 금액이 송금할 계좌의 잔액보다 크지 않은지 확인한다.</p> <p>(S) 비밀번호 입력 창을 띄운다.</p> <p>(A) 비밀번호를 입력한다.</p> <p>(S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다.</p> <p>(S) 송금 금액과 수수료를 더한 만큼 송금할 계좌의 보유현금을 감소시킨다.</p> <p>(S) 송금 금액만큼 송금받은 계좌의 보유현금을 증가시킨다.</p> <p>(S) 계좌에 돈을 송금한 후 실행한 작업(송금)과 송금한 금액, 계좌의 잔액을 화면에 출력하고 로그에 저장한다.</p>
Alternative Courses of Events	6. 타행 계좌의 경우 수수료를 부과하고, 당행 계좌일 경우 수수료를 부과하지 않는다.
Exceptional Courses of Events	<p>2. 송금할 금액이 음수일 경우 Error</p> <p>5. 유효하지 않은 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력을 받는다.(3번부터 다시 진행한다.)</p> <p>7. 송금하려는 금액과 수수료보다 잔액이 적을 시 Error</p> <p>10. 비밀번호가 일치하지 않을 시 유효하지 않다고 알림을 보낸후 다시 8번 부터 진행한다.</p>

# 2-2. Stage 2030 \_Act 2031

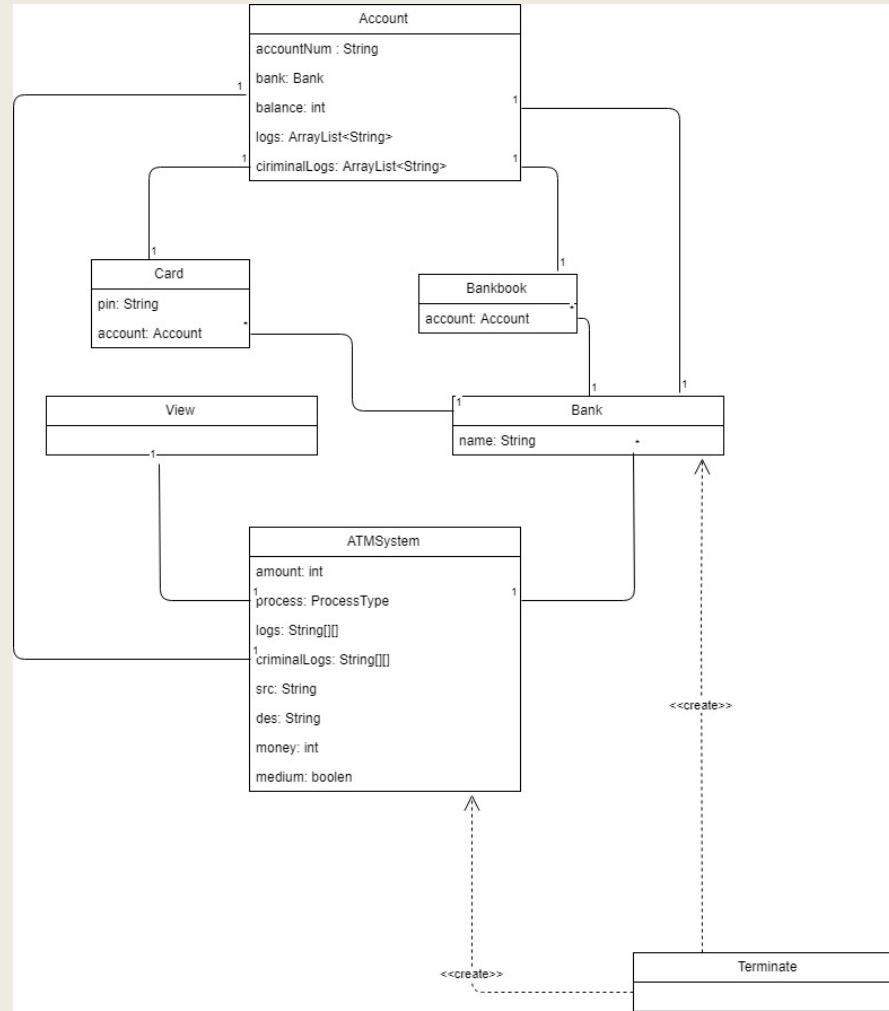
Use Case	6. CheckTransactionHistory
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: "CheckTransactionHistory"
Pre-Requisites	MediumCheck
Typical Courses of Events	(A): Actor , (S):System (S) 비밀번호 입력창을 띄운다. (A) 비밀번호를 입력한다. (S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다. . (S) MediumCheck를 통해 입력받은 계좌의 거래내역 log파일을 불러와 화면에 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. 비밀번호가 일치하지 않을 시 1번 부터 다시 진행한다.



# 2-2. Stage 2030 \_Act 2031

Use Case	7. CheckCriminalHistory
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: "CheckCriminalHistory"
Pre-Requisites	SelectMenu
Typical Courses of Events	(A): Actor , (S):System (S) 계좌번호 입력창을 띄운다. (A) 계좌번호를 입력한다. (S) 입력된 계좌번호가 DB에 존재하는 계좌인지 확인한다. (S) 입력된 계좌에 대한 범죄 이력 log를 가져와 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. 유효하지 않은 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다(1번부터 다시 진행)

# 2-2. Stage 2030 \_Act 2033



# 2-2. Stage 2030 \_Act 2034

Term	Category	Remarks
Account	Class	카드나 통장에 연결된 계좌
Bank	Class	ATM과 연결된 은행 데이터
Card	Class	카드
Passbook	Class	통장
ATM	Class	ATM
User	Class	ATM을 사용하는 유저
Account.accountNum	Attribute	계좌 번호
Account.bank	Attribute	계좌의 은행
Account.password	Attribute	계좌의 비밀번호

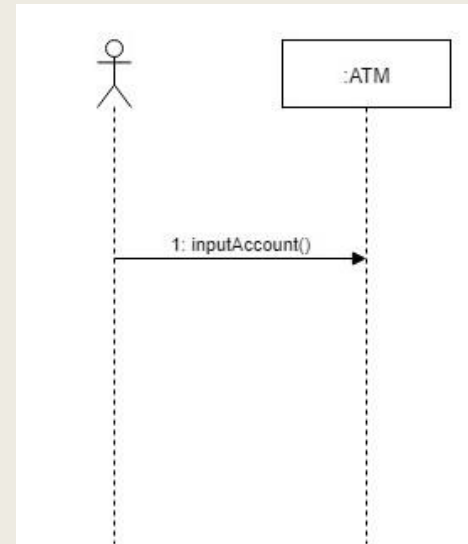
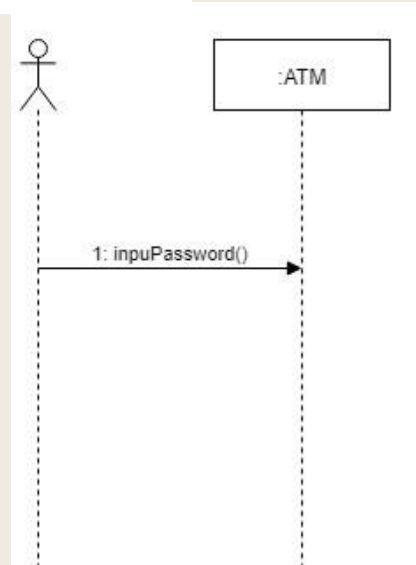
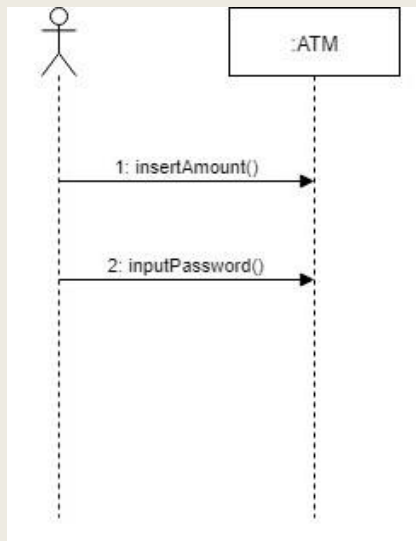
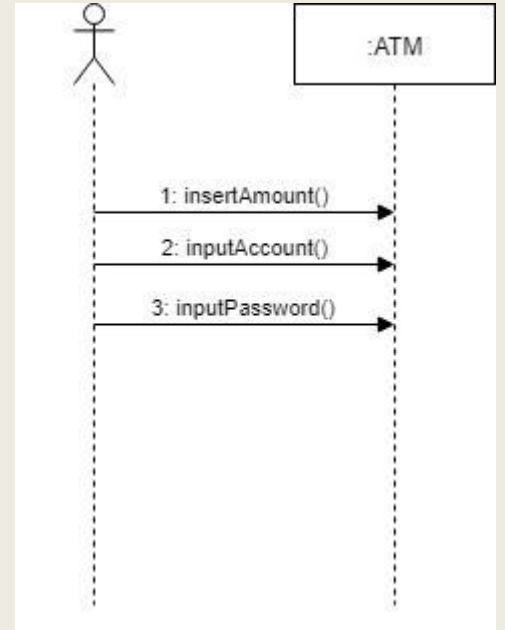
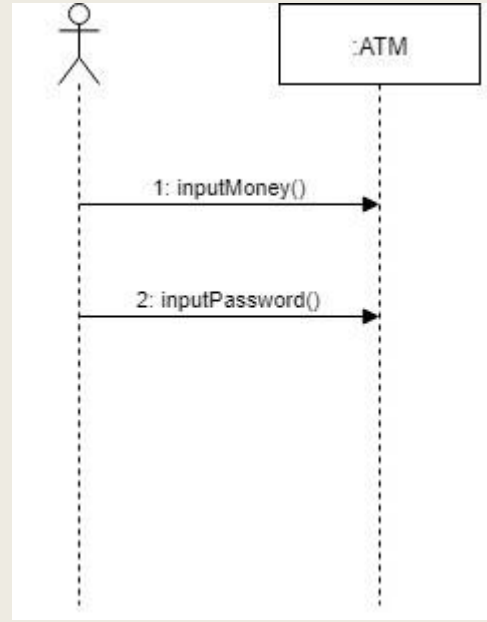
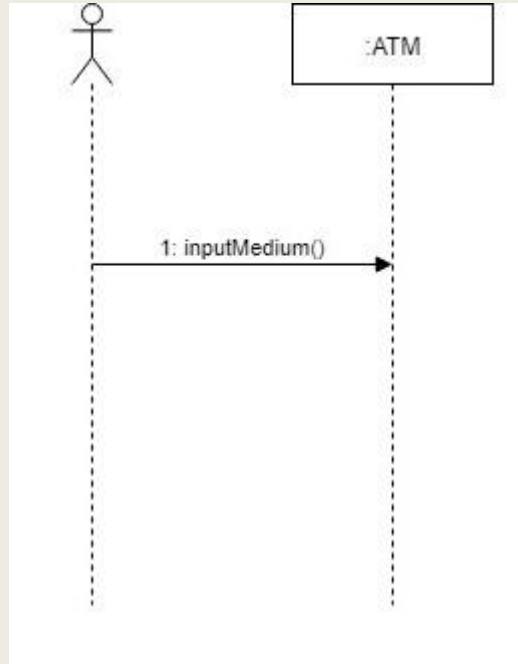
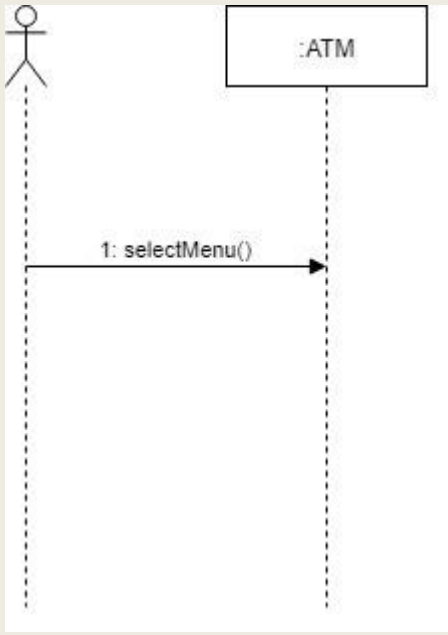
# 2-2. Stage 2030 \_Act 2034

Term	Category	Remarks
AtmSystem.own	Attribute	ATM의 소속은행
AtmSystem.src	Attribute	ATM을 이용중인 계좌
AtmSystem.des	Attribute	송금받을 계좌
AtmSystem.amount	Attribute	출금 또는 송금할 금액
AtmSystem.money	Attribute	입금할 금액
AtmSystem.fee	Attribute	서비스 이용시 수수료
AtmSystem.medium	Attribute	ATM 이용시 사용한 매체
AtmSystem.banks	Attribute	이용 가능한 은행 리스트
AtmSystem.process	Attribute	이용중인 서비스

# 2-2. Stage 2030 \_Act 2034

Term	Category	Remarks
Bank.name	Attribute	은행의 이름
Bank.path	Attribute	Bank DB의 주소
Bank.cards	Attribute	Bank의 당행 카드 리스트
Bank.bankbooks	Attribute	Bank의 당행 통장 리스트
Bankbook.account	Attribute	통장과 연결된 계좌
Card.account	Attribute	카드와 연결된 계좌
Card.pin	Attribute	카드번호
Card.password	Attribute	카드 비밀번호

# 2-2. Stage 2030 \_Act 2035



# 2-2. Stage 2030 \_Act 2036

Use Case	Name of Actor-Activated Events	System Operations
1. selectMenu	1: selectMenu()	1. selectMenu()
2. MediumCheck	2: inputMedium()	2. inputMedium()
3. Deposit	1: inputMoney()	3. inputMoney()
	2: inputPassword()	4. inputPassword()
4. Withdraw	1: insertAmount()	5. inputAmount()
	2: inputPassword()	4. inputPassword()
5. Transfer	1: insertAmount()	5. inputAmount()
	2: inputAccount()	6. inputAccount()
	3: inputPassword()	4. inputPassword()
6. CheckTransaction History	1: inputPassword()	4. inputPassword()
7. CheckCriminal History	1: inputAccount()	6. inputAccount()

# 2-2. Stage 2030 \_Act 2036

Name	selectMenu()
Responsibilities	입금, 출금, 송금, 거래내역확인, 범죄이력확인 버튼 중 하나를 선택한다.
Type	System
Cross Reference	System functions: R1 Use Case: "selectMenu"
Notes	
Exceptions	
Output	선택된 메뉴
Pre-conditions	프로그램이 시작되어있는 상태여야 한다.
Post-conditions	선택된 버튼에 따라 입금, 출금, 송금, 거래내역확인, 범죄이력확인이 진행되고 입금, 출금, 송금, 거래내역확인의 경우 매체를 입력받는다.

Name	inputMedium()
Responsibilities	카드 번호를 입력할지 계좌 번호를 입력할지 선택할 수 있으며, 선택에 따라 카드번호나 계좌번호를 입력받는다.
Type	System
Cross Reference	System functions: R2 Use Case: "MediumCheck"
Notes	
Exceptions	
Output	입력받은 카드 번호, 계좌 번호
Pre-conditions	입금, 출금, 송금, 거래내역확인 중 하나를 선택해야만 한다.
Post-conditions	입력된 계좌번호가 DB에 존재하는 계좌인지 확인한다.



# 2-2. Stage 2030 \_Act 2036

<b>Name</b>	inputMoney()
<b>Responsibilities</b>	입금할 돈을 입력받는다.
<b>Type</b>	System
<b>Cross Reference</b>	System functions: R3 Use Case: "Deposit"
<b>Notes</b>	
<b>Exceptions</b>	
<b>Output</b>	입금할 돈의 양
<b>Pre-conditions</b>	계좌가 유효한지 확인되어 있는 상태여야 한다.
<b>Post-conditions</b>	비밀번호를 입력받는다.

<b>Name</b>	inputPassword()
<b>Responsibilities</b>	비밀번호를 입력받는다.
<b>Type</b>	System
<b>Cross Reference</b>	System functions: R3, R4, R5, R6 Use Case: "Deposit", "Withdraw", "Transfer", "CheckTransactionHistory"
<b>Notes</b>	
<b>Exceptions</b>	
<b>Output</b>	입력받은 비밀번호
<b>Pre-conditions</b>	입금, 출금, 송금, 거래내역확인 중 하나를 선택한 상태이다.
<b>Post-conditions</b>	비밀번호가 계좌에 해당되는 비밀번호인지 확인한다.

# 2-2. Stage 2030 \_Act 2036

<b>Name</b>	inputAmount()
<b>Responsibilities</b>	출금하거나 송금할 돈의 양을 입력한다.
<b>Type</b>	System
<b>Cross Reference</b>	System functions: R4, R5 Use Case: “Withdraw”, “Transfer”
<b>Notes</b>	
<b>Exceptions</b>	
<b>Output</b>	출금일 경우 출금할 돈, 송금일 경우 송금할 돈
<b>Pre-conditions</b>	계좌가 유효한 상태인지 확인되어 있어야한다.
<b>Post-conditions</b>	출금일 경우 출금이 되고 송금일 경우 송금할 계좌를 입력받는다.

<b>Name</b>	inputAccount()
<b>Responsibilities</b>	돈을 보낼 계좌나 범죄 이력을 조회할 계좌 번호를 입력한다.
<b>Type</b>	System
<b>Cross Reference</b>	System functions: R5, R7 Use Case: “Transfer”, “CheckCriminalHistory”
<b>Notes</b>	
<b>Exceptions</b>	
<b>Output</b>	돈을 보낼 계좌 번호나 범죄 이력을 조회할 계좌 번호
<b>Pre-conditions</b>	계좌번호가 DB에 존재하는 계좌인지 확인한다.
<b>Post-conditions</b>	계좌번호가 유효한지 확인한다.

# 2-2. Stage 2030 \_Act 2038

Identifier	Functional Requirements	Description
ATM_STC_001_001	selectMenu	사용자가 입금을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.
ATM_STC_001_002	selectMenu	사용자가 출금을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.
ATM_STC_001_003	selectMenu	사용자가 송금을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.
ATM_STC_001_004	selectMenu	사용자가 거래내역확인을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.
ATM_STC_001_005	selectMenu	사용자가 범죄이력조회를 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.
ATM_STC_002_001	MediumCheck	카드 번호를 입력할지 선택하고 이에 따라 유효한 카드 번호를 입력했을 때 계좌 확인이 정상적으로 진행되는지 확인한다.
ATM_STC_002_002	MediumCheck	카드 번호를 입력할지 선택하고 이에 따라 유효하지 않은 카드 번호를 입력했을 때 다시 입력받을 수 있는지 확인한다.
ATM_STC_002_003	MediumCheck	통장 번호를 입력할지 선택하고 이에 따라 유효한 통장 번호를 입력했을 때 계좌 확인이 정상적으로 진행되는지 확인한다.
ATM_STC_002_004	MediumCheck	통장 번호를 입력할지 선택하고 이에 따라 유효하지 않은 통장 번호를 입력했을 때 다시 입력받을 수 있는지 확인한다.

# 2-2. Stage 2030 \_Act 2038

Identifier	Functional Requirements	Description
ATM_STC_003_001	Deposit	입금할 금액을 잘 입력받을 수 있는지 확인한다.
ATM_STC_003_002	Deposit	계좌 번호에 해당되는 비밀번호가 입력되었을 때 수수료를 제외하고 입금되는지 확인한다.
ATM_STC_003_003	Deposit	계좌 번호에 해당되는 비밀번호가 아니라 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다.
ATM_STC_003_004	Deposit	입금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.
ATM_STC_004_001	Withdraw	출금할 금액을 잘 입력받을 수 있는지 확인한다.
ATM_STC_004_002	Withdraw	수수료를 포함하여 출금할 계좌의 잔액이 충분하지 않을 경우 출금 진행이 멈추는지 확인한다.
ATM_STC_004_003	Withdraw	계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 제외하고 출금되는지 확인한다.
ATM_STC_004_004	Withdraw	계좌번호에 해당되는 비밀번호가 아니라 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다.
ATM_STC_004_005	Withdraw	출금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.

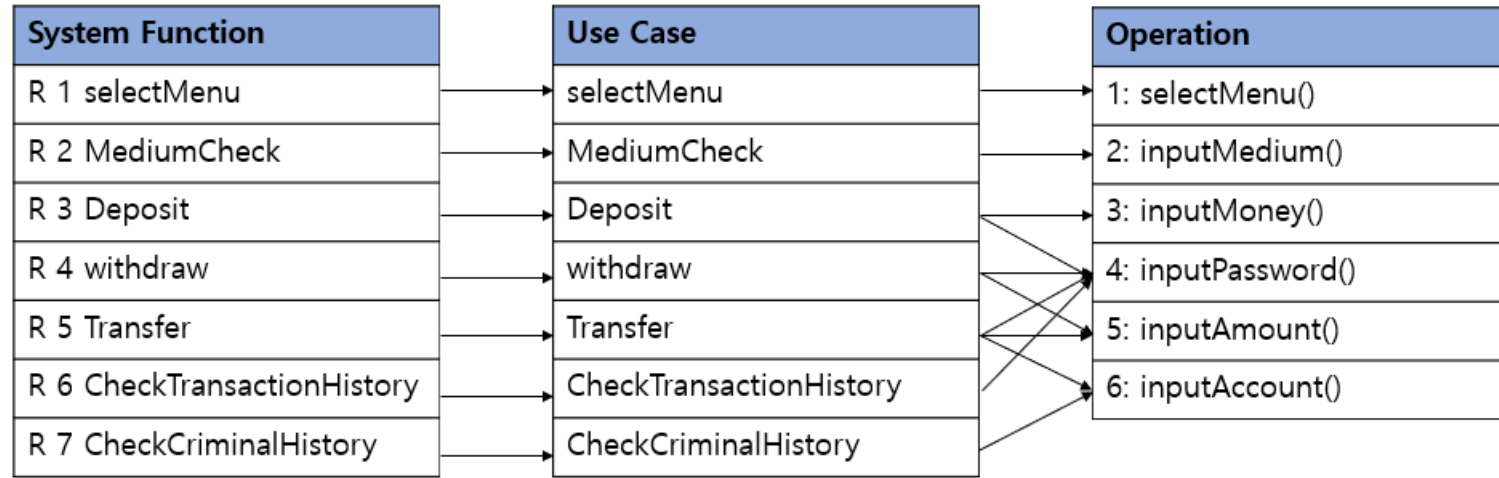
# 2-2. Stage 2030 \_Act 2038

Identifier	Functional Requirements	Description
ATM_STC_005_001	Transfer	송금할 금액을 잘 입력받을 수 있는지 확인한다.
ATM_STC_005_002	Transfer	돈을 보낼 계좌를 입력하고 유효하지 않은 계좌를 입력했을 때 경고를 받고 다시 입력받는지 확인한다.
ATM_STC_005_003	Transfer	수수료를 포함하여 송금할 계좌의 잔액이 충분하지 않을 경우 송금 진행이 멈추는지 확인한다.
ATM_STC_005_004	Transfer	계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 제외하고 송금되는지 확인한다.
ATM_STC_005_005	Transfer	계좌 번호에 해당되는 비밀번호가 아니라 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다.
ATM_STC_005_006	Transfer	송금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.
ATM_STC_006	CheckTransaction History	입력한 계좌의 거래내역이 화면에 정상적으로 출력되는지 확인한다.
ATM_STC_007_001	CheckCriminal History	범죄이력을 조회하고 싶은 계좌를 입력하고 유효하지 않은 계좌를 입력했을 때 경고를 받고 다시 입력받는지 확인한다.
ATM_STC_007_002	CheckCriminal History	입력한 계좌의 범죄이력내역이 화면에 정상적으로 출력되는지 확인한다.

# 2-2. Stage 2030 \_Act 2038

Identifier	Functional Requirements	Description
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 1초 이내로 반응하는지 확인한다.
ATM_STC_010	OS - Independent	프로그램이 OS에 무관하게 정상적으로 동작하는지 확인한다.

# 2-2. Stage 2030 \_Act 2039



# 2-3. Stage 2040 \_Act 2041

Use Case	1. Select Menu
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: R1
Pre-Requisites	Power On
Typical Courses of Events	(A): Actor , (S):System (S) 화면에 메뉴(입금, 출금, 송금, 거래내역, 범죄이력확인 버튼)를 띄운다. (A) 5개의 버튼 중 하나를 누른다. (S) 입력 받은 버튼에 따라 입금, 출금, 송금, 거래내역 확인, 범죄이력 조회가 진행된다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A



# 2-3. Stage 2040 \_Act 2041

Use Case	2. MediumCheck
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: R2
Pre-Requisites	SelectMenu
Typical Courses of Events	(A): Actor , (S):System (S) 화면에 매체 선택 버튼(카드 / 통장)을 띄운다. (A) 둘 중에 이용할 매체에 해당하는 버튼을 입력한다. (S) 번호 입력창을 띄운다. (A) 선택에 따라 카드 번호나 계좌 번호를 입력한다. (S) 입력된 계좌 번호나 입력된 카드번호에 연결된 계좌 번호가 DB에 존재하는 계좌인지 확인한다. .
Alternative Courses of Events	N/A
Exceptional Courses of Events	5. 유효하지 않은 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다.(1번부터 다시 진행)

# 2-3. Stage 2040 \_Act 2041

<b>Use Case</b>	3. Deposit
<b>Actor</b>	User
<b>Purpose</b>	As in the business use case
<b>Overview</b>	As in the business use case
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	Use Cases: R3
<b>Pre-Requisites</b>	MediumCheck
<b>Typical Courses of Events</b>	<p>(A): Actor , (S):System            (S) 화면에 금액 입력 창을 띄운다.            (A) 입금할 금액을 입력한다.            (S) 비밀번호 입력 창을 띄운다.            (A) 비밀번호를 입력한다.            (S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다.            (S) 비밀번호가 일치할 시 ATM의 소속은행과 계좌의 소속은행을 비교하여 수수료를 계산한다.            (S) 수수료를 제외한 금액만큼 계좌의 보유현금을 증가시킨다.            (S) 계좌에 돈을 입금한 후 실행한 작업(입금)과 입금한 금액, 계좌의 잔액을 화면에 출력하고 로그에 저장한다.</p>
<b>Alternative Courses of Events</b>	6. 타행 계좌의 경우 수수료를 부과하고, 당행 계좌일 경우 수수료를 부과하지 않는다.
<b>Exceptional Courses of Events</b>	<p>2. 입금할 금액이 음수일 경우 Error            5. 비밀번호가 일치하지 않을 시 유효하지 않다고 알림을 보낸후 다시 3번 부터 진행한다.</p>

# 2-3. Stage 2040 \_Act 2041

Use Case	4. Withdraw
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: R4
Pre-Requisites	MediumCheck
Typical Courses of Events	<p>(A): Actor , (S):System</p> <p>(S) 화면에 금액 입력창을 띄운다.</p> <p>(A) 출금할 금액을 입력한다.</p> <p>(S) ATM의 소속은행과 계좌의 소속은행을 비교하여 수수료를 계산한다.</p> <p>(S) 입력받은 금액과 수수료를 더한 만큼의 금액이 계좌의 잔액보다 크지 않은지 확인한다.</p> <p>(S) 비밀번호 입력 창을 띄운다.</p> <p>(A) 비밀번호를 입력한다.</p> <p>(S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다.</p> <p>(S) 출금 금액과 수수료만큼 계좌의 보유현금을 감소시킨다.</p> <p>(S) 계좌에 돈을 출금한 후 실행한 작업(출금)과 출금한 금액, 계좌의 잔액을 화면에 출력하고 로그에 저장한다.</p>
Alternative Courses of Events	2. 타행 계좌의 경우 수수료를 부과하고, 당행 계좌일 경우 수수료를 부과하지 않는다.
Exceptional Courses of Events	<p>2. 출금할 금액이 음수이면 Error</p> <p>4. 출금하려는 금액과 수수료보다 잔액이 적을 시 Error</p> <p>7. 비밀번호가 일치하지 않을 시 유효하지 않다고 알림을 보낸후 다시 5번 부터 진행한다.</p>

# 2-3. Stage 2040 \_Act 2041

Use Case	5. Transfer
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: R5
Pre-Requisites	MediumCheck
Typical Courses of Events	<p>(A): Actor , (S):System            (S) 화면에 금액 입력 창을 띄운다.            (A) 송금할 금액을 입력한다.            (S) 화면에 계좌번호 입력 창을 띄운다.            (A) 송금할 계좌번호를 입력한다.            (S) 입력된 계좌 번호가 DB에 존재하는 계좌인지 확인한다.            (S) 송금할 계좌와 송금 받을 계좌의 소속은행을 비교하여 수수료를 계산한다.            (S) 입력받은 금액과 수수료를 더한 만큼의 금액이 송금할 계좌의 잔액보다 크지 않은지 확인한다.            (S) 비밀번호 입력 창을 띄운다.            (A) 비밀번호를 입력한다.            (S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다.            (S) 송금 금액과 수수료를 더한 만큼 송금할 계좌의 보유현금을 감소시킨다.            (S) 송금 금액만큼 송금받은 계좌의 보유현금을 증가시킨다.            (S) 계좌에 돈을 송금한 후 실행한 작업(송금)과 송금한 금액, 계좌의 잔액을 화면에 출력하고 로그에 저장한다.</p>
Alternative Courses of Events	6. 타행 계좌의 경우 수수료를 부과하고, 당행 계좌일 경우 수수료를 부과하지 않는다.
Exceptional Courses of Events	<p>2. 송금할 금액이 음수일 경우 Error            5. 유효하지 않은 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력을 받는다.(3번부터 다시 진행한다.)            7. 송금하려는 금액과 수수료보다 잔액이 적을 시 Error            10. 비밀번호가 일치하지 않을 시 유효하지 않다고 알림을 보낸후 다시 8번 부터 진행한다.</p>

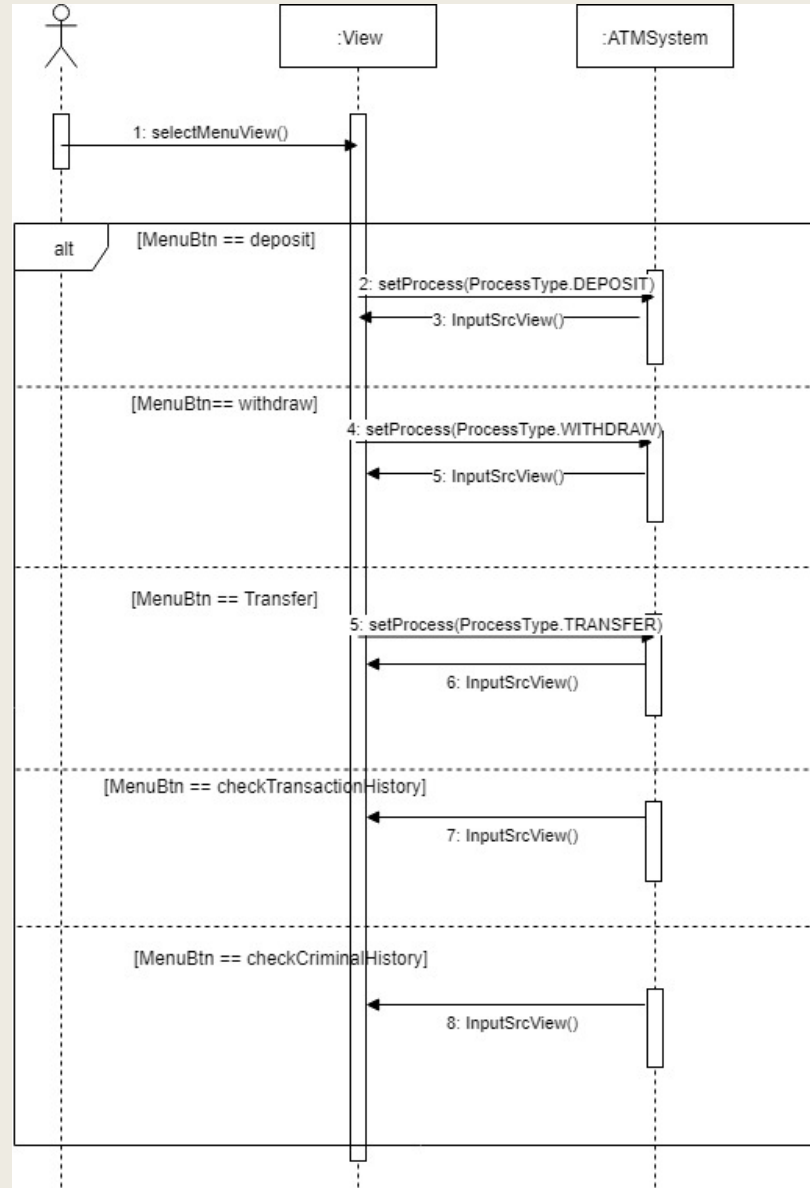
# 2-3. Stage 2040 \_Act 2041

Use Case	6. CheckTransactionHistory
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: R6
Pre-Requisites	MediumCheck
Typical Courses of Events	(A): Actor , (S):System (S) 비밀번호 입력창을 띄운다. (A) 비밀번호를 입력한다. (S) 입력받은 비밀번호가 MediumCheck에서 입력받은 계좌의 비밀번호와 일치하는지 확인한다. (S) MediumCheck를 통해 입력받은 계좌의 거래내역 log파일을 불러와 화면에 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. 비밀번호가 일치하지 않을 시 1번 부터 다시 진행한다.

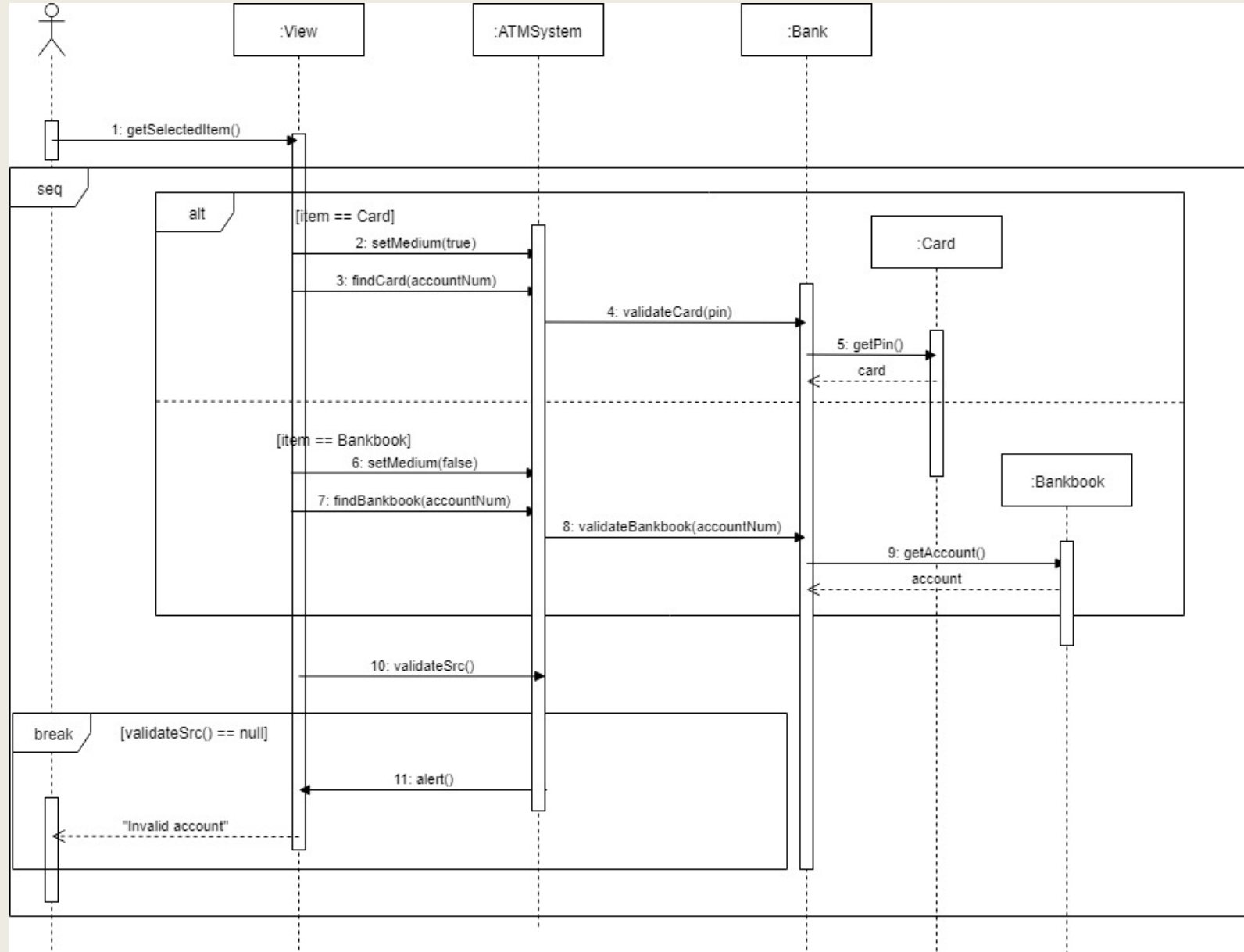
# 2-3. Stage 2040 \_Act 2041

Use Case	7. CheckCriminalHistory
Actor	User
Purpose	As in the business use case
Overview	As in the business use case
Type	Primary and Essential
Cross Reference	Use Cases: R7
Pre-Requisites	SelectMenu
Typical Courses of Events	(A): Actor , (S):System (S) 계좌번호 입력창을 띄운다. (A) 계좌번호를 입력한다. (S) 입력된 계좌번호가 DB에 존재하는 계좌인지 확인한다. (S) 입력된 계좌에 대한 범죄 이력 log를 가져와 출력한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	3. 유효하지 않은 경우 계좌가 유효하지 않다고 알림을 보낸 후 다시 입력받는다(1번부터 다시 진행)

# 2-3. Stage 2040 \_Act 2044

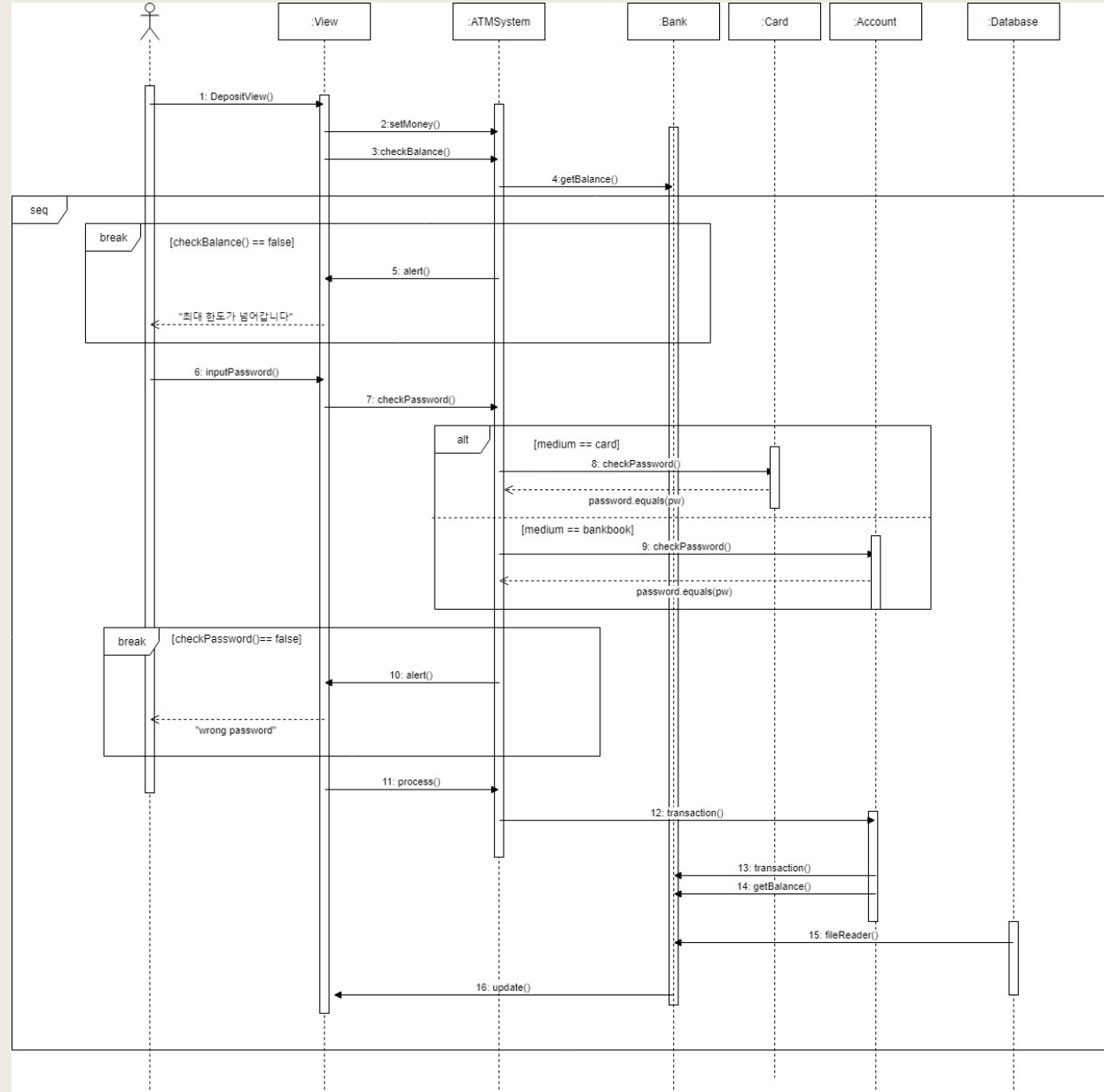


# 2-3. Stage 2040 \_Act 2044

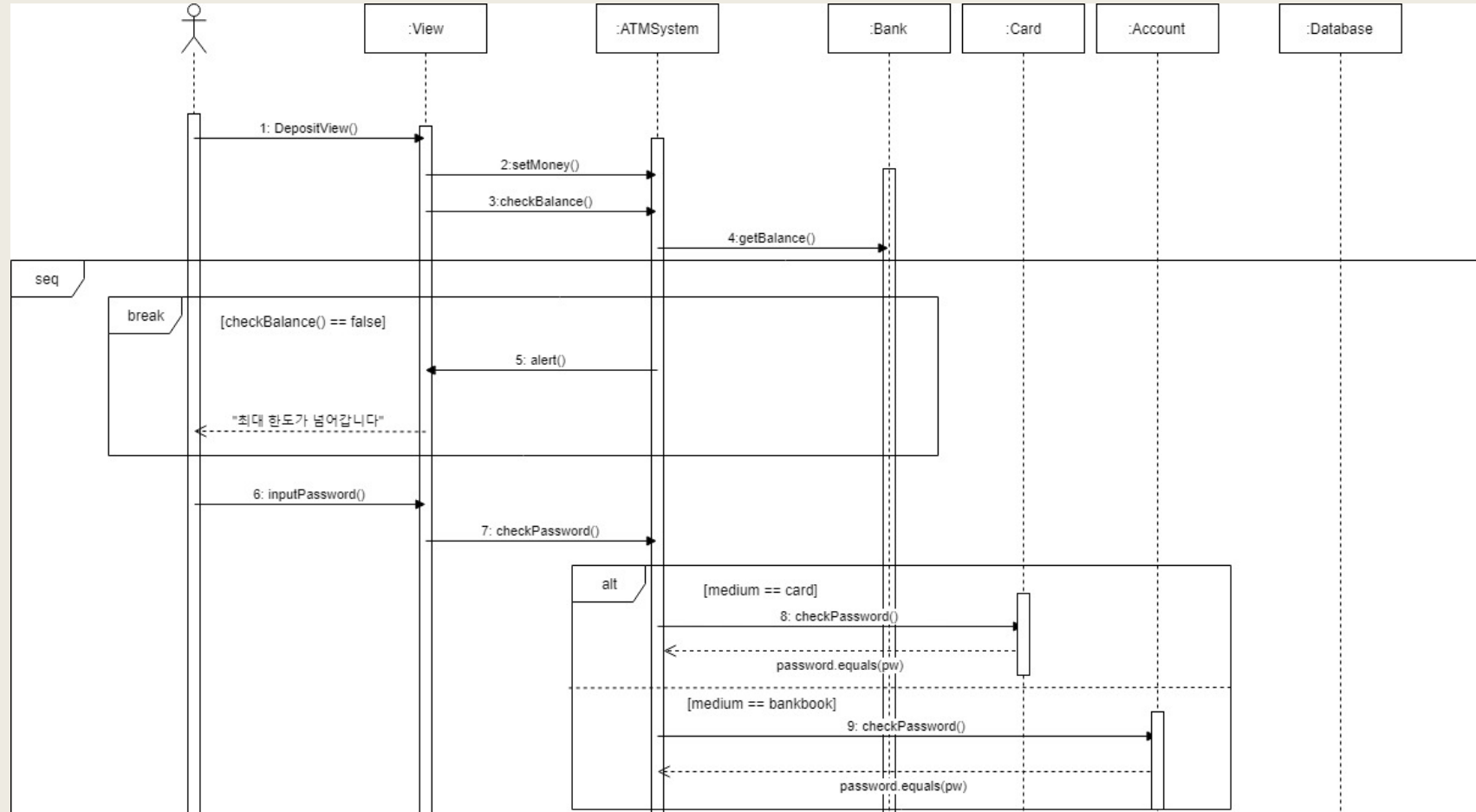




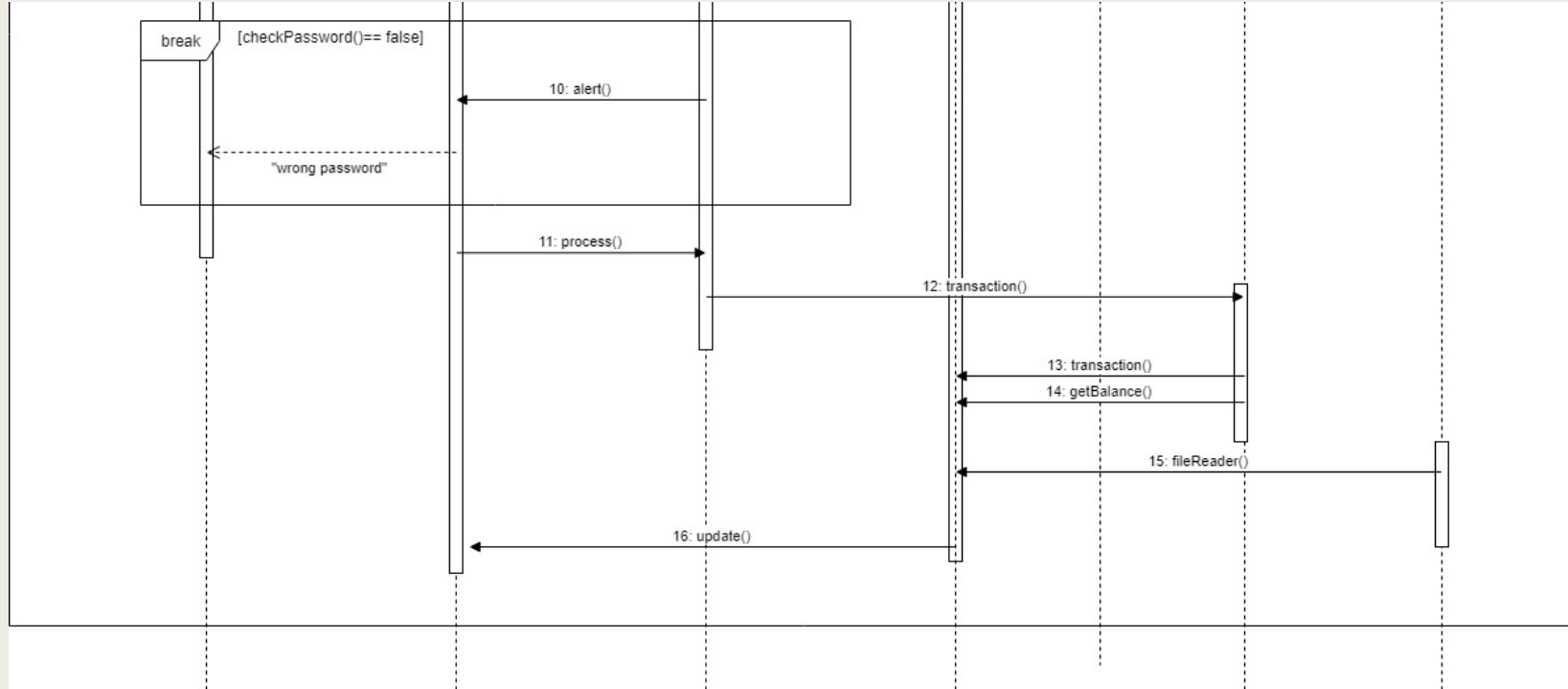
# 2-3. Stage 2040 \_Act 2044



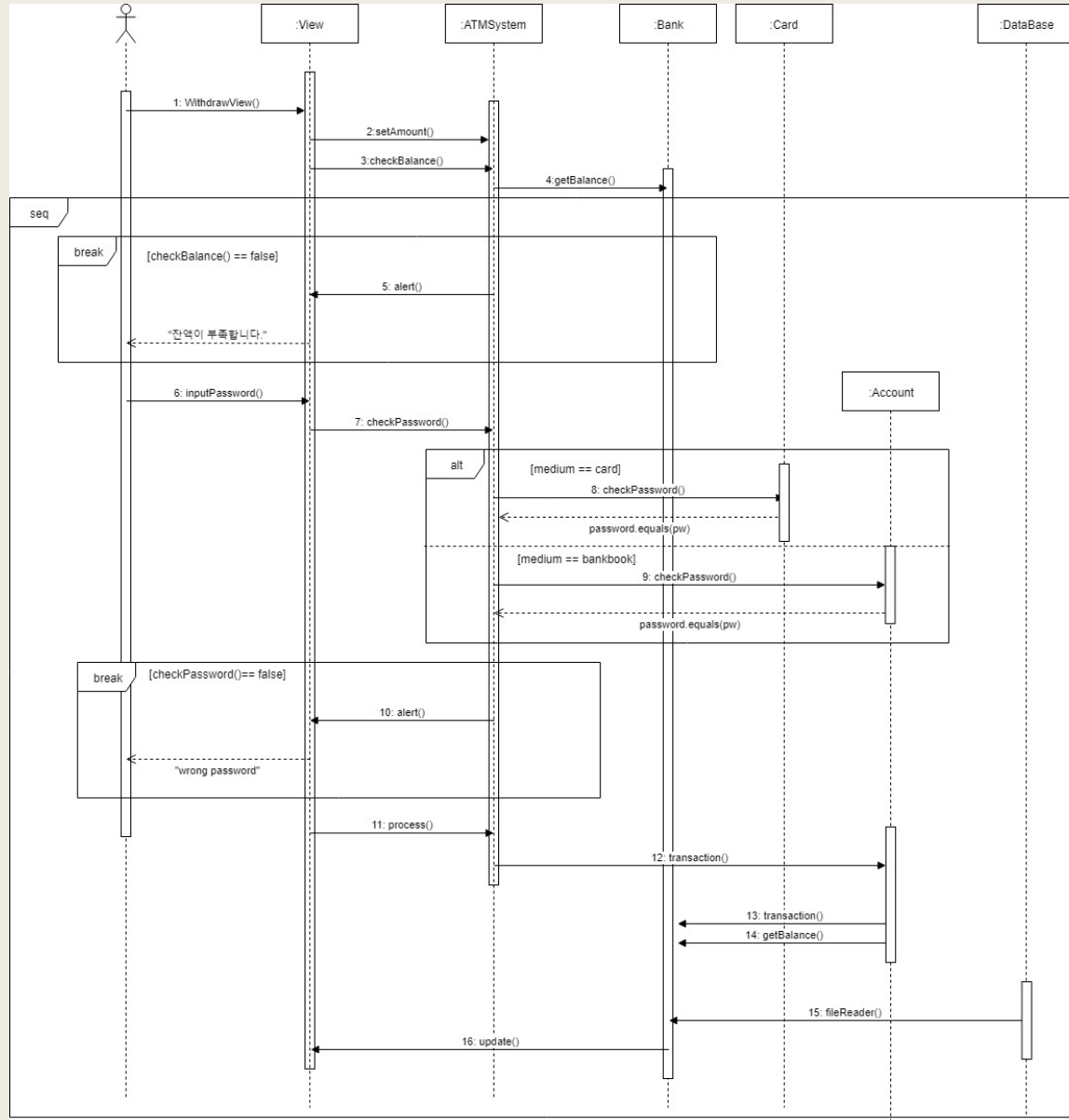
# 2-3. Stage 2040 \_Act 2044



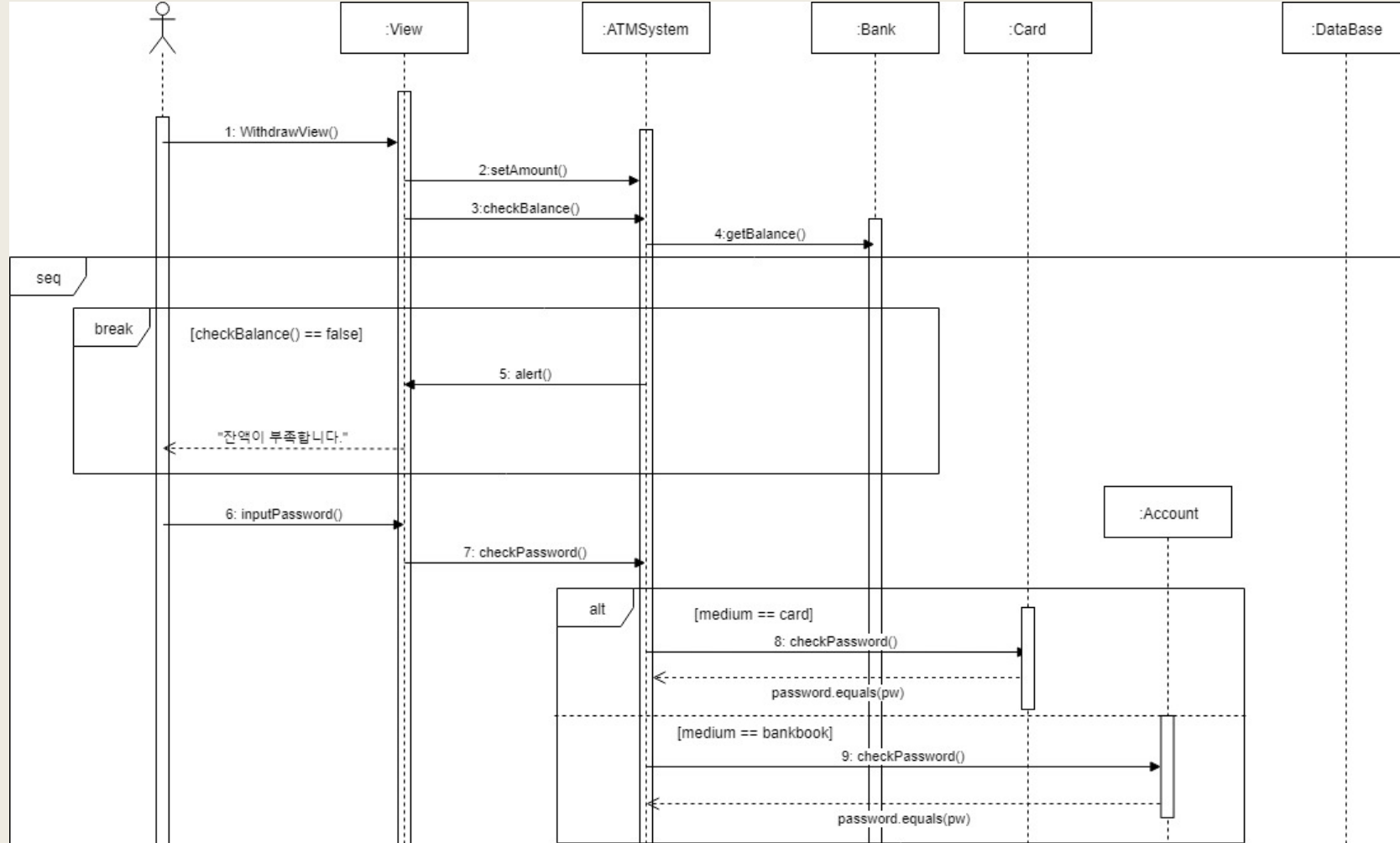
# 2-3. Stage 2040 \_Act 2044



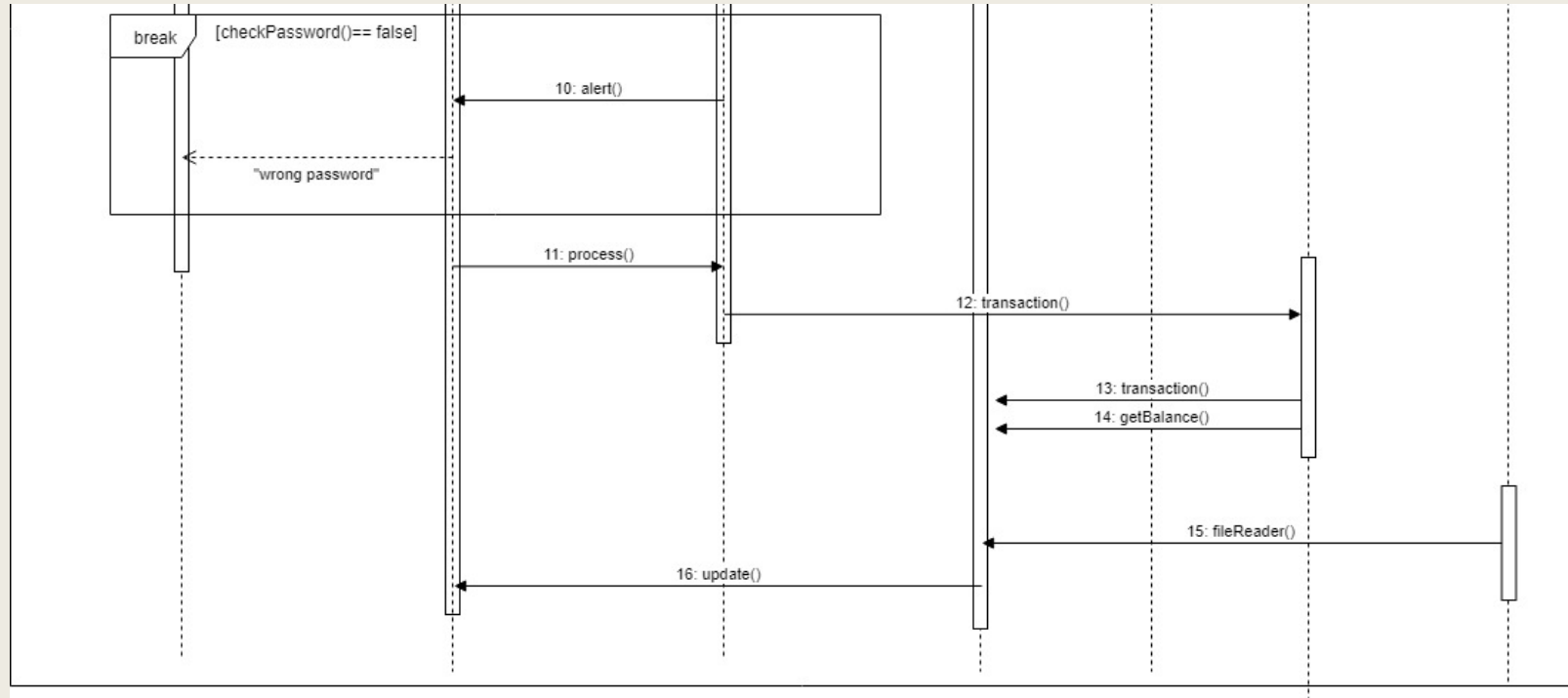
# 2-3. Stage 2040 \_Act 2044



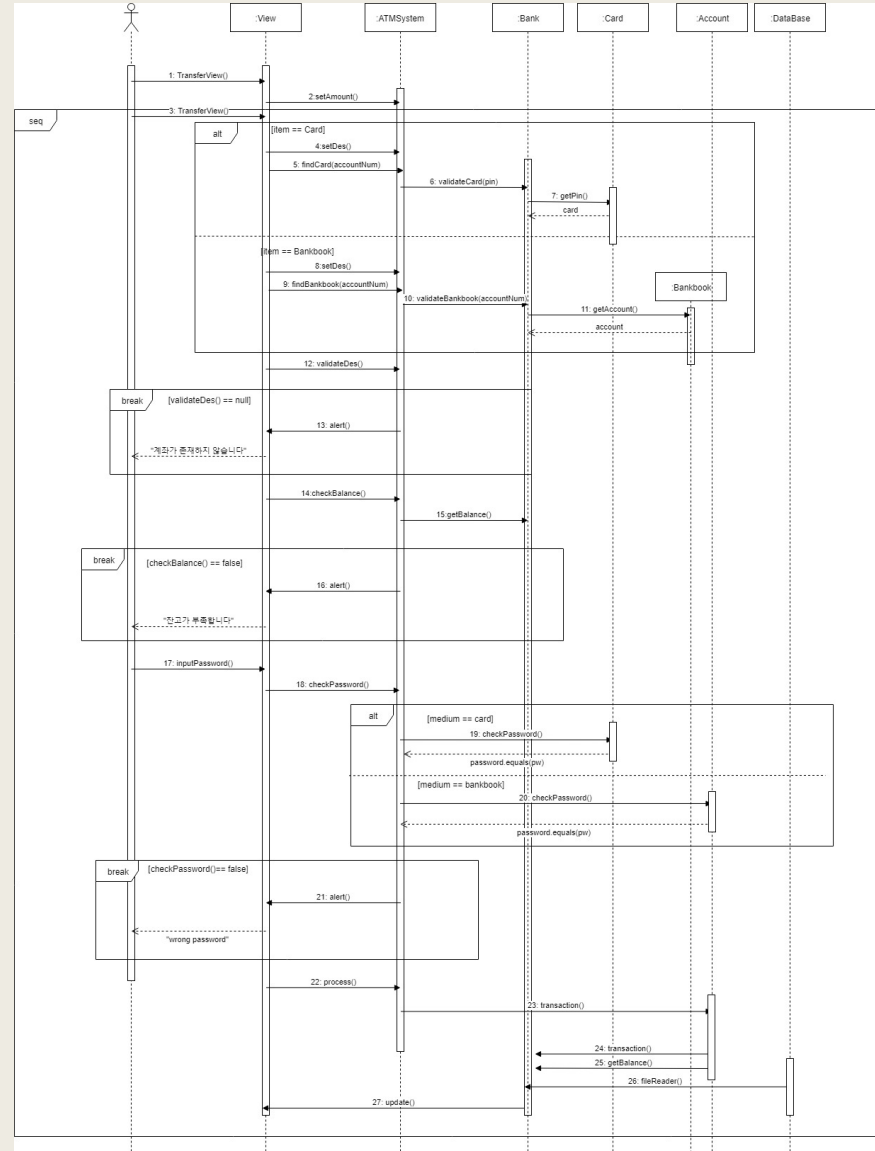
# 2-3. Stage 2040 \_Act 2044



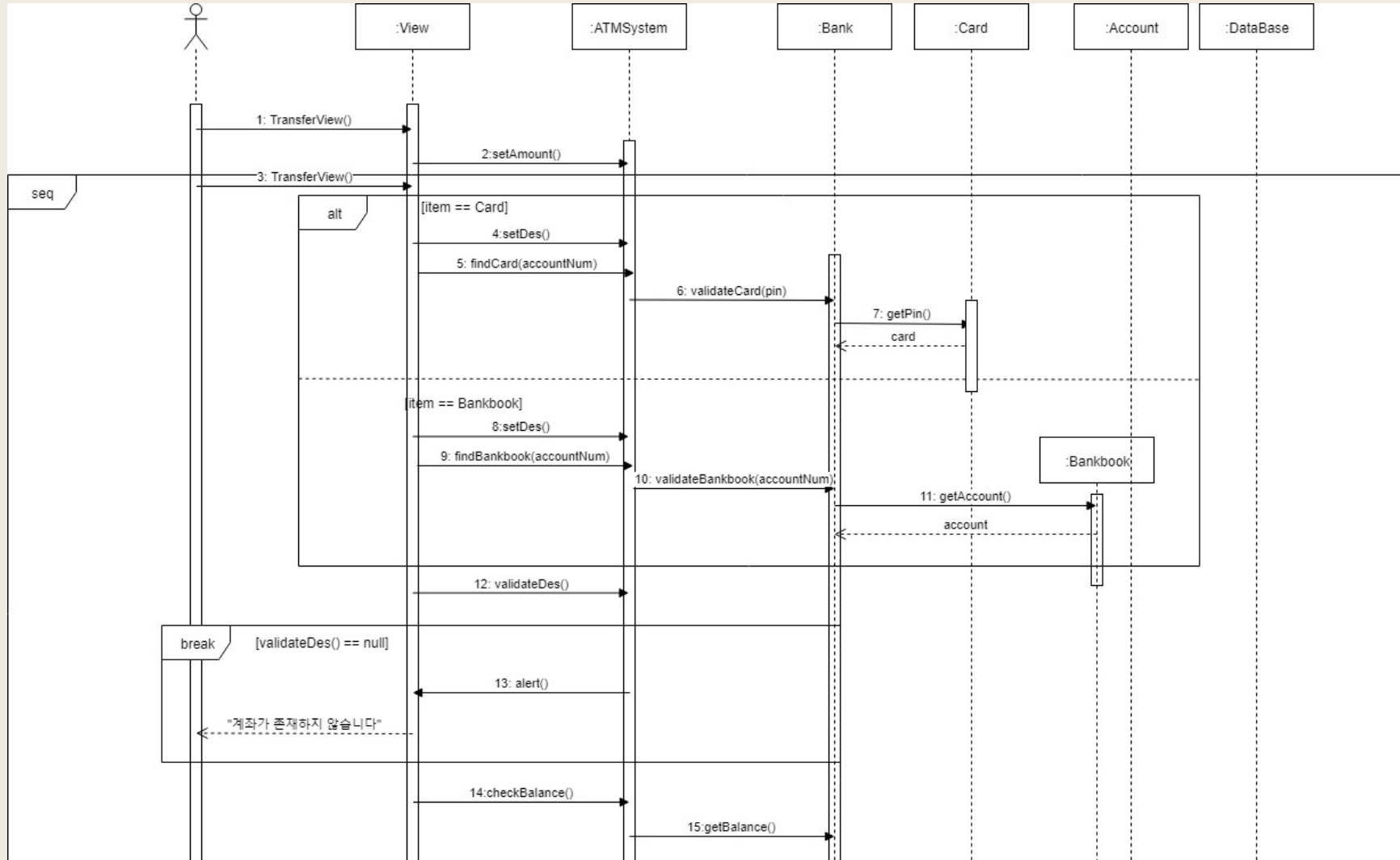
# 2-3. Stage 2040 \_Act 2044



# 2-3. Stage 2040 \_Act 2044

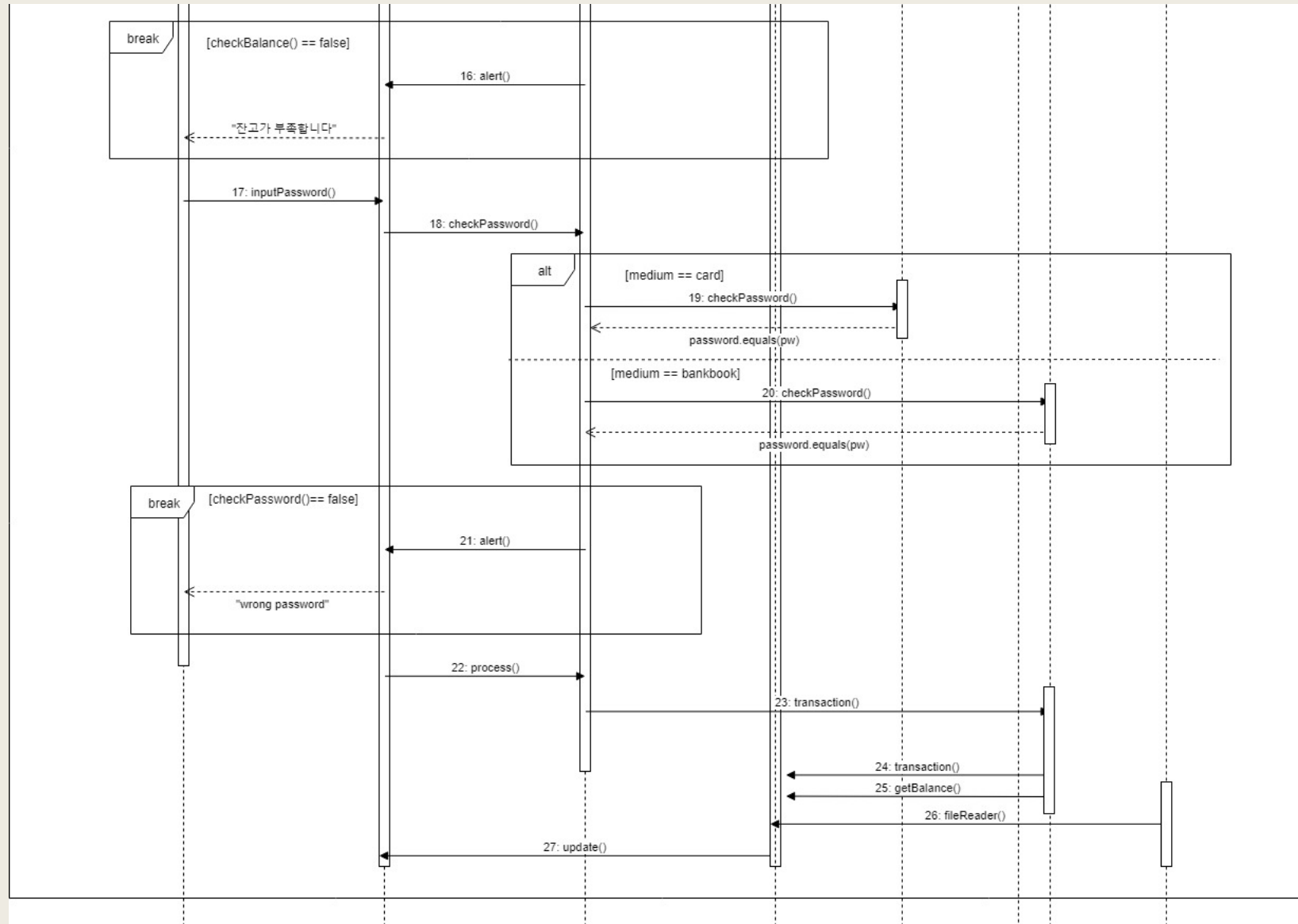


# 2-3. Stage 2040 \_Act 2044

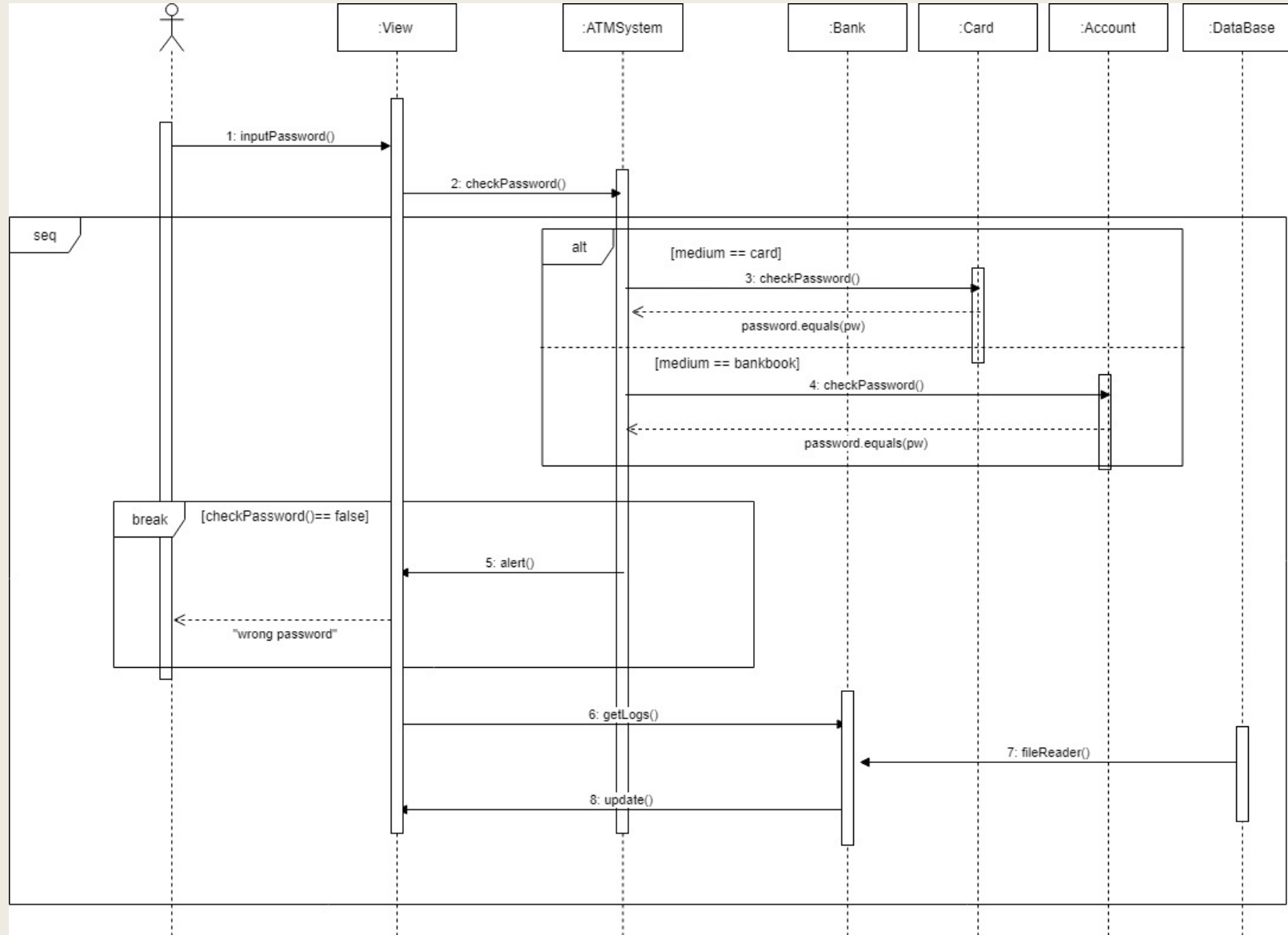




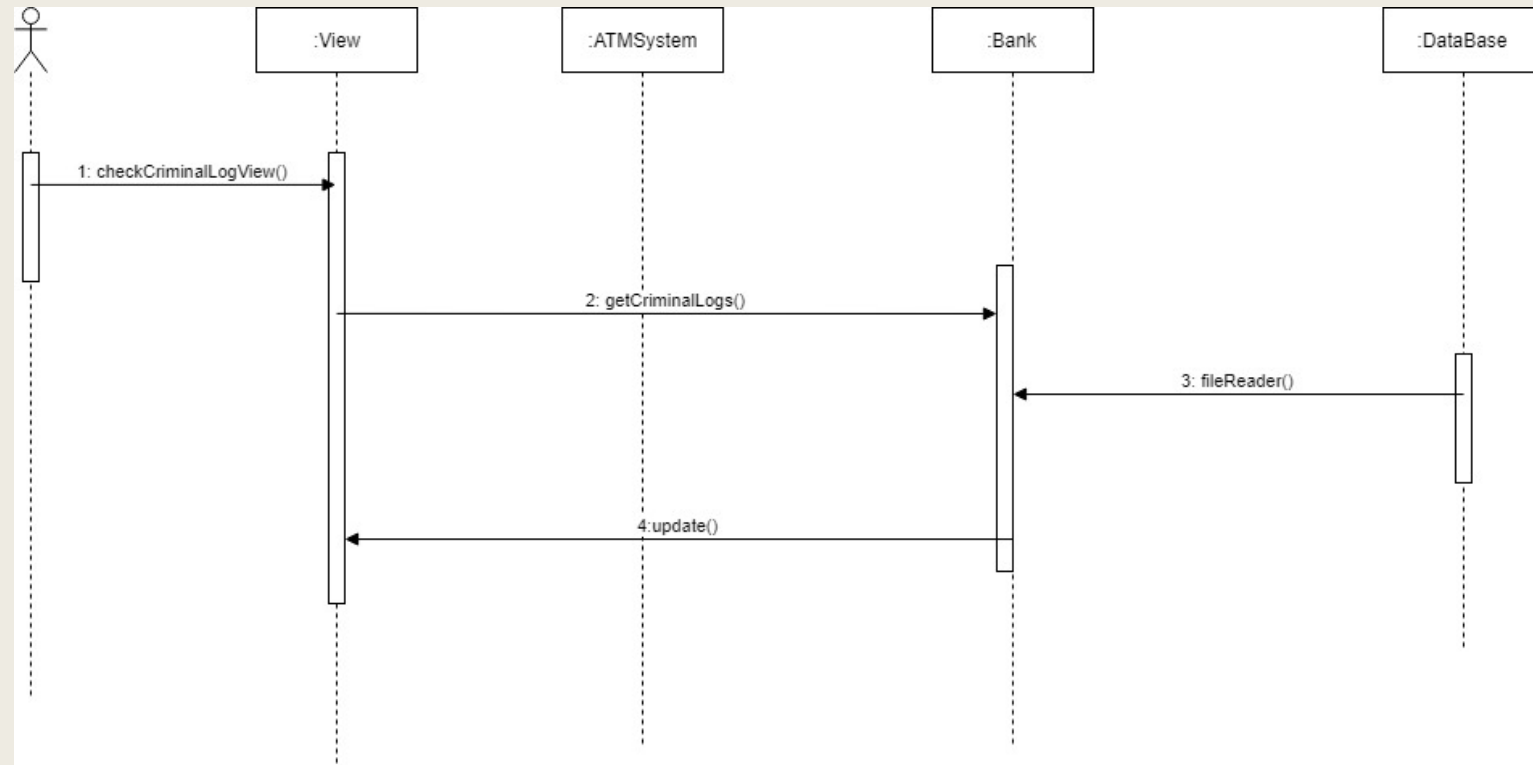
# 2-3. Stage 2040 \_Act 2044



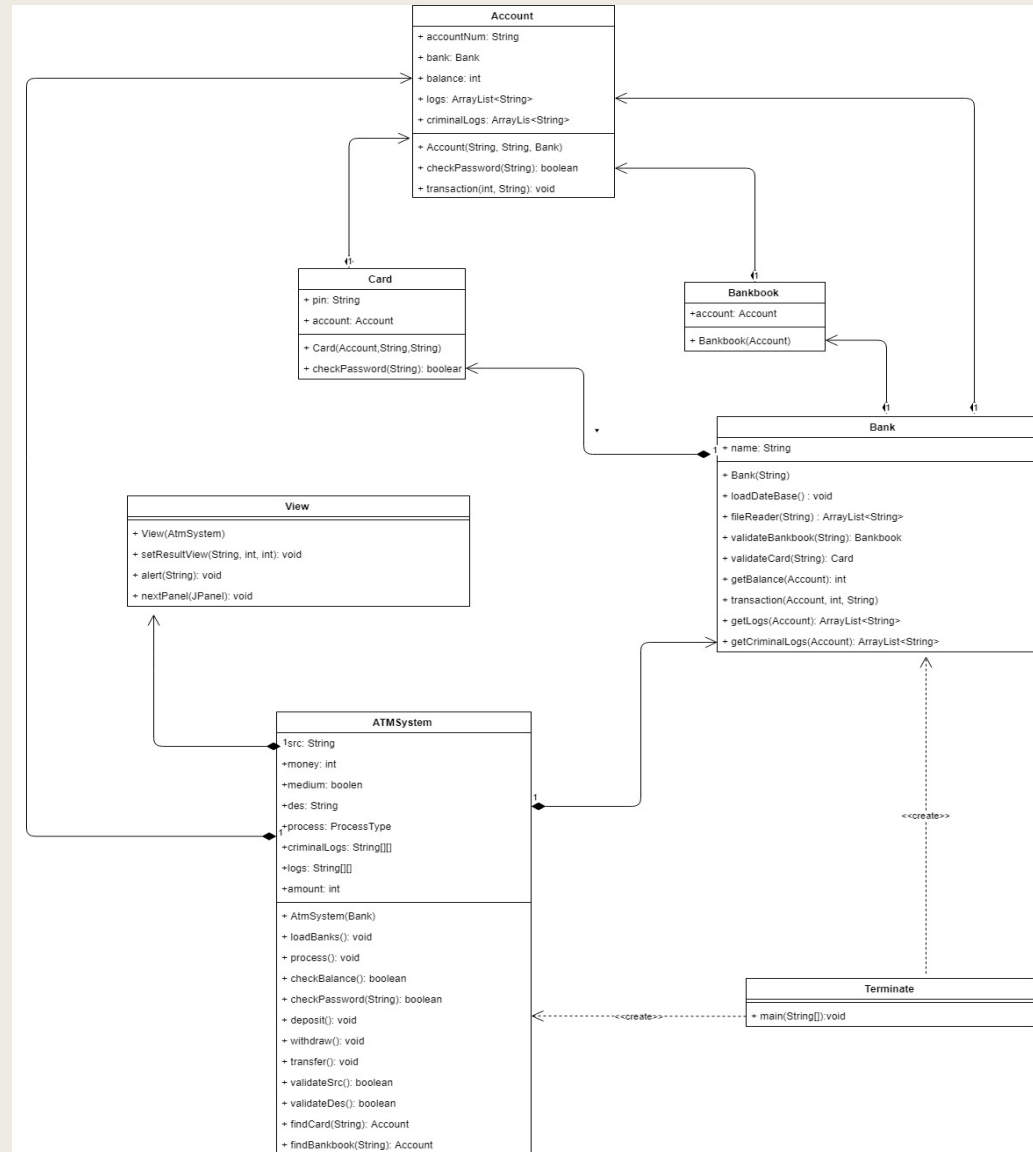
# 2-3. Stage 2040 \_Act 2044



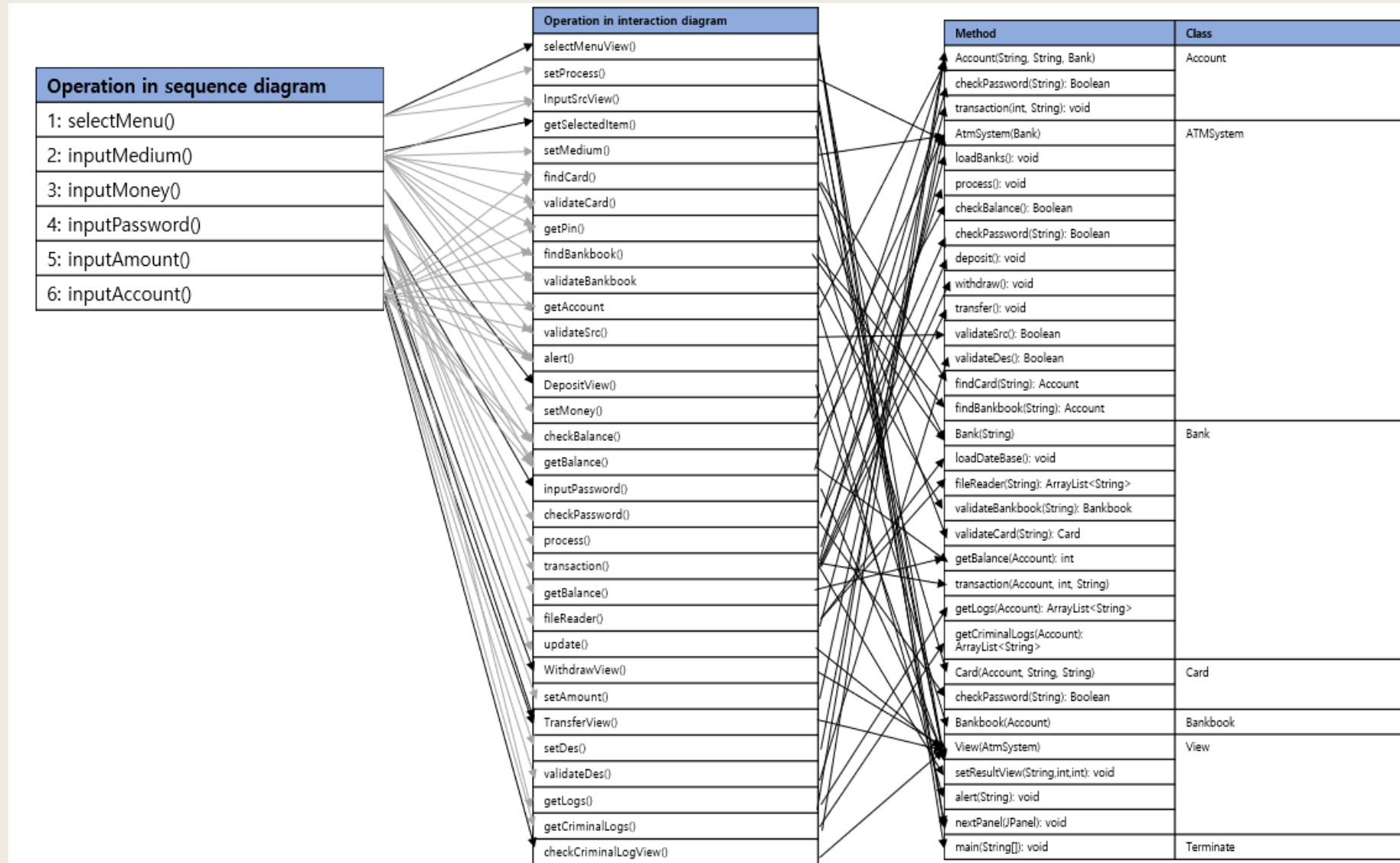
# 2-3. Stage 2040 \_Act 2044



# 2-3. Stage 2040 \_Act 2045



# 2-3. Stage 2040 \_Act 2046



# 2-4. Stage 2050 \_Act 2051

## 1) Account Class

<b>Type</b>	class
<b>Name</b>	Account
<b>Purpose</b>	계좌 정보를 모아두는 클래스
<b>Overview(class)</b>	Attribute : String accountNum, Bank bank , String password Method : Account(), getBank(), getAccountNum(), getBalance(), checkPassword(), transaction(), getLogs(), getCriminalLogs()
<b>CrossReference</b>	Use Case : R2, R3, R4, R5, R6, R7
<b>Exceptional Courses of Events</b>	N/A

# 2-4. Stage 2050 \_Act 2051

## 1) Account Class

Type	Method
Name	Account()
Purpose	Account의 객체 생성한다.
CrossReference	Use Case :R2, R3, R4, R5, R6, R7
Input(Method)	String accountNum, Bank bank
Output(Method)	-
Abstract Operation(Method)	Account의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getBank()
Purpose	Account의 소속 bank를 확인한다.
CrossReference	Use Case R3, R4, R5
Input(Method)	-
Output(Method)	Bank bank
Abstract Operation(Method)	account.bank를 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 1) Account Class

Type	Method
Name	getAccountNum
Purpose	Account의 계좌번호를 확인한다.
CrossReference	Use Case R2, R5
Input(Method)	-
Output(Method)	String accountNum
Abstract Operation(Method)	account.accountNum을 반환한다.
Exceptional Courses of Events	

Type	Method
Name	getBalance
Purpose	계좌 잔액을 확인한다.
CrossReference	Use Case R2, R3, R4, R5, R6
Input(Method)	-
Output(Method)	int bank.getBalance(account: this)
Abstract Operation(Method)	bank DB에서 balance를 불러와 반환한다.
Exceptional Courses of Events	-



# 2-4. Stage 2050 \_Act 2051

## 1) Account Class

Type	Method
Name	checkPassword()
Purpose	비밀번호가 유효한지 확인한다.
CrossReference	Use Case R3, R4, R5
Input(Method)	String pw
Output(Method)	boolean bank.checkPassword(account: this, pw)
Abstract Operation(Method)	bankDB에서 비밀번호 일치여부를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	transaction()
Purpose	거래(입금/출금/송금)
CrossReference	Use Case R3, R4, R5
Input(Method)	int amount, Sting msg
Output(Method)	bank.transaction(account: this, amount, msg)
Abstract Operation(Method)	이 계좌에서 amount만큼 거래하고 log에 msg를 저장한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 1) Account Class

Type	Method
Name	getLogs()
Purpose	거래내역을 조회한다.
CrossReference	Use Case R6
Input(Method)	-
Output(Method)	ArrayList<String> bank.getLogs(this)
Abstract Operation(Method)	해당 계좌의 log를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getCriminalLogs()
Purpose	범죄이력을 조회한다.
CrossReference	Use Case R7
Input(Method)	-
Output(Method)	ArrayList<String> bank.getCriminalLogs(this)
Abstract Operation(Method)	계좌의 범죄 이력과 횟수를 반환한다,
Exceptional Courses of Events	

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

<b>Type</b>	class
<b>Name</b>	AtmSystem
<b>Purpose</b>	ATM 정보를 모아두는 클래스
<b>Overview(class)</b>	<p>enum MediumType{CARD, BANKBOOK}, enum ProcessType{DEPOSIT, WITHDRAW, TRANSFER}</p> <p>Attribute : Bank own, Account src, Account des, int amount, int money, int fee, ArrayList&lt;Bank&gt; banks, MediumType medium, ProcessType process, Card srcCard</p> <p>Method : AtmSystem(), loadBanks(), process(), checkBalance(), checkPassword(), deposit(), withdraw(), transfer(), getLogs(), getCriminalLogs(), setProcess(), setMedium(), setSrc(), validateSrc(), setDes(), validateDes(), setAmount(), setMoney(), findCard(), findBankbook()</p>
<b>CrossReference</b>	Use Case R1, R2, R3, R4, R5, R6, R7
<b>Exceptional Courses of Events</b>	N/A

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	AtmSystem()
Purpose	AtmSystem의 객체를 생성한다.
CrossReference	Use Case R1, R2, R3, R4, R5, R6, R7
Input(Method)	Bank own
Output(Method)	-
Abstract Operation(Method)	AtmSystem의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	loadBanks()
Purpose	은행 목록에 이용가능한 은행을 추가한다.
CrossReference	Use Case R1, R2, R3, R4, R5, R6, R7
Input(Method)	Bank own
Output(Method)	-
Abstract Operation(Method)	banks 배열에 존재하는 Bank를 추가한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	process()
Purpose	이용할 서비스를 선택한다.
CrossReference	Use Case R 1
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	button 입력에 따라 선택한 서비스를 실행한다.
Exceptional Courses of Events	-

Type	Method
Name	checkBalance()
Purpose	계좌 잔고가 출금 또는 송금할 금액보다 많은지 확인한다.
CrossReference	Use Case R4, R5
Input(Method)	-
Output(Method)	boolean (amount + fee) < arc.getBalance()
Abstract Operation(Method)	amount + fee와 src.balance를 비교한 결과를 반환한다,
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	checkPassword
Purpose	비밀번호를 확인한다.
CrossReference	Use Case R3, R4, R5, R6
Input(Method)	String pw
Output(Method)	boolean src.checkPassword(pw);
Abstract Operation(Method)	입력한 비밀번호가 src의 비밀번호와 일치하는지 확인한다.
Exceptional Courses of Events	-

Type	Method
Name	deposit()
Purpose	src계좌에 입금한다.
CrossReference	Use Case R3
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	계좌의 balance를 amount - fee만큼 증가시키고 log에 msg와 해당 거래내용을 업데이트한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	withdraw()
Purpose	src에서 출금한다.
CrossReference	Use Case R4
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	계좌의 balance를 amount + fee만큼 감소시키고 log에 msg와 해당 거래내용을 업데이트한다.
Exceptional Courses of Events	-

Type	Method
Name	transfer()
Purpose	src에서 des로 송금한다.
CrossReference	Use Case R5
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	src 계좌의 balance를 amount + fee만큼 감소시키고 log에 msg와 해당 거래내용을 업데이트한다 des 계좌의 balance를 amount 만큼 증가시키고 log에 msg와 해당 거래내용을 업데이트한다
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	getLogs()
Purpose	계좌 거래내역을 조회한다.
CrossReference	Use Case R6
Input(Method)	-
Output(Method)	String[][] contents
Abstract Operation(Method)	해당 계좌의 거래내역 contents를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getCriminalLogs()
Purpose	계좌 범죄내역을 조회한다.
CrossReference	Use Case R7
Input(Method)	-
Output(Method)	String[][] contents
Abstract Operation(Method)	해당 계좌의 범죄내역 contents를 반환한다.
Exceptional Courses of Events	-



# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	setProcess
Purpose	이용할 서비스를 선택한다.
CrossReference	Use Case R1
Input(Method)	ProcessType process
Output(Method)	-
Abstract Operation(Method)	ProcessType process를 입력받아 AtmSystem의 process를 세팅한다.
Exceptional Courses of Events	-

Type	Method
Name	setMedium()
Purpose	현재 이용할 매체를 선택한다.
CrossReference	Use Case R2
Input(Method)	boolean isCard
Output(Method)	-
Abstract Operation(Method)	isCard가 true이면 ATMSystem의 medium을 MediumType.CARD로, false이면 MediumType.BANKBOOK으로 세팅한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	setSrc()
Purpose	이용할 계좌를 세팅한다.
CrossReference	Use Case R3, R4, R5, R6
Input(Method)	String accountNum
Output(Method)	-
Abstract Operation(Method)	매체에 따라 findCard(accountNum)또는 findBankBook(accountNum)의 결과를 src에 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	validateSrc()
Purpose	src계좌의 유효성을 확인한다.
CrossReference	Use Case R3, R4, R5, R6, R7
Input(Method)	-
Output(Method)	boolean src != null
Abstract Operation(Method)	src계좌의 유효성에 따라 boolean 값을 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	setDes
Purpose	송금 대상 계좌를 세팅한다.
CrossReference	Use Case R5
Input(Method)	String accountNum
Output(Method)	-
Abstract Operation(Method)	매체에 따라 findCard(accountNum)또는 findBankBook(accountNum)의 결과를 des에 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	validateDes()
Purpose	des계좌의 유효성을 확인한다.
CrossReference	Use Case R5
Input(Method)	-
Output(Method)	boolean des != null
Abstract Operation(Method)	des계좌의 유효성에 따라 boolean 값을 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	setAmount()
Purpose	서비스(출금 또는 송금) 이용 금액을 세팅한다.
CrossReference	Use Case R4, R5
Input(Method)	int amount
Output(Method)	-
Abstract Operation(Method)	this.amount에 amount값을 저장한다.
Exceptional Courses of Events	.-

Type	Method
Name	setMoney()
Purpose	입금 금액을 세팅한다.
CrossReference	Use Case R3
Input(Method)	int money
Output(Method)	-
Abstract Operation(Method)	this.money에 money값을 저장한다.
Exceptional Courses of Events	.-

# 2-4. Stage 2050 \_Act 2051

## 2) ATMSystem Class

Type	Method
Name	findCard()
Purpose	각 bank DB에 card와 연결된 account가 존재하는지 찾는다.
CrossReference	Use Case R2, R7
Input(Method)	String pin
Output(Method)	Account ret.getAccount
Abstract Operation(Method)	각 bank DB 에서 pin을 이용해 카드와 연결된 계좌의 유효성 유무를 검색하고 존재하면 ret.getAccount을 반환한다.
Exceptional Courses of Events	어느 bank DB에도 존재하지 않는 pin의 경우 null을 반환한다.

Type	Method
Name	findBankbook()
Purpose	각 bank DB에 bankbook과 연결된 account 가 존재하는지 찾는다.
CrossReference	Use Case R2, R7
Input(Method)	String accountNum
Output(Method)	Account ret.getAccount
Abstract Operation(Method)	각 bank DB 에서 accountNum을 이용해 통장과 연결된 계좌의 유효성 유무를 검색하고 존재하면 ret.getAccount을 반환한다.
Exceptional Courses of Events	어느 bank DB에도 존재하지 않는 accountNum의 경우 null을 반환한다.

# 2-4. Stage 2050 \_Act 2051

## 3) Bank Class

<b>Type</b>	class
<b>Name</b>	Bank
<b>Purpose</b>	Bank 정보를 모아두는 클래스
<b>Overview(class)</b>	Attribute : String name, String path, ArrayList<Card> cards, ArrayList<Bankbook> bankbooks  Method : Bank(), loadDataBase() fileReader(), validateBankbook(), validateCard(), getBalance(), transaction(), getLogs(), getCriminalLogs(), getName()
<b>CrossReference</b>	Use Case R2, R3, R4, R5, R6, R7
<b>Exceptional Courses of Events</b>	N/A

# 2-4. Stage 2050 \_Act 2051

## 3) Bank Class

Type	Method
Name	Bank()
Purpose	Bank의 객체를 생성한다.
CrossReference	Use Case R2, R3, R4, R5, R6, R7
Input(Method)	String name
Output(Method)	-
Abstract Operation(Method)	Bank의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	loadDataBase()
Purpose	은행의 DB를 읽어온다.
CrossReference	Use Case R2
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	Bank의 DB 내에 저장되어 있는 account와 card, bankbook의 데이터를 불러온다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 3) Bank Class

Type	Method
Name	fileReader()
Purpose	파일로 부터 정보를 읽어온다.
CrossReference	Use Case R6, R7
Input(Method)	String path
Output(Method)	AttrayList<String> lines
Abstract Operation(Method)	파일로 저장된 log를 읽어와 반환한다.
Exceptional Courses of Events	path가 올바르지 않을시 Error

Type	Method
Name	validateBankbook()
Purpose	bankbook이 유효한지 확인한다.
CrossReference	Use Case R2, R7
Input(Method)	String accountNum
Output(Method)	Bankbook bankbook
Abstract Operation(Method)	bank DB의 통장중 accountNum에 해당하는 통장이 존재하는지 확인하고 존재하면 해당 bankbook을 반환한다.
Exceptional Courses of Events	bank DB의 통장중 accountNum에 해당하는 통장이 존재하는지 확인하고 존재하지 않으면 null을 반환한다.



# 2-4. Stage 2050 \_Act 2051

## 3) Bank Class

Type	Method
Name	validateCard()
Purpose	Card가 유효한지 확인한다.
CrossReference	Use Case R2
Input(Method)	String pin
Output(Method)	Card card
Abstract Operation(Method)	bank DB의 통장중 accountNum에 해당하는 카드가 존재하는지 확인하고 존재하면 해당 card를 반환한다.
Exceptional Courses of Events	bank DB의 통장중 accountNum에 해당하는 카드가 존재하는지 확인하고 존재하지 않으면 null을 반환한다.

Type	Method
Name	getBalance()
Purpose	계좌의 잔액을 확인한다.
CrossReference	Use Case R3, R4, R5, R6
Input(Method)	Account account
Output(Method)	int balance
Abstract Operation(Method)	account의 balance를 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 3) Bank Class

Type	Method
Name	transaction()
Purpose	거래 서비스를 실행한다.
CrossReference	Use Case R3, R4, R5, R6, R7
Input(Method)	Account account, int amount, String msg
Output(Method)	-
Abstract Operation(Method)	account의 log파일에 거래 시간, 거래량, 거래 유형을 기록한다.
Exceptional Courses of Events	-

Type	Method
Name	getLogs()
Purpose	거래내역을 출력한다.
CrossReference	Use Case R6
Input(Method)	Account account
Output(Method)	ArrayList<String> this.fileReader(path)
Abstract Operation(Method)	account의 거래내역 로그파일을 읽어와 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 3) Bank Class

Type	Method
Name	getCriminalLogs()
Purpose	범죄이력을 출력한다.
CrossReference	Use Case R 7
Input(Method)	Account account
Output(Method)	ArrayList<String> this.fileReader(path)
Abstract Operation(Method)	account의 범죄 이력 로그파일을 읽어와 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getName()
Purpose	은행의 이름을 확인한다.
CrossReference	Use Case R3, R4, R5
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	Bank의 name을 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 4) Bankbook Class

<b>Type</b>	class
<b>Name</b>	Bankbook
<b>Purpose</b>	Bankbook 정보를 모아두는 클래스
<b>Overview(class)</b>	Attribute : account Method : Bankbook(), getAccount()
<b>CrossReference</b>	Use Case R2, R5
<b>Exceptional Courses of Events</b>	N/A

# 2-4. Stage 2050 \_Act 2051

## 4) Bankbook Class

Type	Method
Name	Bankbook()
Purpose	Bankbook의 객체를 생성한다.
CrossReference	Use Case R2
Input(Method)	Account account
Output(Method)	-
Abstract Operation(Method)	Bankbook의 객체를 생성하고, this.account를 input된 account로 설정한다.
Exceptional Courses of Events	-

Type	Method
Name	getAccount()
Purpose	통장과 연결된 계좌를 확인한다.
CrossReference	Use Case R2
Input(Method)	-
Output(Method)	Account account
Abstract Operation(Method)	Bankbook의 account를 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

## 5) Card Class

<b>Type</b>	class
<b>Name</b>	Card
<b>Purpose</b>	Card 정보를 모아두는 클래스
<b>Overview(class)</b>	Attribute : Account account, String pin, String password Method : Card(), getAccount, getPin, checkPassword
<b>CrossReference</b>	Use Case R2
<b>Exceptional Courses of Events</b>	N/A

# 2-4. Stage 2050 \_Act 2051

## 5) Card Class

Type	Method
Name	Card()
Purpose	Card의 객체를 생성한다.
CrossReference	Use Case R2
Input(Method)	Account account String pin String password
Output(Method)	-
Abstract Operation(Method)	Bankbook의 객체를 생성하고, this.account를 input된 account로, this.cardNumber을 input된 cardNumber로 설정한다.
Exceptional Courses of Events	-

Type	Method
Name	getAccount()
Purpose	카드와 연결된 계좌를 확인한다.
CrossReference	Use Case R2
Input(Method)	-
Output(Method)	Account account
Abstract Operation(Method)	Card의 account를 반환한다.
Exceptional Courses of Events	-

# 2-4. Stage 2050 \_Act 2051

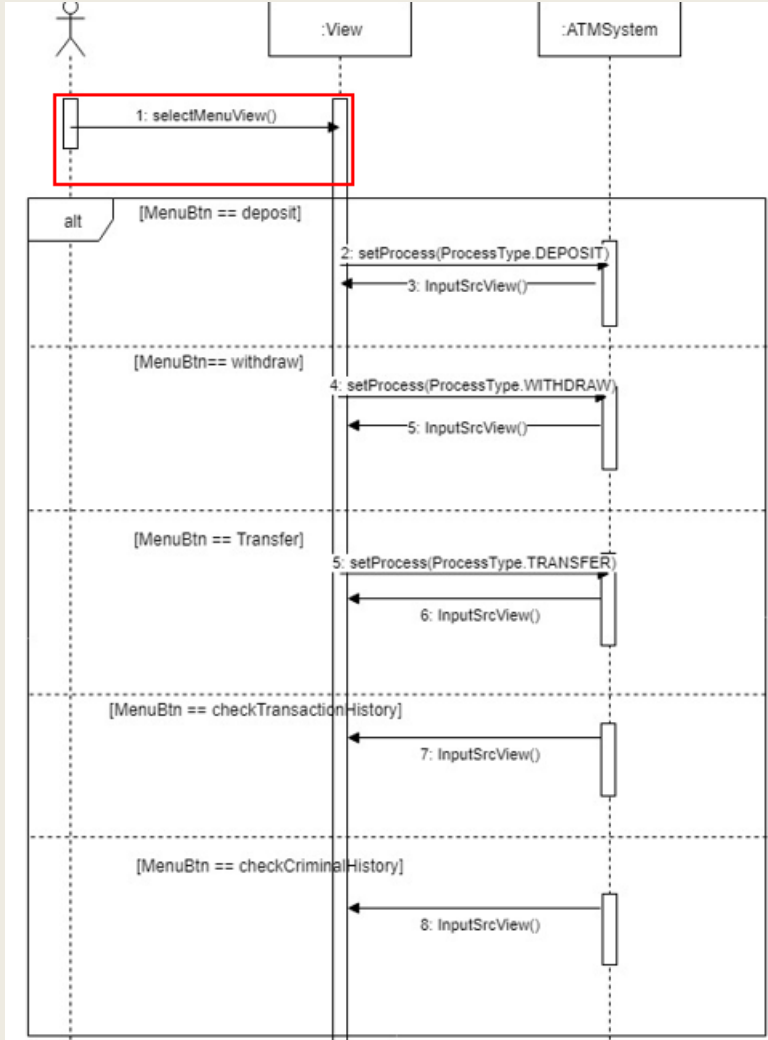
## 5) Card Class

Type	Method
Name	getPin()
Purpose	카드번호를 확인한다.
CrossReference	Use Case R2
Input(Method)	-
Output(Method)	String pin
Abstract Operation(Method)	Card의 pin을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	checkPassword
Purpose	비밀번호가 맞는지 확인한다.
CrossReference	Use Case R2
Input(Method)	String pw
Output(Method)	this.password.equals(pw)
Abstract Operation(Method)	비밀번호 일치여부를 boolean값으로 반환한다.
Exceptional Courses of Events	-

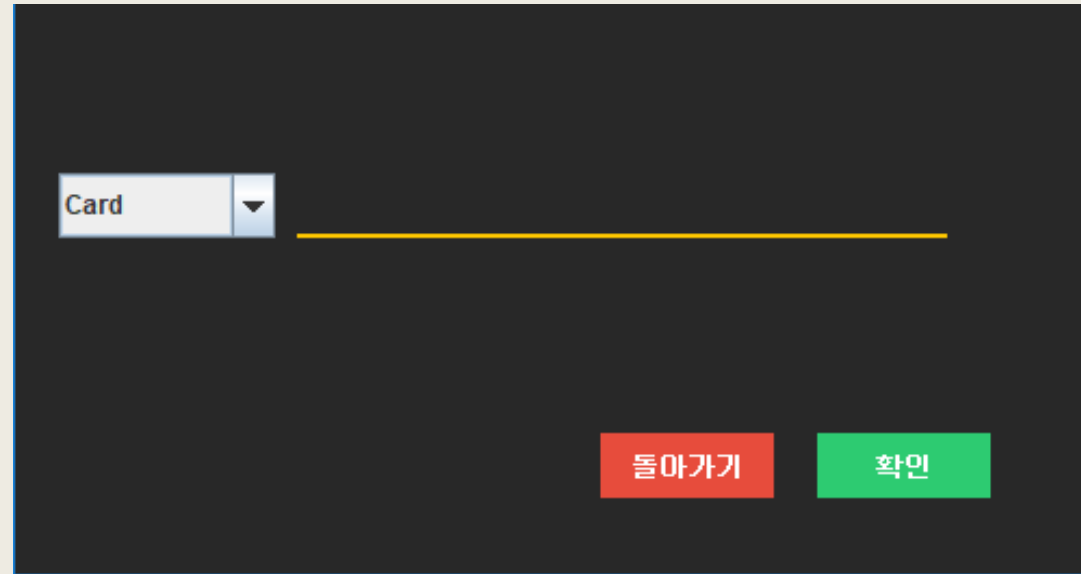
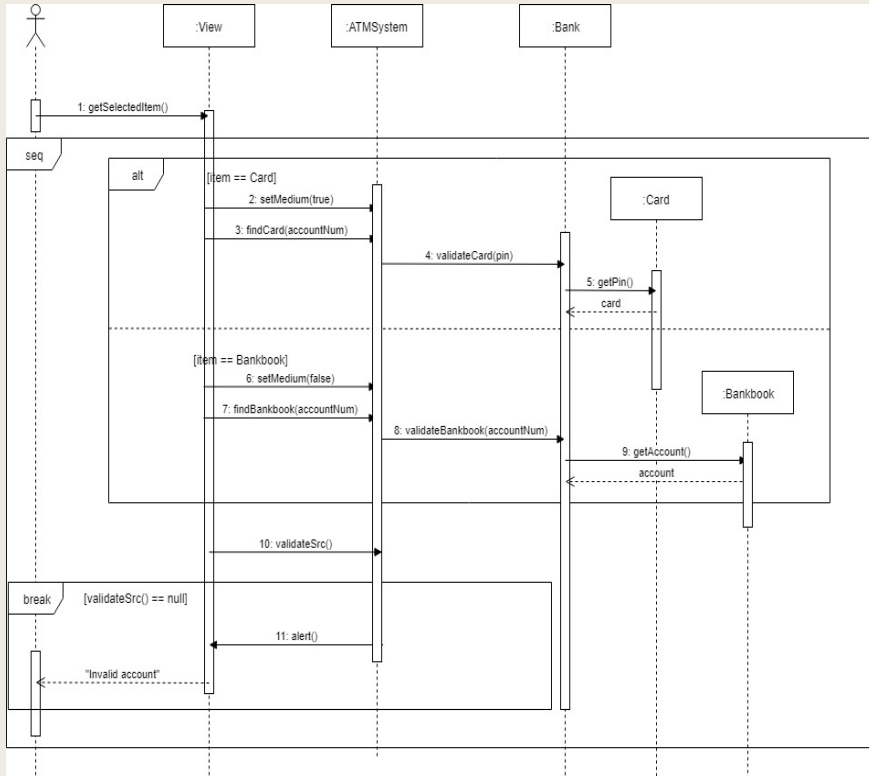


# 2-4. Stage 2050 \_Act 2052



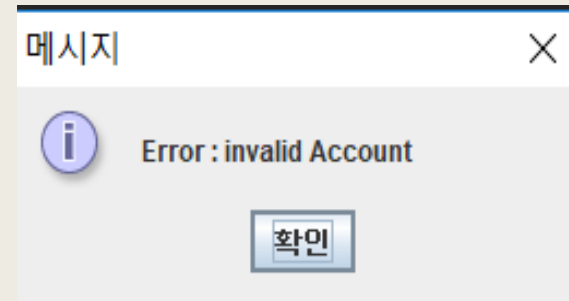
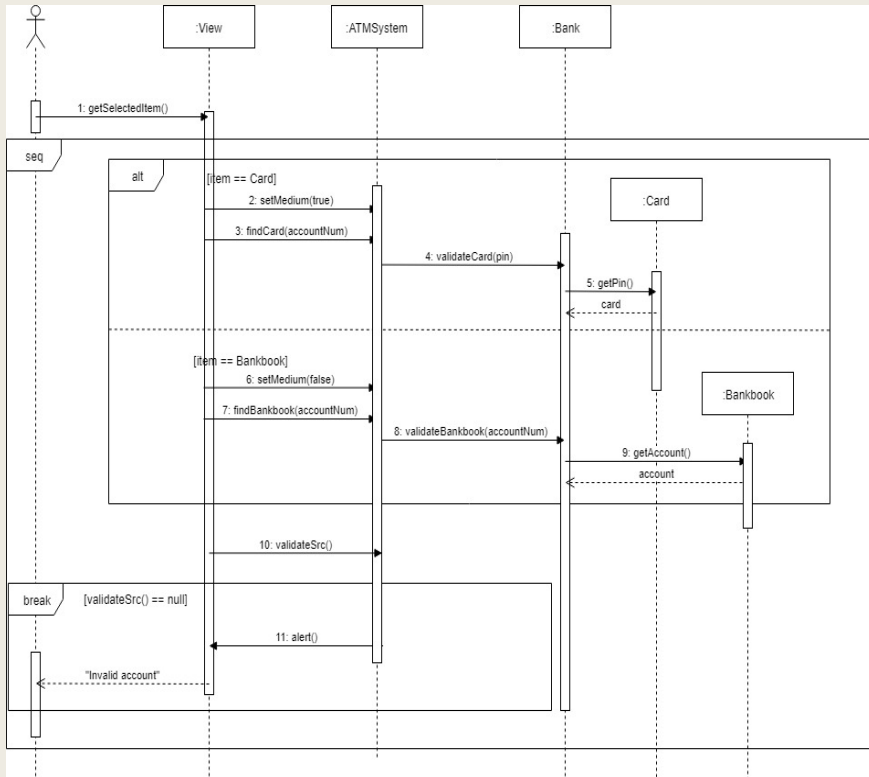
<b>Name</b>	SelectMenuView
<b>Type</b>	GUI
<b>Responsibilities</b>	입금, 출금, 송금, 조회, 범죄이력 버튼 중 하나를 누른다.
<b>Cross Reference</b>	Functional Requirement: R 1
<b>Notes</b>	입금, 출금, 송금, 조회, 범죄이력 조회 버튼 중 하나를 누른다.
<b>Post-Condition</b>	선택한 버튼의 기능으로 진행하고 계좌를 입력할 수 있다.
<b>Pre-Condition</b>	ATM이 켜진 상황이어야 한다.

# 2-4. Stage 2050 \_Act 2052



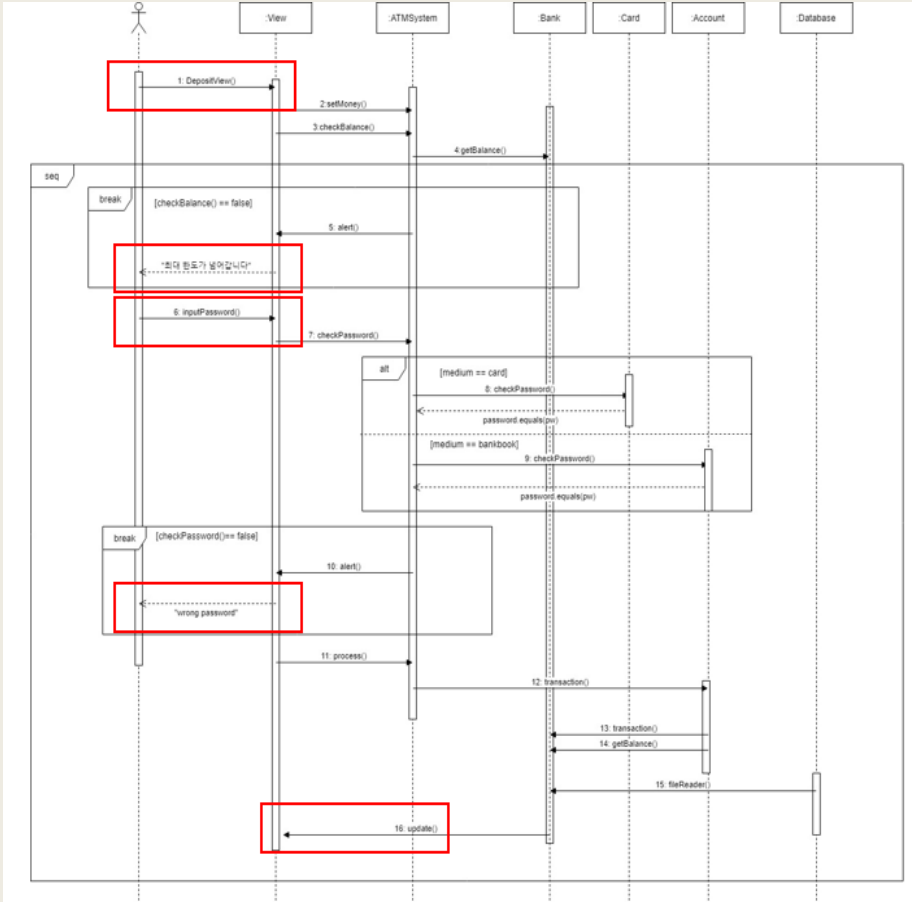
<b>Name</b>	InputSrcView()
<b>Type</b>	GUI
<b>Responsibilities</b>	카드 번호를 입력할지 통장 번호를 입력할지 선택하고 그에 따라 카드 번호 또는 통장 번호를 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 2
<b>Notes</b>	카드와 통장 중 하나를 선택한 뒤 번호를 입력한다.
<b>Post-Condition</b>	선택한 버튼의 기능으로 계속 진행한다.
<b>Pre-Condition</b>	메뉴 선택이 되어 있다.

# 2-4. Stage 2050 \_Act 2052



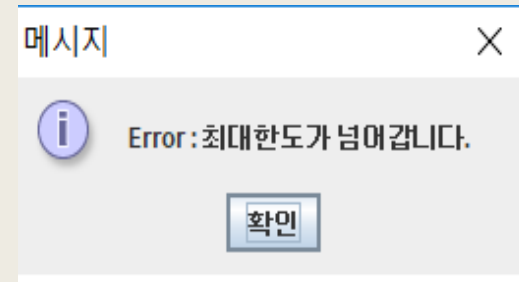
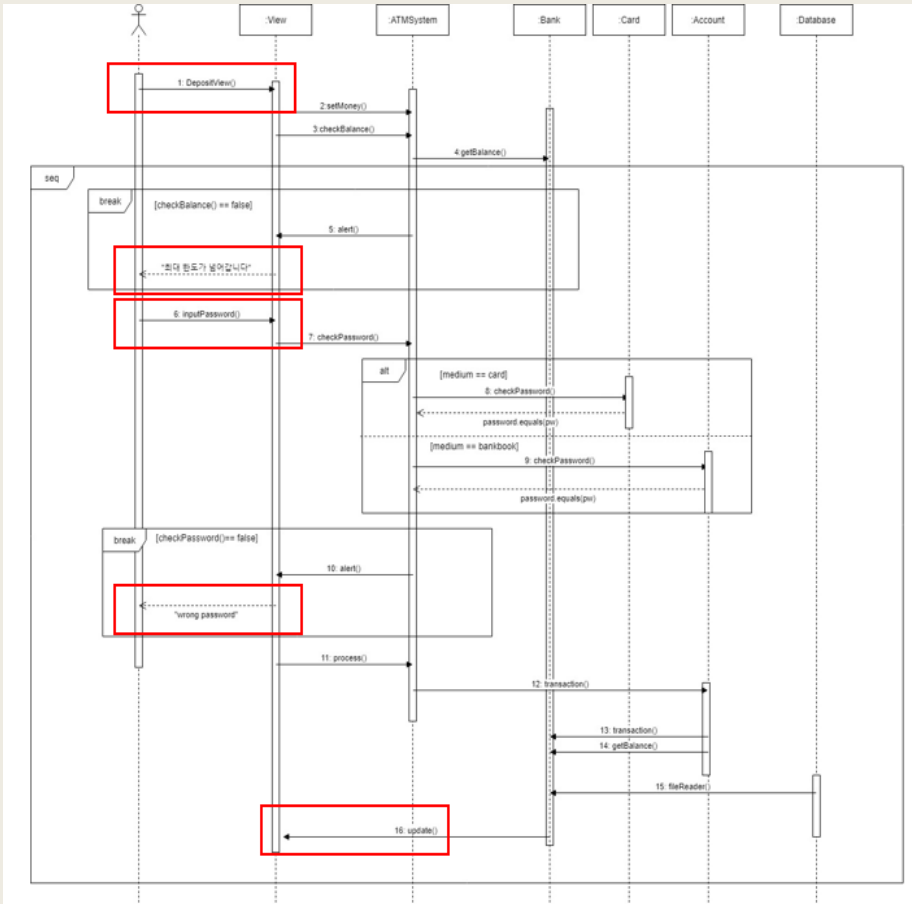
<b>Name</b>	alert "Invalid Account"
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌가 유효하지 않음을 알린다.
<b>Cross Reference</b>	R 2
<b>Notes</b>	계좌가 유효하지 않음을 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효하지 않은 계좌를 입력한다.

# 2-4. Stage 2050 \_Act 2052



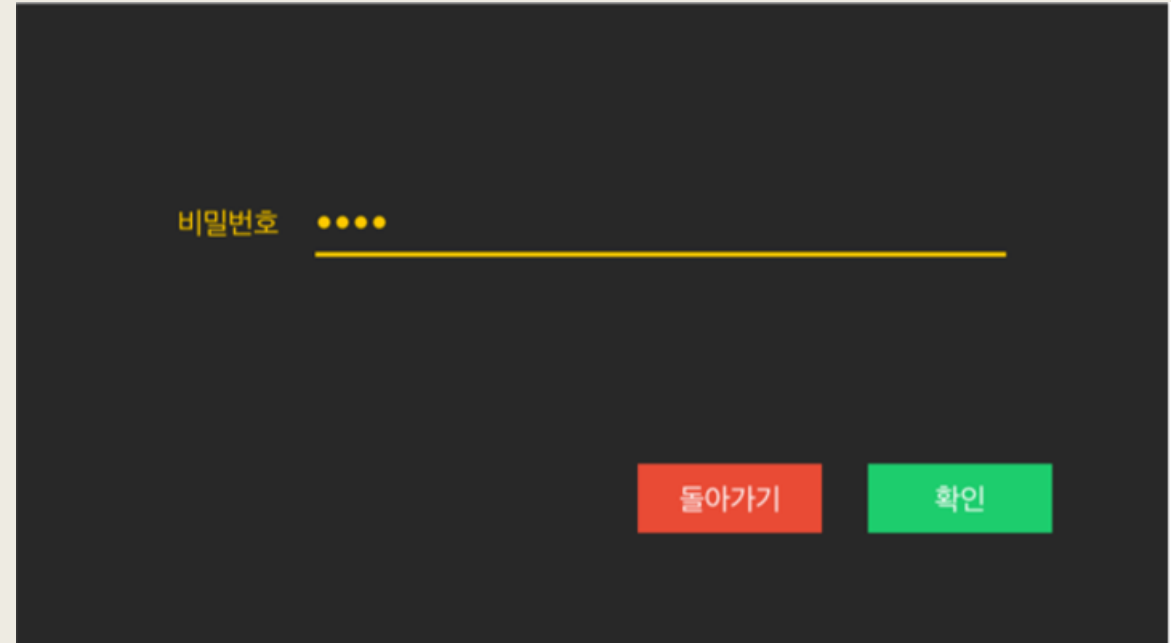
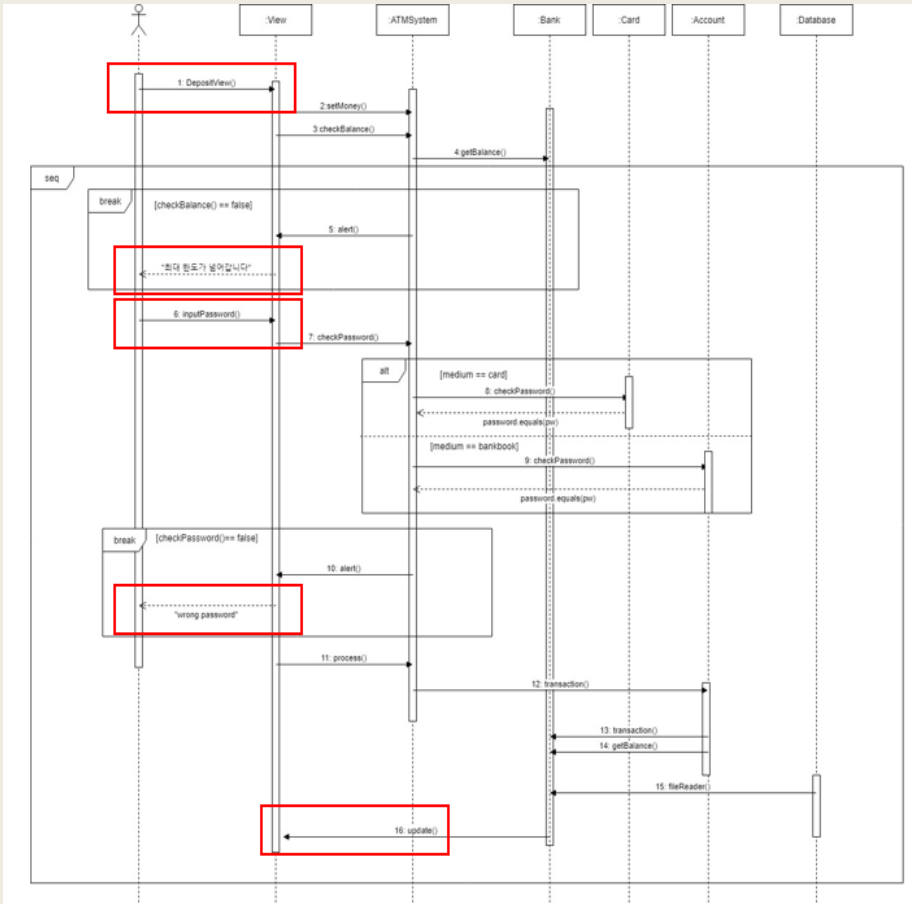
<b>Name</b>	DepositView()
<b>Type</b>	GUI
<b>Responsibilities</b>	입금할 금액을 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 3
<b>Notes</b>	
<b>Post-Condition</b>	비밀번호를 입력할 수 있다.
<b>Pre-Condition</b>	카드 번호나 통장 번호를 입력하고 계좌 번호가 유효한 상태여야 한다.

# 2-4. Stage 2050 \_Act 2052



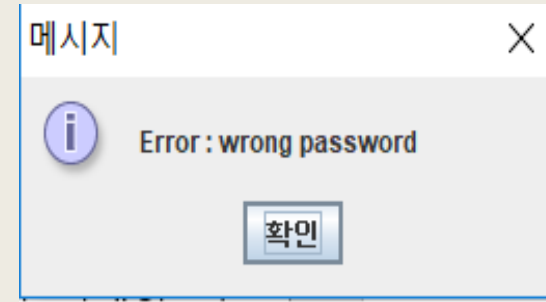
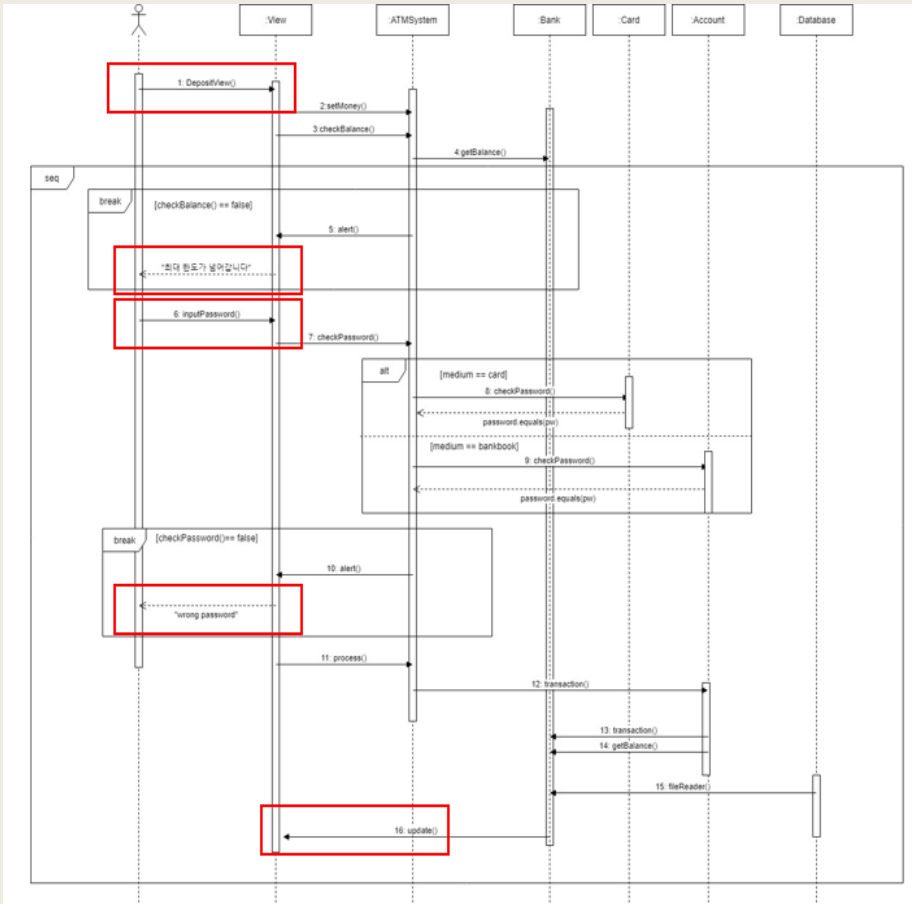
<b>Name</b>	alert “최대한도가 넘어갑니다”
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌에 들어갈 돈의 최대 한도가 넘어갈 때 알린다.
<b>Cross Reference</b>	Functional Requirement: R 3
<b>Notes</b>	계좌에 들어갈 돈의 최대 한도가 넘어갈 때 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	최대 한도 이상의 돈을 입력한다.

# 2-4. Stage 2050 \_Act 2052



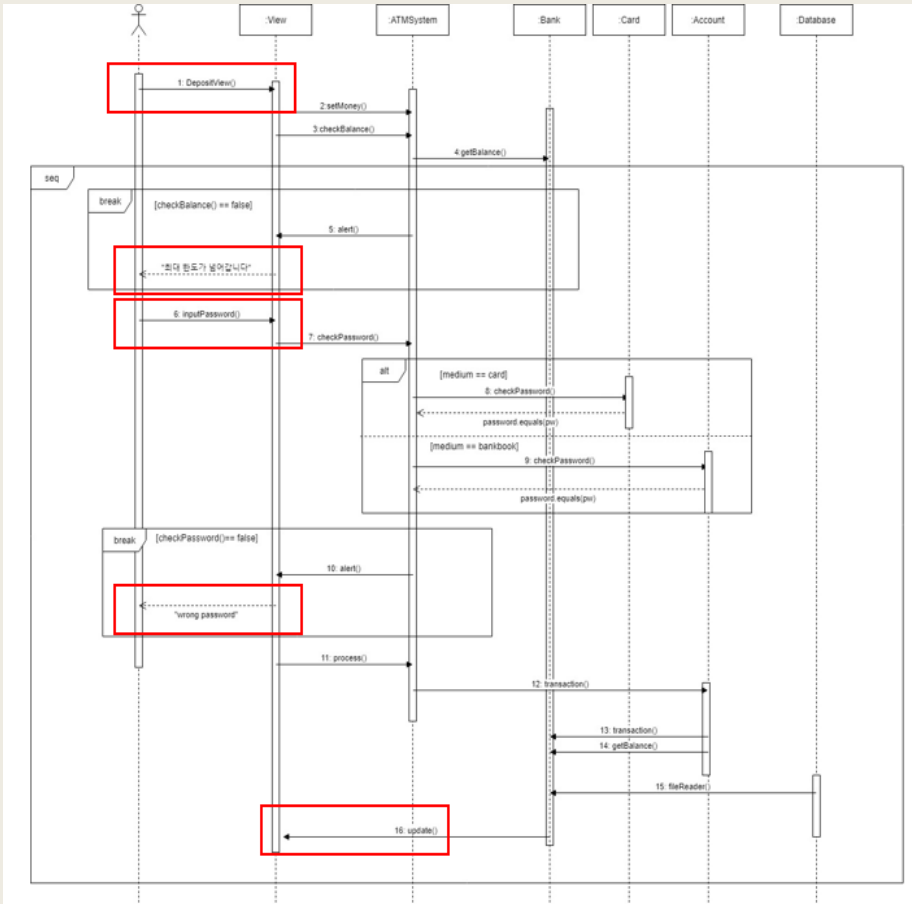
<b>Name</b>	inputPassword
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌의 비밀번호를 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 3
<b>Notes</b>	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
<b>Post-Condition</b>	비밀번호가 유효할 경우 입금이 계속 진행되고 결과창이 뜬다.
<b>Pre-Condition</b>	입금할 돈을 입력받은 상황이어야 한다.

# 2-4. Stage 2050 \_Act 2052



<b>Name</b>	alert "wrong password"
<b>Type</b>	GUI
<b>Responsibilities</b>	비밀번호가 유효하지 않음을 알린다.
<b>Cross Reference</b>	Functional Requirement: R 3
<b>Notes</b>	비밀번호가 유효하지 않음을 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효하지 않은 비밀번호를 입력한다.

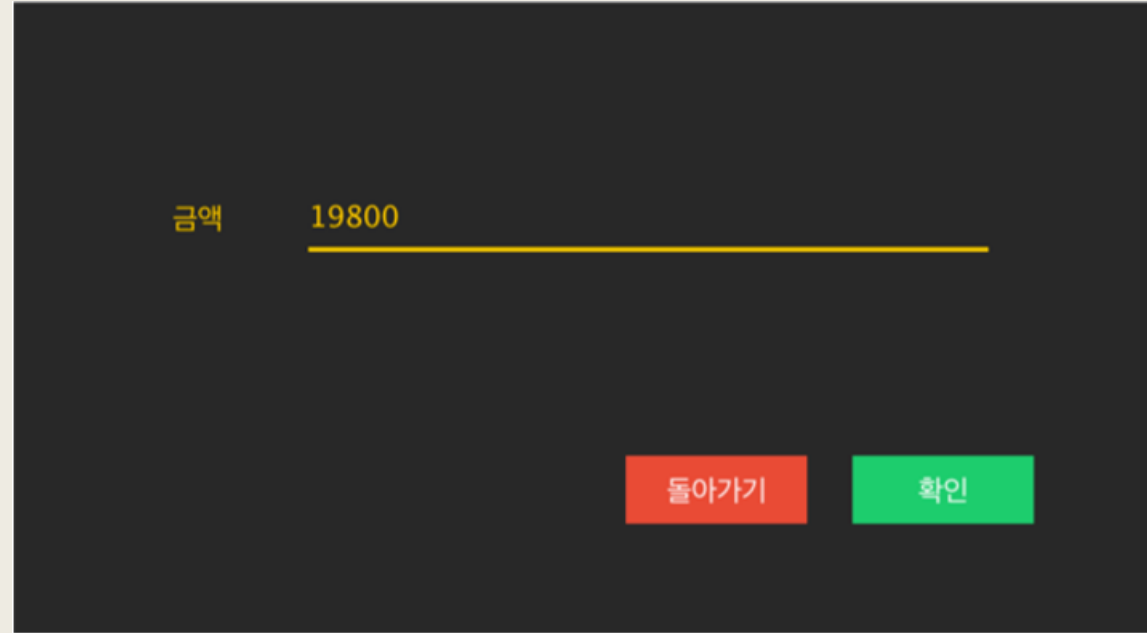
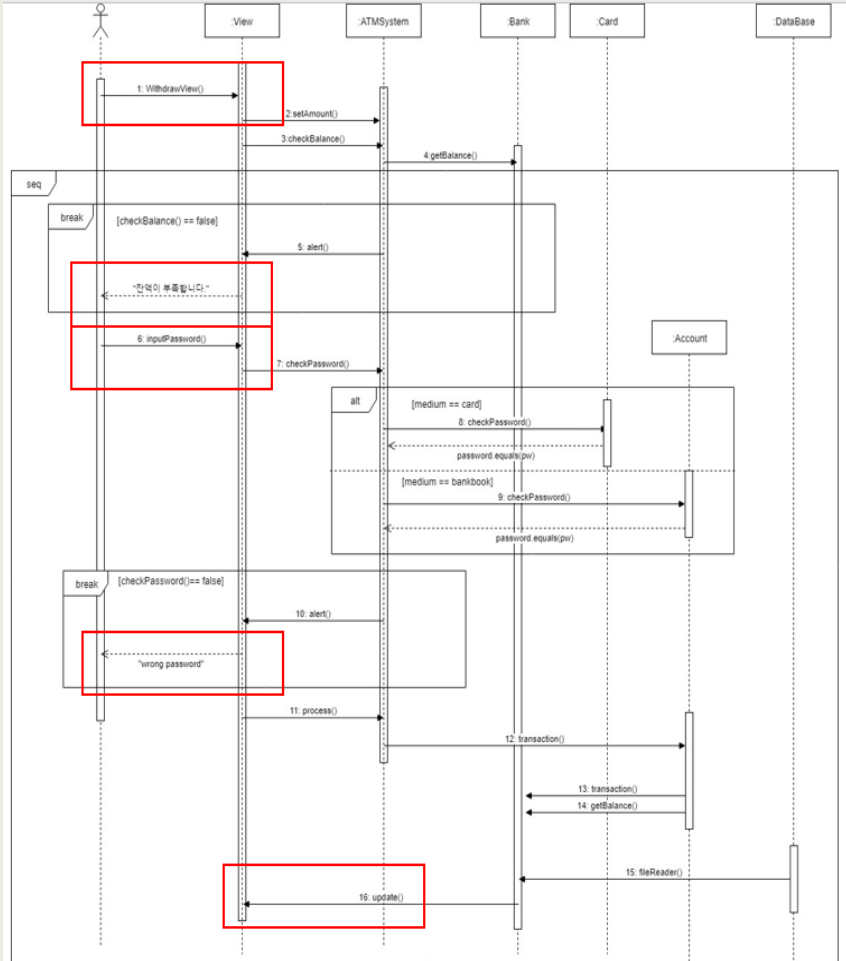
# 2-4. Stage 2050 \_Act 2052



Name	Update
Type	GUI
Responsibilities	입금이 완료된 후 결과와 내역을 보여준다.
Cross Reference	R 3
Notes	작업, 금액, 잔액의 입금 내역을 보여준다.
Post-Condition	돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다.
Pre-Condition	유효한 비밀번호를 입력하고 입금이 계속 진행된다.

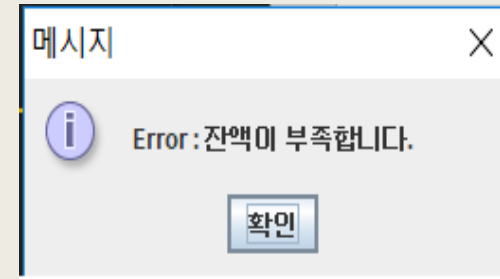
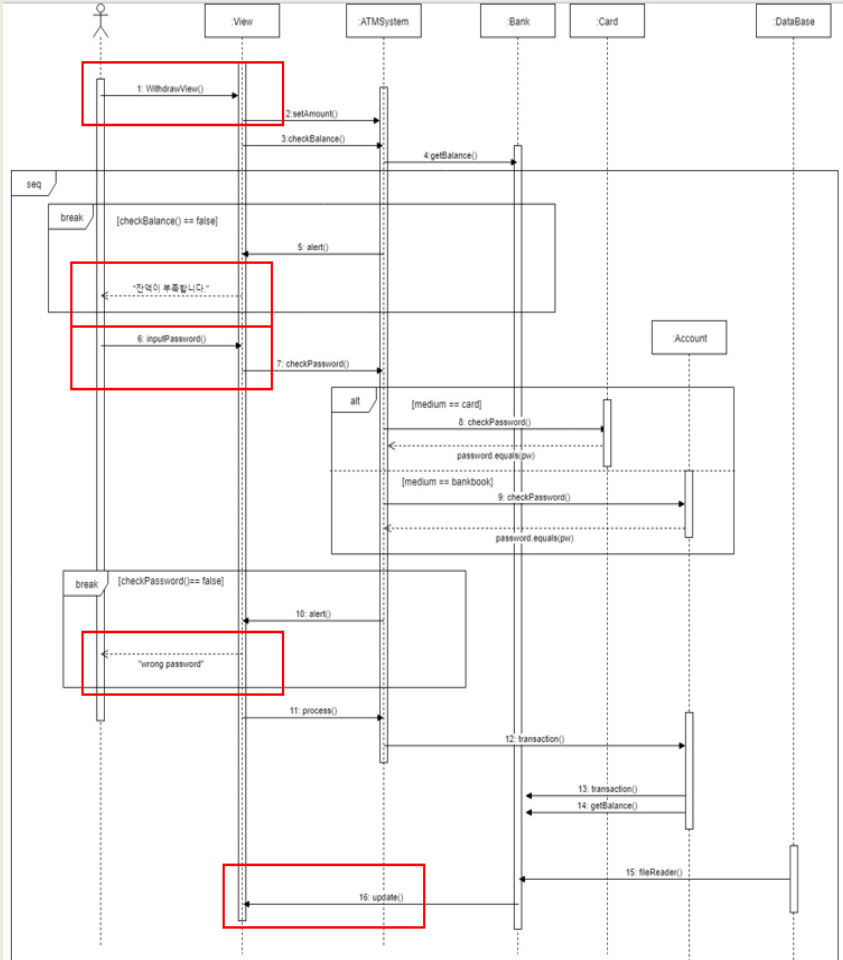


# 2-4. Stage 2050 \_Act 2052



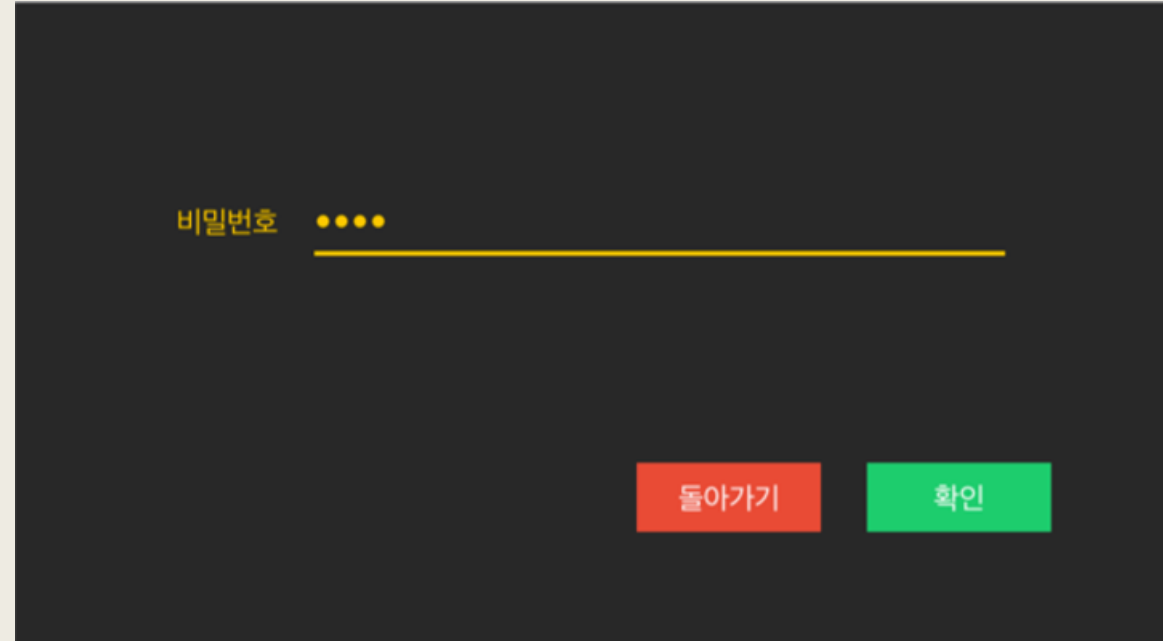
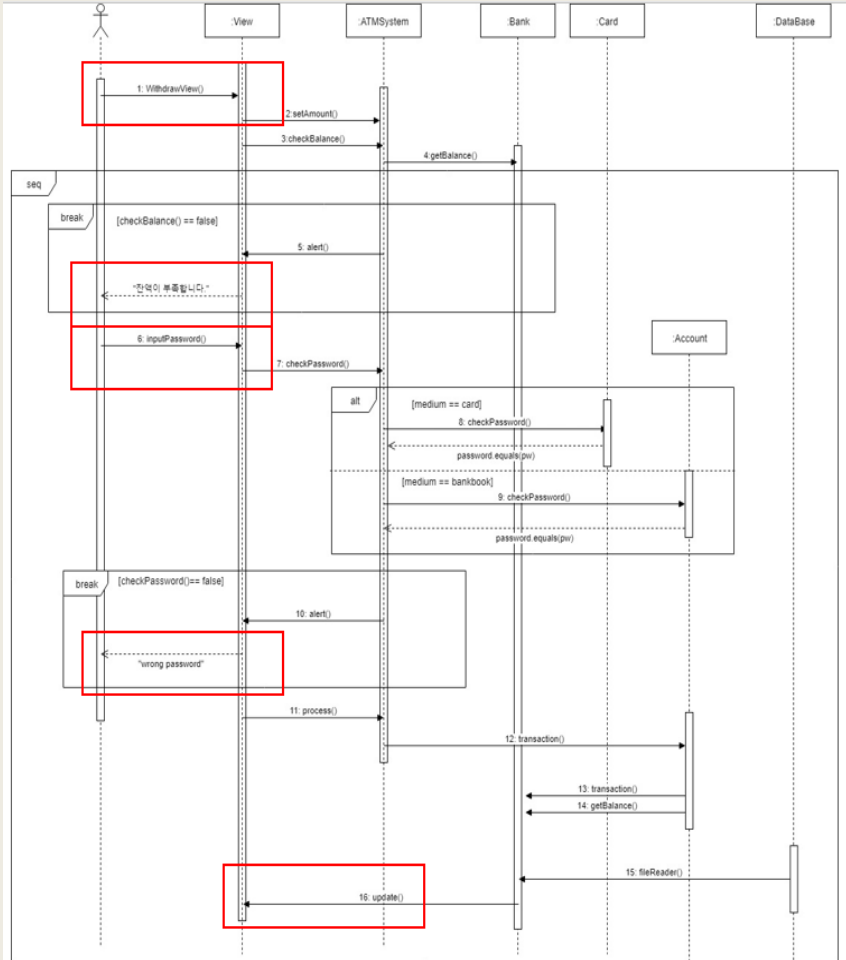
<b>Name</b>	WithdrawView()
<b>Type</b>	GUI
<b>Responsibilities</b>	출금할 금액을 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 4
<b>Notes</b>	
<b>Post-Condition</b>	비밀번호를 입력할 수 있다.
<b>Pre-Condition</b>	카드 번호나 통장 번호를 입력하고 계좌 번호가 유효한 상태여야 한다.

# 2-4. Stage 2050 \_Act 2052



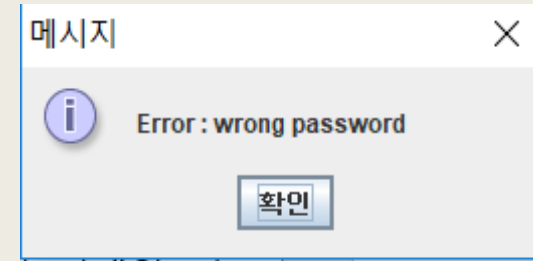
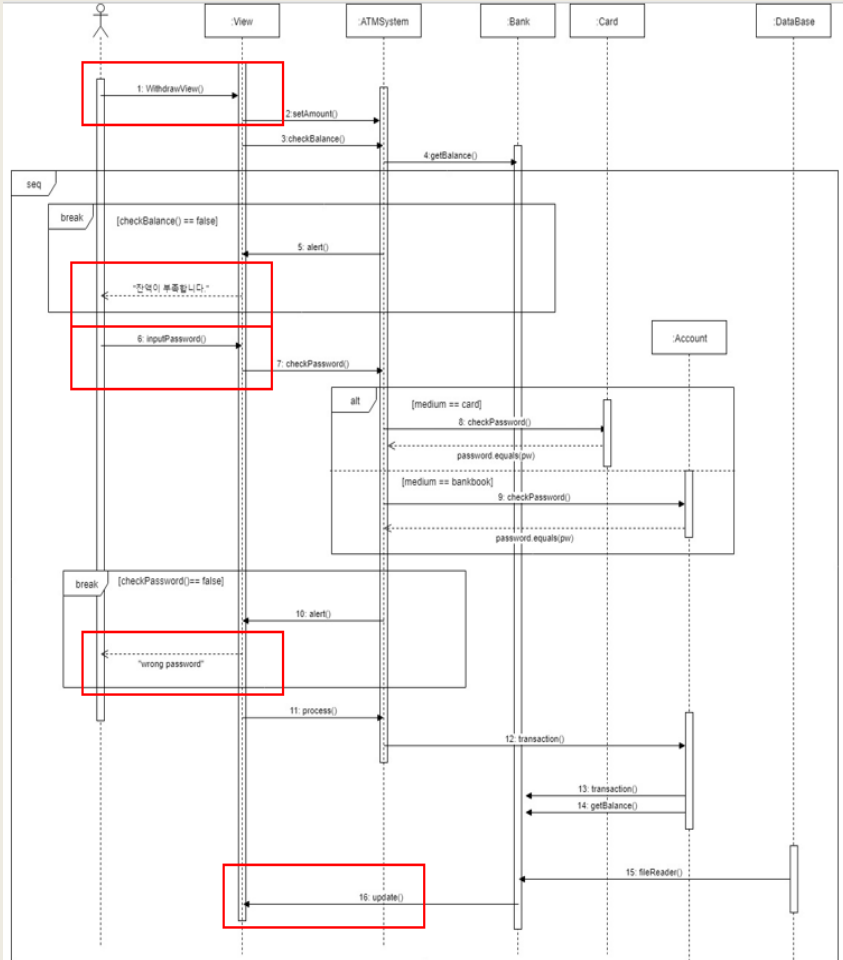
<b>Name</b>	alert “잔액이 부족합니다”
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌에서 출금할 돈이 잔고보다 더 클 때 알린다.
<b>Cross Reference</b>	Functional Requirement: R 4
<b>Notes</b>	계좌에서 출금할 돈이 잔고보다 더 클 때 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	잔고보다 더 큰 돈을 입력한다.

# 2-4. Stage 2050 \_Act 2052



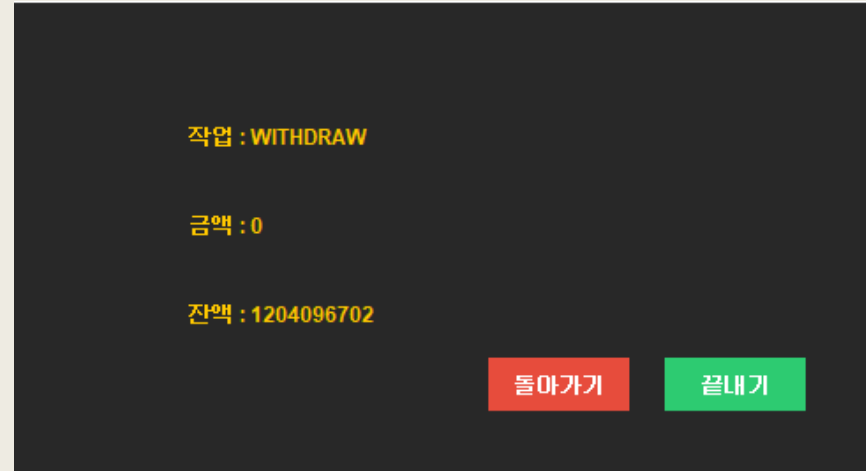
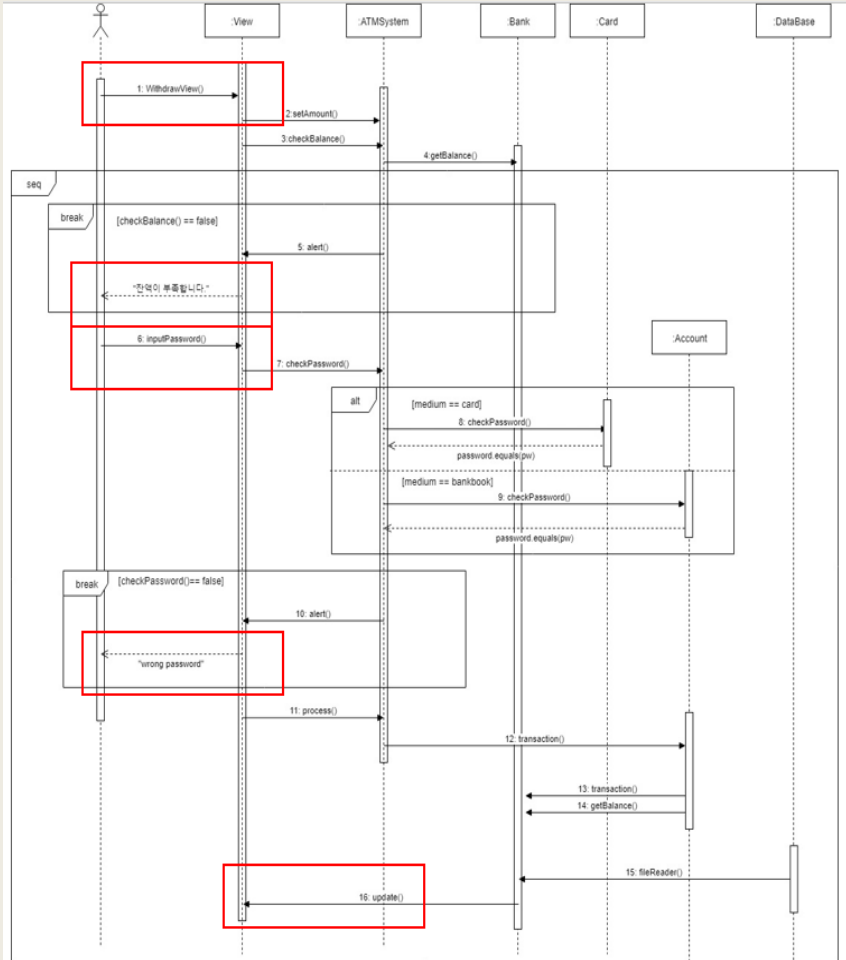
<b>Name</b>	inputPassword
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌의 비밀번호를 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 4
<b>Notes</b>	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
<b>Post-Condition</b>	비밀번호가 유효할 경우 출금이 계속 진행되고 결과창이 뜬다.
<b>Pre-Condition</b>	출금할 돈을 입력받은 상황이어야 한다.

# 2-4. Stage 2050 \_Act 2052



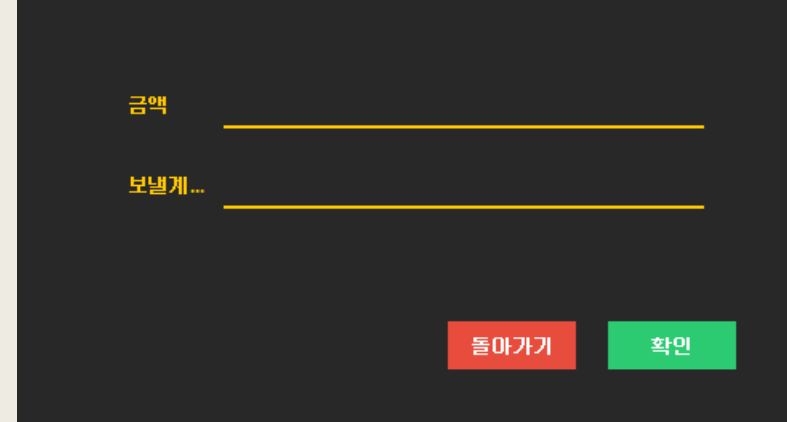
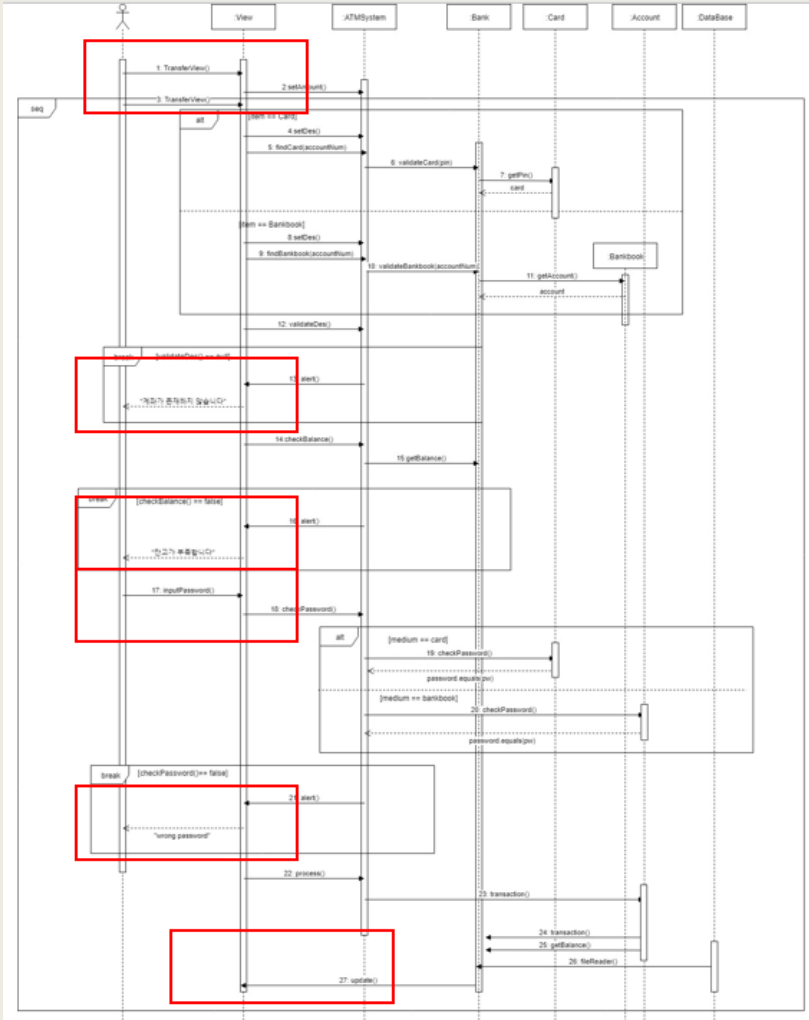
<b>Name</b>	alert “wrong password”
<b>Type</b>	GUI
<b>Responsibilities</b>	비밀번호가 유효하지 않음을 알린다.
<b>Cross Reference</b>	Functional Requirement: R 4
<b>Notes</b>	비밀번호가 유효하지 않음을 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효하지 않은 비밀번호를 입력한다.

# 2-4. Stage 2050 \_Act 2052



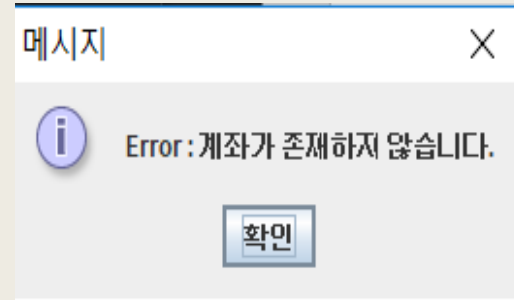
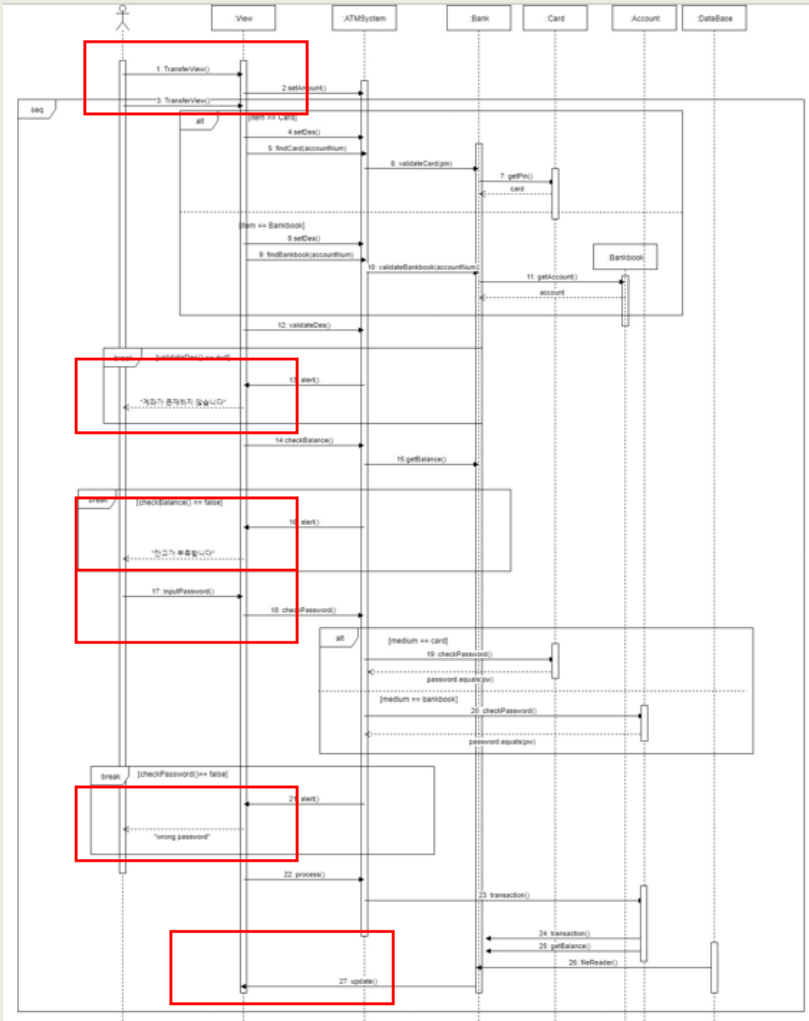
Name	Update
Type	GUI
Responsibilities	출금이 완료된 후 결과와 내역을 보여준다.
Cross Reference	R 4
Notes	작업, 금액, 잔액의 출금 내역을 보여준다.
Post-Condition	돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다.
Pre-Condition	유효한 비밀번호를 입력하고 출금이 계속 진행된다.

# 2-4. Stage 2050 \_Act 2052



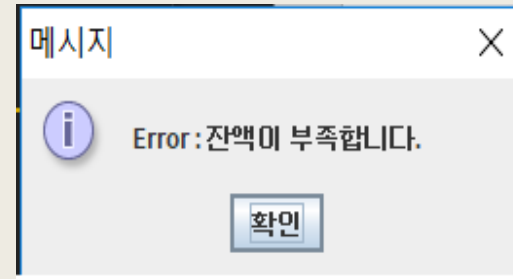
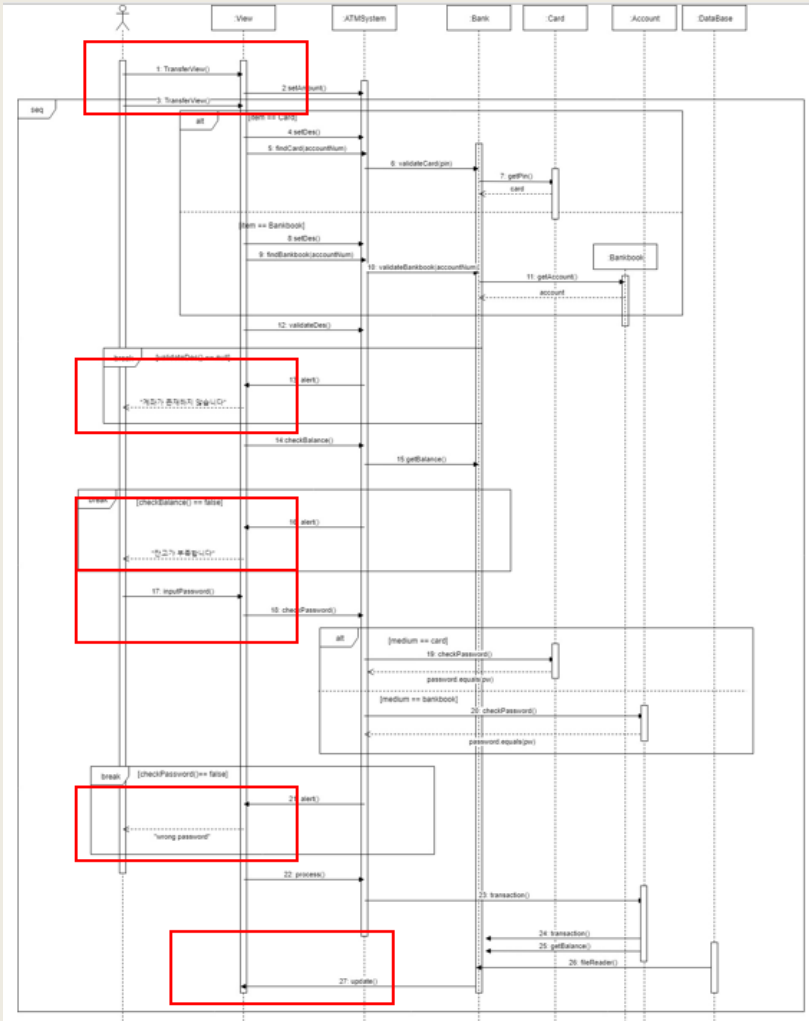
<b>Name</b>	TransferView()
<b>Type</b>	GUI
<b>Responsibilities</b>	송금할 금액과 보낼 계좌 번호를 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 5
<b>Notes</b>	
<b>Post-Condition</b>	비밀번호를 입력할 수 있다.
<b>Pre-Condition</b>	카드 번호나 통장 번호를 입력하고 계좌 번호가 유효한 상태여야 한다.

# 2-4. Stage 2050 \_Act 2052



<b>Name</b>	alert “계좌가 존재하지 않습니다”
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌가 유효하지 않음을 알린다.
<b>Cross Reference</b>	Functional Requirement: R 5
<b>Notes</b>	계좌가 유효하지 않음을 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효하지 않은 계좌를 입력한다.

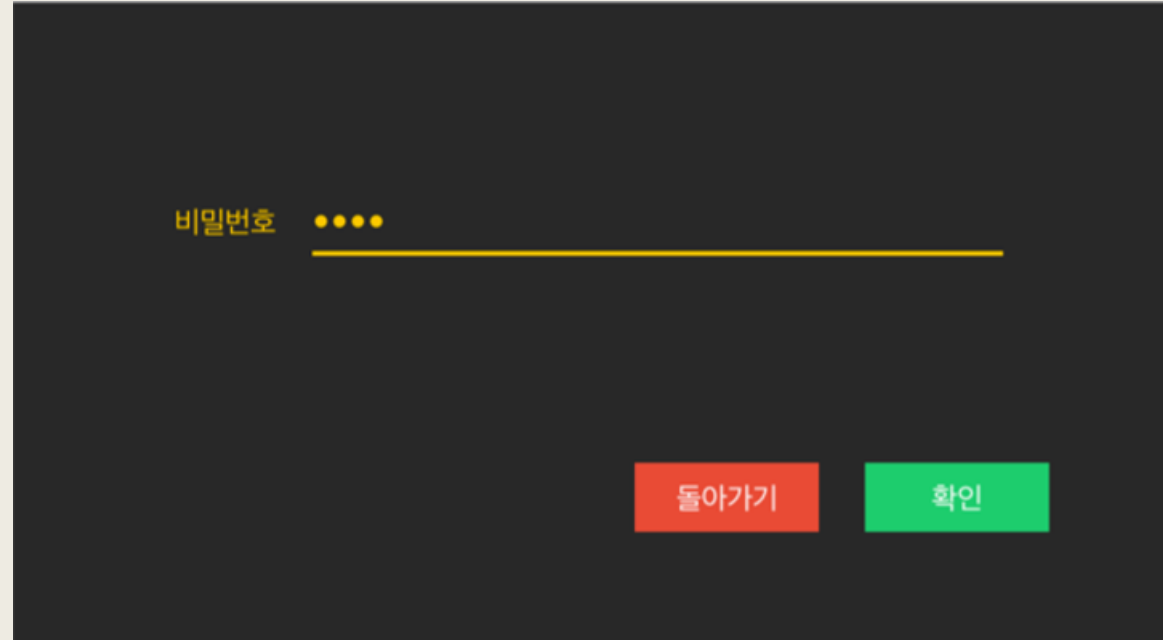
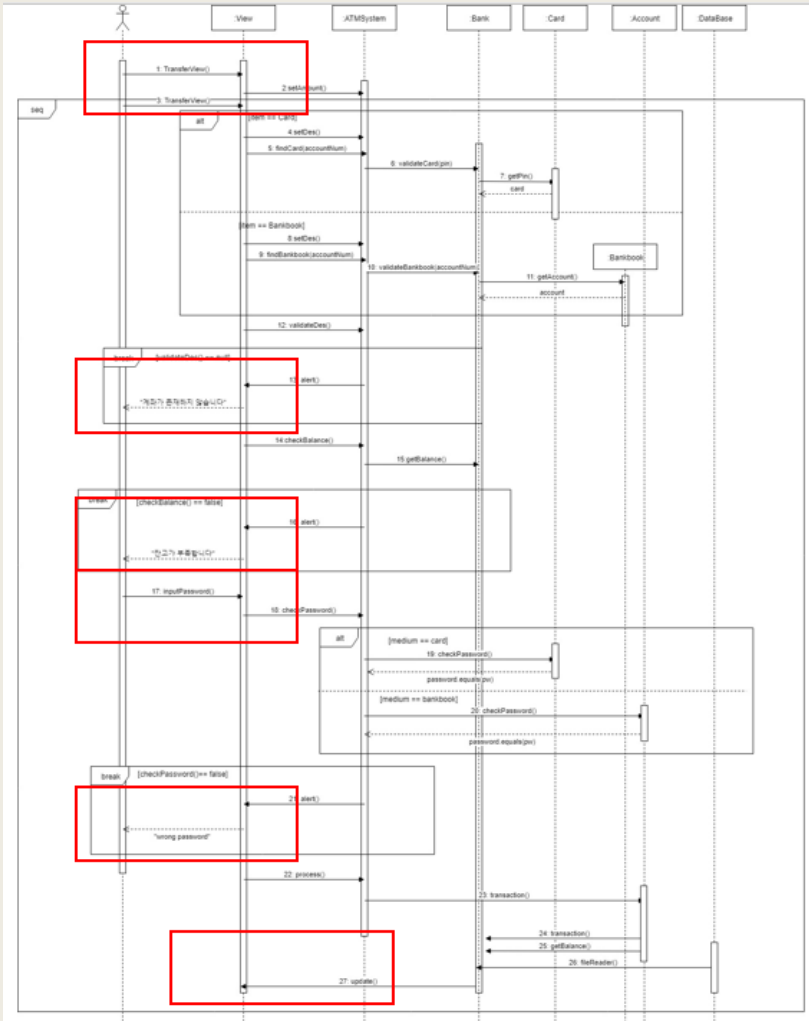
# 2-4. Stage 2050 \_Act 2052



<b>Name</b>	alert “잔고가 부족합니다”
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌에서 송금할 돈이 잔고보다 더 클 때 알린다.
<b>Cross Reference</b>	Functional Requirement: R 5
<b>Notes</b>	계좌에서 송금할 돈이 잔고보다 더 클 때 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	잔고보다 더 큰 돈을 입력한다.

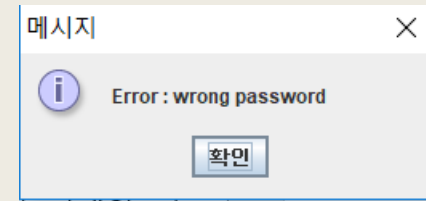
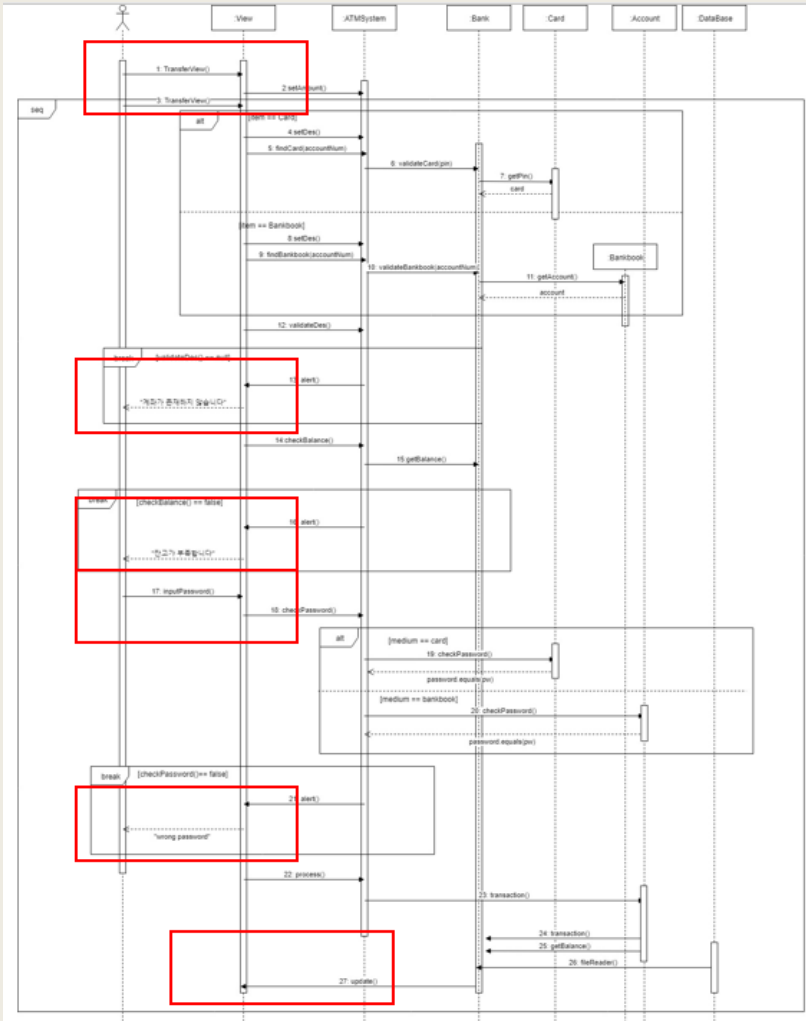


# 2-4. Stage 2050 \_Act 2052



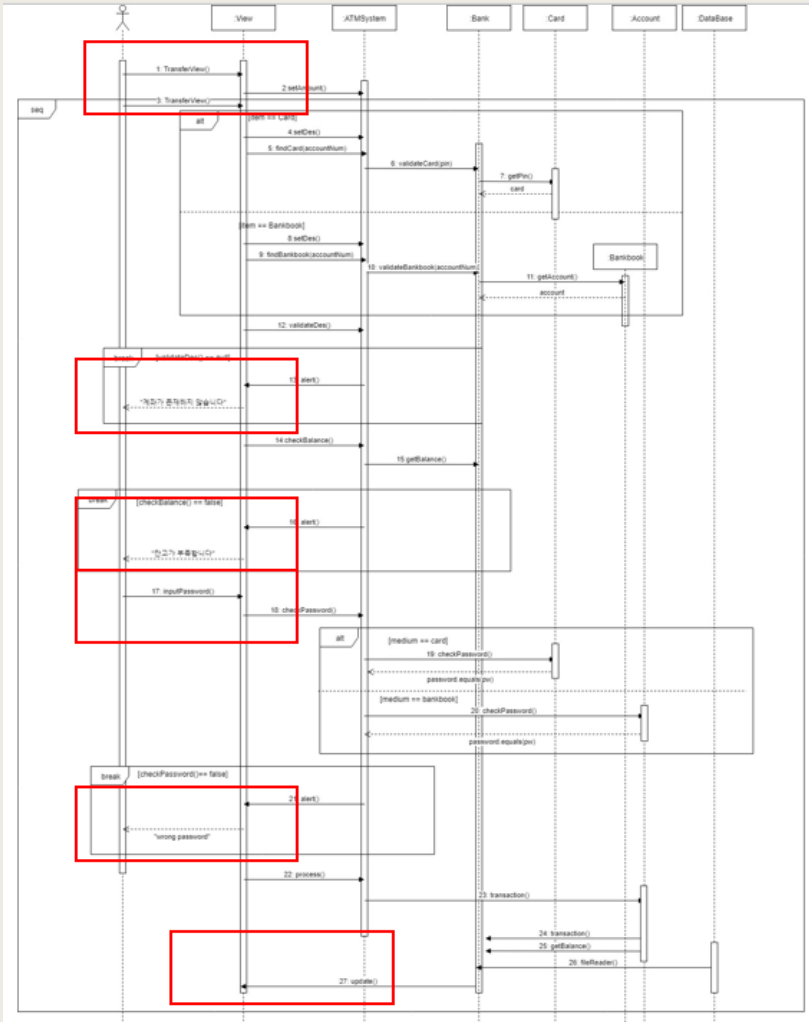
<b>Name</b>	inputPassword
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌의 비밀번호를 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 5
<b>Notes</b>	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
<b>Post-Condition</b>	비밀번호가 유효할 경우 송금이 계속 진행되고 결과창이 뜬다.
<b>Pre-Condition</b>	송금할 돈을 입력받은 상황이어야 한다.

# 2-4. Stage 2050 \_Act 2052



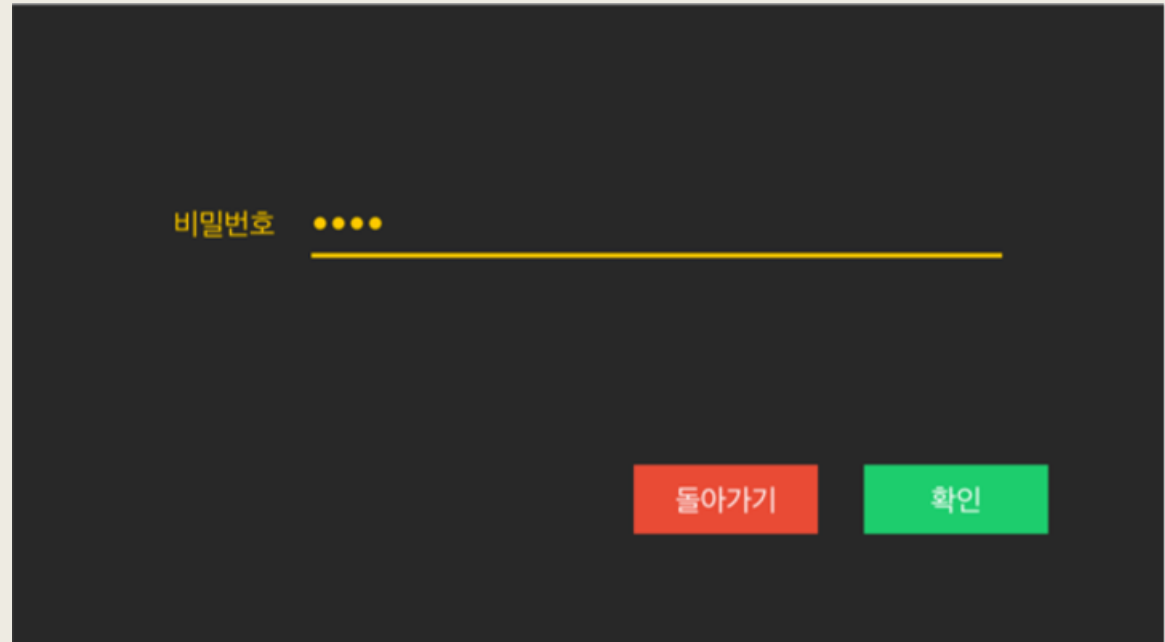
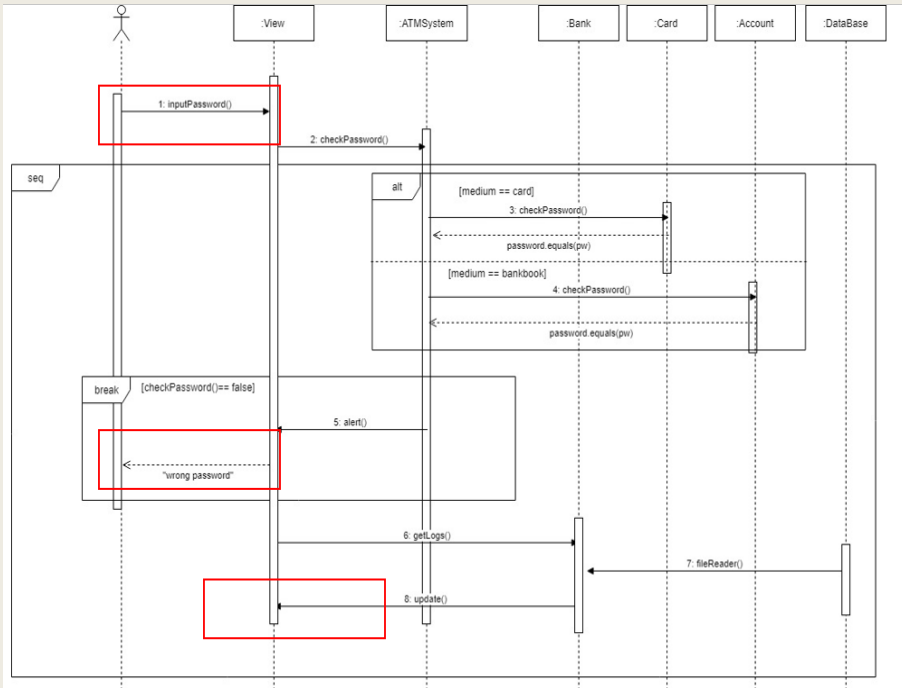
<b>Name</b>	alert “wrong password”
<b>Type</b>	GUI
<b>Responsibilities</b>	비밀번호가 유효하지 않음을 알린다.
<b>Cross Reference</b>	Functional Requirement: R 5
<b>Notes</b>	비밀번호가 유효하지 않음을 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효하지 않은 비밀번호를 입력한다.

# 2-4. Stage 2050 \_Act 2052



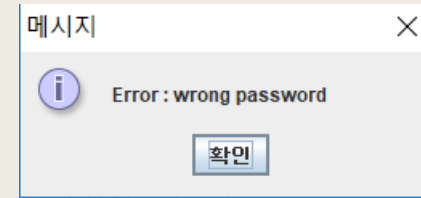
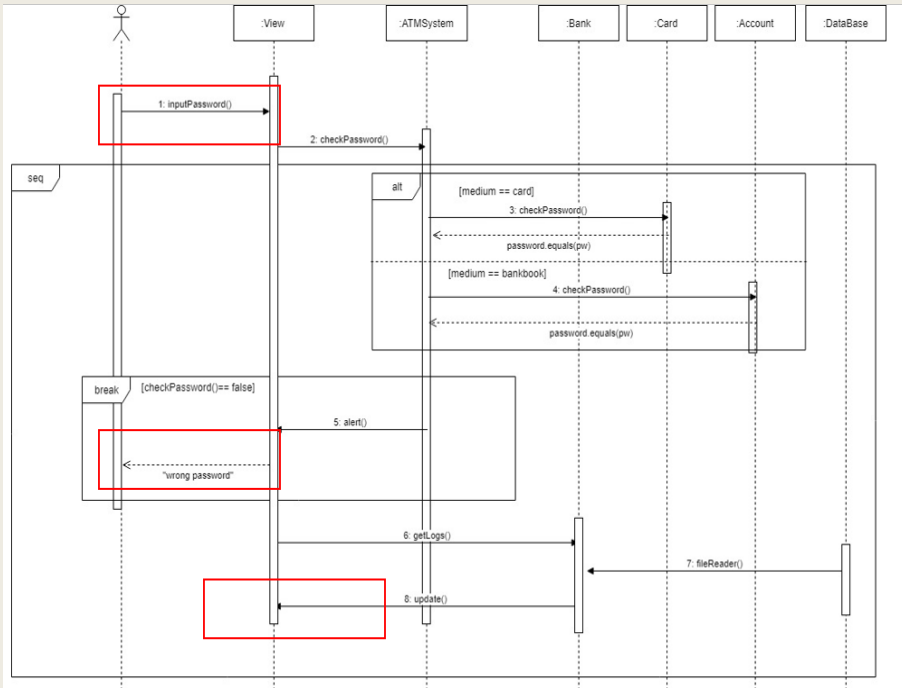
<b>Name</b>	Update
<b>Type</b>	GUI
<b>Responsibilities</b>	송금이 완료된 후 결과와 내역을 보여준다.
<b>Cross Reference</b>	R 5
<b>Notes</b>	작업, 금액, 잔액의 송금 내역을 보여준다.
<b>Post-Condition</b>	돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효한 비밀번호를 입력하고 송금이 계속 진행된다.

# 2-4. Stage 2050 \_Act 2052



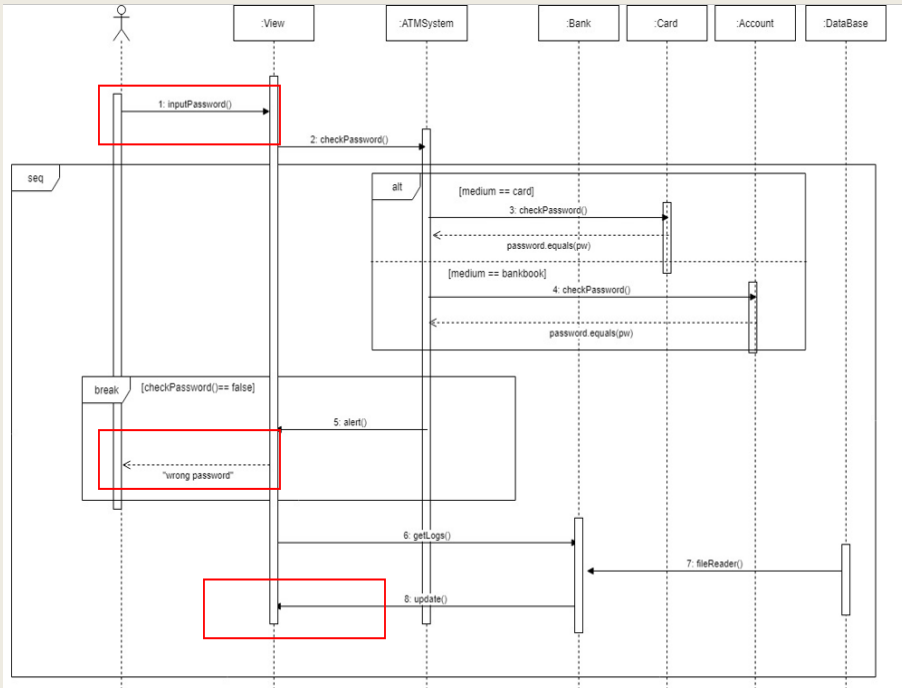
<b>Name</b>	inputPassword
<b>Type</b>	GUI
<b>Responsibilities</b>	계좌의 비밀번호를 입력할 수 있다.
<b>Cross Reference</b>	Functional Requirement: R 6
<b>Notes</b>	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
<b>Post-Condition</b>	비밀번호가 유효할 경우 거래내역조회가 계속 진행되고 결과창이 뜬다.
<b>Pre-Condition</b>	거래내역 조회를 선택한 상황이어야 한다.

# 2-4. Stage 2050 \_Act 2052



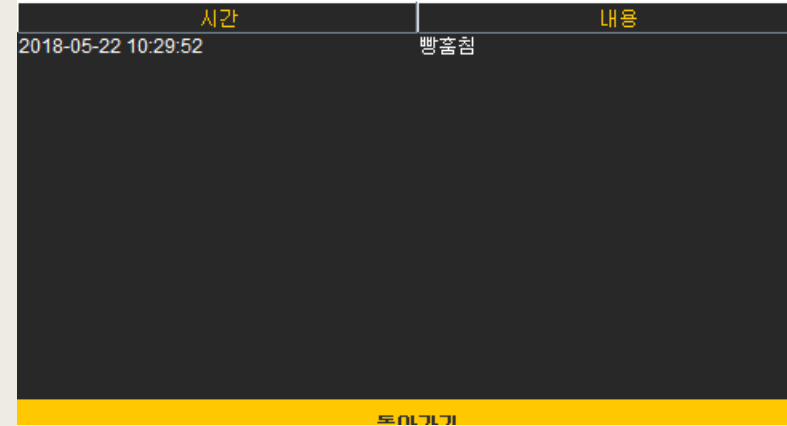
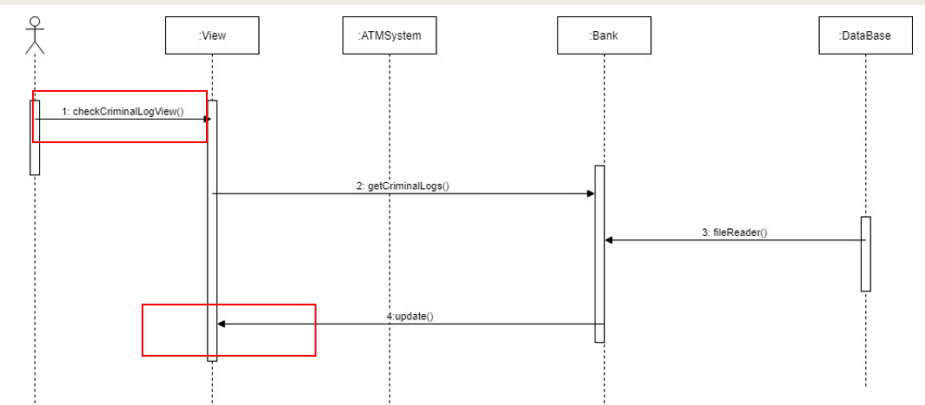
<b>Name</b>	alert "wrong password"
<b>Type</b>	GUI
<b>Responsibilities</b>	비밀번호가 유효하지 않음을 알린다.
<b>Cross Reference</b>	Functional Requirement: R 6
<b>Notes</b>	비밀번호가 유효하지 않음을 알린다.
<b>Post-Condition</b>	확인 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효하지 않은 비밀번호를 입력한다.

# 2-4. Stage 2050 \_Act 2052



<b>Name</b>	Update
<b>Type</b>	GUI
<b>Responsibilities</b>	거래 내역을 보여준다.
<b>Cross Reference</b>	R 6
<b>Notes</b>	계좌의 거래 내역을 보여준다.
<b>Post-Condition</b>	돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효한 비밀번호를 입력하고 거래내역조회가 계속 진행된다.

# 2-4. Stage 2050 \_Act 2052



<b>Name</b>	Update
<b>Type</b>	GUI
<b>Responsibilities</b>	범죄 이력 내역을 보여준다.
<b>Cross Reference</b>	R 7
<b>Notes</b>	계좌의 범죄 이력 내역을 보여준다.
<b>Post-Condition</b>	돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다.
<b>Pre-Condition</b>	유효한 계좌를 입력하고 범죄 이력 조회가 계속 진행된다.

# 2-4. Stage 2050 \_Act 2063

Identifier	Test 항목	Description	System Function	Pass/Fail
ATM_STC_001_001	selectMenu	사용자가 입금을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.	R 1	P
ATM_STC_001_002	selectMenu	사용자가 출금을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.	R 1	P
ATM_STC_001_003	selectMenu	사용자가 송금을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.	R 1	P
ATM_STC_001_004	selectMenu	사용자가 거래내역확인을 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.	R 1	P
ATM_STC_001_005	selectMenu	사용자가 범죄이력조회를 선택할 때 정상적으로 선택된 메뉴가 진행되는지 확인한다.	R 1	P
ATM_STC_002_001	MediumCheck	카드 번호를 입력할지 선택하고 이에 따라 유효한 카드 번호를 입력했을 때 계좌 확인이 정상적으로 진행되는지 확인한다.	R 2	P
ATM_STC_002_002	MediumCheck	카드 번호를 입력할지 선택하고 이에 따라 유효하지 않은 카드 번호를 입력했을 때 다시 입력받을 수 있는지 확인한다.	R 2	P
ATM_STC_002_003	MediumCheck	통장 번호를 입력할지 선택하고 이에 따라 유효한 통장 번호를 입력했을 때 계좌 확인이 정상적으로 진행되는지 확인한다.	R 2	P
ATM_STC_002_004	MediumCheck	통장 번호를 입력할지 선택하고 이에 따라 유효하지 않은 통장 번호를 입력했을 때 다시 입력받을 수 있는지 확인한다.	R 2	P



# 2-4. Stage 2050 \_Act 2063

Identifier	Test 항목	Description	System Function	Pass/Fail
ATM_STC_003_001	Deposit	입금할 금액을 잘 입력받을 수 있는지 확인한다.	R 3	P
ATM_STC_003_002	Deposit	계좌 번호에 해당되는 비밀번호가 입력되었을 때 수수료를 제외하고 입금되는지 확인한다.	R 3	P
ATM_STC_003_003	Deposit	계좌 번호에 해당되는 비밀번호가 아니라 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다.	R 3	P
ATM_STC_003_004	Deposit	입금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.	R 3	P
ATM_STC_004_001	Withdraw	출금할 금액을 잘 입력받을 수 있는지 확인한다.	R 4	P
ATM_STC_004_002	Withdraw	수수료를 포함하여 출금할 계좌의 잔액이 충분하지 않을 경우 출금 진행이 멈추는지 확인한다.	R 4	P
ATM_STC_004_003	Withdraw	계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 제외하고 출금되는지 확인한다.	R 4	P
ATM_STC_004_004	Withdraw	계좌번호에 해당되는 비밀번호가 아니라 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다.	R 4	P
ATM_STC_004_005	Withdraw	출금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.	R 4	P

# 2-4. Stage 2050 \_Act 2063

Identifier	Test 항목	Description	System Function	Pass/Fail
ATM_STC_005_001	Transfer	송금할 금액을 잘 입력받을 수 있는지 확인한다.	R 5	P
ATM_STC_005_002	Transfer	돈을 보낼 계좌를 입력하고 유효하지 않은 계좌를 입력했을 때 경고를 받고 다시 입력받는지 확인한다.	R 5	P
ATM_STC_005_003	Transfer	수수료를 포함하여 송금할 계좌의 잔액이 충분하지 않을 경우 송금 진행이 멈추는지 확인한다.	R 5	P
ATM_STC_005_004	Transfer	계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 제외하고 송금되는지 확인한다.	R 5	P
ATM_STC_005_005	Transfer	계좌 번호에 해당되는 비밀번호가 아니라 유효하지 않은 비밀번호가 입력되었을 때 다시 입력받는지 확인한다.	R 5	P
ATM_STC_005_006	Transfer	송금이 종료된 후 결과가 정상적으로 출력되는지 확인한다.	R 5	P
ATM_STC_006	CheckTransaction History	입력한 계좌의 거래내역이 화면에 정상적으로 출력되는지 확인한다.	R 6	P
ATM_STC_007_001	CheckCriminal History	범죄이력을 조회하고 싶은 계좌를 입력하고 유효하지 않은 계좌를 입력했을 때 경고를 받고 다시 입력받는지 확인한다.	R 7	P
ATM_STC_007_002	CheckCriminal History	입력한 계좌의 범죄이력내역이 화면에 정상적으로 출력되는지 확인한다.	R 7	P

# 2-4. Stage 2050 \_Act 2063

Identifier	Test 항목	Non-Description	System Function	Pass/Fail
ATM_STC_008	Safe Transaction	범죄 이력이 있는 계좌의 조회가 가능한지 확인한다.	R 8	P
ATM_STC_009	Performance	모든 버튼 입력 및 이벤트에 대하여 1초 이내로 반응하는지 확인한다.	R 9	F
ATM_STC_010	OS - Independent	프로그램이 OS에 무관하게 정상적으로 동작하는지 확인한다.	R 10	P



Q & A

---

