

Software Verification

1st Testing

Team 5

201410373 고예은

201411266 김수현

201411268 김아름

Index

- 1. Specification Review**
- 2. Category-Partition Testing**
- 3. Pairwise Testing**
- 4. Brute Force Testing**
- 5. Overall**

Specification Review - Stage 1000

- 1001. Define Draft Plan
 - 4. Non Functional Requirements
 - “금융 거래 프로그램이므로 보안이 잘 되어야 한다.”
 - 이후 문서에서 **보안에 대한 언급이 존재하지 않음**
- 1003. Define Requirements
 - 1. Functional Requirements
 - “성공한 거래의 송금금액/입금계좌/이름을 보여줌”
 - 이후 단계에서 거래 후 잔고를 보여주는 **명세가 존재하지 않음**
- 문서 전체에서 “withdraw”에 대한 **해석이 통일되어있지 않음(출금/인출)**

Specification Review - Stage 2030

- 2031. Define Essential use cases
 - 2010까지 존재하던 ‘수수료’와 ‘명세서 출력’에 대한 서술이 모두 사라짐
 - 1. Withdraw
 - “출금 화면을 출력한다”, “출금 금액을 입력한다” 표현이 모호함
 - ”ATM의 잔고와 출금 금액을 비교한다” 내용 추가

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

Specification Review - Stage 2030

- 2031. Define Essential use cases
 - 2. Deposit
 - "계좌 잔액을 표시한다" 내용 추가

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

Specification Review - Stage 2030

- 2031. Define Essential use cases
 - 3. Remittance
 - ”계좌 잔액을 표시한다” 내용 추가
 - Exceptional courses of Events : “자기 자신의 계좌에는 송금할 수 없다”
내용 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 송금 버튼을 누른다. 2.(S) 계좌 입력 화면을 출력한다. 3.(A) User는 계좌를 입력한다. 4.(S) 계좌가 유효한지 확인한다. 5.(S) 송금받을 계좌 입력 화면을 출력한다. 6.(A) User는 송금받을 계좌를 입력한다. 7.(S) 계좌가 유효한지 확인한다. 8.(A) User는 송금 금액을 입력한다. 9.(A) User는 비밀번호를 입력한다. 10.(S) 비밀번호가 일치하는지 확인한다. 11.(S) User의 잔고와 송금금액을 비교한다. 12.(S) 받는사람의 통장잔고를 증가시킨다. 13.(S) 보내는 사람의 통장잔고를 감소시킨다. 14.(S) ATM 잔고를 증가시킨다.
Alternative Courses of Events	N/A

Specification Review - Stage 2030

- 2031. Define Essential use cases
 - 5. Random_Jackpot
 - "user가 돈을 인출했을 때" 표현이 모호함
 - pre-requisites : 사용자가 인출 중이어야한다.

Use Case	Random_Jackpot
Actor	User
Purpose	인출 기능을 이용하는 임의의 사용자에게 추가로 5만원을 지급한다.
Overview	00시가 되면 ATM기의 몇번째 인출 고객에게 추가로 5만원이 인출될지 정해진다. 해당 순서의 사용자가 인출시에 5만원이 추가로 인출된다.
Type	Primary and Essential
Cross Reference	System Function R.1.1
Pre-Requisites	N/A
Typical Courses of Events	(A):Actor, (S):System 1.(S) 00시가 되면 당첨되는 순서를 랜덤하게 선택한다. 2.(S) User가 돈을 인출했을때 해당 User가 해당 순서의 사용자인지 확인한다. 3.(S) 해당 User에게 돈을 5만원 더 추가로 인출해준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

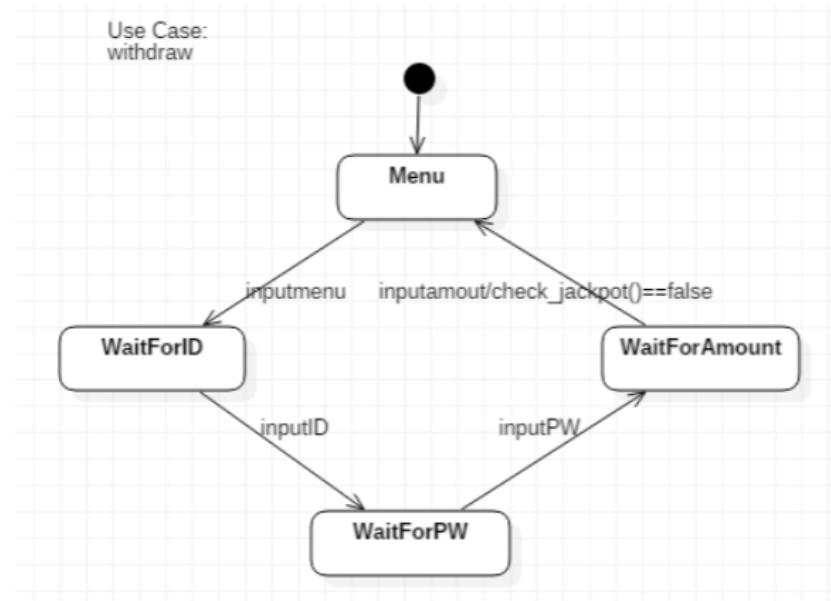
Specification Review - Stage 2030

- 2031. Define Essential use cases
 - 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 ATM의 잔고를 추가하거나 뺄 수 있다
Overview	ATM의 잔고가 일정 수준 이하로 낮아지면 관리자가 ATM의 잔고를 추가하고 잔고가 일정 수준 이상으로 많아지면 ATM의 잔고를 뺀다
Type	Primary and Essential
Cross Reference	System Function R.5
Pre-Requisites	N/A
Typical Courses of Events	(A)Actor, (S)System 1.(A) 관리자가 관리 메뉴 버튼을 누른다. 2.(S) ID와 입력 화면을 출력한다 3.(A) 관리자가 ID를 입력한다 4.(S) 유효한 관리자 아이디인지 확인한다. 5.(A) PW 입력화면을 출력한다. 6.(A) 관리자의 PW를 입력한다 7.(S) 관리자의 PW를 확인한다 8.(S) 잔고를 추가할 것인지 뺄 것인지 선택하는 메뉴를 출력한다. 9.(A) 관리자가 잔고를 추가할 것인지 뺄 것인지 메뉴를 선택한다. 10.(S) 금액을 입력하는 메뉴를 출력한다. 11.(A)변화시킬 잔고 금액을 입력한다. 12.(S) ATM에 변화된 잔고를 갱신한다
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 관리자의 ID와 PW가 잘못된 경우 에러 메시지를 출력한다 E2. 해낼 잔고가 ATM 잔고보다 많을 경우 에러 메시지를 출력한다

Specification Review - Stage 2030

- 2037. Define State Diagram
 - WaitForAmount와 Menu 사이에 잔고를 보여주는 상태가 존재



- 2038. Refine System Test Case
 - 1009에서 정의한 System test case와 항목 및 내용이 완전히 달라짐

Specification Review - Stage 2030

- 2038. Refine System Test Case
 - 1009에서 정의한 System test case와 항목 및 내용이 완전히 달라짐

Activity 1009. Define System Test Case

Test Number	Test 항목	Description	Use Case	System Function
1	출금	유효한 계좌를 입력받으면 사용자가 입력한 금액을 계좌의 잔고와 ATM 잔고를 확인해서 제대로 사용자에게 지급하는지 확인하고, 명세서를 출력하는지 확인	withdraw	
2	입금	유효한 계좌를 입력받으면 사용자가 넣은 금액을 통장 잔고에 제대로 더하고, ATM 잔고를 확인하고 명세서를 출력하는지 확인	deposit	
3	송금	유효한 계좌를 입력받으면 사용자가 입력한 금액을 알맞은 계좌에 입금하는지 확인하고, 이용자의 계좌잔고가 줄어드는지 확인	remittance	
4	조회	유효한 계좌를 입력받으면 계좌의	view_account_det	

Activity2038. Refine System Test Case

Test Num	Test 항목	Description	Use case
1-1	input_menu Test	사용자가 입력한 해당 메뉴 화면으로 올바르게 진입하는지 확인한다.	R1.2 ,R.2, R.3, R.4, R.5
2-1	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하지 않으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2,R3,R.4
2-2	input_ID	사용자가 입력한 계좌 혹은 카드번호가 유효하면 다음 단계로 넘어가는지 확인한다.	R1.2, R.2,R3,R.4
3-1	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 아니면 오류메시지를 출력하는지 확인한다.	R1.2, R.3, R.4
3-2	input_PW	사용자가 입력한 비밀번호가 계좌번호의 비밀번호가 맞으면 다음단계로 넘어가는지 확인한다.	R1.2, R.3, R.4
4-1	input_amount	출금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-2	input_amount	출금의 경우 사용자가 입력한 금액이 ATM 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3
4-3	input_amount	송금의 경우 사용자가 입력한 금액이 계좌 잔고보다 많으면 오류메시지를 출력하는지 확인한다.	R1.2, R.2, R.3

Specification Review - Stage 2040

- 2041. Design Real Use Cases
 - 2031을 바탕으로 **전체적인 수정 필요**
 - 1. Withdraw
 - TotalMoney뿐만 아니라 지폐권에 따른 각 잔고도 감소시킨다는 내용 추가
 - 2. Deposit
 - TotalMoney뿐만아니라 지폐권에 따른 각 잔고도 감소시킨다는 내용 추가
 - 6. manage_ATM
 - TotalMoney뿐만 아니라 지폐권에 따른 각 잔고도 증가/감소시킨다는 내용 추가

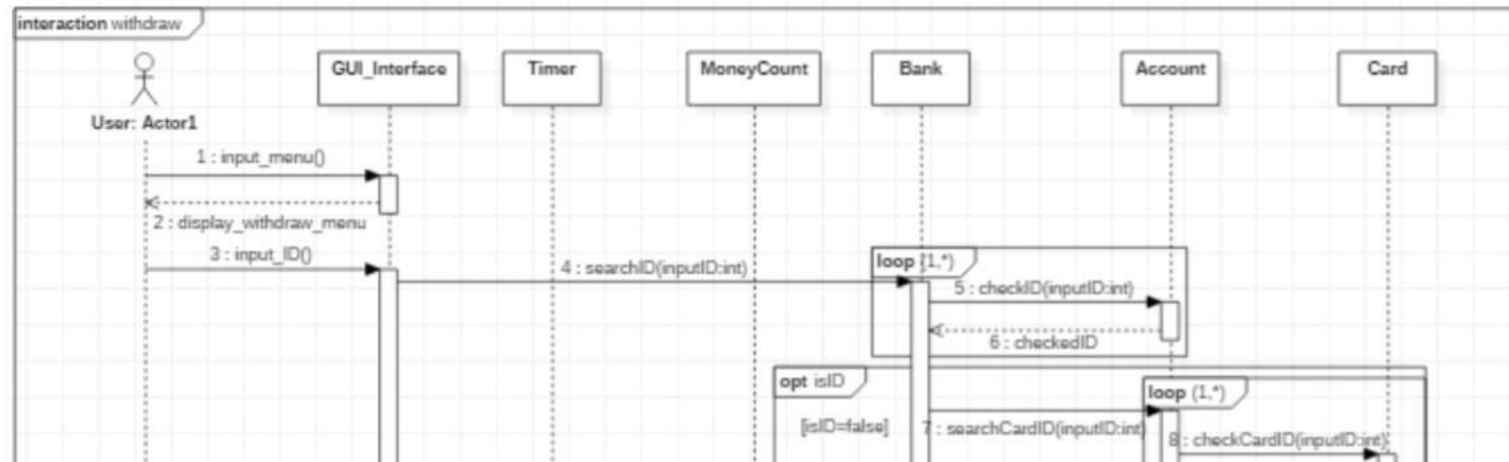
Specification Review - Stage 2040

- 2042. Define Reports UI, and Storyboards
 - '돌아가기'버튼에 대한 명세가 전혀 없었으나 UI 단계에서 등장
 - 금액을 입력할 때 지폐권을 나누어서 입력한다는 명세가 없었음



Specification Review - Stage 2040

- 2044. Define Interaction Diagrams
 - 실제로 GUI_Interface라는 클래스는 존재하지 않음
 - 대신 System_Control & MainFrame 존재
 - Timer 클래스를 사용하지않는 diagrams에서는 생략 필요



Specification Review - Stage 2050

- 2051. Implement Class & Method Definitions
 - Class Definitions - System_Control
 - 명세에 정의되지 않은 메소드가 소스코드 상에 존재(get ATMcount, get_inputMoney, inputRID, get_isjackpot, set_usercount)
 - Method Definitions
 - 1. System_Control
 - 생성자 메소드에 대한 명세 누락
 - 2. input_menu
 - 명세와 소스코드의 input 값 자료형 불일치
 - Abstract Operation : “사용자가 입력한 기능”이라는 표현이 모호함
 - 3. input_ID
 - 명세와 소스코드의 input, output 값 자료형 불일치
 - 4. input_amount
 - 명세와 소스코드의 input, output 값 자료형 불일치
 - Purpose, Abstract Operation에서는 inputAmount attribute가 존재하나 소스코드상에서는 존재하지 않음

Specification Review - Stage 2050

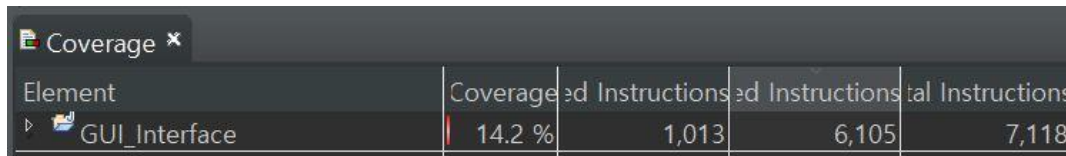
- 2051. Implement Class & Method Definitions
 - Method Definitions
 - 5. input_PW
 - 명세와 소스코드의 input, output값 자료형 불일치
 - Purpose, Abstract Operation에서는 inputPW attribute가 존재하나 소스코드상에서는 존재하지 않음.
 - 6. updateBalance
 - 명세와 소스코드의 output값 자료형 불일치
 - Abstract Operation에서는 Account.Balance가 존재하나 소스코드 상에서는 존재하지 않음.
 - 7. input_MID
 - 명세와 소스코드의 input, output값 자료형 불일치
 - Abstract Operation에서는 inputID가 존재하나 소스코드 상에서는 존재하지 않음

Specification Review - Stage 2050

- 2051. Implement Class & Method Definitions
 - Method Definitions
 - 8. input_POM
 - 명세와 소스코드의 input값 자료형 불일치
 - Abstract Operation에는 inpuPW가 존재하나 소스코드 상에는 존재하지 않음.
 - 오타 : 입력력 > 입력, inpuPW > inputPW
 - 9. input_MPW
 - 명세와 소스코드의 input, output값 자료형 불일치

Specification Review - Stage 2050

- 2051. Implement Class & Method Definitions
 - Method Definitions
 - 1. Manager
 - 생성자 메소드에 대한 명세누락
 - 2. Manager.checkID
 - 소스코드상에서는 checkID
 - Abstract Operation **오타** : inputMID > inputID, ManagerIID > ManagerID
 - 3. Manager.CheckPW
 - 소스코드상에서는checkPW
 - Abstract Operation **오타** : inputMPW > inputPW
- 2055. Write Unit Test Code
 - Unit Test를 거치지 않은 함수들 존재



Element	Coverage	Covered Instructions	Uncovered Instructions	Total Instructions
GUI_Interface	14.2 %	1,013	6,105	7,118

Specification Review - Stage 2060

- 2063. System Testing
 - System Testing **결과차이존재**(2-1, 3-1, 5-1, 6-1, 7-1: Pass -> Fail)
 - 숫자가 아닌 다른 character를 입력하거나 Integer 범위 밖에 있을 때는 오류메시지를 출력하지 않는다.
 - 8-1 “plus를 입력 하였을 때” **표현 매우 모호**
 - 8-2 “minus를 입력 하였을 때” **표현 매우 모호**
 - test case의 내용 자체가 굉장히 **광범위하게 작성됨**

Category-partition Testing

- Test Cases
- User/Manager를 분리

User

Environments

Program Mode

Parameters

Account ID

Receiver ID

Bill

Amount

Password input

Return

Manager

Environments

Program Mode

Parameters

ID Input

Password Input

Select

Bill

Amount

Return

Category-partition Testing

- User
- Environments
 - Program Mode
 - Main Menu [single]
 - Withdraw [property Withdraw]
 - Deposit [property Deposit]
 - Transfer [property Transfer]
 - View [property View]

Category-partition Testing

- User
- Parameters
 - Account ID
 - Invalid ID [error]
 - Correct ID
 - Incorrect ID [error]
 - Receiver ID
 - Invalid ID [error][if Transfer]
 - Correct ID [if Transfer]
 - Incorrect ID [error][if Transfer]

Category-partition Testing

- User
- Parameters
 - Bill
 - 50000 [if Deposit || Withdraw]
 - 10000 [if Deposit || Withdraw]
 - 5000 [if Deposit]
 - 1000 [if Deposit]
 - Amount
 - Invalid [error]
 - None [error]
 - One less than Integer boundary [if !View]
 - More than One [if Withdraw || Deposit]
 - Over balance [error][if Withdraw || Transfer]
 - Over Integer boundary [if !View]

Category-partition Testing

- User
- Parameters
 - Password input
 - Invalid Password [error]
 - Correct Password [if !Deposit]
 - Incorrect Password [error]
 - Return
 - Click Return [single]

Category-partition Testing

- Manger
- Environments
 - Program Mode
 - Main Menu [single]
 - Manage ATM [property Manage_ATM]

Category-partition Testing

- **Manager**
- **Parameters**
 - **ID Input**
 - Invalid ID [error]
 - Correct ID [if Manage_ATM]
 - Incorrect ID [error]
 - **Password Input**
 - Invalid Password [error]
 - Correct Password [if Manage_ATM]
 - Incorrect Password [error]
 - **Select**
 - Click Deposit [property Deposit]
 - Click Withdraw [property Withdraw]

Category-partition Testing

- **Manager**
- **Parameters**
 - **Bill**
 - 50000
 - 10000
 - 5000
 - 1000
 - **Amount**
 - Invalid [error]
 - None [error]
 - One less than Integer boundary
 - More than One
 - Over balance [error][if Withdraw]
 - Over Integer boundary
 - **Return**
 - Click Return [single]

Test Case > 32개

Category-partition Testing

- Testing Result - User

User	atm-5:Test Case 1	Passed	User	atm-21:Test Case 17	Failed
User	atm-6:Test Case 2	Failed	User	atm-22:Test Case 18	Failed
User	atm-7:Test Case 3	Failed	User	atm-23:Test Case 19	Failed
User	atm-8:Test Case 4	Failed	User	atm-24:Test Case 20	Failed
User	atm-9:Test Case 5	Failed	User	atm-25:Test Case 21	Failed
User	atm-10:Test Case 6	Failed	User	atm-26:Test Case 22	Failed
User	atm-11:Test Case 7	Passed	User	atm-27:Test Case 23	Failed
User	atm-12:Test Case 8	Failed	User	atm-28:Test Case 24	Failed
User	atm-13:Test Case 9	Failed	User	atm-29:Test Case 25	Failed
User	atm-14:Test Case 10	Failed	User	atm-30:Test Case 26	Failed
User	atm-15:Test Case 11	Passed	User	atm-31:Test Case 27	Failed
User	atm-16:Test Case 12	Passed	User	atm-32:Test Case 28	Failed
User	atm-17:Test Case 13	Passed	User	atm-33:Test Case 29	Failed
User	atm-18:Test Case 14	Failed	User	atm-34:Test Case 30	Failed
User	atm-19:Test Case 15	Passed	User	atm-39:Test Case 31	Failed
User	atm-20:Test Case 16	Passed	User	atm-40:Test Case 32	Passed

Category-partition Testing

- Testing Result - Manager

Manager	atm-41:Test Case 1	Passed	Manager	atm-57:Test Case 17	Failed
Manager	atm-42:Test Case 2	Failed	Manager	atm-58:Test Case 19	Passed
Manager	atm-43:Test Case 3	Failed	Manager	atm-59:Test Case 20	Failed
Manager	atm-44:Test Case 4	Failed	Manager	atm-60:Test Case 21	Failed
Manager	atm-45:Test Case 5	Failed	Manager	atm-61:Test Case 22	Passed
Manager	atm-46:Test Case 6	Failed	Manager	atm-62:Test Case 23	Passed
Manager	atm-47:Test Case 7	Passed	Manager	atm-63:Test Case 24	Failed
Manager	atm-48:Test Case 8	Failed	Manager	atm-64:Test Case 25	Passed
Manager	atm-49:Test Case 9	Passed	Manager	atm-65:Test Case 26	Passed
Manager	atm-50:Test Case 10	Failed	Manager	atm-66:Test Case 27	Failed
Manager	atm-51:Test Case 11	Failed	Manager	atm-67:Test Case 28	Passed
Manager	atm-52:Test Case 12	Failed	Manager	atm-68:Test Case 29	Passed
Manager	atm-53:Test Case 13	Failed	Manager	atm-69:Test Case 30	Failed
Manager	atm-54:Test Case 14	Failed	Manager	atm-70:Test Case 31	Passed
Manager	atm-55:Test Case 15	Failed	Manager	atm-71:Test Case 32	Passed
Manager	atm-56:Test Case 16	Passed	Manager	atm-72:Test Case 33	Failed

21/64 = 32% pass

Pairwise Testing

- User
- 입력 데이터

Program Mode	Account ID	Receiver ID	Bill	Amount	Password Input
Withdraw	Correct ID	Correct ID	50000	One less than Integer boundary	
Deposit			10000	More than One	Correct Password
Transfer			5000	Over Integer boundary	
View			1000		

- 테스트 케이스

TEST CASES							
case	Program Mode	Account ID	Receiver ID	Bill	Amount	Password Input	pairings
1	Withdraw	Correct ID	Correct ID	50000	One less than Integer boundary	Correct Password	15
2	Deposit	Correct ID	Correct ID	10000	More than One	Correct Password	12
3	Transfer	Correct ID	Correct ID	5000	Over Integer boundary	Correct Password	12
4	View	Correct ID	Correct ID	1000	One less than Integer boundary	Correct Password	9
5	Withdraw	~Correct ID	~Correct ID	10000	Over Integer boundary	~Correct Password	3
6	Deposit	~Correct ID	~Correct ID	50000	Over Integer boundary	~Correct Password	3
7	Transfer	~Correct ID	~Correct ID	1000	More than One	~Correct Password	3
9	Withdraw	~Correct ID	~Correct ID	5000	One less than Integer boundary	~Correct Password	2
10	Deposit	~Correct ID	~Correct ID	1000	One less than Integer boundary	~Correct Password	2
11	Transfer	~Correct ID	~Correct ID	50000	More than One	~Correct Password	2
13	Withdraw	~Correct ID	~Correct ID	1000	More than One	~Correct Password	2
14	Transfer	~Correct ID	~Correct ID	10000	One less than Integer boundary	~Correct Password	3
15	Deposit	~Correct ID	~Correct ID	5000	~Over Integer boundary	~Correct Password	1
17	~Withdraw	~Correct ID	~Correct ID	1000	Over Integer boundary	~Correct Password	1

Pairwise Testing

- Manager
- 입력 데이터

Program Mode	ID Input	Password Input	Select	Bill	Amount
Manage ATM	Correct ID	Correct Password	Click Deposit	50000	One less than Integer boundary
			Click Withdraw	10000	More than One
				5000	Over Integer boundary
				1000	

- 테스트 케이스

TEST CASES							
case	Program Mode	ID Input	Password Input	Select	Bill	Amount	pairings
1	Manage ATM	Correct ID	Correct Password	Click Deposit	50000	One less than Integer boundary	15
2	Manage ATM	Correct ID	Correct Password	Click Withdraw	10000	More than One	12
3	Manage ATM	Correct ID	Correct Password	Click Deposit	5000	Over Integer boundary	9
4	Manage ATM	Correct ID	Correct Password	Click Withdraw	1000	One less than Integer boundary	6
5	~Manage ATM	~Correct ID	~Correct Password	Click Deposit	50000	More than One	2
6	~Manage ATM	~Correct ID	~Correct Password	Click Withdraw	10000	Over Integer boundary	2
7	~Manage ATM	~Correct ID	~Correct Password	Click Withdraw	5000	One less than Integer boundary	2
8	~Manage ATM	~Correct ID	~Correct Password	Click Deposit	1000	More than One	2
9	~Manage ATM	~Correct ID	~Correct Password	Click Withdraw	50000	Over Integer boundary	2
10	~Manage ATM	~Correct ID	~Correct Password	Click Deposit	10000	One less than Integer boundary	2
11	~Manage ATM	~Correct ID	~Correct Password	~Click Withdraw	5000	More than One	1
12	~Manage ATM	~Correct ID	~Correct Password	~Click Deposit	1000	Over Integer boundary	1

Pairwise Testing

- Testing Result

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

4/26 = 15% pass

Brute Force Testing

- Testing Result

Test Suite	Test Case	Result
Manager	관리자 프로세스에서 5 번 반복하여 입/출금 할 수 있다.	Failed
Manager	관리자 프로세스에서 10 번 반복하여 입/출금 할 수 있다.	Failed
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
User	1 일 1 명만 Jackpot 이 터진다.	Failed
User	출금 시에만 Jackpot 이 터진다.	Failed
User	하나의 통장으로 여러 번 출금할 시 Jackpot 당첨이 여러 번 되지 않게 한다.	Failed
User	모든 입력에 대한 기기의 반응은 1 초 이내로 이루어진다.	Failed
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
User	세 번 이상 연속적으로 송금할 수 있다.	Failed
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
User	입금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed
User	송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.	Failed
Manager	ATM 잔고가 일정수준(50 만원) 이하가 되면 잔고부족 메시지를 출력한다.	Failed

8/19 = 42% Pass

Brute Force Testing

- Failed Case Report

Test Suite	Test Case
Manager	잔고가 어느 일정 범위 이상 커지면 "잔고 확인하세요"라는 오류 메시지가 출력된다. (잔고 최대값 명세에 존재하지 않음)
User	Jackpot 이 터져도 ATM 의 잔고에는 변화가 생기지 않는다.
User	초기 실행 후 Jackpot 이 한 번 터진 후에도 다시 터진다.
User	송금 시 사용자의 비밀번호를 입력하고 나서도 Jackpot 이 터지는 경우 존재
User	integer 이내의 정수값이 아니라면, 1 초 이내로 반응이 없다(반응이 아예 없다)
User	같은 계좌로 출금을 5 회 이상 반복 했을 때, 연속으로 2 번 잭팟이 터지는 경우가 있다
User	잔고가 어느 일정 범위 이상 커지면 "비밀번호 확인하세요"라는 오류 메시지가 출력된다.
User	3 번 이상 송금 하려고 하면 알맞은 비밀번호여도 오류 메시지가 나온다
User	출금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 증가한다. 예외처리 필요
User	입금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 감소한다. 예외처리 필요
User	송금 시 minus 정수 값을 금액으로 입력하면 해당 정수의 절대값만큼 계좌의 잔액이 증가한다. 예외처리 필요
Manager	ATM 잔고가 50 만원 이하가 되어도 아무 메시지도 출력되지 않는다.

1st System Testing Result

- Category-partition Test
 - $21/64 = 32\%$ pass
- Parwise Test
 - $4/26 = 15\%$ pass
- Brute Force Test
 - $6/19 = 31\%$ Pass

Summary

- 문서의 단계마다 일치하지 않는 부분이 존재
- Specification이 명확하지 않고 모호함, 추가 작성 필요
- 문서와 실제 구현된 소스코드와의 차이점이 존재
- 초반 설계한 Functional Requirements 중 구현하지 않은 부분 있음
- 프로그램 자체의 기능이 부족함
 - 실생활에 활용되는 ATM이라는 점을 고려했을 때 기능이 매우 부실
 - 키보드 입력에 대한 예외 처리가 없음

In CTIP

- Testlink

The screenshot displays the Testlink interface. On the left is a tree view of test cases under 'Jackpot_ATM / test (109)(0, 32, 77, 0)'. The main area shows 'Test Results on Build test version0' with a table of execution results and a detailed view of a specific test case.

Test Results on Build test version0

Print Show complete execution history Import Results (XML)

Test Plan test

Build test version0

Test Suite : User/

Details

Latest execution (any build)

Date : 26/05/2018 17:33:27 - Tested by : admin - Build : test version0 - Status : Passed

Latest execution (current build) - Build : test version0

Date	Build	Tested by	Status	Exec (min)	Version	BUG management
26/05/2018 17:33:27	test version0	admin	Passed		1	

Test Case atm-5 :: Version : 1 :: Test Case 1

Assigned to : admin

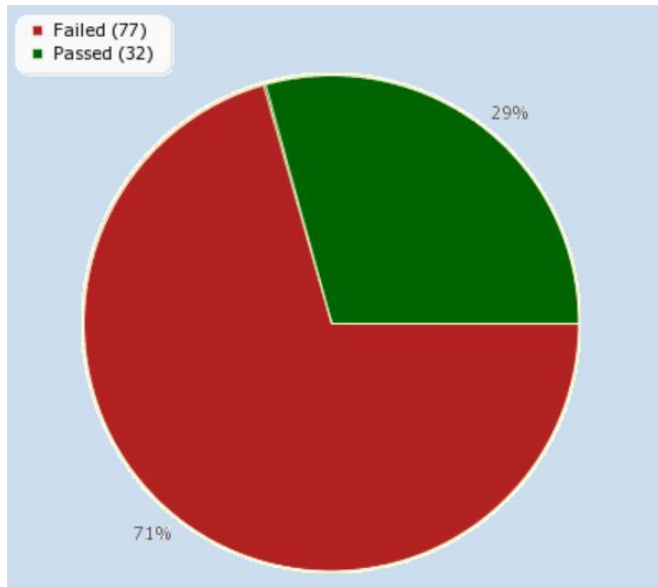
Summary

Preconditions

#	Step actions	Expected Results	Execution	Execution notes
1	Program Mode : Main Menu		Manual	

In CTIP

- Testlink



In CTIP

- Redmine

The screenshot shows the Redmine interface for a project named 'Jackpot_ATM'. The top navigation bar includes tabs for '개요' (Overview), '작업내역' (Work History), '일감' (Issues), '소요 시간' (Time Spent), 'Gantt 차트' (Gantt Chart), '달력' (Calendar), '문서' (Documents), '위키' (Wiki), '파일' (Files), '지점소' (Locations), 'Jenkins', and '설정' (Settings). The '일감' tab is active, displaying a list of issues.

Filters for the issue list are set to: State: 진행중 (In Progress), Type: 이다 (is), and Category: Bug. There are 43 issues listed in total.

#	유형	상태	우선순위	제목
34	Bug	New	Normal	[Brute Force Test] Jackpot이 터지면 해당 User에게 5만원을 추가 인출하며 ATM의 잔고에서 5만원이 빠져나간다.
33	Bug	New	Normal	[Category Partition Test] User : Account ID : Incorrect ID
32	Bug	New	Normal	[Category Partition Test] 관리자 ID에서 integer외의 것을 입력해도 오류 메시지가 없음
31	Bug	New	Normal	[BruteForce Testing] 송금 3번 연속 불가 (동일 카드로는 아예 송금이 되지 않고, 다른 카드로도 3번 이상 송금 하려고 하면 알맞은 바 호여도 오류 메시지가 나온다)
30	Bug	New	Normal	[BruteForce Testing] 같은 계좌로 출금을 5회 이상 반복 했을 때, 연속으로 2번 찍땀이 터지는 경우가 있다
29	Bug	New	Normal	[BruteForce Testing] integer 이내의 정수값이 아니라면, 1초 이내로 반응이 없다(반응이 아예 없다)
28	Bug	New	Normal	[Category Partition Test] 특정 범위(integer)를 벗어나는 금액을 입력하면 입금 불가
27	Bug	New	Normal	[Category Partition Test] 특정 범위(integer) 밖의 값 입력하면 출금 불가.
26	Bug	New	Normal	[Category Partition Test] 잔액 보다 높은 금액을 입력해도 integer 범위를 벗어나면 예외 메시지가 없음
25	Bug	New	Normal	[Brute Force Test] 출금 시에만 Jackpot이 터진다.
24	Bug	New	Normal	[Category Partition Test] User : Amount : Invalid
23	Bug	New	Normal	[Category Partition Test] Manager : Amount : Over Integer boundary
22	Bug	New	Normal	[Category Partition Test] Manager : Amount : Invalid
21	Bug	New	Normal	[Category Partition Test] Manager : ID Input : Incorrect ID
20	Bug	New	Normal	[Brute Force Test] 1일 1명만 Jackpot이 터진다.
19	Bug	New	Normal	[Brute Force Test] 세 번 이상 연속적으로 입력할 수 있다.
18	Bug	New	Normal	[Category Partition Test] User : Deposit 1000
17	Bug	New	Normal	[Category Partition Test] User : Deposit 5000
16	Bug	New	Normal	[Category Partition Test] User : Deposit 10000
15	Bug	New	Normal	[Category Partition Test] User : Deposit 50000
14	Bug	New	Normal	[Brute Force Test] 송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.
13	Bug	New	Normal	[Brute Force Test] 송금 시 금액 입력 값에 minus 정수를 입력하면 다음 단계로 넘어가지 않거나, 오류 메시지가 출력된다.

THANK YOU