

# OOPT STAGE 2050-2060

## -Implementation & Unit Test

### **Project Team**

Team T4

### **Date**

2018 / 06 / 10

### **Team Members**

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

## Activity 2051. Implement Class & Methods Definition

### 1) Account class

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

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

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

Type	Method
Name	getAccountNum()
Purpose	Account 의 계좌번호를 확인한다.
CrossReference	Use Case R2, R5
Input(Method)	-
Output(Method)	String accountNum
Abstract Operation(Method)	account.accountNum 을 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 에서 balance 를 가져와 반환한다.
Exceptional Courses of Events	-

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)	Account 의 비밀번호와의 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 amount 만큼 거래하고 log 에 msg 를 저장한다.
Exceptional Courses of Events	-

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 log 를 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 내용과 범죄 시간을 반환한다,

Exceptional Courses of Events	
-------------------------------	--

## 2) ATMSystem Class

Type	class
Name	AtmSystem
Purpose	ATM 정보를 모아두는 클래스
Overview(class)	<p>enum MediumType{CARD, BANKBOOK}, enum ProcessType{DEPOSIT, WITHDRAW, TRANSFER}</p> <p>Attribute : Bank own, View view, 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>
CrossReference	Use Case R1, R2, R3, R4, R5, R6, R7
Exceptional Courses of Events	N/A

Type	Method
Name	AtmSystem()
Purpose	AtmSystem 의 Attribute 를 초기화한다.

CrossReference	Use Case R1, R2, R3, R4, R5, R6, R7
Input(Method)	Bank own
Output(Method)	-
Abstract Operation(Method)	AtmSystem 의 Attribute 를 초기화한다.
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)	
Output(Method)	-
Abstract Operation(Method)	banks 배열에 존재하는 Bank 를 추가한다.
Exceptional Courses of Events	-

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)	if(money == 0) boolean (amount + fee) <=src.getBalance() else boolean money-fee+src.getBalance() >= 0
Abstract Operation(Method)	money= 0 일 경우 amount + fee 와 src.getbalance 를 비교한 결과를 반환한다. 아닐 경우 money-fee+src.getBalance()를 비교한 결과를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	checkPassword
Purpose	비밀번호를 확인한다.
CrossReference	Use Case R3, R4, R5, R6
Input(Method)	String pw

Output(Method)	if(medium == MediumType.CARD) boolean srcCard.checkPassword(pw) else boolean src.checkPassword(pw)
Abstract Operation(Method)	입력한 비밀번호가 src 의 비밀번호와 일치하는지 확인한다. 매체가 카드일 경우 srcCard 의 비밀번호와 일치하는지 확인한다.
Exceptional Courses of Events	-

Type	Method
Name	deposit()
Purpose	src 계좌에 입금한다.
CrossReference	Use Case R3
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	계좌의 balance 를 money - fee 만큼 증가시키고 log 에 msg 와 해당 거래내용을 업데이트한다.
Exceptional Courses of Events	-

Type	Method
Name	withdraw()
Purpose	src 에서 출금한다.

CrossReference	Use Case R4
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	계좌의 balance 를 amount + fee 만큼 감소시키고 log 에 msg 와 해당 거래내용을 업데이트한다.
Exceptional Courses of Events	-

Type	Method
Name	transfer()
Purpose	src 에서 des 로 송금한다.
CrossReference	Use Case R5
Input(Method)	-
Output(Method)	-
Abstract Operation(Method)	src 계좌의 balance 를 amount + fee 만큼 감소시키고 log 에 msg 와 해당 거래내용을 업데이트한다 des 계좌의 balance 를 amount 만큼 증가시키고 log 에 msg 와 해당 거래내용을 업데이트한다
Exceptional Courses of Events	-

Type	Method
Name	getLogs()

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

Type	Method
Name	setProcess
Purpose	이용할 서비스를 선택한다.
CrossReference	Use Case R1

Input(Method)	ProcessType process
Output(Method)	-
Abstract Operation(Method)	ProcessType process 를 입력받아 AtmSystem 의 process 를 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)	isCard 가 true 이면 ATMSystem 의 medium 을 MediumType.CARD 로, false 이면 MediumType.BANKBOOK 으로 세팅한다.
Exceptional Courses of Events	-

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 src 에 저장한다.
Exceptional Courses of Events	-

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

Type	Method
Name	setDes
Purpose	송금 대상 계좌를 세팅한다.
CrossReference	Use Case R5
Input(Method)	String accountNum

Output(Method)	-
Abstract Operation(Method)	findBankBook(accountNum)의 결과를 des 에 저장한다.
Exceptional Courses of Events	-

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

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.amount 에 amount 값을 저장한다. 타행 계좌일 경우 fee = amount*0.01 로 세팅한다.
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.money 에 money 값을 저장한다. 타행 계좌일 경우 fee = money*0.01 로 세팅한다.
Exceptional Courses of Events	-

Type	Method
Name	findCard()
Purpose	banks 에서 card 와 연결된 account 가 존재하는지 찾는다.
CrossReference	Use Case R2, R7
Input(Method)	String pin
Output(Method)	Account ret.getAccount

Abstract Operation(Method)	banks 에서 pin 을 이용해 카드와 연결된 계좌의 유효성 유무를 검색하고 존재하면 ret.getAccount 을 반환한다.
Exceptional Courses of Events	어느 banks 에도 존재하지 않는 pin 의 경우 null 을 반환한다.

Type	Method
Name	findBankbook()
Purpose	banks 에서 bankbook 과 연결된 account 가 존재하는지 찾는다.
CrossReference	Use Case R2, R7
Input(Method)	String accountNum
Output(Method)	Account ret.getAccount
Abstract Operation(Method)	banks 에서 accountNum 을 이용해 통장과 연결된 계좌의 유효성 유무를 검색하고 존재하면 ret.getAccount 을 반환한다.
Exceptional Courses of Events	banks 에도 존재하지 않는 accountNum 의 경우 null 을 반환한다.

### 3) Bank class

Type	class
Name	Bank
Purpose	Bank 정보를 모아두는 클래스
Overview(class)	Attribute : String name, String path, ArrayList<Card> cards, ArrayList<Bankbook> bankbooks

	Method : Bank(), loadDataBase() fileReader(), validateBankbook(), validateCard(), getBalance(), transaction(), getLogs(), getCriminalLogs(), getName()
CrossReference	Use Case R2, R3, R4, R5, R6, R7
Exceptional Courses of Events	N/A

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

Type	Method
Name	loadDataBase()
Purpose	은행의 정보가 기록된 File 로부터 Card 와 Bankbook 을 읽어온다.
CrossReference	Use Case R2
Input(Method)	-
Output(Method)	-

Abstract Operation(Method)	Bank 에 관련된 File 내에 저장되어 있는 account 와 card, bankbook 의 데이터를 불러온다.
Exceptional Courses of Events	-

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 log 를 읽어와 반환한다.
Exceptional Courses of Events	path 가 올바르지 않을시 Error

Type	Method
Name	validateBankbook()
Purpose	bankbook 이 유효한지 확인한다.
CrossReference	Use Case R2, R7
Input(Method)	String accountNum
Output(Method)	Bankbook bankbook

Abstract Operation(Method)	banks 의 통장중 accountNum 에 해당하는 통장이 존재하는지 확인하고 존재하면 해당 bankbook 을 반환한다.
Exceptional Courses of Events	banks 의 통장중 accountNum 에 해당하는 통장이 존재하는지 확인하고 존재하지 않으면 null 을 반환한다.

Type	Method
Name	validateCard()
Purpose	Card 가 유효한지 확인한다.
CrossReference	Use Case R2
Input(Method)	String pin
Output(Method)	Card card
Abstract Operation(Method)	cards 에서 pin 이 일치하는 카드가 존재하는지 확인하고 존재하면 해당 card 를 반환한다.
Exceptional Courses of Events	cards 에서 pin 이 일치하는 카드가 존재하는지 확인하고 존재하지 않으면 null 을 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)	account 의 balance 를 반환한다.
Exceptional Courses of Events	-

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)	account 의 log 파일에 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)	account 의 거래내역 로그파일을 읽어와 반환한다.
Exceptional Courses of Events	-

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)	account 의 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)	Bank 의 name 을 반환한다.
Exceptional Courses of Events	-

#### 4) Bankbook class

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

Type	Method
Name	Bankbook()
Purpose	Bankbook 의 Attribute 를 설정한다.
CrossReference	Use Case R2
Input(Method)	Account account
Output(Method)	-
Abstract Operation(Method)	Bankbook 의 Attribute 를 설정하고, this.account 를 input 된 account 로 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)	Bankbook 의 account 를 반환한다.
Exceptional Courses of Events	-

**5) Card class**

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

Type	Method
Name	Card()
Purpose	Card 의 Attribute 를 초기화한다.

CrossReference	Use Case R2
Input(Method)	Account account String pin String password
Output(Method)	-
Abstract Operation(Method)	Card 의 Attribute 를 초기화하고, this.account 를 input 된 account 로, this.password 를 input 된 password 로, this.pin 을 input 된 pin 로 설정한다.
Exceptional Courses of Events	-

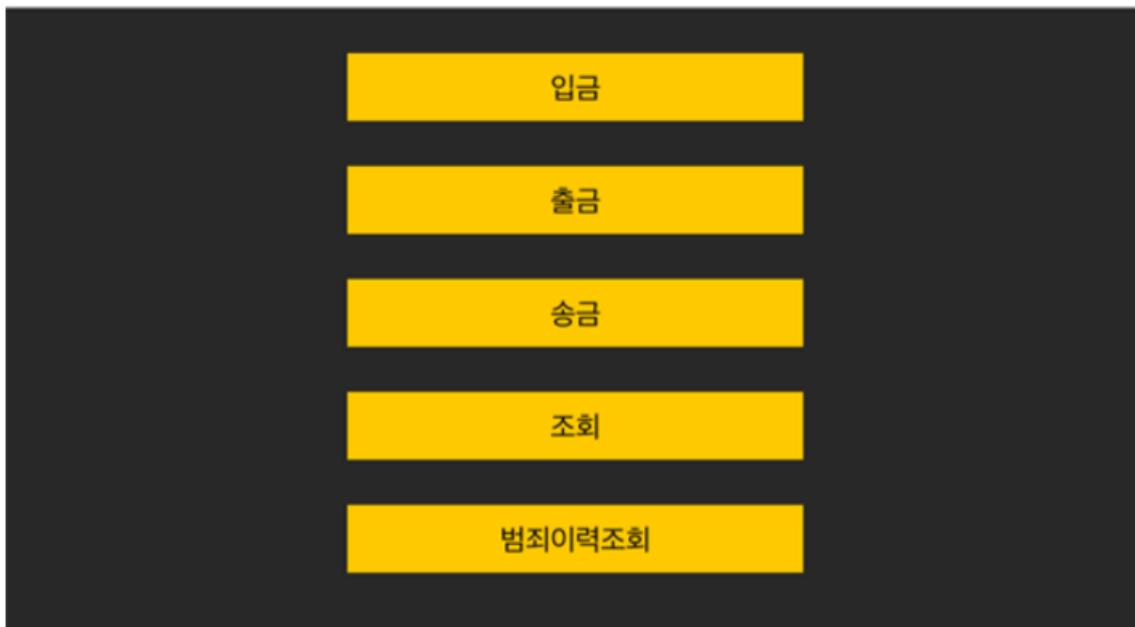
Type	Method
Name	getAccount()
Purpose	카드와 연결된 계좌를 확인한다.
CrossReference	Use Case R2
Input(Method)	-
Output(Method)	Account account
Abstract Operation(Method)	Card 의 account 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getPin()

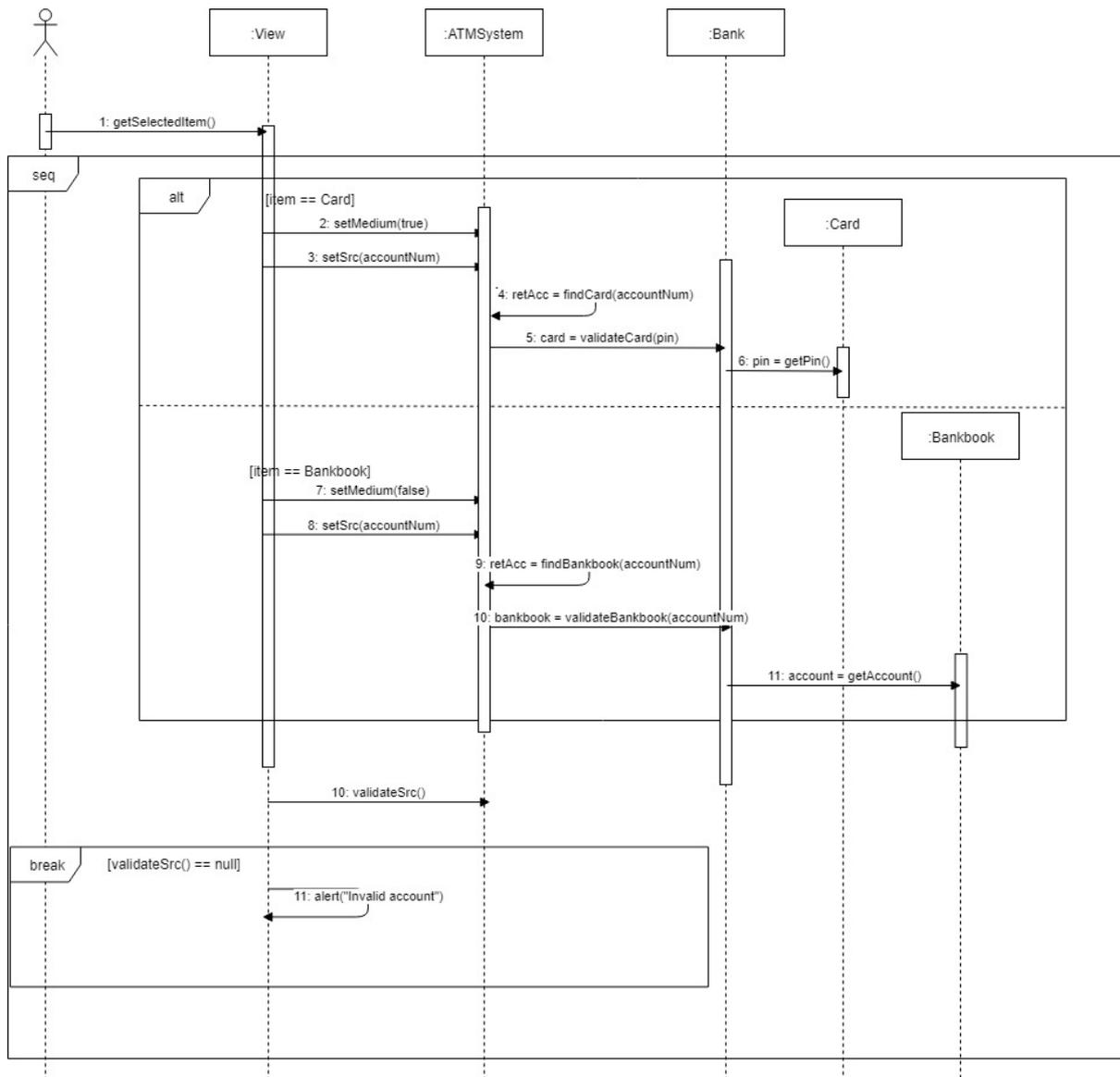
Purpose	카드번호를 확인한다.
CrossReference	Use Case R2
Input(Method)	-
Output(Method)	String pin
Abstract Operation(Method)	Card 의 pin 을 반환한다.
Exceptional Courses of Events	-

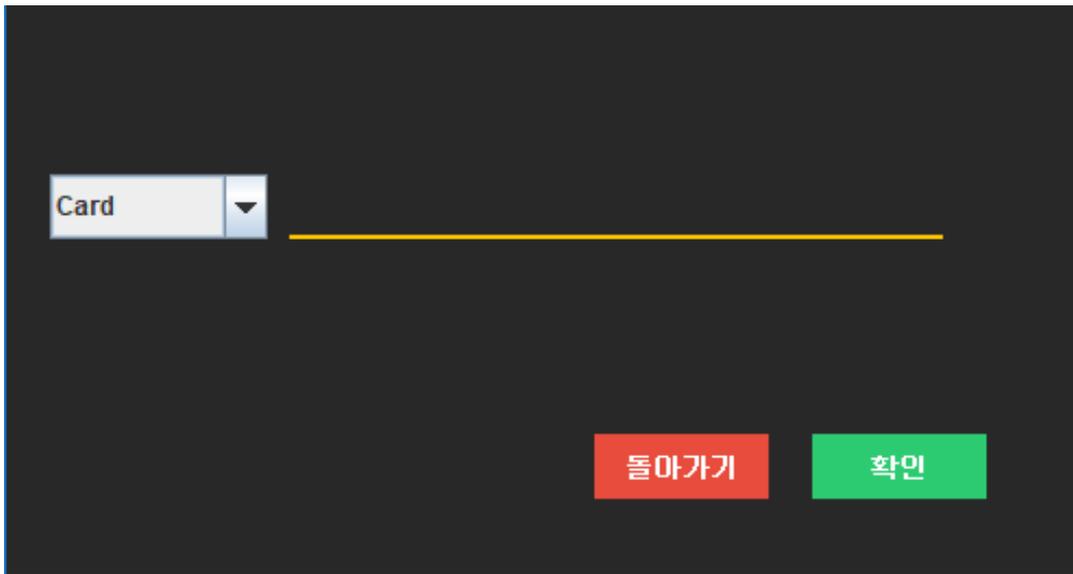
Type	Method
Name	checkPassword
Purpose	비밀번호가 맞는지 확인한다.
CrossReference	Use Case R2
Input(Method)	String pw
Output(Method)	this.password.equals(pw)
Abstract Operation(Method)	비밀번호 일치여부를 boolean 값으로 반환한다.
Exceptional Courses of Events	-

### Activity 2052. Implements Windows

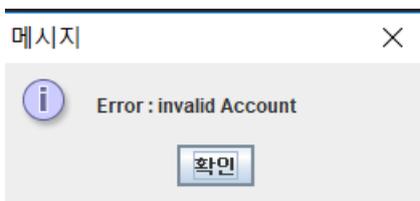


Name	SelectMenuView
Type	GUI
Responsibilities	입금, 출금, 송금, 거래 내역 조회, 범죄이력 버튼 중 하나를 누른다.
Cross Reference	Functional Requirement: R 1
Notes	입금, 출금, 송금, 조회, 범죄이력 조회 버튼 중 하나를 누른다.
Post-Condition	선택한 버튼의 기능으로 진행한다.
Pre-Condition	ATM이 켜진 상황이어야 한다.



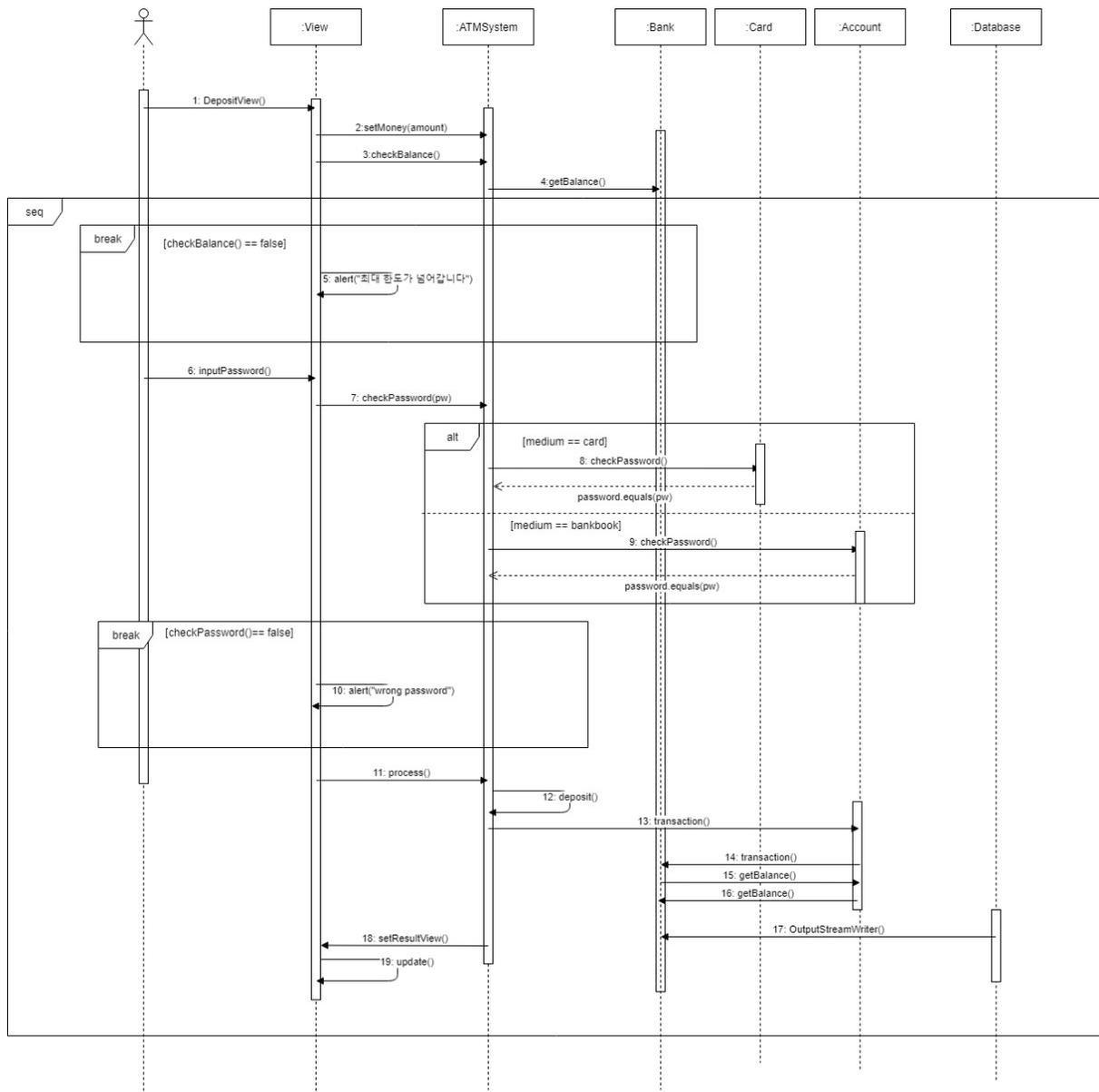


Name	InputSrcView()
Type	GUI
Responsibilities	카드 번호를 입력할지 통장 번호를 입력할지 선택하고 그에 따라 카드 번호 또는 통장 번호를 입력할 수 있다.
Cross Reference	Functional Requirement: R 2
Notes	카드와 통장 중 하나를 선택한 뒤 번호를 입력한다.
Post-Condition	이전 단계에서 선택한 기능의 다음 단계로 계속 진행한다.
Pre-Condition	메뉴 선택이 되어 있다.



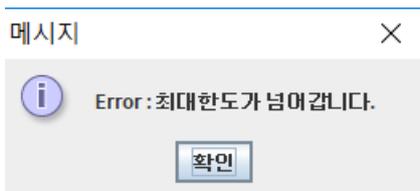
Name	alert "Invalid Account"
------	-------------------------

Type	GUI
Responsibilities	계좌가 유효하지 않음을 알린다.
Cross Reference	R 2
Notes	계좌가 유효하지 않음을 알린다.
Post-Condition	확인 버튼을 누를 수 있다.
Pre-Condition	유효하지 않은 계좌를 입력한다.

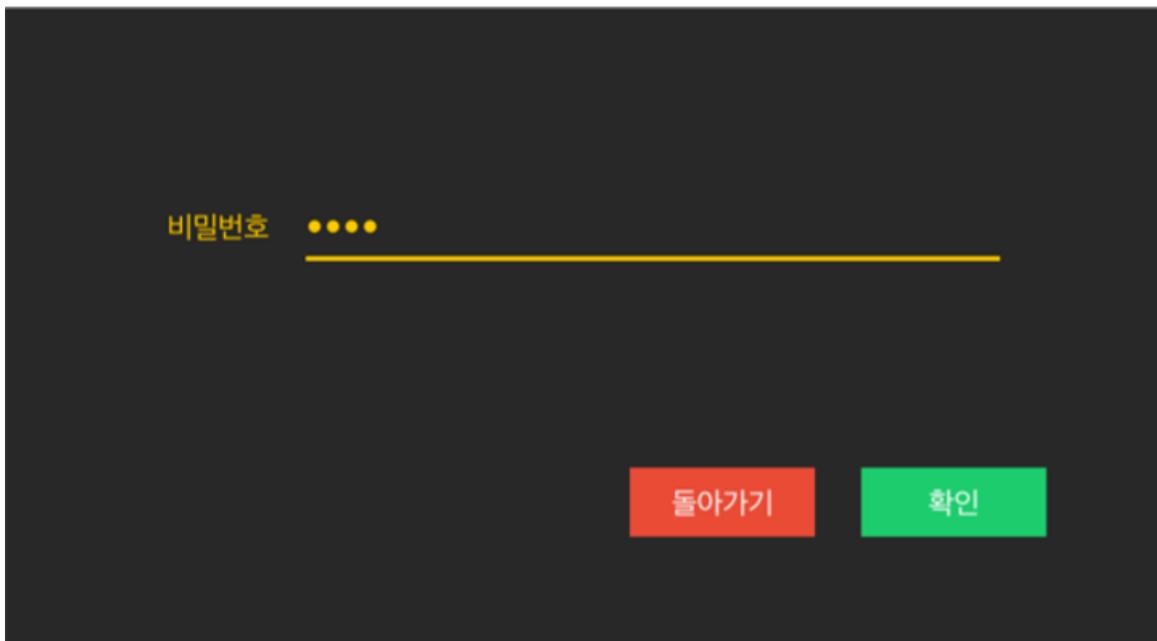




Name	DepositView()
Type	GUI
Responsibilities	입금할 금액을 입력할 수 있다.
Cross Reference	Functional Requirement: R 3
Notes	
Post-Condition	비밀번호를 입력할 수 있다.
Pre-Condition	카드 번호나 통장 번호를 입력하고 해당 번호가 유효한 상태여야 한다.



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



Name	inputPassword
Type	GUI
Responsibilities	이전 단계에서 선택한 매체(통장 혹은 카드)에 대한 비밀번호를 입력할 수 있다.

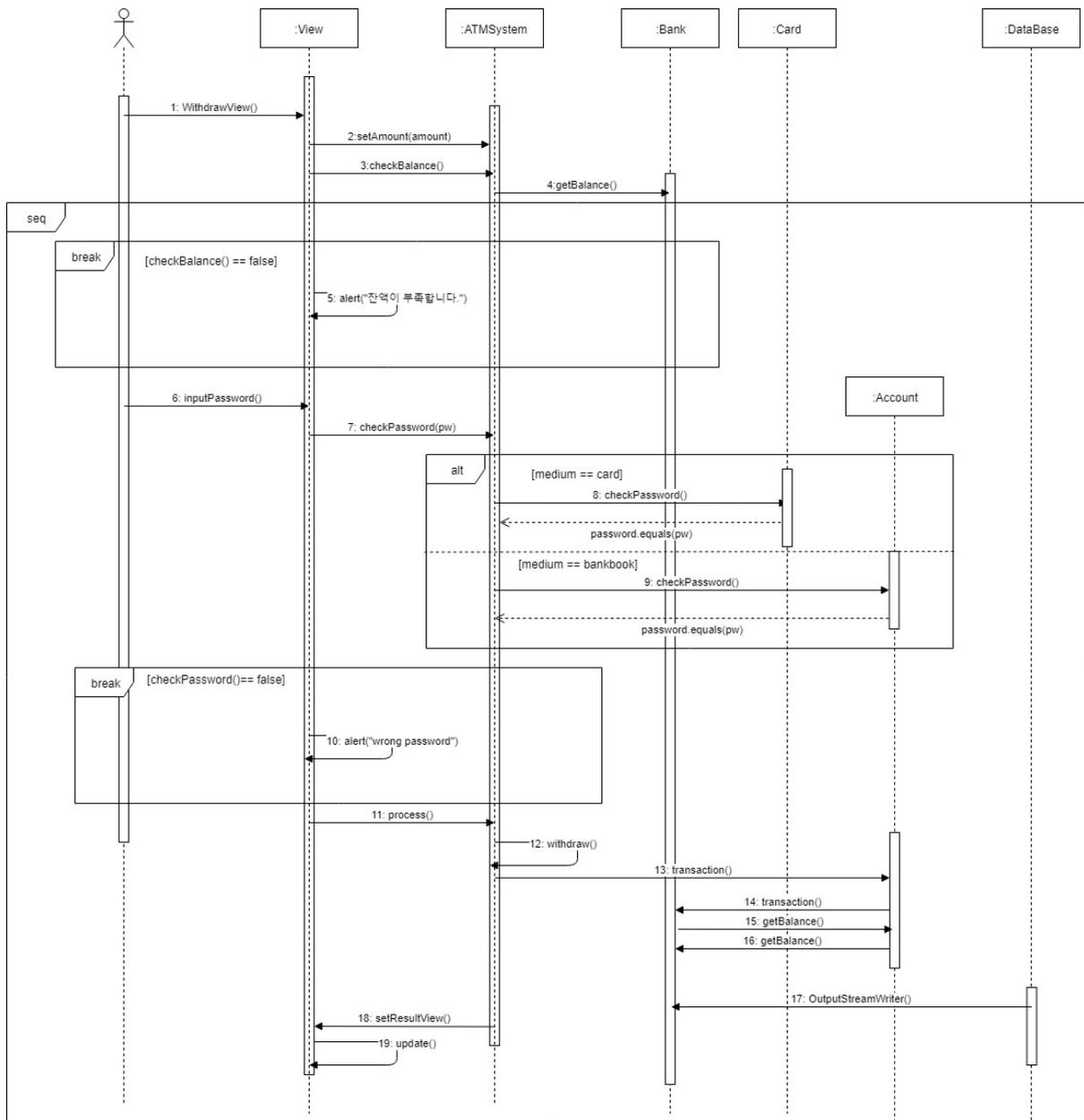
Cross Reference	Functional Requirement: R 3
Notes	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
Post-Condition	비밀번호가 유효할 경우 입금이 계속 진행되고 결과창이 뜬다.
Pre-Condition	입금할 돈을 입력받은 상황이어야 한다.



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

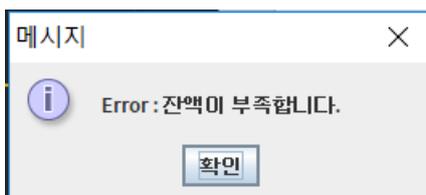


Name	update
Type	GUI
Responsibilities	입금이 완료된 후 결과와 내역을 보여준다.
Cross Reference	R 3
Notes	작업, 금액, 잔액의 입금 내역을 보여준다.
Post-Condition	돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다.
Pre-Condition	유효한 비밀번호를 입력하고 입금이 계속 진행된다.

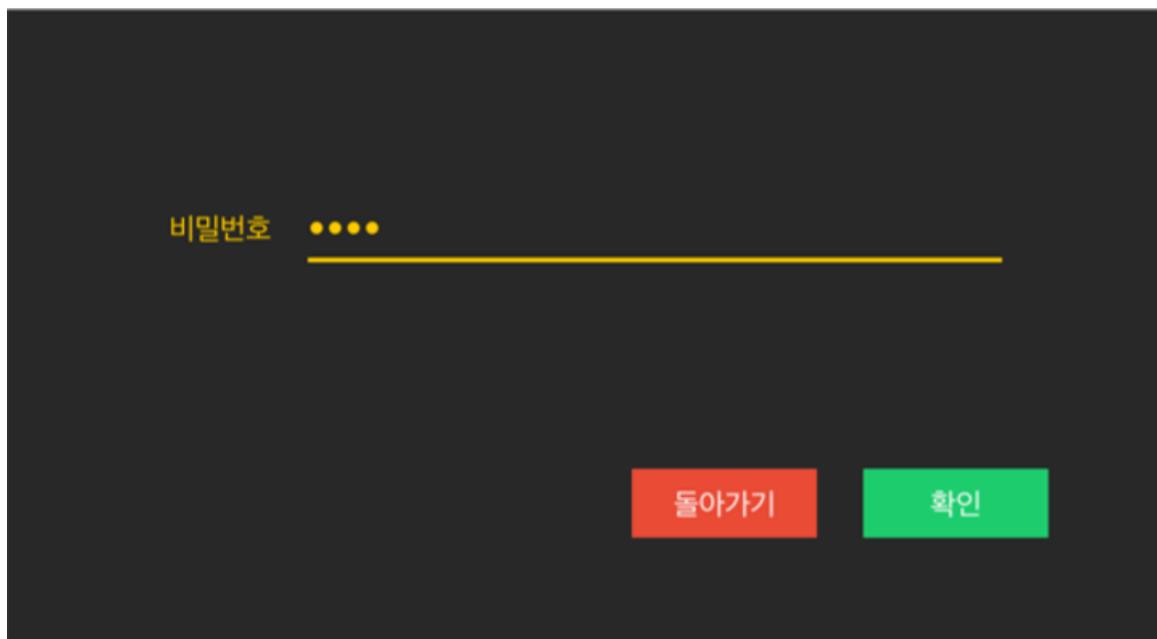




Name	WithdrawView()
Type	GUI
Responsibilities	출금할 금액을 입력할 수 있다.
Cross Reference	Functional Requirement: R 4
Notes	
Post-Condition	비밀번호를 입력할 수 있다.
Pre-Condition	카드 번호나 통장 번호를 입력하고 해당 번호가 유효한 상태여야 한다.

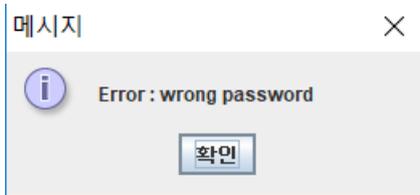


Name	alert "잔액이 부족합니다"
Type	GUI
Responsibilities	계좌에서 출금할 돈이 잔고보다 더 클 때 알린다.
Cross Reference	Functional Requirement: R 4
Notes	
Post-Condition	확인 버튼을 누를 수 있다.
Pre-Condition	잔고보다 더 큰 돈을 입력한다.



Name	inputPassword
Type	GUI

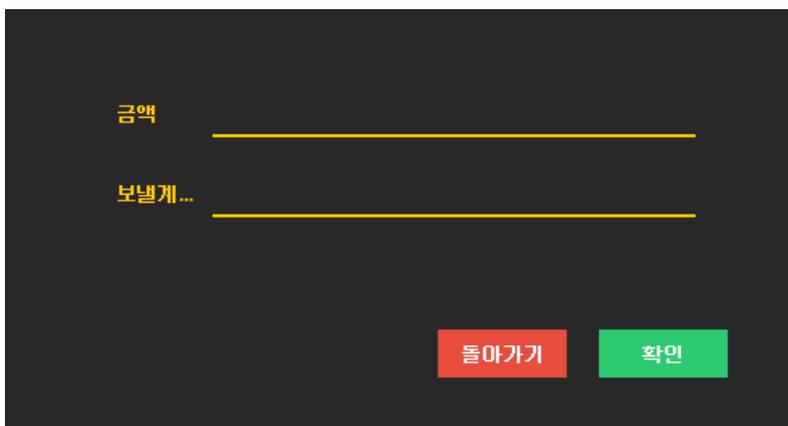
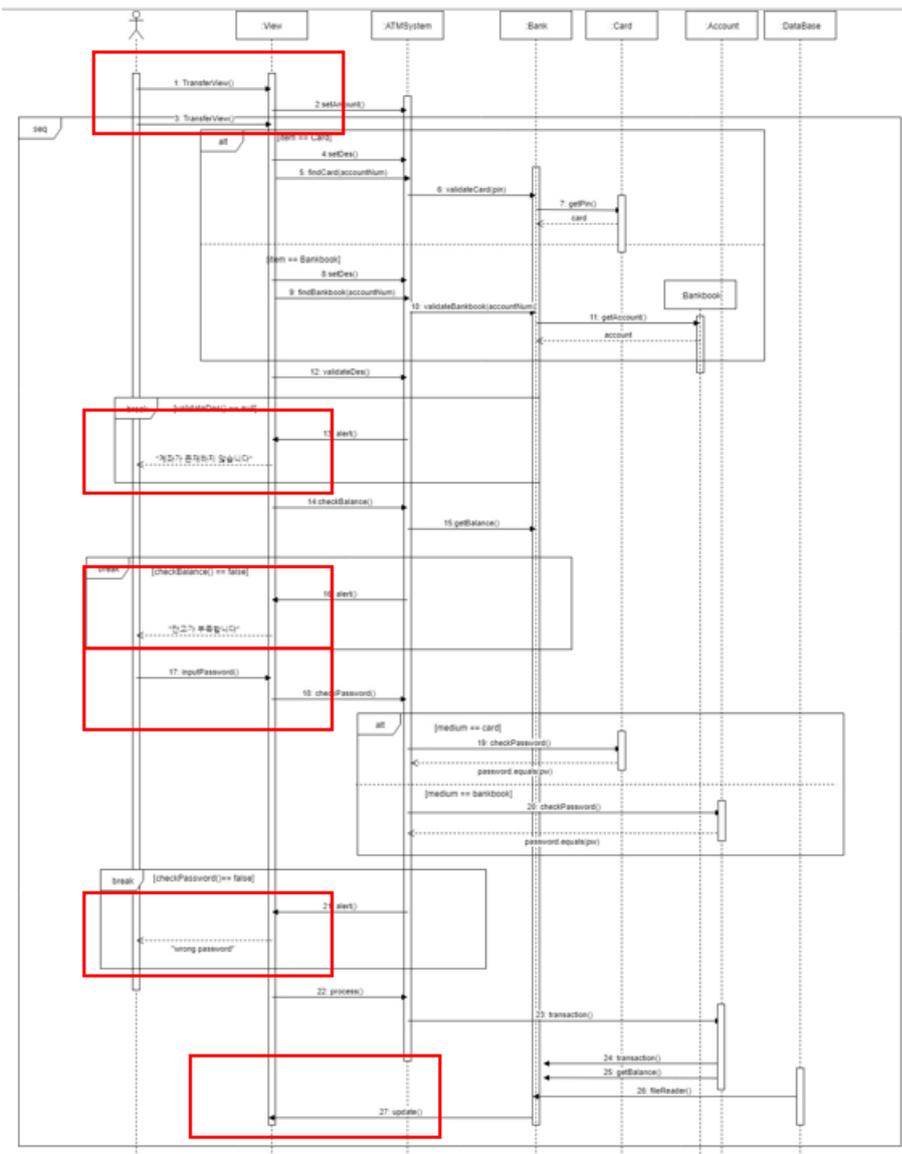
Responsibilities	계좌의 비밀번호를 입력할 수 있다.
Cross Reference	Functional Requirement: R 4
Notes	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
Post-Condition	비밀번호가 유효할 경우 출금이 계속 진행되고 결과창이 뜬다.
Pre-Condition	출금할 돈을 입력받은 상황이어야 한다.



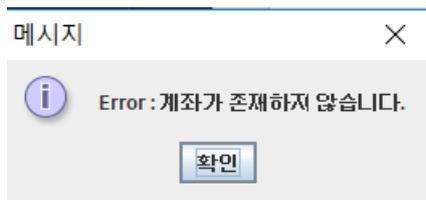
Name	alert "wrong password"
Type	GUI
Responsibilities	비밀번호가 유효하지 않음을 알린다.
Cross Reference	Functional Requirement: R 4
Notes	비밀번호가 유효하지 않음을 알린다.
Post-Condition	확인 버튼을 누를 수 있다.
Pre-Condition	유효하지 않은 비밀번호를 입력한다.



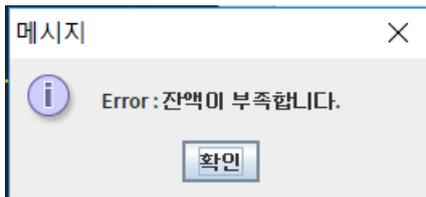
Name	update
Type	GUI
Responsibilities	출금이 완료된 후 결과와 내역을 보여준다.
Cross Reference	R 4
Notes	작업, 금액, 잔액의 출금 내역을 보여준다.
Post-Condition	돌아가기 버튼 또는 끝내기 버튼을 누를 수 있다.
Pre-Condition	유효한 비밀번호를 입력하고 출금이 계속 진행된다.



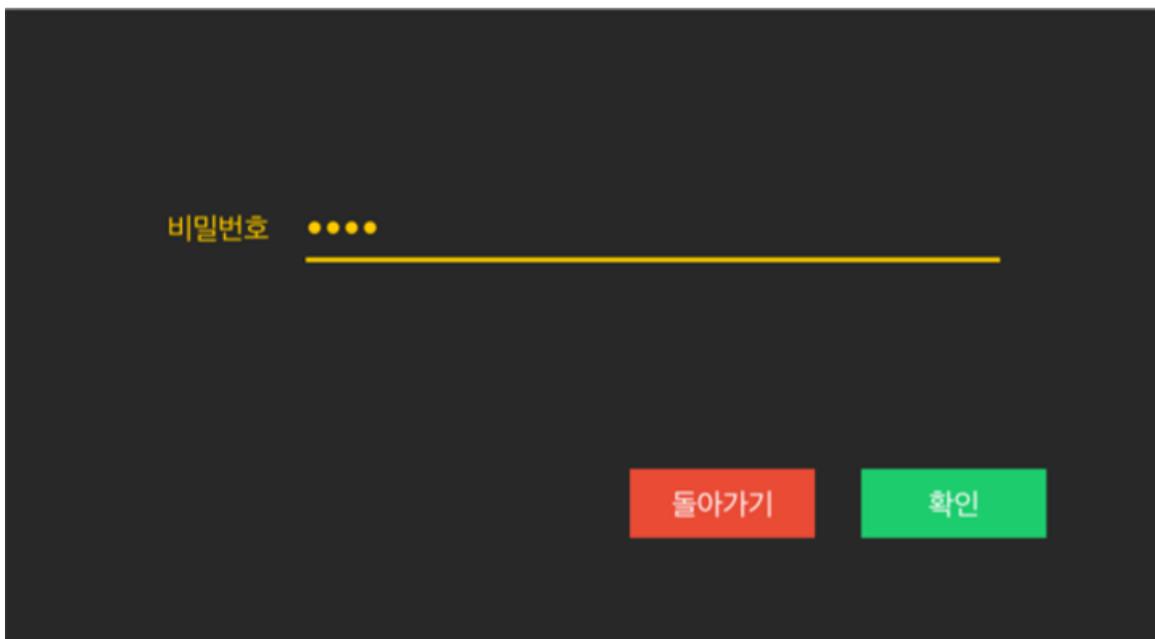
Name	TransferView()
Type	GUI
Responsibilities	송금할 금액과 보낼 계좌 번호를 입력할 수 있다.
Cross Reference	Functional Requirement: R 5
Notes	
Post-Condition	비밀번호를 입력할 수 있다.
Pre-Condition	카드 번호나 통장 번호를 입력하고 해당 번호가 유효한 상태여야 한다.



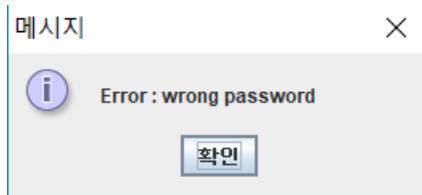
Name	alert "계좌가 존재하지 않습니다"
Type	GUI
Responsibilities	계좌가 유효하지 않음을 알린다.
Cross Reference	Functional Requirement: R 5
Notes	계좌가 유효하지 않음을 알린다.
Post-Condition	확인 버튼을 누를 수 있다.
Pre-Condition	유효하지 않은 계좌를 입력한다.



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

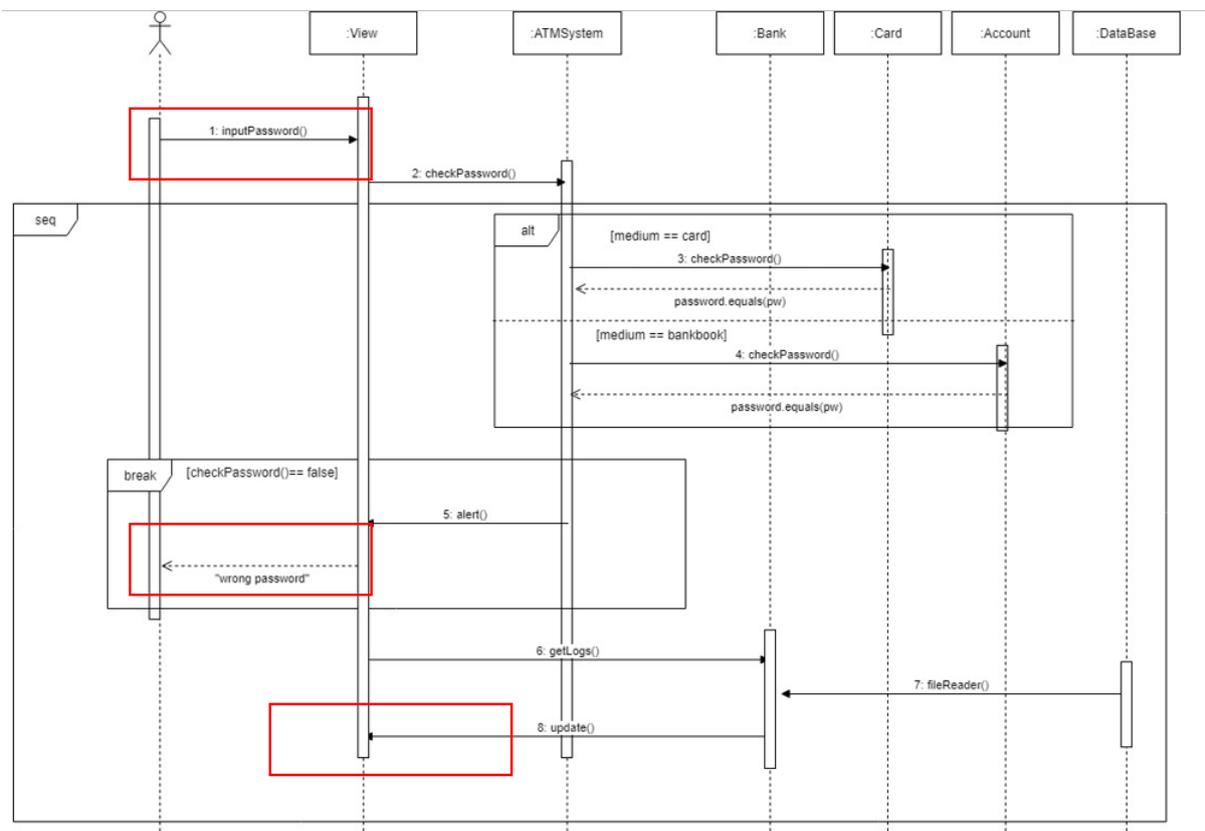


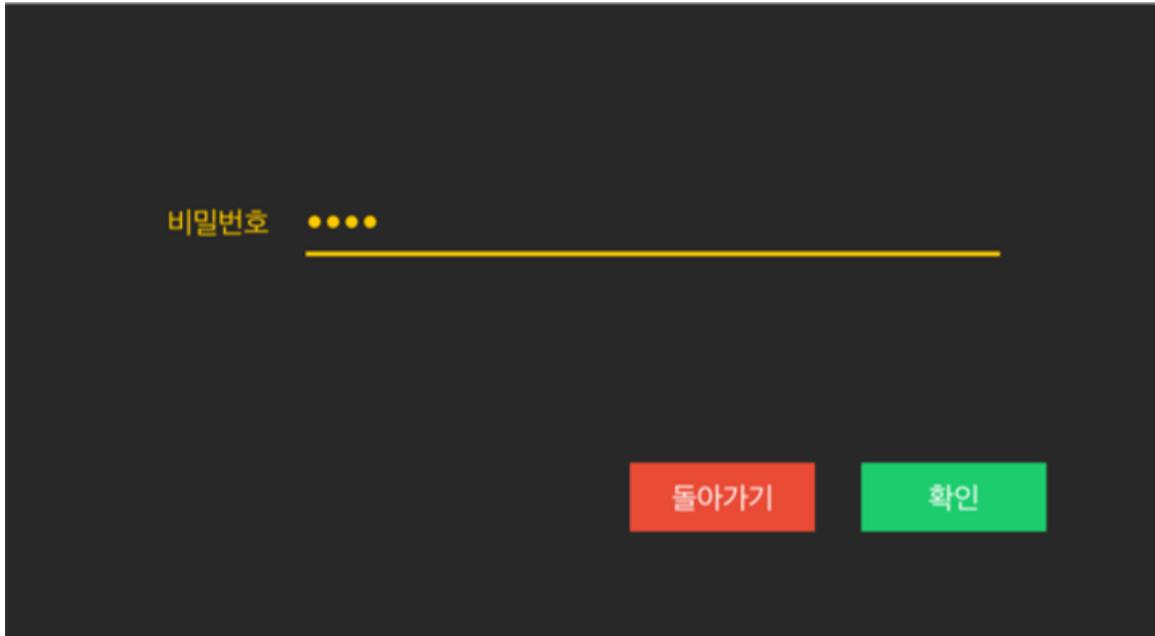
Name	inputPassword
Type	GUI
Responsibilities	계좌의 비밀번호를 입력할 수 있다.
Cross Reference	Functional Requirement: R 5
Notes	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
Post-Condition	비밀번호가 유효할 경우 송금이 계속 진행되고 결과창이 뜬다.
Pre-Condition	송금할 돈을 입력받은 상황이어야 한다.



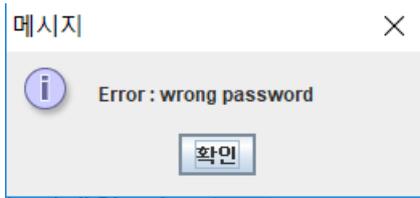
Name	alert "wrong password"
Type	GUI
Responsibilities	비밀번호가 유효하지 않음을 알린다.
Cross Reference	Functional Requirement: R 5
Notes	비밀번호가 유효하지 않음을 알린다.
Post-Condition	확인 버튼을 누를 수 있다.
Pre-Condition	유효하지 않은 비밀번호를 입력한다.

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



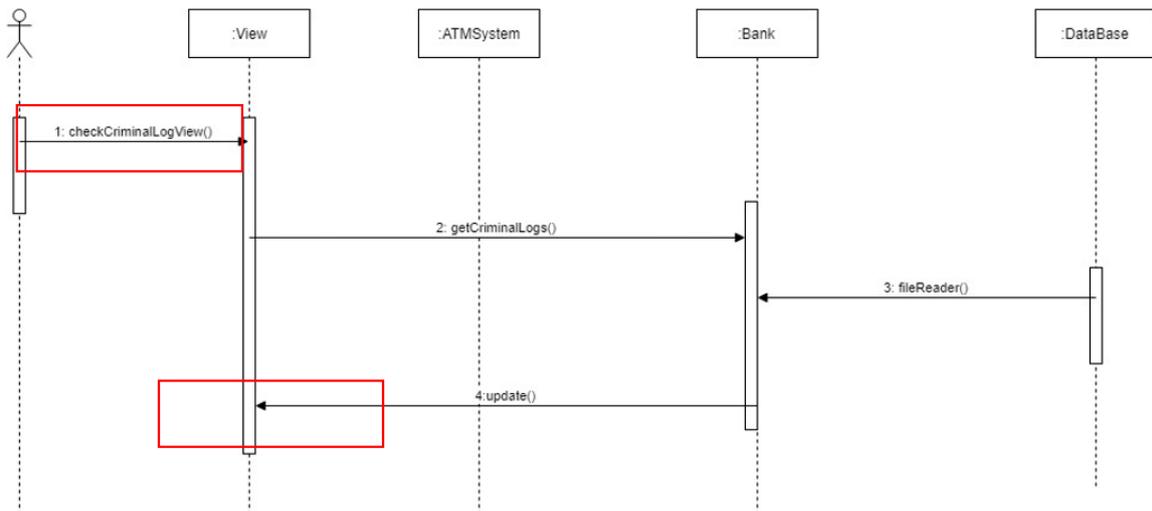


Name	inputPassword
Type	GUI
Responsibilities	계좌의 비밀번호를 입력할 수 있다.
Cross Reference	Functional Requirement: R 6
Notes	비밀번호를 입력받은 뒤 유효하지 않을 경우 에러 메시지를 출력하는 화면으로 넘어간다.
Post-Condition	비밀번호가 유효할 경우 거래내역조회가 계속 진행되고 결과창이 뜬다.
Pre-Condition	거래내역 조회를 선택한 상황이어야 한다.



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

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



Name	update
Type	GUI
Responsibilities	범죄 이력 내역을 보여준다.
Cross Reference	R 7
Notes	계좌의 범죄 이력 내역을 보여준다.
Post-Condition	돌아가기 버튼을 누를 수 있다.
Pre-Condition	유효한 계좌를 입력하고 범죄 이력 조회가 계속 진행된다.

## Activity 2063. System Testing

Test Number	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	MediumCheck	카드 번호를 입력할지	R 2	P

_002		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 진행 되는지 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
ATM_STC_003_001	Deposit	입금할 금액을 잘 입력받을 수 있는지 확인한다.	R 3	P
ATM_STC_003_002	Deposit	계좌 번호에 해당되는 비밀번호를 입력했을 때 수수료를 더하고 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 결과 정상적으로 출력	R 3	P

		되는지 확인한다.		
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 입력했을 때 수수료를 더하고 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 결과로 정상적으로 출력되는지 확인한다.	R 4	P
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	Transfer	수수료를 포함하여 송	R 5	P

_003		금할 계좌의 잔액이 충분하지 않을 경우 송금 진행이 멈추는지 확인한다.		
ATM_STC_005_004	Transfer	계좌번호에 해당되는 비밀번호를 입력되었을 때 수수료를 더하고 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_001	CheckTransactionHistory	입력한 계좌의 거래내역이 화면에 정상적으로 출력되는지 확인한다.	R 6	P
ATM_STC_006_002	CheckTransactionHistory	입력한 카드번호의 거래내역이 화면에 정상적으로 출력되는지 확인한다.	R 6	p
ATM_STC_007_001	CheckCriminalHistory	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
ATM_STC_007_003	CheckCriminal History	입력한 카드번호의 범죄이력내역이 화면에 정상적으로 출력되는지 확인한다.	R 7	p
Identifier	Non-Functional Requirements	Description		
ATM_STC_008	Performance	모든 버튼 입력 및 이벤트에 대하여 1초 이내로 반응하는지 확인한다.	R 9	F
ATM_STC_009_001	OS Independent	프로그램이 window10에서 정상적으로 동작하는지 확인한다.	R 10	P
ATM_STC_009_002	OS-Independent	프로그램이 Mac OS에서 정상적으로 동작하는지 확인한다.	R 10	p