JackPot ATM

System Test Response Report

Team 5

Date

2018-06-08

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

System test Modified Part

Static Analysis Modified Part

Stage 1000. Modified Part

1.1. Stage 1000 Planning1001. Define Draft Plan

2. Project Objective

- Jackpot 기능의 악용가능성(ex. 한사람이 만원씩 지속적으로 출금)을 제거

-> 이후 단계에서 관련 명세가 존재하지 않음

이후 단계에서 시간 차가 있는 출금만 카운트하거나, 5 만원 이상 출금만 카운트하는 식 등의 명확한 방법으로 명세 추가 필요

Modified : 1일 1/1000 의 확률로 Jackpot 당첨이 가능하며, 수수료 1,200원을 도입하여 지속출금 50번만 하여도 Jackpot 으로 얻을수 있는 수익보다 ATM 수익이 높아지기 때문에 ATM 기기 입장에서 이득을 얻을수 있다고 판단하여 추가적으로 기능구현을 하지 않았으며, 이에따라 문서를 수정하였음.

변경 전 : - Jackpot 기능의 악용가능성(ex. 한사람이 만원씩 지속적으로 출금)을 제거

변경 후 : - Jackpot 기능의 악용 가능성(1만원씩 지속출금)은 수수료를 통해 ATM에 이득이 되도록 구현

Stage 2030. Modified Part

2031. Define Essential Use Cases

2010 까지는 존재하는 '수수료'와 '명세서 출력'에 대한 서술이 모두 사라짐

1. Withdraw

- "ATM의 잔고와 출금 금액을 비교한다" 는 내용 추가해야 함- Typical Courses of Events 항목에 13번이 두 개임.

- 수수료에 대한 서술 추가 필요

Modified : 수수료에 대한 내용 모두 추가, 명세서에 대한 내용은 1000 부터 제거함을 2nd cycle 발표에서 공지

Use Case	withdraw
Actor	User
Purpose	사용자 계좌에 있는 돈을 출금한다.
Overview	사용자가 ATM기의 출금 버튼을 누르고 출금할 계좌를 입력한다. 계좌가 확인되면 금액을 입력받아 잔고보다 적을 경우 출금을 진행한다. 그 후 사용자가 돈을 갖고 떠난다.
Type	Primary and Essential
Cross Reference	System Function R.1.2
Pre-Requisites	ATM에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 출금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User는 계좌를 입력한다. 4.(S) 유효한 계좌인지 확인한다. 5.(S) 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) ATM 잔고와 출금금액을 비교한다. 11.(S) User의 잔고와 출금금액+수수료 를 비교한다. 12.(S) User의 잔고를 출금금액+수수료만큼 감소시킨다. 13.(S) ATM 잔액을 감소시킨다. 14.(S) ATM 각 지폐별 잔고를 감소시킨다. 15.(S) 거래후 사용자의 계좌 잔고를 출력한다.

5. Random_Jackpot

- “당첨되는 순서를 랜덤하게 선택한다.”

-> Random 범위 설명 필요

- “ATM 의 잔고에서 5 만원을 감소시킨다” 는 내용 추가해야 함

Modified : ATM 잔고에서 5만원을 감소시킨다는 내용 추가

Use Case	Random_Jackpot
Actor	User
Purpose	출금 기능을 이용하는 임의의 사용자에게 추가로 5만원을 지급한다.
Overview	OO시가 되면 ATM기의 몇번째 출금 고객에게 추가로 5만원이 출금될지 정해진다. 해당 순서의 사용자가 출금시에 5만원이 추가로 출금된다.
Type	Primary and Essential
Cross Reference	System Function R.1.1
Pre-Requisites	User가 출금 거래중이어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(S) OO시가 되면 당첨되는 순서를 랜덤하게(1/1000의 확률) 선택한다. 2.(S) User가 ATM을 통해 출금거래를 완료하였을때 해당 User가 해당 순서의 사용자인지 확인한다. 3.(S) 해당 User에게 돈을 5만원 더 추가로 출금해준다. 4.(S) ATM 의 잔고에서 5 만원을 감소시킨다
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

2034. Refine Glossary

- 2040단계에서 View의 부재로 인해 GUI_Interface 관련 항목 수정 필요

Modified ：이 단계는 View(GUI 에 대한것을 고려하지 않고 시스템 내부적으로 사용자와 interaction 만을 고려하며 뒤의 2040 GUI 를 다루었기 때문에 변경된 클래스 네임만을 반영

수정 전 :

GUI_Interface	Class	사용자로부터 메뉴를 입력 받아 처리하고 컨트롤 하는 클래스

수정 후 :

System_control	Class	사용자로부터 메뉴를 입력 받아 처리하고 컨트롤 하는 클래스

2038. Refine System Test Case

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

-> 관련된 test case 끼리 묶고 넘버링 필요

Modified ：Refine System Test Case 이기 때문에 기존 1000 단계에서 구체적이지 못한 항목에서 구체적으로 분석된 프로그램을 기반으로 새롭게 테스트 케이스를 변경하고

관련된 test case 는 N-1, N-2… 로 명시하여 묶음

Test Num	Test 항목	Description	Use case
1-1	input_menu Test	사용자가 입력한 해당 메뉴 화면으로 올바르게 진입하는지 확인한다.	R1.2 ,R.2, R.3, R.4, R.5

2-1	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하지 않으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2,R.3,R.4
2-2	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하면 다음 단계로 넘어가는지 확인한다.	R1.2, R.2,R.3,R.4
3-1	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 아니면 오류메시지를 출력하는지 확인한다	R1.2, R.3, R.4
3-2	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 맞으면 다음단계로 넘어가는지 확인한다.	R1.2, R.3, R.4
4-1	input_amount	출금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-2	input_amount	출금의 경우 사용자가 입력한 금액이ATM 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-3	input_amount	송금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-4	input_amount	사용자의 입력금액이 계좌 잔고보다 작고 ATM 잔고보다 작을때 다음 단계로 넘어가는지 확인한다.	R1.2, R.2, R.3

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

Stage 2040. Modified Part

2041. Design Real Use Cases

3. remittance

- Exceptional Courses of Events 에 2030 단계에서 존재하는

"E5: 자기 자신의 계좌에는 송금할 수 없다." 는 내용 누락.

Modified : 구현상에서 추가된 예외 상황에 대한 명시 추가

Use Case	remittance
Actor	User
Purpose	사용자 계좌의 돈을 다른 사용자 계좌로 송금한다.
Overview	사용자가 ATM기의 송금 버튼을 누르고 자신의 계좌를 입력한다. 계좌가 확인되면 송금할 계좌와 금액을 입력받는다. 송금 계좌가 맞고 통장 잔고가 송금액보다 많으면 송금 계좌의 잔고가 액수만큼 증가한다.
Type	Primary and Essential
Cross Reference	System Function R.3
Pre-Requisites	ATM에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.

Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 remittance 버튼을 누른다. 2.(S) 계좌 ID 입력 화면을 출력한다. 3.(A) User는 계좌(inputID)를 입력한다. 4.(S) Account.ID와 inputID를 비교 확인한다. 5.(S) 송금받을 계좌 입력 화면을 출력한다. 6.(A) User는 송금받을 계좌(inputID)를 입력한다. 7.(S) Account.ID와 inputID를 비교한다. 8.(A) User는 송금 금액(inputMoney)을 입력한다. 9.(S) User의 잔고(Account.Balance)와 송금액(inputMoney)을 비교 확인한다. 10.(A) User는 비밀번호(inputPW)를 입력한다. 11.(S) Account.password와 inputPW가 일치하는지 확인한다. 12.(S) 받는사람의 통장잔고(Account.Balance)를 증가시킨다. 13.(S) User의 통장잔고(Account.Balance)를 감소시킨다. 14.(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 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

6. manage_ATM

- 다음 단계에서 기술되는 GUI_Interface 가 존재함

Modified ：GUI_Interface 를 제거하고 어트리뷰트 명인Total money 만을 명시

Use Case	manage_ATM
Actor	Manager
Purpose	관리자가 ATM의 잔고를 추가하거나 뺄 수 있다
Overview	ATM의 TotalMoney가 일정 수준 이하로 낮아지면 관리자가TotalMoney를 추가하고 잔고가 일정 수준 이상으로 많아지면 TotalMoney를 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) Manager가 manage 버튼을 누른다. 2.(S) 관리자 ID 입력화면을 출력한다. 3.(A) Manager가 관리자 ID(inputID)를 입력한다 4.(S) ManagerID와 inputID를 비교 확인한다 5.(A) Manager가 관리자 Password(inputPW)를 입력한다 6.(S) ManagerPW와 inputPW를 비교 확인한다 5.(S) 확인이 끝나면 Manager가 화폐별 잔고에 접근할수 있도록 한다. 6.(A) Manager가 화폐별 잔고를 추가하거나 뺀다 7.(S) TotalMoney를 갱신한다 8.(S) Count50000, Count10000, Count5000, Count1000 각 화폐별 금액을 갱신한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1.MID와 MPW가 잘못된 경우 에러 메시지를 출력한다

Stage 2050. Modified Part

2051. Implement Class & Method Definitions

Class Definitions – Manager

1. Manager.checkID

- Abstract Operation 오타: inputMID > inputID, ManagerIID > ManagerID

2. Manager.CheckPW

- Abstract Operation 오타 : inputMPW > inputPW

Modified : MID,MPW는 오타가 아닌 매니저의 정보임을 나타내기 위해 M을 붙여 표현한 것이므로 수정하지 않음

Class Definitions – Timer

1. Timer

- 메소드 start 에 대한 명세는 존재하나 소스코드 상에 존재하지 않음.

Modified : 멀티쓰레드 사용을 위해 run 대신 start를 사용함

Class Definitions – MoneyCount

1. ATMBalanceUpdate

- 명세와 소스코드의 output 값의 자료형 불일치. Abstract Operation 에서는 inputcount 이나 소스코드 상에서는 inputcount 임

Modified : 소스코드와 일치하도록 문서상의 내용들을 수정함.

Type	Method
Name	ATMBalanceUpdate
Purpose	ATM 자체가 가지고 있는 각 화폐별 잔고를 업데이트 해주는 메소드
Overview	N/A

Cross Reference	jackPot, withdraw, deposit, manage_ATM
Input	inputMoney: int, inputcount: int, inputPOM: int
Output	true or false
Abstract Operation	1.사용자로부터 입력받은 화폐의 inputcount 를 Count화폐 에 반영
Exceptional Course of Event	N/A

Class Definitions – Bank

1. SearchID

- 명세와 소스코드의 output 값의 자료형 불일치

2. getAccountBalance

- 명세와 소스코드의 output 값의 자료형 불일치

Modified : 소스코드와 일치하도록 문서상의 내용들을 수정함

Type	Method
Name	searchID
Purpose	Bank 내부에 개설된 통장 혹은 카드의 정보를 찾는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail
Input	inputID:int
Output	int
Abstract Operation	Account 에 있는 checkID 를 호출

Exceptional Course of Event	N/A

Type	Method
Name	getAccountBalance
Purpose	사용자의 계좌 잔고를 받아오는 메소드
Overview	N/A
Cross Reference	view_account_detail
Input	void
Output	int
Abstract Operation	Account에 있는 Balance를 호출
Exceptional Course of Event	N/A

Class Definitions – Account

1. Account.checkID

- Abstract Operation 에서 '찾음'이라는 표현이 모호함.

2. searchCardID

- Abstract Operation 에서 '찾음'이라는 표현이 모호함.

Modified : '찾음' 이라는 표현을 함수의 기능에 따라 좀 더 구체적으로 수정함.

Type	Method
Name	Account.checkID

Purpose	사용자로부터 입력된 ID 계좌번호를 찾는 메소드
Overview	N/A
Cross Reference	withdraw, deposit,remittance, view_account_detail
Input	inputID: int
Output	int
Abstract Operation	inputID와 일치하는 계좌번호를 찾은 경우와 못 찾은 경우 각각 다른 값 반환
Exceptional Course of Event	N/A

Type	Method
Name	searchCardID
Purpose	사용자가 입력한 ID가 계좌번호가 아닌 경우 일치하는 카드번호가 있는지 찾기위한 메소드
Overview	N/A
Cross Reference	withdraw,remittance,deposit,view_account_detail
Input	inputID: int
Output	int
Abstract Operation	inputID 와 일치하는 Card의 ID를 찾은 경우와 못 찾은 경우 각각 다른 값 반환
Exceptional Course of Event	N/A

3. balanceAccount

- Abstract Operation 에서 '변화시킨다'는 표현이 증가인지 감소인지 불분명함.

Modified : input으로 받는 inputMoney가 양수와 음수 모두 들어올 수 있기에 불분명하지 않다고 판단해서 수정하지 않음

1. View Account Detail

- Insert_ID 의 Pre-Conditions 가 적합하지 않음.

Modified : 적합하도록 수정함

Name	Insert_ID
Responsibilities	MainGUI를 통해 사용자의 ID를 입력한다.
Type	MainGUI
Cross Reference	R.5
Notes	MainGUI를 통해 사용자의 ID를 입력한다.
Pre-Conditions	InputMenu의 값에 remittance(3)이 입력된 상태여야 한다.
Post-Conditions	ID 입력 화면을 출력한다.

System test Modified Part

4. Brute Force Testing

4.1. Testing Result

Test Suite	Test Case	Result
Manager	관리자 프로세스에서 5번 반복하여 입/출금 할 수 있다.	Passed
Manager	관리자 프로세스에서 10번 반복하여 입/출금 할 수 있다.	Passed
User	프로그램 초기 실행 시 난수가 제대로 생성된다.	Passed
User	출금 10번 이내에 Jackpot이 터지는 지 확인한다.	Passed
User	Jackpot이 터지면 해당 User 에게 5만원을 추가 인출하며 ATM의 잔고에서 5만원이 빠져나간다.	Passed
User	1일 1 명만 Jackpot 이 터진다.	Passed
User	출금 시에만 Jackpot 이 터진다.	Passed
User	하나의 통장으로 여러 번 출금할 시 Jackpot 당첨이 여러 번 되지 않게 한다.	Failed
User	모든 입력에 대한 기기의 반응은 1초 이내로 이루어진다.	Passed
User	거래는 1분 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.	Passed
User	세 번 이상 연속적으로 송금할 수 있다.	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.	Passed
User	입금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Passed
User	송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Passed
Manager	ATM 잔고가 일정수준(50 만원) 이하가 되면 잔고부족 메시지를 출력한다.	Passed
User	출금 시 수수료 1200원을 더하여 출금한다.	Passed
User	송금 시 수수료 1200원을 더하여 출금한다.	Passed
User	입금 시 ATM 잔고 지폐 별 500 개를 초과하면 오류 메시지를 출력한다.	Passed
User	입금 시 계좌 잔고 10억원을 초과하면 오류 메시지를 출력한다.	Passed
User	송금 시 상대방 계좌 잔고 10억원을 초과하면 오류 메시지를 출력한다.	Failed
User	Jackpot이 터질 때 ATM 잔고가 부족하면 사용자의 계좌로 5만원을 입금하고 알림 메시지를 출력한다.	Passed
User	복수의 화폐에서 최대치(500매)를 넘어선 매수를 입금하려고 하면, 초과된 모든 지폐에 대해 오류 메시지를 출력한다.	Failed
Manager	복수의 화폐에서 최대치(500매)를 넘어선 매수를 입금하려고 하면, 초과된 모든 지폐에 대해 오류 메시지를 출력한다.	Failed

23/27 = 85% Pass

문서 변경으로 인하여 이전 Case에서 8개 추가됨

Test Suite	Test Case	How to fix	Developer Result
User	여러 화폐의 최대 수량(500)을 동시에 넘었을 때, 복수의 오류 메시지를 출력하지 않는다	각각의 화폐 수량 초과에 대하여 오류메시지를 출력하도록 변경	pass
User	같은 계좌로 출금을 여러 번 반복했을 때, 다시 Jackpot이 터진다.	잭팟 당첨 횟수에 대한 제한은 없으므로 같은 계좌라고 하더라도 당첨 당일만 아니라면 당첨이 가능함-> 수정하지 않음	예외
User	송금 시 상대방 계좌 잔고 10억원을 초과하면 잘못된 오류 메시지가 출력된다.	송금시, 입금계좌 10억 초과시 "일반계좌 최대 입금한도는 10억입니다" 메시지를 출력	pass
Manage	여러 화폐의 최대 수량(500)을 동시에 넘었을 때, 복수의 오류 메시지를 출력하지 않는다	각각의 화폐 수량 초과에 대하여 오류메시지를 출력하도록 변경	pass

수정후 RE-Test case 테스트

2. Category-partitionTesting

Test Suite	Test Case	Latest Exec Result
User	atm-5:Test Case 1	Passed
User	atm-6:Test Case 2	Passed
User	atm-7:Test Case 3	Passed
User	atm-8:Test Case 4	Passed
User	atm-9:Test Case 5	Passed
User	atm-10:Test Case 6	Passed
User	atm-11:Test Case 7	Passed
User	atm-12:Test Case 8	Passed
User	atm-13:Test Case 9	Passed
User	atm-14:Test Case 10	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	Passed
User	atm-19:Test Case 15	Passed
User	atm-20:Test Case 16	Passed
User	atm-21:Test Case 17	Passed
User	atm-22:Test Case 18	Passed
User	atm-23:Test Case 19	Passed
User	atm-24:Test Case 20	Passed
User	atm-25:Test Case 21	Passed
User	atm-26:Test Case 22	Passed
User	atm-27:Test Case 23	Passed
User	atm-28:Test Case 24	Passed
User	atm-29:Test Case 25	Passed
User	atm-30:Test Case 26	Passed
User	atm-31:Test Case 27	Passed
User	atm-32:Test Case 28	Passed
User	atm-33:Test Case 29	Passed
User	atm-34:Test Case 30	Passed
User	atm-39:Test Case 31	Passed
User	atm-40:Test Case 32	Passed
Manager	atm-41:Test Case 1	Passed
Manager	atm-42:Test Case 2	Passed
Manager	atm-43:Test Case 3	Passed
Manager	atm-44:Test Case 4	Passed
Manager	atm-45:Test Case 5	Passed
Manager	atm-46:Test Case 6	Passed
Manager	atm-47:Test Case 7	Passed

Manager	atm-48:Test Case 8	Passed
Manager	atm-49:Test Case 9	Passed
Manager	atm-50:Test Case 10	Passed
Manager	atm-51:Test Case 11	Passed
Manager	atm-52:Test Case 12	Passed
Manager	atm-53:Test Case 13	Passed
Manager	atm-54:Test Case 14	Passed
Manager	atm-55:Test Case 15	Passed
Manager	atm-56:Test Case 16	Passed
Manager	atm-57:Test Case 17	Passed
Manager	atm-58:Test Case 19	Passed
Manager	atm-59:Test Case 20	Passed
Manager	atm-60:Test Case 21	Passed
Manager	atm-61:Test Case 22	Passed
Manager	atm-62:Test Case 23	Passed
Manager	atm-63:Test Case 24	Passed
Manager	atm-64:Test Case 25	Passed
Manager	atm-65:Test Case 26	Passed
Manager	atm-66:Test Case 27	Passed
Manager	atm-67:Test Case 28	Passed
Manager	atm-68:Test Case 29	Passed
Manager	atm-69:Test Case 30	Passed
Manager	atm-70:Test Case 31	Passed
Manager	atm-71:Test Case 32	Passed
Manager	atm-72:Test Case 33	Passed

64/64 = 100% Pass

3. PairwiseTesting

Test Suite	Test Case	Latest Exec Result
User	atm-73:[Pairwise]Test Case1	Passed
User	atm-74:[Pairwise]Test Case2	Passed
User	atm-75:[Pairwise]Test Case3	Passed
User	atm-76:[Pairwise]Test Case4	Passed
User	atm-77:[Pairwise]Test Case5	Passed
User	atm-78:[Pairwise]Test Case6	Passed
User	atm-79:[Pairwise]Test Case7	Passed
User	atm-80:[Pairwise]Test Case8	Passed
User	atm-81:[Pairwise]Test Case9	Passed
User	atm-82:[Pairwise]Test Case10	Passed
User	atm-83:[Pairwise]Test Case11	Passed
User	atm-84:[Pairwise]Test Case12	Passed
User	atm-85:[Pairwise]Test Case13	Passed
User	atm-86:[Pairwise]Test Case14	Passed
Manager	atm-87:[Pairwise]Test Case1	Passed
Manager	atm-88:[Pairwise]Test Case2	Passed
Manager	atm-89:[Pairwise]Test Case3	Passed
Manager	atm-90:[Pairwise]Test Case4	Passed
Manager	atm-91:[Pairwise]Test Case5	Passed
Manager	atm-92:[Pairwise]Test Case6	Passed
Manager	atm-93:[Pairwise]Test Case7	Passed
Manager	atm-94:[Pairwise]Test Case8	Passed
Manager	atm-95:[Pairwise]Test Case9	Passed
Manager	atm-96:[Pairwise]Test Case10	Passed
Manager	atm-97:[Pairwise]Test Case11	Passed
Manager	atm-98:[Pairwise]Test Case12	Passed

26/26 = 100% Pass

Test Suite	Test Case	Result
Manager	관리자 프로세스에서 5 번 반복하여 입/출금 할 수 있다.	Passed
Manager	관리자 프로세스에서 10 번 반복하여 입/출금 할 수 있다.	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.	Passed
User	1 일 1 명만 Jackpot 이 터진다.	Passed
User	출금 시에만 Jackpot 이 터진다.	Passed
User	하나의 통장으로 여러 번 출금할 시 Jackpot 당첨이 여러 번 되지 않게 한다.	제외
User	모든 입력에 대한 기기의 반응은 1 초 이내로 이루어진다.	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.	Passed
User	세 번 이상 연속적으로 송금할 수 있다.	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.	Passed
User	입금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Passed
User	송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Passed
Manager	ATM 잔고가 일정수준(50 만원) 이하가 되면 잔고부족 메시지를 출력한다.	Passed
User	출금 시 수수료 1200 원을 더하여 출금한다.	Passed
User	송금 시 수수료 1200 원을 더하여 출금한다.	Passed
User	입금 시 ATM 잔고 지폐 별 500 개를 초과하면 오류 메시지를 출력한다.	Passed
User	입금 시 계좌 잔고 10 억원을 초과하면 오류 메시지를 출력한다.	Passed
User	송금 시 상대방 계좌 잔고 10 억원을 초과하면 오류 메시지를 출력한다.	Passed
User	Jackpot 이 터질 때 ATM 잔고가 부족하면 사용자의 계좌로 5 만원을 입금하고 알림 메시지를 출력한다.	Passed
User	복수의 화폐에서 최대치(500 매)를 넘어선 매수를 입금하려고 하면, 초과된 모든 지폐에 대해 오류 메시지를 출력한다.	Passed
Manager	복수의 화폐에서 최대치(500 매)를 넘어선 매수를 입금하려고 하면, 초과된 모든 지폐에 대해 오류 메시지를 출력한다.	Passed

5.1. System Test Result

- Category Partition Test

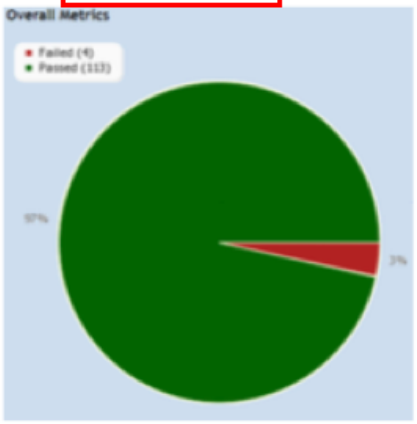
64/64 = 100% Pass

- Pairwise Test

26/26 = 100% Pass

- Brute Force Test

23/27 = 85% Pass

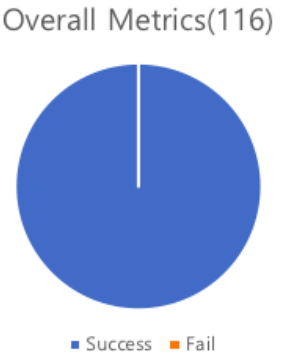


-Category Partition Test

64/64 = 100% Pass

-Pairwise Test

26/26 = 100% Pass

-Brute Force Test

26/26 = 100% Pass