

# JackPot ATM

## OOPT Stage 2nd cycle

### Team 5

Date

2018-04-16

201312259 백만일

201112052 방민석

201211383 조영래

**Stage 1000. Modified Part**

**Stage 2030. Modified Part**

**Stage 2040. Modified Part**

**Stage 2050. Modified Part**

**Stage 2060. Modified Part**

**Bug Fix**

## Stage 1000. Modified Part

### 1001. Define Draft Plan

#### - 4. Non Functional Requirements

- “금융 거래 프로그램이므로 보안이 잘 되어야 한다.”
- 이후문서에서보안에대한언급이존재하지않음

How to fix -> Non Functional Requirements 자체는 다른 스테이지 상에서 구현에 있어 구체적으로 어떻게 구현할 것인가를 명세하는 내용이 존재하지 않기때문에 배재하였으며 “Private”, “Public” 의 적절한 사용을 통하여 시스템에서 중요한 정보를 아무 사용자나 읽을 수 없도록 하였다.

### 1003. Define Requirements - 1. Functional Requirements

- “성공한 거래의 송금금액/입금계좌/이름을 보여줌”
- 이후단계에서거래후잔고를보여주는명세가존재하지않음

How to fix -> ‘거래 후 잔고를 출력’ 하는것으로 구현 명세를 수정하였다.

- 문서 전체에서 “withdraw”에 대한 해석이 통일되어있지 않음(출금/인출)

How to fix -> 전체 문서 내용을 출금으로 통일하였다.

## Stage 2030. Modified Part

### 2031. Define Essential use cases

- 2010 까지 존재하던 ‘수수료’와 ‘명세서 출력’에 대한 서술이 모두 사라짐
- 1. Withdraw
  - “출금 화면을 출력한다”, “출금 금액을 입력한다” 표현이 모호함
  - ”ATM 의 잔고와 출금 금액을 비교한다” 내용 추가

Use Case	withdraw
Actor	User
Purpose	사용자 계좌에 있는 돈을 출금한다.
Overview	사용자가 ATM 기의 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	ATM 에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User 가 출금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User 는 계좌를 입력한다. 4.(S) 유효한 계좌인지 확인한다. 5.(S) 출금하고자 하는 금액 입력 화면을 출력한다. 6.(A) 출금하고자 하는 금액을 입력한다. 7.(S) 비밀번호 입력 화면을 출력한다. 8.(A) 비밀번호를 입력한다. 9.(S) 비밀번호가 일치하는지 확인한다. 10.(S) User 의 잔고와 출금금액을 비교한다. 11.(S) User 의 잔고를 출금금액만큼 감소시킨다. 12.(S) ATM 잔액을 감소시킨다. 13.(S) ATM 각 지폐별 잔고를 감소시킨다. 13.(S) 거래후 사용자의 계좌 잔고를 출력한다.
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

• 2031. Define Essential use cases - 2. Deposit

”계좌 잔액을 표시한다” 내용 추가

Use Case	deposit
Actor	User
Purpose	사용자 계좌에 돈을 입금한다.
Overview	사용자가 ATM 기의 입금 버튼을 누르고 입금할 계좌를 입력한다. 사용자가 입금할 금액을 기기에 넣으면 ATM 이 금액을 세고, 그 액수만큼 입력된 계좌의 잔고를 증가시킨다.

Type	Primary and Essential
Cross Reference	System Function R.2
Pre-Requisites	ATM 에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User 가 입금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User 는 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. 9.(S) ATM 각 지폐별 잔고를 증가시킨다. 10.(S) 거래 후 계좌 잔액을 표시한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1: 잘못된 계좌를 입력받으면 에러 메시지를 출력한다

- 2031. Define Essential use cases - 3. Remittance
- "계좌 잔액을 표시한다" 내용 추가
- Exceptional courses of Events : "자기 자신의 계좌에는 송금할 수 없다"

Use Case	remittance
Actor	User
Purpose	사용자 계좌의 돈을 다른 사용자 계좌로 송금한다.
Overview	사용자가 ATM 기의 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	ATM 에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.

Typical Courses of Events	(A):Actor, (S):System 1.(A) User 가 송금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User 는 계좌를 입력한다. 4.(S) 계좌가 유효한지 확인한다. 5.(S) 송금받을 계좌 입력 화면을 출력한다. 6.(A) User 는 송금받을 계좌를 입력한다. 7.(S) 계좌가 유효한지 확인한다. 8.(A) User 는 송금 금액을 입력한다. 9.(A) User 는 비밀번호를 입력한다. 10.(S) 비밀번호가 일치하는지 확인한다. 11.(S) User 의 잔고와 송금금액을 비교한다. 12.(S) 받는사람의 통장잔고를 증가시킨다. 13.(S) 보내는 사람의 통장잔고를 감소시킨다. 14.(S) 거래후 계좌 잔고를 출력한다.
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 E5: 자기 자신의 계좌에는 송금할 수 없다.

- 5. Random\_Jackpot

- "user 가 돈을 인출했을 때" 표현이 모호함
- pre-requisites : 사용자가 인출 중이어야한다.

Use Case	Random_Jackpot
Actor	User
Purpose	출금 기능을 이용하는 임의의 사용자에게 추가로 5 만원을 지급한다.
Overview	00 시가 되면 ATM 기의 몇번째 출금 고객에게 추가로 5 만원이 출금될지 정해진다. 해당 순서의 사용자가 출금시에 5 만원이 추가로 출금된다.
Type	Primary and Essential

Cross Reference	System Function R.1.1
Pre-Requisites	User 가 출금 거래중이어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(S) 00 시가 되면 당첨되는 순서를 랜덤하게 선택한다. 2.(S) User 가 ATM 을 통해 출금거래를 완료하였을때 해당 User 가 해당 순서의 사용자인지 확인한다. 3.(S) 해당 User 에게 돈을 5 만원 더 추가로 출금해준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

- 6. manage\_ATM

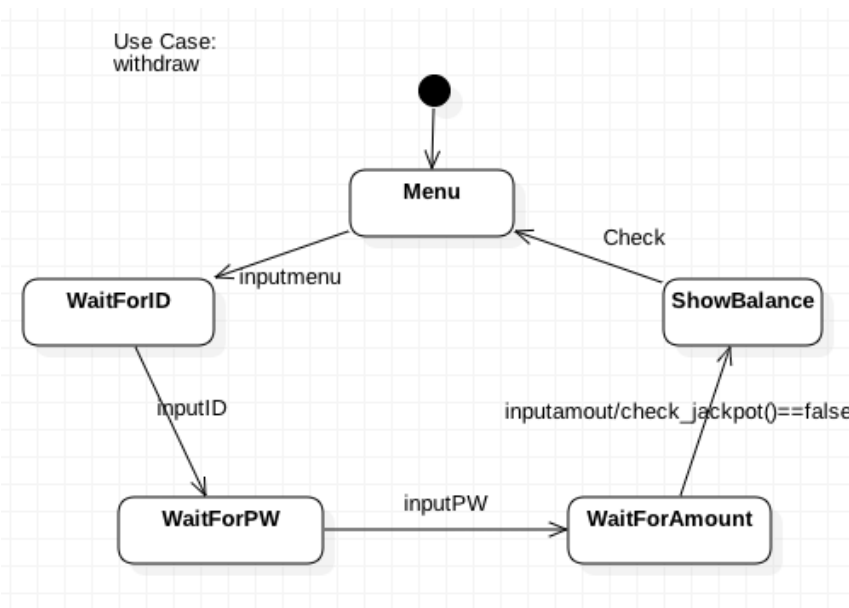
- “갱신 후 관리자의 입출금 선택화면으로 돌아간다” 내용 추가
- Exceptional Courses of Events: ATM 잔고의 최댓값에 대한 내용 추가

Use Case	manage_ATM
Actor	Manager
Purpose	관리자가 ATM 의 잔고를 추가하거나 뺄 수 있다
Overview	ATM 의 잔고가 일정 수준 이하로 낮아지면 관리자가 ATM 의 잔고를 추가하고 잔고가 일정 수준 이상으로 많아지면 ATM 의 잔고를 뺀다
Type	Primary and Essential
Cross Reference	System Function R.5
Pre-Requisites	N/A
Typical Courses of Events	(A):Actor, (S):System 1.(A) 관리자가 관리 메뉴 버튼을 누른다. 2.(S) ID 와 입력 화면을 출력한다 3.(A) 관리자가 ID 를 입력한다 4.(S) 유효한 관리자 아이디인지 확인한다. 5.(A) PW 입력화면을 출력한다. 6.(A) 관리자의 PW 를 입력한다 7.(S) 관리자의 PW 를 확인한다 8.(S) 잔고를 추가할 것인지 뺄것인지 선택하는 메뉴를

	<p>출력한다.</p> <p>9.(A) 관리자가 잔고를 추가할 것인지 뺄 것인지 메뉴를 선택한다.</p> <p>10.(S) 금액을 입력하는 메뉴를 출력한다.</p> <p>11.(A) 입금 혹은 출금하고자 하는 금액을 입력한다.</p> <p>12.(S) 입금 혹은 출금하고자 하는 금액과 ATM 잔고를 비교하여 ATM 각 지폐 잔고의 최솟값(0) 또는 최댓값(500)을 벗어나지 않는지 확인한다.</p> <p>12.(S) ATM 에 변화된 잔고를 갱신한다</p> <p>13.(S) 갱신 후 관리자의 입출금 선택화면으로 돌아간다.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>E1. 관리자의 ID 와 PW 가 잘못된 경우 에러 메시지를 출력한다</p> <p>E2. 빼낼 잔고가 ATM 잔고보다 많을 경우 에러 메시지를 출력한다</p> <p>E3. ATM 각 화폐별 최대잔고는 500 매로 하며 이를 초과할 경우 관리자 혹은 일반 입금에 있어서 오류를 출력한다.</p>

• 2037. Define State Diagram

- WaitForAmount 와 Menu 사이에 잔고를 보여주는 상태가 존재



• 2038. Refine System Test Case



- 1009 에서 정의한 System test case 와 항목 및 내용이 완전히 달라짐

How to fix -> 해당 Refine System Test Case 단계는 1000 단계를 바탕으로 2030 에서 더욱 세부적으로 시스템을 분석하여 나오는 테스트 케이스들만큼 세부적으로 테스트케이스의 항목들이나 내용이 새롭게 정의될 수 있다.

## Stage 2040. Modified Part

- 2041. Design Real Use Cases
- 2031 을 바탕으로 전체적인 수정 필요
- 1. Withdraw
  - TotalMoney 뿐만 아니라 지폐권에 따른 각 잔고도 감소시킨다는 내용 추가
- 2. Deposit
  - TotalMoney 뿐만아니라 지폐권에 따른 각 잔고도 감소시킨다는 내용 추가
- 6. manage\_ATM
  - TotalMoney 뿐만 아니라 지폐권에 따른 각 잔고도 증가/감소시킨다는 내용 추가

Use Case	withdraw
Actor	User
Purpose	사용자 계좌에 있는 돈을 출금한다.
Overview	사용자가 ATM 기의 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	ATM 에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.

Typical Courses of Events	<p>(A):Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) User 가 withdraw 버튼을 누른다.</li> <li>2.(S) 계좌 ID 입력 화면을 출력한다.</li> <li>3.(A) User 는 계좌(inputID)를 입력한다.</li> <li>4.(S) Account.ID 와 inputID 를 비교 확인한다.</li> <li>5.(S) 출금 화면을 출력한다.</li> <li>6.(A) 출금 금액(inputMoney)을 입력한다.</li> <li>7.(S) 계좌 password 입력 화면을 출력한다.</li> <li>8.(A) 비밀번호(inputPW)를 입력한다.</li> <li>9.(S) Account.password 와 inputPW 가 일치하는지 확인한다.</li> <li>10.(S) Account.Balance 와 inputMoney 을 비교한다.</li> <li>11.(S) Account.Balance 를 inputMoney 만큼 감소시킨다.</li> <li>12.(S) ATM 잔액(TotalMoney)을 감소시킨다.</li> <li>13.(S) ATM 각 지폐별 잔고를 감소시킨다.</li> <li>14.(S) 거래후 사용자의 계좌 잔고(Account.Balance)를 출력한다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>E1: 잘못된 계좌를 입력받으면 에러 메시지를 출력한다  E2: 비밀번호가 틀릴 경우 에러 메시지를 출력한다  E3: 계좌 잔고가 출금액보다 적을 경우 에러 메시지를 출력한다</p>

Use Case	deposit
Actor	User
Purpose	사용자 계좌에 돈을 입금한다.
Overview	사용자가 ATM 기의 입금 버튼을 누르고 입금할 계좌를 입력한다. 사용자가 입금할 금액을 기기에 넣으면 ATM 이 금액을 세고, 그 액수만큼 입력된 계좌의 잔고를 증가시킨다.

Type	Primary and Essential
Cross Reference	System Function R.2
Pre-Requisites	ATM 에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User 가 deposit 버튼을 누른다. 2.(S) 계좌 ID 입력 화면을 출력한다. 3.(A) User 는 계좌(inputID)를 입력한다. 4.(S) Account.ID 와 inputID 를 비교 확인한다. 5.(S) 금액 입력 화면을 출력한다. 6.(A) 입금 금액(inputMoney)을 입력한다. 7.(S) Account.Balance 를 inputMoney 만큼 증가시킨다. 8.(S) ATM 잔액(TotalMoney)을 증가시킨다. 9.(S) ATM 각 지폐별 잔고를 증가시킨다. 10. (S) 거래 후 계좌 잔액(Account.Balance)을 표시한다.
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 ATM 기의 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	ATM 에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	<p>(A):Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) User 가 remittance 버튼을 누른다.</li> <li>2.(S) 계좌 ID 입력 화면을 출력한다.</li> <li>3.(A) User 는 계좌(inputID)를 입력한다.</li> <li>4.(S) Account.ID 와 inputID 를 비교 확인한다.</li> <li>5.(S) 송금받을 계좌 입력 화면을 출력한다.</li> <li>6.(A) User 는 송금받을 계좌(inputID)를 입력한다.</li> <li>7.(S) Account.ID 와 inputID 를 비교한다.</li> <li>8.(A) User 는 송금 금액(inputMoney)을 입력한다.</li> <li>9.(S) User 의 잔고(Account.Balance)와 송금액(inputMoney)을 비교 확인한다.</li> <li>10.(A) User 는 비밀번호(inputPW)를 입력한다.</li> <li>11.(S) Account.password 와 inputPW 가 일치하는지 확인한다.</li> <li>12.(S) 받는사람의 통장잔고(Account.Balance)를 증가시킨다.</li> <li>13.(S) User 의 통장잔고(Account.Balance)를 감소시킨다.</li> <li>14.(S) 거래 후 계좌 잔액(Account.Balance)을 표시한다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>E1: 잘못된 계좌를 입력받으면 에러 메시지를 출력한다</p> <p>E2: 비밀번호가 틀릴 경우 에러 메시지를 출력한다</p> <p>E3: 송금 계좌가 잘못된 경우 에러 메시지를 출력한다</p> <p>E4: 계좌 잔고가 송금액보다 적을 경우 에러 메시지를 출력한다</p>
Use Case	manage_ATM
Actor	Manager

Purpose	관리자가 ATM 의 잔고를 추가하거나 뺄 수 있다
Overview	ATM 의 TotalMoney 가 일정 수준 이하로 낮아지면 관리자가 TotalMoney 를 추가하고 잔고가 일정 수준 이상으로 많아지면 TotalMoney 를 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"> <li>1.(A) Manager 가 manage 버튼을 누른다.</li> <li>2.(S) 관리자 ID 입력화면을 출력한다.</li> <li>3.(A) Manager 가 관리자 ID(inputID)를 입력한다</li> <li>4.(S) ManagerID 와 inputID 를 비교 확인한다</li> <li>5.(A) Manager 가 관리자 Password(inputPW)를 입력한다</li> <li>6.(S) ManagerPW 와 inputPW 를 비교 확인한다</li> <li>5.(S) 확인이 끝나면 Manager 가 화폐별 잔고에 접근할수 있도록 한다.</li> <li>6.(A) Manager 가 화폐별 잔고를 추가하거나 뺀다</li> <li>7.(S) GUI_Interface.TotalMoney 를 갱신한다</li> <li>8.(S) Count50000, Count10000, Count5000, Count1000 각 화폐별 금액을 갱신한다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>E1.MID 와 MPW 가 잘못된 경우 에러 메시지를 출력한다</p> <p>E2. inputAmount 가 ATM 잔고보다 많을 경우 에러 메시지를 출력한다</p> <p>E3. ATM ount50000, Count10000, Count5000, Count1000 화폐별 최대잔고는 500 매로 하며 이를 초과할 경우 관리자 혹은 일반 입금에 있어서 오류를 출력한다.</p>

- 2042. Define Reports UI, and Storyboards

- ‘돌아가기’버튼에 대한 명세가 전혀 없었으나 UI 단계에서 등장

How to fix -> 돌아가기 버튼의 경우 시스템 내부에서 동작하는 것이 아니라 GUI 로 출력되는 사용자 화면만을 처음으로 돌려서 사용자의 거래를 처음으로 돌리게 하는 것이기 때문에 GUI 단계 이전에 따로 나올 수 없음.

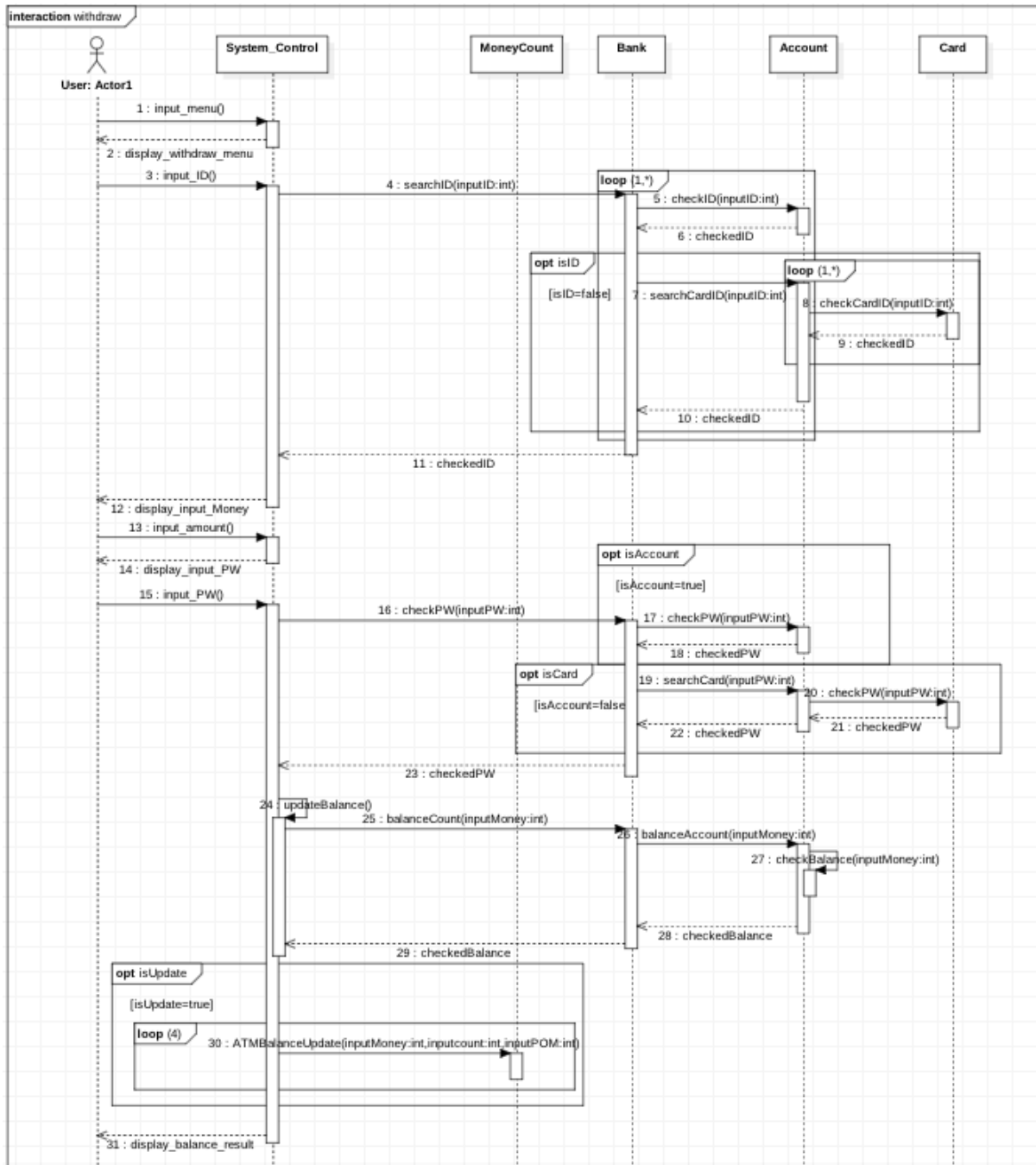
#### 2044. Define Interaction Diagrams

- 실제로 GUI\_Interface 라는 클래스는 존재하지 않음

- 대신 System\_Control & MainFrame 존재

- Timer 클래스를 사용하지않는 diagrams 에서는 생략 필요

How to fix -> 해당 단계는 GUI interface 는 시스템 컨트롤로 변경 MainFrame 의 경우에는 최종 코드이기 때문에 확인할수 있지만 2044 단계에서는 GUI 에 대한 diagram 을 고려하는 단계가 아니므로 존재할수 없음 그러므로 MainFrame 은 배제(하단 그림 참조)



## Stage 2050. Modified Part

Class Definitions - Syste\_Control

- 명세에 정의되지 않은 메소드가 소스코드 상에 존재(get ATMcount, get\_inputMoney, inputRID, get\_isjackpot, set\_usercount)

Type	Method
Name	getATMcount
Purpose	ATM 이 가지고있는 화폐별 수량을 확인하는 메소드

Overview	N/A
Cross Reference	jackPot, withdraw, deposit, manage_ATM
Input	Void
Output	int
Abstract Operation	ATM 이 가지고있는 각각의 50,000 원 10,000 원 5,000 원 1,000 원 지폐별 수량을 확인하는 메소드
Exceptional Course of Event	N/A

Type	Method
Name	get_inputMoney
Purpose	사용자가 입력한 금액을 반환하는 메소드
Overview	N/A
Cross Reference	
Input	Void
Output	int
Abstract Operation	사용자가 시스템에 입력한 금액을 반환하는 메소드
Exceptional Course of Event	N/A

Type	Method
Name	InutRID
Purpose	송금받는 사용자의 계좌가 유효한지 확인하고 저장하는 메소드
Overview	N/A
Cross Reference	Remittance
Input	Void
Output	Boolean
Abstract Operation	송금받는 사용자의 계좌가 유효한지 확인하고 저장하는 메소드
Exceptional Course of Event	N/A



Type	Method
Name	get_isjackpot
Purpose	출금시 해당 사용자가 jackpot 당첨자인지를 확인하는 메소드
Overview	N/A
Cross Reference	Withdraw
Input	Void
Output	int
Abstract Operation	출금시 해당 사용자가 jackpot 당첨자인지를 확인하는 메소드
Exceptional Course of Event	N/A

Type	Method
Name	set_usercount
Purpose	withdraw
Overview	N/A
Cross Reference	당일 ATM 사용자 숫자를 초기화하는 메소드
Input	int
Output	Void
Abstract Operation	당일 ATM 사용자 숫자를 초기화하는 메소드
Exceptional Course of Event	N/A

- Method Definitions

Type	Class
Name	System_Control
Purpose	사용자(User 혹은 Manager)로부터 데이터를 입력받아 저장하고 실제 데이터의 처리가 동작하는 Bank, Account 등 다양한 클래스와 데이터를 주고받는 클래스

Overview	N/A
Cross Reference	jackPot, withdraw, deposit,remittance, view_account_detail, manage_ATM
Exceptional Course of Event	N/A

Type	Method
Name	input_menu
Purpose	사용자로부터 진입하고자하는 ATM 의 기능을 입력받아 inputMenu 어트리뷰트에 저장하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail, manage_ATM
Input	int
Output	Void
Abstract Operation	사용자가 입력한 메뉴의 번호를 inputMenu 어트리뷰트에 저장한다
Exceptional Course of Event	N/A

Type	Method
Name	input_ID
Purpose	사용자로부터 ID 를 입력받아 inputID 어트리뷰트에 저장하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail
Input	Int
Output	True or false
Abstract Operation	사용자로부터 입력받은 input 값을 account 클래스로 전달해서 ID 가 유효한 값을 갖고 있는지를 확인한다.
Exceptional Course of Event	N/A

Type	Method
------	--------

Name	input_amount
Purpose	사용자로부터 금액을 입력받아 각각의 지폐별 변수(input50000,input10000,...)에 저장하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit, remittance
Input	ArrayList<Integer>
Output	Int
Abstract Operation	사용자로부터 입력받은 값을 각각의 지폐별 변수(input50000,input10000,...)에 저장한다.
Exceptional Course of Event	N/A

Type	Method
Name	input_PW
Purpose	사용자로부터 비밀번호를 입력받아 Account 에 저장된 비밀번호와 비교하여 유효성을 확인하는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail
Input	int
Output	int
Abstract Operation	사용자로부터 입력받은 비밀번호를 Account 에 저장된 비밀번호와 비교한다.
Exceptional Course of Event	N/A

Type	Method
Name	updateBalance
Purpose	사용자의 거래에 따른 계좌의 잔고를 변화시켜주는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance
Input	void

Output	int
Abstract Operation	사용자의 거래에 따라 계좌가 가지고있는 Balance 를 변화시켜준다.
Exceptional Course of Event	N/A

Type	Method
Name	input_MID
Purpose	관리자의 ID 를 입력받아 유효한 MID 인지 반환하는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	int
Output	True of false
Abstract Operation	관리자의 ID 를 입력받아 유효한 MID 인지 boolean 값을 반환하는 메소드
Exceptional Course of Event	N/A

Type	Method
Name	input_POM
Purpose	관리자가 ATM 의 금액을 빼줄것인지 더해줄것인지를 선택받아 inputPOM 에 저장해 주는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	int
Output	Void
Abstract Operation	사용자의 입력을 inputPOM 에 저장한다.
Exceptional Course of Event	N/A

Type	Method
------	--------

Name	input_MPW
Purpose	관리자의 PW 를 입력받아 유효한 MPW 인지 boolean 값을 반환하는 메소드
Overview	N/A
Cross Reference	manage_ATM
Input	int
Output	True of false
Abstract Operation	관리자의 PW 를 입력받아 유효한 MPW 인지 boolean 값을 반환하는 메소드
Exceptional Course of Event	N/A

## 2051. Implement Class & Method Definitions - Method Definitions

- 1. Manager
  - 생성자 메소드에 대한 명세누락
- 2. Manager.checkID
  - 소스코드상에서는 checkID
  - Abstract Operation 오타 : inputMID > inputID, ManagerID > ManagerID
- 3. Manager.CheckPW
  - 소스코드상에서는 checkPW
  - Abstract Operation 오타 : inputMPW > inputPW

How to fix -> 생성자 메소드의 경우 명세가 불필요하기때문에 생략 하였음, 발견된 오타 수정

## 2055. Write Unit Test Code

- Unit Test 를 거치지 않은 함수들 존재

How to fix -> Unit Test 는 테스트 상에서 중요하다고 판단되는 Code 에 대해서만 수행하는 것이므로 테스트를 거치지 않은 함수들이 당연히 존재

## Stage 2060. Modified Part

- System Testing 결과차이존재(2-1, 3-1, 5-1, 6-1, 7-1: Pass -> Fail)

How to fix -> Timer 오류로 인하여 Fail 과 Pass 가 번갈아 가며 나왔으나 Timer 오류 수정으로 Pass

- 숫자가 아닌 다른 character 를 입력하거나 Integer 범위 밖에 있을 때는 오류메시 지를 출력하지 않는다.

How to fix -> 입력값에 대한 예외처리 모두 적용

- 8-1 “plus 를 입력하였을때” 표현매우모호

How to fix -> ‘관리자가 관리자 메뉴에서 입금을 선택하고 화폐별 증가수량을 입력하였을때, 해당 금액만큼을 ATM 잔고에 더해주는지 확인한다.’ 로 변경

- 8-2 “minus 를 입력 하였을 때” 표현 매우 모호

How to fix -> ‘관리자가 관리자 메뉴에서 출금을 선택하고 화폐별 증가수량을 입력하였을 때, 해당 금액만큼을 ATM 잔고에서 빼주는지를 확인한다’ 로 변경

## Category Partition Test. Bug Fix

Test Suite	Test Case	Test Case Details	Latest Exec Result	How to fix	Developer Exec Result
User	atm-5:Test Case 1		Passed		
User	atm-6:Test Case 2	입금/출금/송금 시에 계좌번호 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-7:Test Case 3	입금/출금/송금 시에 계좌번호 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-8:Test Case 4	송금 시에 수취인 계좌번호 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-9:Test Case 5	송금 시에 수취인 계좌번호 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-10:Test Case 6	송금 시에 금액 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-11:Test Case 7		Passed		
User	atm-12:Test Case 8	송금 시에 금액 입력값에 계좌 잔고보다 큰 값을 입력하면 오류 메시지가 출력된다.	Failed	잔고확인 메시지 출력 잔고확인 은 송금계좌의 비밀번호를 알기전까지 확인할수 없으므로 비밀번호 입력까지 해보면 잔고가 부족하다고 뜨는것을 확인할 수 있음	Passed
User	atm-13:Test Case 9	송금 시에 비밀번호 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-14:Test Case 10	송금 시에 비밀번호 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-15:Test Case 11		Passed		
User	atm-16:Test Case 12		Passed		
User	atm-17:Test Case 13		Passed		

User	atm-18:Test Case 14	출금 시에 5 만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-19:Test Case 15		Passed		
User	atm-20:Test Case 16		Passed		
User	atm-21:Test Case 17	출금 시에 1 만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-22:Test Case 18	입금 시에 5 만원권 입력값에 Integer 범위 이내의 수를 입력하고 올바른 비밀번호를 입력하면 입금에 성공한다.	Failed	ATM 기기가 받을수 있는 최대한도는 5 만원권 500 매이며, 입금시에는 비밀번호를 받지 않습니다.(명세해답)	Passed
User	atm-23:Test Case 19	입금 시에 5 만원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-24:Test Case 20	입금 시에 5 만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-25:Test Case 21	입금 시에 1 만원권 입력값에 Integer 범위 이내의 수를 입력하고 올바른 비밀번호를 입력하면 입금에 성공한다.	Failed	ATM 기기가 받을수 있는 최대한도는 1 만원권 500 매이며, 입금시에는 비밀번호를 받지 않습니다.(명세해답)	Passed
User	atm-26:Test Case 22	입금 시에 1 만원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-27:Test Case 23	입금 시에 1 만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-28:Test Case 24	입금 시에 5 천원권 입력값에 Integer 범위 이내의 수를 입력하고 올바른 비밀번호를 입력하면 입금에 성공한다.	Failed	ATM 기기가 받을수 있는 최대한도는 5 천원권 500 매이며, 입금시에는 비밀번호를 받지 않습니다.(명세해답)	Passed
User	atm-29:Test Case 25	입금 시에 5 천원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-30:Test Case 26	입금 시에 5 천원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed



User	atm-31:Test Case 27	입금 시에 1천원권 입력값에 Integer 범위 이내의 수를 입력하고 올바른 비밀번호를 입력하면 입금에 성공한다.	Failed	ATM 기기가 받을수 있는 최대한도는 1천원권 500매이며, 입금시에는 비밀번호를 받지 않습니다.(명세해답)	Passed
User	atm-32:Test Case 28	입금 시에 1천원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-33:Test Case 29	입금 시에 1천원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-34:Test Case 30	송금 시에 금액 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-39:Test Case 31	송금 시에 금액 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
User	atm-40:Test Case 32		Passed		
Manager	atm-41:Test Case 1		Passed		
Manager	atm-42:Test Case 2	관리자 메뉴 진입 시에 ID 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-43:Test Case 3	관리자 메뉴 진입 시에 ID 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-44:Test Case 4	관리자 메뉴 진입 시에 비밀번호 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-45:Test Case 5	관리자 메뉴 진입 시에 비밀번호 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-46:Test Case 6	관리자 입금/출금 시에 금액 입력 값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-47:Test Case 7		Passed		
Manager	atm-48:Test Case 8	관리자 출금 시에 금액 입력값에 계좌 잔고보다 큰 값을 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-49:Test Case 9		Passed		

Manager	atm-50:Test Case 10	관리자 입금 시에 5 만원권 입력값에 Integer 범위 이내의 수를 입력하고 올바른 비밀번호를 입력하면 입금에 성공한다.	Failed	ATM 기기가 받을수 있는 최대한도는 5 만원권 500 매이며, 관리자 계정은 이미 MPW의 유효성을 확인한 상태입니다.(명세해답)	Passed
Manager	atm-51:Test Case 11	관리자 입금 시에 5 만원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-52:Test Case 12	관리자 입금 시에 5 만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-53:Test Case 13	관리자 입금 시에 1 만원권 입력값에 Integer 범위 이내의 수를 입력하고 올바른 비밀번호를 입력하면 입금에 성공한다.	Failed	ATM 기기가 받을수 있는 최대한도는 1 만원권 500 매이며, 관리자 계정은 이미 MPW의 유효성을 확인한 상태입니다.(명세해답)	Passed
Manager	atm-54:Test Case 14	관리자 입금 시에 1 만원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-55:Test Case 15	관리자 입금 시에 1 만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-56:Test Case 16		Passed		
Manager	atm-57:Test Case 17	관리자 입금 시에 5 천원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	ATM 기기가 받을수 있는 최대한도는 5 천원권 500 매이며, 관리자 계정은 이미 MPW의 유효성을 확인한 상태입니다.(명세해답)	Passed
Manager	atm-58:Test Case 19		Passed		
Manager	atm-59:Test Case 20	관리자 입금 시에 1 천원권 입력값에 정수가 아닌 문자를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-60:Test Case 21	관리자 입금 시에 1 천원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-61:Test Case 22		Passed		
Manager	atm-62:Test Case 23		Passed		
Manager	atm-63:Test Case 24	관리자 출금 시에 5 만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-64:Test Case 25		Passed		

Manager	atm-65:Test Case 26		Passed		
Manager	atm-66:Test Case 27	관리자 출금 시에 1만원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-67:Test Case 28		Passed		
Manager	atm-68:Test Case 29		Passed		
Manager	atm-69:Test Case 30	관리자 출금 시에 5천원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed
Manager	atm-70:Test Case 31		Passed		
Manager	atm-71:Test Case 32		Passed		
Manager	atm-72:Test Case 33	관리자 출금 시에 1천원권 입력값에 Integer 범위를 벗어난 수를 입력하면 오류 메시지가 출력된다.	Failed	예외처리	Passed

## Brute Force Test. Issues Fix

Test Suite	Test Case	Result	Summary	How to fix	Developer Exec Result
Manager	관리자 프로세스에서 5 번 반복하여 입/출금 할 수 있다.	Failed	잔고가 어느 일정 범위 이상 커지면 "잔고 확인하세요"라는 오류 메시지가 출력된다. (잔고 최대값 명세에 존재하지 않음)	잔고는 지폐별 500 매로 최댓값 설정,	Passed
Manager	관리자 프로세스에서 10 번 반복하여 입/출금 할 수 있다.	Failed	잔고가 어느 일정 범위 이상 커지면 "잔고 확인하세요"라는 오류 메시지가 출력된다. (잔고 최대값 명세에 존재하지 않음)	잔고는 지폐별 500 매로 최댓값 설정,	Passed
User	프로그램 초기 실행 시 난수가 제대로 생성된다.	Passed			
User	출금 10 번 이내에 Jackpot 이 터지는 지 확인한다.	Passed			
User	Jackpot 이 터지면 해당 User 에게 5 만원을 추가 인출하며 ATM 의 잔고에서 5 만원이 빠져나간다.	Failed	Jackpot 이 터져도 ATM 의 잔고에는 변화가 생기지 않는다. ATM 잔고가 0 일 때도 예외처리 필요	ATM 잔고가 부족하면 사용자의 계좌로 입력된다는 메시지와 함께 사용자의 계좌로 당첨금을 입금	Passed

User	1 일 1 명만 Jackpot 이 터진다.	Failed	초기 실행 후 Jackpot 이 한 번 터진 후에도 다시 터진다.	Timer 오류로 초기화가 되지 않아 생겼던 문제를 해결	Passed
User	출금 시에만 Jackpot 이 터진다.	Failed	송금 시 사용자의 비밀번호를 입력하고 나서도 Jackpot 이 터지는 경우 존재	출금 상태에서만 Jackpot 이 터지도록 수정	Passed
User	하나의 통장으로 여러 번 출금할 시 Jackpot 당첨이 여러 번 되지 않게 한다.	Failed	같은 계좌로 출금을 5 회 이상 반복 했을 때, 연속으로 2 번 잭팟이 터지는 경우가 있다	하루 한번만 Jackpot 이 터지도록 수정	Passed
User	모든 입력에 대한 기기의 반응은 1 초 이내로 이루어진다.	Failed	integer 이내의 정수값이 아니라면, 1 초 이내로 반응이 없다(반응이 아예 없다)	예외처리	Passed
User	거래는 1 분 이내에 이루어져야 한다.	Passed			
User	사용자가 보기 편한 화면을 제공해야 한다.	Passed			
User	세 번 이상 연속적으로 출금할 수 있다.	Passed			
User	세 번 이상 연속적으로 입금할 수 있다.	Failed	잔고가 어느 일정 범위 이상 커지면 "비밀번호 확인하세요"라는 오류 메시지가 출력된다.	일반계좌의 잔고를 10 억 이내로 고정	Passed
User	세 번 이상 연속적으로 송금할 수 있다.	Failed	3 번 이상 송금하려고 하면 알맞은 비밀번호여도 오류 메시지가 나온다	3 번이상 송금이 가능하도록 오류 수정	Passed
User	세 번 이상 연속적으로 조회할 수 있다.	Passed			
User	출금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed	출금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 증가한다. 예외처리 필요	예외처리	Passed
User	입금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed	입금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 감소한다. 예외처리 필요	예외처리	Passed
User	송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed	송금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 증가한다. 예외처리 필요	예외처리	Passed

<p><b>Manager</b></p>	<p>ATM 잔고가 일정수준(50 만원) 이하가 되면 잔고부족 메시지를 출력한다.</p>	<p><b>Failed</b></p>	<p>ATM 잔고가 50 만원 이하가 되어도 아무 메시지도 출력되지 않는다.</p>	<p><b>시스템 메인화면(메뉴선택화면) 하단에 1 만원과 5 만원에 대한 부족 메시지 출력(1 천원 5 천원은 출금 대상이 아니므로 출력 x)</b></p>	<p><b>Passed</b></p>
-----------------------	---	----------------------	--	--	----------------------