

A NEW C.S.E. ATM

OOPT Stage 2030

Project Team

T3

Date

2018-04-28

Team Information

201311299 이원오

201311301 이재규

201311309 전홍준

INDEX

- 1. Activity 2010. Revise Plan**
- 2. Activity 2020. Synchronize Artifacts**
- 3. Activity 2031. Define Essential Use Cases**
- 4. Activity 2032. Refine Use Case Diagrams**
- 5. Activity 2033. Define Domain Model**
- 6. Activity 2034. Refine Glossary**
- 7. Activity 2035. Define System Sequence Diagrams**
- 8. Activity 2036. Define Operation Contracts**
- 9. Activity 2037. Define State Diagrams**
- 10. Activity 2038. Refine System Test Case**
- 11. Activity 2039. Analyze Traceability Analysis**

1. Activity 2010. Revise Plan

- Activity 1003의 Functional Requirements
- version1에서 잘못 표시된 use case들을 지우고 Actor의 기준에서 생각할 수 있는 case 추가

| Function | Description |
|----------------|--|
| Deposit | User가 사용할 카드나 통장에서 요청한 금액만큼 돈을 받은 뒤, 입금을 해준다 |
| Withdraw | User가 사용할 카드나 통장에서 요청한 금액만큼 돈을 출금해준다. |
| Transfer | User가 입력한 카드나 통장에서 목적 계좌에 돈을 이체한다. |
| Check Balance | User가 원하는 카드나 통장의 잔고를 확인한다. |
| Robbery Report | User가 도난 신고한 카드의 정보를 저장한다. |
| Add Cash | 관리자가 요청한 금액만큼 ATM의 현금을 추가한다. |
| Check Cash | ATM의 현금 잔액을 확인한다. |

2. Activity 2020. Synchronize Artifacts

- Version OOPT stage 1000 version 1에서 OOPT stage version 2로 수정하였다.

3. Activity 2031. Define Essential Use Case

- 1. Deposit

| | |
|-------------------------------|--|
| Use Case | 1. Deposit |
| Actor | User |
| Purpose | 사용자가 원하는 계좌에 입금을 한다. |
| Overview | User가 입금버튼을 누른다. |
| Type | Primary and Essential |
| Cross Reference | Functional Requirements : R.1. |
| Pre-Requisites | N/A |
| Typical courses of events | (A) : Actor, (S) : System 1. (A) User가 프로그램(UI)에서 Deposit버튼을 누른다. 2. (S) User에게 Deposit UI를 띄워준다. 3. (A) 입금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력창으로 UI 변경 6. (A) User가 입금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에 추가한다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | 1. (S) 잘못된 계좌정보를 입력하거나 입금금액이 일정 금액(100만원)을 초과하면 알림을 띄운다. |

- 2. Withdraw

| | |
|-------------------------------|--|
| Use Case | 2. Withdraw |
| Actor | User |
| Purpose | 사용자가 원하는 계좌에서 출금을 한다. |
| Overview | User가 출금버튼을 누른다. |
| Type | Primary and Essential |
| Cross Reference | Functional Requirements : R.2. |
| Pre-Requisites | N/A |
| Typical courses of events | <p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> 1. (A) User가 프로그램(UI)에서 Withdraw버튼을 누른다. 2. (S) User에게 Withdraw UI를 띄워준다. 3. (A) 출금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 금액 입력창으로 UI 변경 6. (A) User가 출금할 금액을 입력한다. 7. (S) User가 입력한 금액을 해당 계좌나 카드에서 빼준다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | <ol style="list-style-type: none"> 1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 출금금액이 한계 금액(100만원)이나 계좌 잔액을 초과하면 알림을 띄운다. |

- 3. Transfer

| | |
|-------------------------------|--|
| Use Case | 3. Transfer |
| Actor | User |
| Purpose | 사용자가 원하는 계좌로 자신의 계좌에서 돈을 이체한다. |
| Overview | User가 계좌이체 버튼을 누른다. |
| Type | Primary and Essential |
| Cross Reference | Functional Requirements : R.3. |
| Pre-Requisites | N/A |
| Typical courses of events | (A) : Actor, (S) : System 1. (A) User가 프로그램(UI)에서 Transfer버튼을 누른다. 2. (S) User에게 Transfer UI를 띄워준다. 3. (A) 출금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 이체 계좌정보 및 금액 입력창으로 UI 변경 6. (A) User가 이체할 계좌정보와 이체할 금액을 입력한다. 7. (S) User가 입력한 금액을 이체 계좌나 카드에 더해주고 해당 계좌에서는 그 금액만큼 빼준다. 8. (S) 변경 내용을 저장한다. 9. (S) 변경 내용을 User에게 보여준다. |
| Alternative Courses of Events | |
| Exceptional Courses of Events | 1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 이체금액이 한계 금액(100만원)이나 계좌 잔액을 초과하면 알림을 띄운다. |

- 4. Check Balance

| | |
|-------------------------------|--|
| Use Case | 4. Check Balance |
| Actor | User |
| Purpose | 사용자가 원하는 계좌의 잔액을 보여준다. |
| Overview | User가 잔액확인 버튼을 누른다. |
| Type | Primary and Essential |
| Cross Reference | Functional Requirements : R.4. |
| Pre-Requisites | N/A |
| Typical courses of events | (A) : Actor, (S) : System 1. (A) User가 프로그램(UI)에서 Check Balance버튼을 누른다. 2. (S) User에게 Check Balance UI를 띄워준다. 3. (A) 확인할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 해당계좌나 카드의 잔액을 화면에 띄운다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | 1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하면 알림을 띄운다. |

-
-
-
-
-
-
-
-
-
-

- 5. Robbery Report

| | |
|-------------------------------|--|
| Use Case | 5. Robbery Report |
| Actor | User |
| Purpose | 사용자가 도난 신고를 한다. |
| Overview | User가 도난 신고 버튼을 누른다. |
| Type | Primary and Essential |
| Cross Reference | Functional Requirements : R.5. |
| Pre-Requisites | N/A |
| Typical courses of events | (A) : Actor, (S) : System 1. (A) User가 프로그램(UI)에서 도난 신고버튼을 누른다. 2. (S) User에게 도난신고 UI를 띄워준다. 3. (A) 도난 신고할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다. 4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다. 5. (S) 계좌정보(카드)를 도난 상태로 변경. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | 1. (S) 이미 있는 도난 카드의 번호를 입력했을 시에는 화면에 알림을 띄운다. |

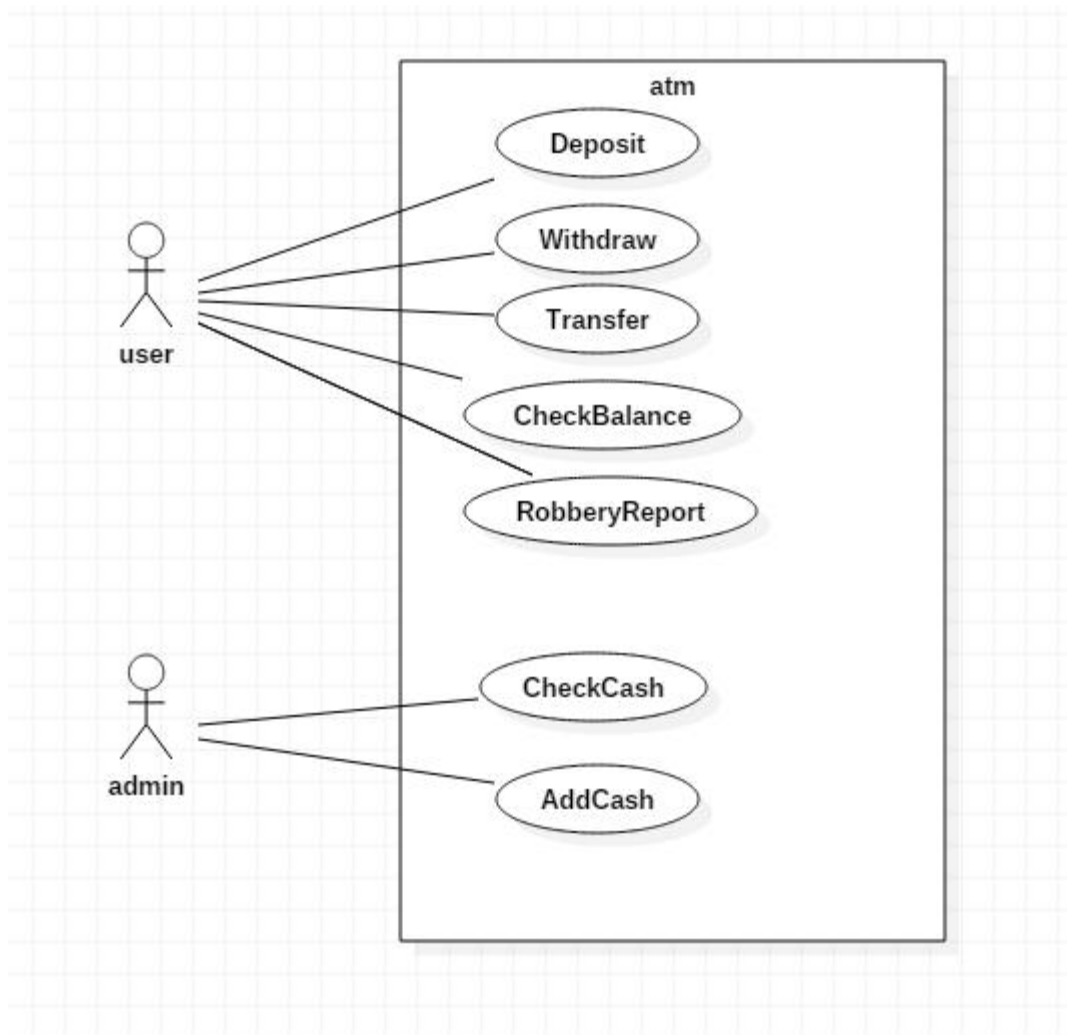
- 6. Add Cash

| | |
|-------------------------------|--|
| Use Case | 6. Add Cash |
| Actor | Admin |
| Purpose | 관리자가 ATM에 현금을 추가한다. |
| Overview | 관리자가 비밀번호를 입력해서 현금을 추가한다. |
| Type | Primary and Essential |
| Cross Reference | Functional Requirements : R.6. |
| Pre-Requisites | Actor가 Admin이어야한다. |
| Typical courses of events | (A) : Actor, (S) : System 1. (A) 관리자가 Admin버튼을 누른다. 2. (S) 관리자 확인 UI로 바꿔준다. 3. (A) 관리자 ID/PW를 입력한다. 4. (S) 관리자가 입력한 ID/PW가 올바르게 입력된지 확인한다. 5. (S) 관리자 UI로 바꿔준다. 6. (A) Add Cash 버튼을 누른다. 7. (S) Add Cash UI로 바꾼다. 8. (A) 관리자가 추가할 현금 액수를 입력한다. 9. (S) 입력한 액수만큼 잔액에 더해준다. 10. (S) 변경내용을 저장한다. |
| Alternative Courses of Events | |
| Exceptional Courses of Events | 1. (S) 관리자 모드 정보를 잘못 입력하면 접근이 되지 않는다. |

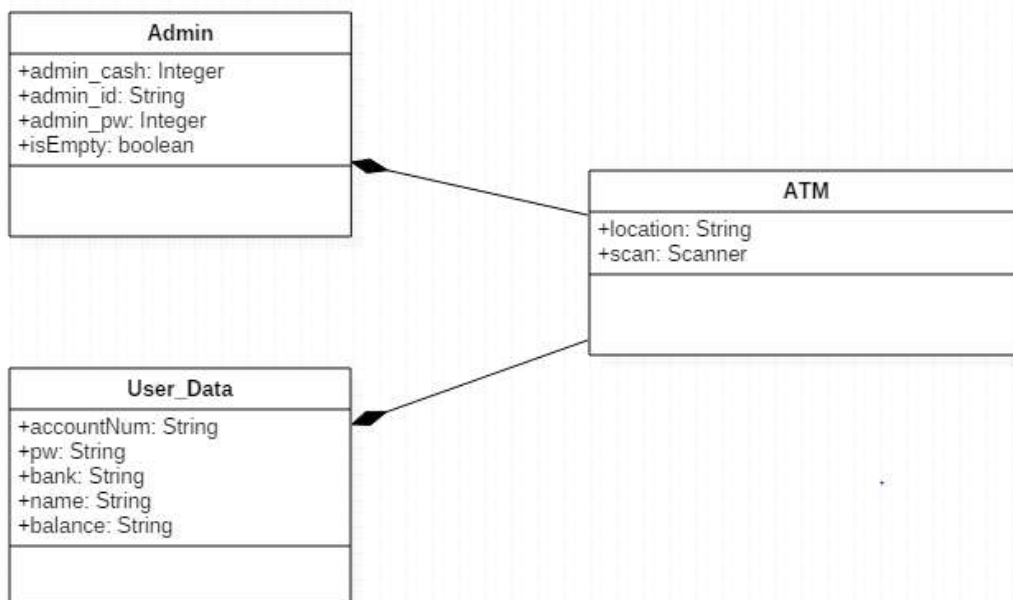
- 7.Check Cash

| | |
|-------------------------------|--|
| Use Case | 7. Check Cash |
| Actor | Admin |
| Purpose | ATM의 현금 잔액을 확인한다. |
| Overview | 관리자 모드에서 ATM잔액확인을 누른다. |
| Type | Primary and Essential |
| Cross Reference | Functional Requirements : R.7. |
| Pre-Requisites | Actor가 Admin이어야한다. |
| Typical courses of events | (A) : Actor, (S) : System 1. (A) 관리자가 Admin버튼을 누른다. 2. (S) 관리자 확인 UI로 바꿔준다. 3. (A) 관리자 ID/PW를 입력한다. 4. (S) 관리자가 입력한 ID/PW가 올바르게 입력된지 확인한다. 5. (S) 관리자 UI로 바꿔준다. 6. (A) Check Cash 버튼을 누른다. 7. (S) ATM의 잔액을 화면에 띄운다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | 1. (S) 관리자 모드 비밀번호를 잘못 입력하면 접근이 되지 않는다. |

4. Activity 2032. Refine Use Case Diagrams



5. Activity 2033. Define Domain Model



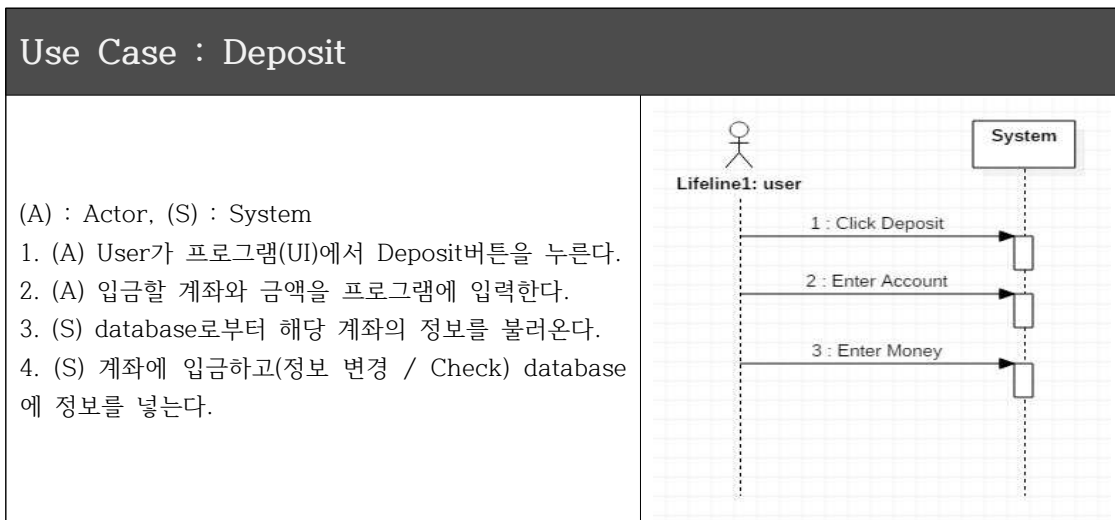
6. Activity 2034. Refine Glossary

| Term | Category | Description |
|------------|------------|---|
| Admin | Class | 관리자의 아이디, 패스워드, 현금을 관리하는 클래스 |
| ATM | Class | 전반적인 ATM시스템 기능들 구현 |
| User_Data | Class | User의 정보들(이름, 은행, 계좌번호, 비밀번호, 잔액)을 관리하는 클래스 |
| admin_cash | Attributes | 관리자 모드의 현금 잔액 (Integer / 초기 : 500만원) |
| Admin_id | Attributes | 관리자 모드의 관리자 아이디 (String / sklee08) |

| Term | Category | Description |
|------------|------------|---|
| Admin_pw | Attributes | 관리자 모드의 관리자 비밀번호 (String / 1234) |
| pw | Attributes | User의 비밀번호를 저장하는 변수 (String / 숫자 4개) |
| accountNum | Attributes | User의 계좌번호를 저장하는 변수 (String / 숫자 6개) |
| bank | Attributes | User의 은행 정보를 나타내는 변수 (String / 국민/신한/농협) |
| name | Attributes | User의 이름을 저장하는 변수 (String / 세 글자의 한글) |
| balance | Attributes | User의 잔액을 저장하는 변수 (String) |

7. Activity 2035. Define System Sequence Diagrams

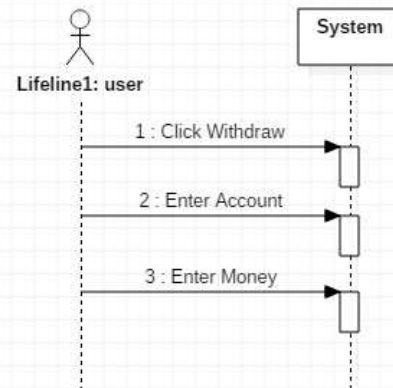
| Uss Case | Name of Actor-Activated Event | System Operation |
|----------------|-------------------------------|------------------|
| Deposit | Deposit | Deposit() |
| Withdraw | Withdraw | Withdraw() |
| Transfer | Transfer | Transfer() |
| Check Balance | Check Balance | Check Balance() |
| Robbery Report | Robbery Report | Robbery Report() |
| Add Cash | Add Cash | Add Cash() |
| Check Cash | Check Cash | Check Cash() |



Use Case : Withdraw

(A) : Actor, (S) : System

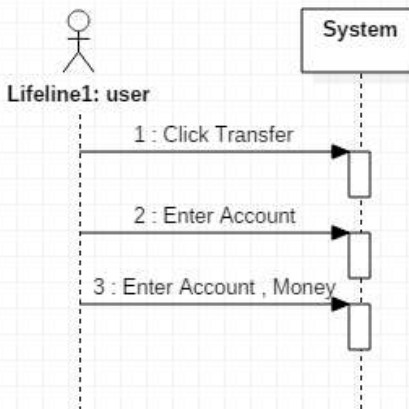
1. (A) User가 프로그램(UI)에서 Withdraw버튼을 누른다.
2. (A) 출금할 계좌와 금액을 프로그램에 입력한다.
3. (S) database로부터 해당 계좌의 정보를 불러온다.
4. (S) 계좌로부터 출금하고(정보 변경 / Check) database에 변경된 정보를 넣는다.



Use Case : Transfer

(A) : Actor, (S) : System

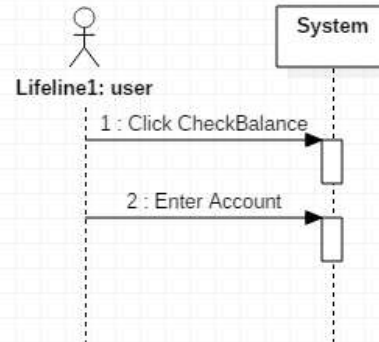
1. (A) User가 프로그램(UI)에서 Transfer버튼을 누른다.
2. (A) 자신의 카드정보와 이체할 계좌 번호를 입력한다.
3. (S) database로부터 해당 계좌와 카드의 정보를 불러온다.
4. (S) 계좌로부터 입력받은 만큼의 돈을 빼준 뒤 이체계좌에 돈을 더해준다.
5. (S) 계좌들의 변경된 정보를 database에 다시 입력한다.



Use Case : Check Balance

(A) : Actor, (S) : System

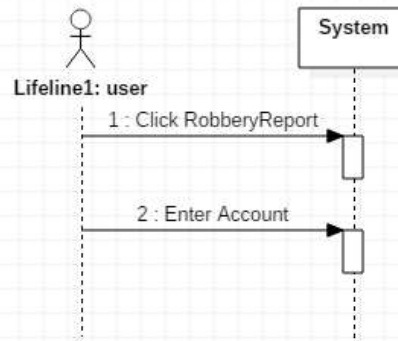
1. (A) User가 프로그램(UI)에서 Check Balance버튼을 누른다.
2. (A) User가 확인하고 싶은 계좌나 카드의 번호를 입력하고 비밀번호도 입력한다.
3. (S) database로부터 해당 계좌나 카드의 정보를 불러온다.
4. (S) 화면에 잔액을 보여준다.



Use Case : Robbery Report

(A) : Actor, (S) : System

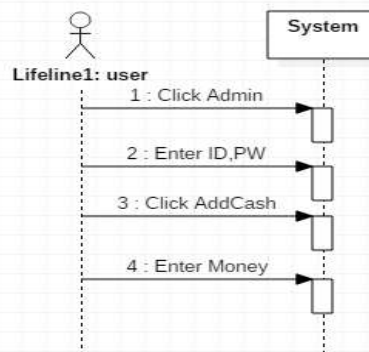
1. (A) User가 프로그램(UI)에서 도난신고 버튼을 누른다.
2. (A) User가 신고할 계좌나 카드의 번호를 입력한다.
3. (S) User가 입력한 카드나 계좌의 번호를 database에 입력한다.



Use Case : Add Cash

(A) : Actor, (S) : System

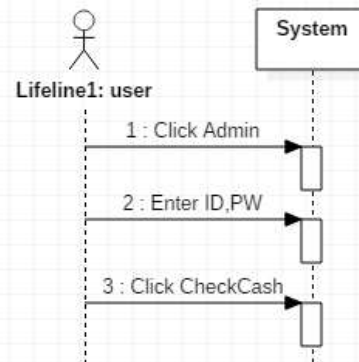
1. (A) Admin이 관리자 모드로 접근한다(비밀번호 입력).
2. (A) 입금할 금액을 넣어준다.
3. (S) database로부터 ATM의 잔액정보에 입금된 금액을 더해 준 뒤 다시 database에 넣는다.



Use Case : Check Cash

(A) : Actor, (S) : System

1. (A) Admin이 관리자 모드로 접근한다(비밀번호 입력).
2. (S) database로부터 ATM의 잔액을 확인한다.
3. (S) 잔액을 화면에 출력한다.



8. Activity 2036. Define Operation Contracts

| | |
|-----------------|--|
| Name | Deposit() |
| Responsibility | 해당 계좌에서 입금을 한다. |
| Type | System |
| Cross Reference | R.1 |
| Exception | 계좌가 없을 시 에러 메시지 출력 금액이 한도 금액을 넘으면 에러 메시지 출력 |
| Output | N/A |
| Pre-Condition | 올바른 계좌정보를 받은 상태 |
| Post-Condition | N/A |

| | |
|-----------------|---|
| Name | Withdraw() |
| Responsibility | 해당 계좌에서 출금을 한다. |
| Type | Actor |
| Cross Reference | R.2, |
| Exception | 계좌가 없을 시 에러 메시지 출력 출금하고자 하는 계좌의 잔액보다 큰 금액이 입력 될 시 에러 메시지 출력 출금하고자 하는 금액이 ATM 기기 내에 현금 잔액보다 큰 값일 시, 에러메세지 출력 |
| Output | N/A |
| Pre-Condition | 올바른 계좌정보를 받은 상태 잔액보다 작은 금액을 받은 상태 |
| Post-Condition | N/A |

| | |
|-----------------|--|
| Name | Transfer() |
| Responsibility | 해당 계좌에서 이체 계좌로 금액을 이체한다. |
| Type | Actor |
| Cross Reference | R.3. |
| Exception | 계좌 또는 이체 계좌가 없을 시 에러 메시지 출력 잔액보다 큰 금액이 입력 될 시 에러 메시지 출력 |
| Output | N/A |
| Pre-Condition | 올바른 계좌정보를 받은 상태 잔액보다 적은 금액을 받은 상태 |
| Post-Condition | N/A |

| | |
|-----------------|-----------------------------|
| Name | Check Balance() |
| Responsibility | 해당 계좌에서 잔액을 조회한다. |
| Type | Actor |
| Cross Reference | R.4. |
| Exception | 계좌 또는 이체 계좌가 없을 시 에러 메시지 출력 |
| Output | N/A |
| Pre-Condition | 올바른 계좌정보를 받은 상태 |
| Post-Condition | N/A |

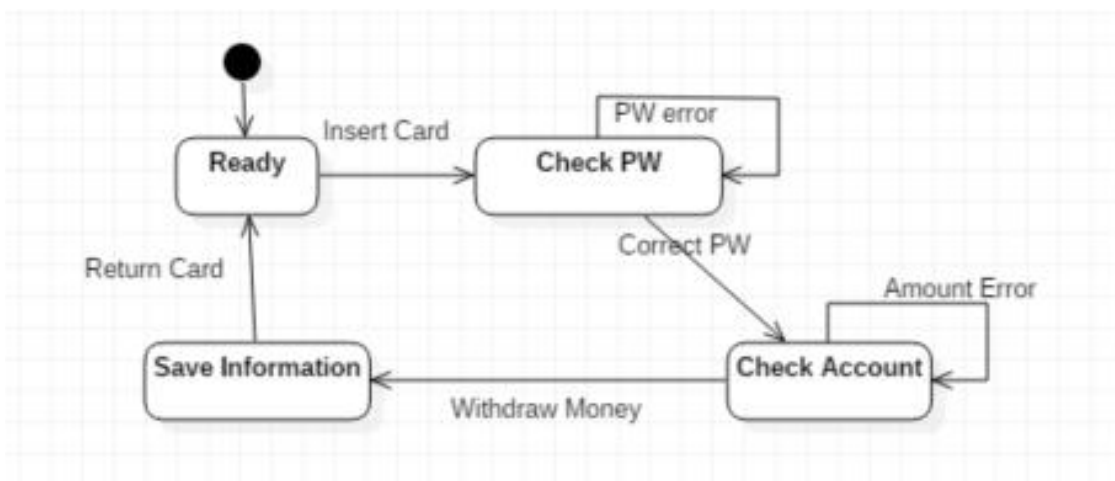
| | |
|-----------------|---|
| Name | RobberyReport() |
| Responsibility | 도난 카드나 계좌의 번호를 저장하고 신고한다. |
| Type | Actor |
| Cross Reference | R.5. |
| Exception | 카드의 계좌정보(계좌번호, 비밀번호)가 잘못 입력되었을 때 오류 메시지 출력 |
| Output | N/A |
| Pre-Condition | 올바른 계좌정보를 받은 상태 |
| Post-Condition | N/A |

| | |
|-----------------|-------------------------------|
| Name | Add Cash() |
| Responsibility | ATM의 현금을 추가해준다. |
| Type | Admin |
| Cross Reference | R.6. |
| Exception | ATM의 관리자 모드 로그인이 실패할 경우 예외 발생 |
| Output | N/A |
| Pre-Condition | 올바른 비밀번호를 받은 상태 |
| Post-Condition | N/A |

| | |
|-----------------|-------------------------------|
| Name | Check Cash() |
| Responsibility | ATM의 현금 잔액을 확인해준다. |
| Type | Admin |
| Cross Reference | R.7. |
| Exception | ATM의 관리자 모드 로그인이 실패할 경우 예외 발생 |
| Output | N/A |
| Pre-Condition | 올바른 비밀번호를 받은 상태 |
| Post-Condition | N/A |

9. Activity 2037. Define State Diagrams

- withdraw



10. Activity 2038. Refine System Test Case

| Test Num | Test 항목 | Description | Use-Case |
|----------|---------------------|---|----------------|
| 1-1 | Deposit Test | 해당 계좌의 번호가 DB에 없을 경우, 오류 메시지가 제대로 출력되는지 확인한다. | Deposit |
| 1-2 | Deposit Test | 해당 계좌로 돈이 잘 입금이 되는 지 확인한다. | Deposit |
| 2-1 | Withdraw Test | 해당 계좌의 번호가 DB에 없을 경우, 그리고 입금 금액이 해당 계좌의 잔액보다 많은 경우에 그에 따른 오류 메시지가 제대로 출력되는지 확인한다. | Withdraw |
| 2-2 | Withdraw Test | 해당 계좌로부터 출금이 잘 되는지 확인한다. | Withdraw |
| 3-1 | Transfer Test | 해당 계좌와 이체 계좌의 정보가 DB에 없을 경우, 그리고 이체 금액이 잔액보다 큰 경우 오류 메시지가 제대로 출력 되는지 확인한다. | Transfer |
| 3-2 | Transfer Test | 이체 금액에 대한 정보가 제대로 업데이트가 되었는지 확인한다. | Transfer |
| 4-1 | Check Balance Test | 해당 계좌의 번호가 DB에 없을 경우, 오류 메시지가 제대로 출력되는지 확인한다. | Check Balance |
| 5-1 | Robbery Report Test | 도난 신고 계좌 정보를 확인하고 올바른 계좌번호와 비밀번호가 입력되었는지 확인한다. | Robbery Report |
| 5-2 | Robbery Report Test | 도난 신고된 계좌의 번호가 DB에서 제대로 삭제되었지 확인한다. | Robbery Report |
| 6-1 | Add Cash Test | ATM에 현금이 제대로 들어갔는지 확인한다. | Add Cash |
| 6-2 | Check Cash Test | ATM의 잔액정보가 올바르게 출력되는지 확인한다. | Check Cash |

11. Activity 2039. Analyze Traceability Analysis

