

Software Modeling

# A NEW C.S.E. ATM

Stage 2030

201311299 | 이원오  
201311301 | 이재규  
201311309 | 전홍준

CrePAS 6<sup>th</sup> the first session

# Back to Basic

아주 상식적인 프레젠테이션 기법

크레파스 1기 | 물고기

A NEW C.S.E. ATM

# INDEX

- Revise Plan
- Define Essential Use Case
- Refine Use Case Diagrams
- Define Domain Model
- Refine Glossary
- Define System Sequence Diagrams
- Define Operation Contracts
- Define State Diagrams
- Refine System Test Case
- Analyze Traceability Analysis

# Revise Plan

Stage 2010

# C.S.E.?



--- . **C : Convenient**



--- . **S : Secure**



--- . **E : Equipment**

01

02

03

04

05

06

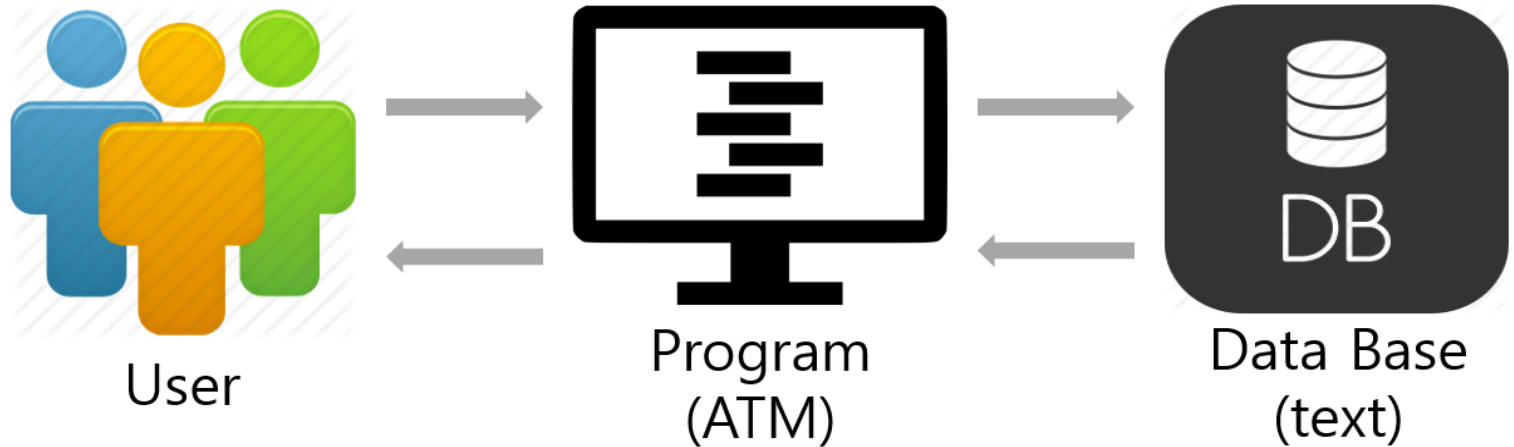
07

08

09

10

# System Architecture



01

02

03

04

05

06

07

08

09

10

# Functional Requirements

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.
Language	화면의 인터페이스를 쓰고 싶은 언어로 선택한다.
Check Balance	User가 원하는 카드나 통장의 잔고를 확인한다.
Size Up	화면의 인터페이스를 확대한다.
Robbery Report	User가 도난 신고한 카드의 정보를 저장한다.
Add Cash	관리자가 요청한 금액만큼 ATM의 현금을 추가한다.
Check Cash	ATM의 현금 잔액을 확인한다.
Load	database로부터 정보를 가져온다.
Store	Database로 바뀐 정보를 올린다.
Check	정보를 비교해서 바뀐 내용을 바꾼다.

01

02

03

04

05

06

07

08

09

10

# Define Essential Use Cases

Stage 2031



## Define Essential Use Case

01	Use Case	1. Deposit
02	Actor	User
03	Purpose	사용자가 원하는 계좌에 입금을 한다.
04	Overview	User가 입금버튼을 누른다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.1,10,11,12
07	Pre-Requisites	N/A
08	Typical courses of events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) User가 프로그램(UI)에서 Deposit버튼을 누른다.</li> <li>2. (A) 입금할 계좌와 금액을 프로그램에 입력한다.</li> <li>3. (S) database로부터 해당 계좌의 정보를 불러온다.</li> <li>4. (S) 계좌에 입금하고(정보 변경 / Check) database에 정보를 넣는다.</li> </ol>
09	Alternative Courses of Events	2. (S) 잘못된 계좌정보를 입력하거나 입금금액이 일정 금액을 초과하면 알림을 띄운다.
10	Exceptional Courses of Events	N/A

# Define Essential Use Case

01	Use Case	2. Withdraw
02	Actor	User
03	Purpose	사용자가 원하는 계좌에서 출금을 한다.
04	Overview	User가 출금버튼을 누른다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.2,10,11,12
07	Pre-Requisites	N/A
08	Typical courses of events	(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.
09	Alternative Courses of Events	2. (S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 출금금액이 일정 금액을 초과하면 알림을 띄운다.
10	Exceptional Courses of Events	N/A

# Define Essential Use Case

01	Use Case	3. Transfer
02	Actor	User
03	Purpose	사용자가 원하는 계좌로 자신의 계좌에서 돈을 이체한다.
04	Overview	User가 계좌이체 버튼을 누른다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.3,10,11,12
07	Pre-Requisites	N/A
08	Typical courses of events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) User가 프로그램(UI)에서 Transfer버튼을 누른다.</li> <li>2. (A) 자신의 카드정보와 이체할 계좌 번호를 입력한다.</li> <li>3. (S) database로부터 해당 계좌와 카드의 정보를 불러온다.</li> <li>4. (S) 계좌로부터 입력받은 만큼의 돈을 빼준 뒤 이체계좌에 돈을 더해준다.</li> <li>5. (S) 계좌들의 변경된 정보를 database에 다시 입력한다.</li> </ol>
09	Alternative Courses of Events	2. (S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 이체금액이 일정 금액을 초과하면 알림을 띄운다.
10	Exceptional Courses of Events	N/A

01

02

03

04

05

06

07

08

09

10

Use Case	4. Language
Actor	User
Purpose	사용자가 원하는 언어로 인터페이스를 바꾼다.
Overview	User가 Language 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 Language 버튼을 누른다. 2. (S) User가 선택한 언어에 따른 인터페이스로 변경한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

# Define Essential Use Case

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Use Case	5. Check Balance
Actor	User
Purpose	사용자가 원하는 계좌의 잔액을 보여준다.
Overview	User가 잔액확인 버튼을 누른다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R.5,10,11,12
Pre-Requisites	N/A
Typical courses of events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>(A) User가 프로그램(UI)에서 Check Balance버튼을 누른다.</li> <li>(A) User가 확인하고 싶은 계좌나 카드의 번호를 입력하고 비밀번호도 입력한다.</li> <li>(S) database로부터 해당 계좌나 카드의 정보를 불러온다.</li> <li>(S) 화면에 잔액을 보여준다.</li> </ol>
Alternative Courses of Events	2. (S) 잘못된 계좌정보 또는 비밀번호를 입력하면 알림을 띄운다.
Exceptional Courses of Events	N/A

# Define Essential Use Case

01	Use Case	6. Size Up
02	Actor	User
03	Purpose	인터페이스의 글자 폰트들을 크게 만든다.
04	Overview	User가 SizeUp버튼을 누른다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.6
07	Pre-Requisites	N/A
08	Typical courses of events	(A) : Actor, (S) : System 1. (A) User가 프로그램(UI)에서 Size Up 버튼을 누른다. 2. (S) 미리 준비된 큰 글자들의 인터페이스로 화면을 변경한다.
09	Alternative Courses of Events	N/A
10	Exceptional Courses of Events	N/A

# Define Essential Use Case

01	Use Case	7. Robbery Report
02	Actor	User
03	Purpose	사용자가 도난 신고를 한다.
04	Overview	User가 도난 신고 버튼을 누른다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.7,10,11,12
07	Pre-Requisites	N/A
08	Typical courses of events	(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.
09	Alternative Courses of Events	2. (S) 이미 있는 도난 카드의 번호를 입력했을 시에는 화면에 알림을 띄운다.
10	Exceptional Courses of Events	N/A

# Define Essential Use Case

01	Use Case	8. Add Cash
02	Actor	Admin
03	Purpose	관리자가 ATM에 현금을 추가한다.
04	Overview	관리자가 비밀번호를 입력해서 현금을 추가한다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.8,10,11,12
07	Pre-Requisites	Actor가 Admin이어야한다. (A) : Actor, (S) : System
08	Typical courses of events	1. (A) Admin이 관리자 모드로 접근한다(비밀번호 입력). 2. (A) 입금할 금액을 넣어준다. 3. (S) database로부터 ATM의 잔액정보에 입금된 금액을 더해준 뒤 다시 database에 넣는다.
09	Alternative Courses of Events	2. (S) 관리자 모드 비밀번호를 잘못 입력하면 접근이 되지 않는다.
10	Exceptional Courses of Events	N/A



## Define Essential Use Case

01	Use Case	9. Check Cash
02	Actor	Admin
	Purpose	ATM의 현금 잔액을 확인한다.
03	Overview	관리자 모드에서 ATM잔액확인을 누른다.
04	Type	Primary and Essential
	Cross Reference	Functional Requirements : R.9,10,11,12
05	Pre-Requisites	Actor가 Admin이어야한다.
06	Typical courses of events	(A) : Actor, (S) : System
07		1. (A) Admin이 관리자 모드로 접근한다(비밀번호 입력).
		2. (S) database로부터 ATM의 잔액을 확인한다. 3. (S) 잔액을 화면에 출력한다.
08	Alternative Courses of Events	2. (S) 관리자 모드 비밀번호를 잘못 입력하면 접근이 되지 않는다.
09	Exceptional Courses of Events	N/A
10		

# Define Essential Use Case

01	Use Case	10. Load
02	Actor	System
03	Purpose	DataBase로 원하는 정보를 불러온다.
04	Overview	원하는 정보의 형식에 따라서 DB에서 검색을 한 뒤 불러온다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.10
07	Pre-Requisites	N/A
08	Typical courses of events	(A) : Actor, (S) : System 1. (S) 원하는 정보의 형식을 확인한다. 2. (S) 형식에 맞는 정보를 Database로 접근해서 검색한 뒤 불러온다.
09	Alternative Courses of Events	2. (S) 원하는 형식의 정보가 Database에 존재하지 않는다면 Error 메시지를 출력한다.
10	Exceptional Courses of Events	N/A

# Define Essential Use Case

01	Use Case	11. Store
02	Actor	System
03	Purpose	데이터베이스에 변경된 정보를 입력한다.
04	Overview	변경된 정보를 DB에 넣는다.
05	Type	Primary and Essential
06	Cross Reference	Functional Requirements : R.11
07	Pre-Requisites	N/A
08	Typical courses of events	(A) : Actor, (S) : System 1. (S) 변경된 정보의 형식을 확인한다. 2. (S) 형식에 맞는 정보를 Database로 접근해서 검색한 뒤 변경된 정보로 대체한다.
09	Alternative Courses of Events	2. (S) DataBase에 존재하지 않는 형식이라면 Error 메시지를 출력한다.
10	Exceptional Courses of Events	N/A

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Use Case	12. Check
Actor	System
Purpose	User의 입력된 정보를 바탕으로 정보가 맞는지 확인하고 필요한 계산을 수행한다.
Overview	변경될 정보가 맞는지 확인하고 금액의 계산을 해준다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R.12
Pre-Requisites	Load가 미리 실행이 되어야한다.
Typical courses of events	(A) : Actor, (S) : System 1. (S) 변경된 정보의 형식을 확인한다.
Alternative Courses of Events	2. (S) 원하는 형식의 정보가 Database에 존재하지 않거나 계산이 범위를 벗어나게 되는 경우 Error 메시지를 출력한다.
Exceptional Courses of Events	N/A

# Refine Use Case Diagrams

Stage 2032

# Refine Use Case Diagrams

01

02

03

04

05

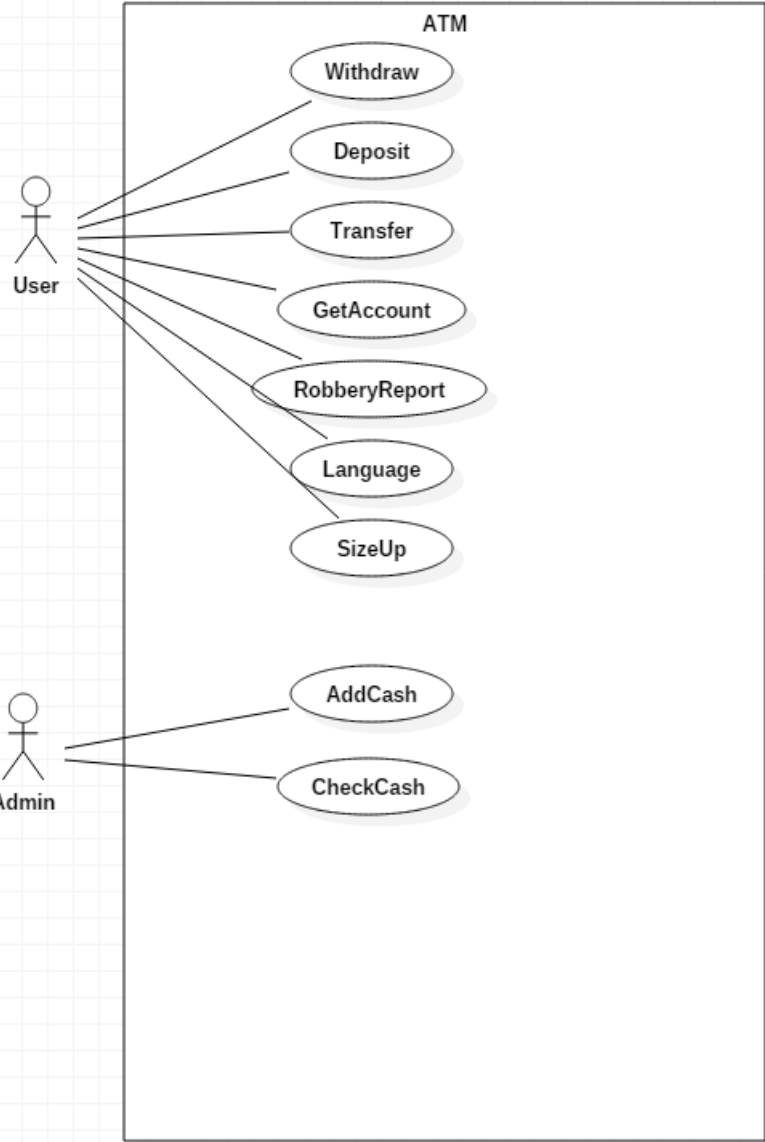
06

07

08

09

10



# Define Domain Model

Stage 2033

# Define Domain Model

01

02

03

04

05

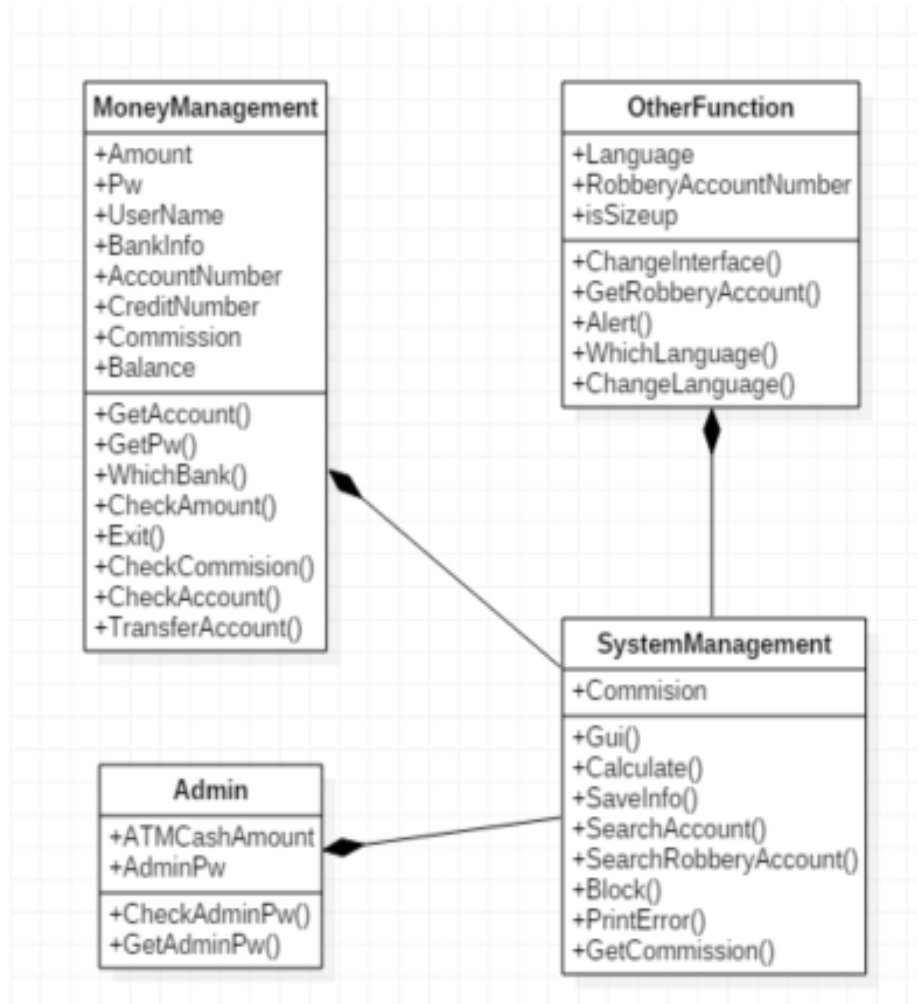
06

07

08

09

10





# Refine Glossary

Stage 2034

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Term	Category	Description
MoneyManagement	Class	전반적인 돈 관리(출금, 입금, 이체, 잔액확인) 실행을 제어하는 클래스
OtherFunction	Class	다른 기능들(도난 신고, 언어 선택, 저시력자를 위한 인터페이스 변경)의 실행을 제어하는 클래스
Admin	Class	관리자 모드의 기능들(Add Cash, Check Cash)을 제어하는 클래스
SystemManagement	Class	System의 전반적인 기능들, 특히 Database의 정보들을 관리하고 변경하는 (load, store, check) 클래스
MoneyManagement.GetAccount	Operation	User로부터 계좌정보를 받아들이는 메소드
MoneyManagement.GetPW	Operation	User로부터 비밀번호를 받는 메소드
MoneyManagement.WhichBank	Operation	어떤 종류의 은행을 선택할지 받아주는 메소드
MoneyManagement.CheckAmount	Operation	User가 입력한 금액을 한도 금액과 비교해주는 메소드
MoneyManagement.Exit	Operation	잘못된 접근이나 Warning이 떴을 때 프로그램을 종료해주는 메소드
MoneyManagement.CheckCommission	Operation	은행과 은행사이의 수수료를 확인해주는 메소드
MoneyManagement.CheckAccount	Operation	계좌정보를 받아들였을 때 맞는 계좌인지 확인해주는 메소드
MoneyManagement.TransferAccount	Operation	이체계좌가 올바른지 확인해주는 메소드
OtherFunction.ChangeInterFace	Operation	User가 저시력자용 UI를 원할 때 인터페이스를 변경해주는 메소드
OtherFunction.GetRobberyAccount	Operation	User로부터 도난신고할 계좌의 정보를 받는 메소드

# Refine Glossary

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Term	Category	Description
OtherFunction.Alert	Operation	User가 도난 카드를 사용했을 시 경고메시지를 출력해주는 메소드
OtherFunction.WhichLanguage	Operation	어떤 언어로 선택할지 결정해주는 메소드
OtherFunction.ChangeLanguage	Operation	User로부터 언어를 선택받은 뒤 그 언어로 이루어진 인터페이스를 바꿔주는 메소드
Admin.CheckAdminPW	Operation	관리자로부터 관리자용 비밀번호와 일치하는지 비교해주는 메소드
Admin.GetAdminPW	Operation	관리자로부터 비밀번호를 입력받는 메소드
SystemManagement.GUI	Operation	시스템의 GUI정보 초기화 메소드
SystemManagement.Calculate	Operation	금액의 변경된 부분을 계산해서 변경하는 메소드
SystemManagement.SaveInfo	Operation	계산되서 바뀐 정보를 DB에 다시 저장해주는 메소드
SystemManagement.SearchAccount	Operation	DB로부터 원하는 계좌의 정보를 검색하는 메소드
SystemManagement.SearchRobberyAccount	Operation	DB로부터 도난 신고된 계좌의 번호를 검색하는 메소드
SystemManagement.Block	Operation	잘못된 접근일 때 DB와 시스템의 연결을 끊는 메소드
SystemManagement.PrintError	Operation	잘못된 접근(ex. 계좌정보 존재X)의 경우 에러메시지 출력하는 메소드
SystemManagement.GetCommission	Operation	DB로부터 특정 은행과 은행사이의 수수료를 불러오는 메소드
MoneyManagement.Amount	Attributes	User가 입력한 금액을 저장하는 변수

# Refine Glossary

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Term	Category	Description
MoneyManagement.PW	Attributes	User가 입력한 비밀번호를 저장하는 변수
MoneyManagement.UserName	Attributes	User의 이름을 저장하는 변수
MoneyManagement.BankInfo	Attributes	어떤 은행인지 정보를 나타내는 변수
MoneyManagement.AccountNum	Attributes	계좌번호를 저장하는 변수
MoneyManagement.CardNum	Attributes	카드 번호를 저장하는 변수
MoneyManagement.Commission	Attributes	특정 수수료들을 저장하는 변수
MoneyManagement.Balance	Attributes	특정 계좌의 잔액을 저장하는 변수
OtherFunction.Language	Attributes	사용할 언어의 정보를 저장하는 변수
OtherFunction.RobberyAccountNumber	Attributes	도난신고된 계좌번호의 정보를 저장하는 변수
OtherFunction.isSizeUp	Attributes	저 시력자를 위한 인터페이스의 사이즈 업에 대한 정보를 저장하는 변수
Admin.ATMCashAmount	Attributes	ATM의 현금의 보유량을 저장하는 변수
Admin.AdminPW	Attributes	관리자의 비밀번호를 저장하는 변수
SystemManagement.Commission	Attributes	DB에 존재하는 수수료를 저장하는 변수
SystemManagement.CalResult	Attributes	계산된 결과값을 저장하는 변수

# Define System Sequence Diagrams

Stage 2035

01

02

03

04

05

06

07

08

09

10

### Use Case : Deposit

(A) : Actor, (S) : System

1. (A) User가 프로그램(UI)에서 Deposit버튼을 누른다.
2. (A) 입금할 계좌와 금액을 프로그램에 입력한다.
3. (S) database로부터 해당 계좌의 정보를 불러온다.
4. (S) 계좌에 입금하고(정보 변경 / Check) database에 정보를 넣는다.

```
sequenceDiagram
    actor User
    participant System
    User->>System: 1: RequestDeposit
    activate System
```

The diagram shows a User actor sending a message labeled '1: RequestDeposit' to a System participant. The System participant has an activation bar indicating it is active during the message exchange.

### 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.

```
sequenceDiagram
    actor User
    participant System
    User->>System: 1: RequestWithdraw
    activate System
```

The diagram shows a User actor sending a message labeled '1: RequestWithdraw' to a System participant. The System participant has an activation bar indicating it is active during the message exchange.

01

02

03

04

05

06

07

08

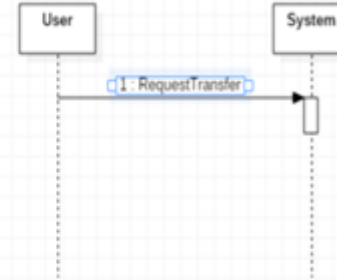
09

10

## Use Case : Transfer

(A) : Actor, (S) : System

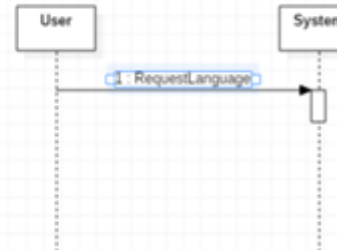
1. (A) User가 프로그램(UI)에서 Transfer버튼을 누른다.
2. (A) 자신의 카드정보와 이체할 계좌 번호를 입력한다.
3. (S) database로부터 해당 계좌와 카드의 정보를 불러온다.
4. (S) 계좌로부터 입력받은 만큼의 돈을 빼준 뒤 이체계좌에 돈을 더해준다.
5. (S) 계좌들의 변경된 정보를 database에 다시 입력한다.



## Use Case : Language

(A) : Actor, (S) : System

1. (A) User가 Language 버튼을 누른다.
2. (S) User가 선택한 언어에 따른 인터페이스로 변경한다



01

02

03

04

05

06

07

08

09

10

## Use Case : Check Balance

(A) : Actor, (S) : System

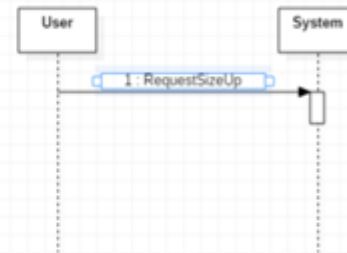
1. (A) User가 프로그램(UI)에서 Check Balance버튼을 누른다.
2. (A) User가 확인하고 싶은 계좌나 카드의 번호를 입력하고 비밀번호도 입력한다.
3. (S) database로부터 해당 계좌나 카드의 정보를 불러온다.
4. (S) 화면에 잔액을 보여준다.



## Use Case : Size Up

(A) : Actor, (S) : System

1. (A) User가 프로그램(UI)에서 Size Up 버튼을 누른다.
2. (S) 미리 준비된 큰 글자들의 인터페이스 화면을 변경한다.





01

02

03

04

05

06

07

08

09

10

## 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.



01

02

03

04

05

06

07

08

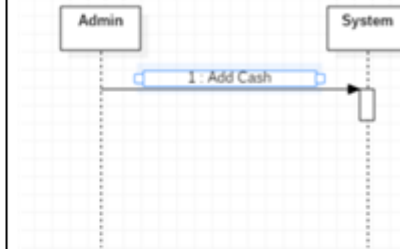
09

10

## Use Case : Add Cash

(A) : Actor. (S) : System

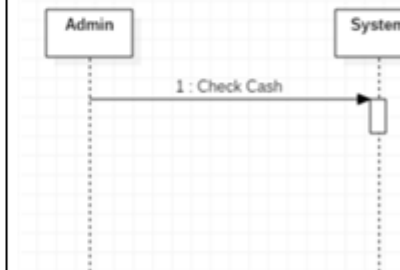
1. (A) Admin이 관리자 모드로 접근한다(비밀번호 입력).
2. (A) 입금할 금액을 넣어준다.
3. (S) database로부터 ATM의 잔액정보에 입금된 금액을 더해준 뒤 다시 database에 넣는다.



## 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.



# Define Operation Contracts

Stage 2036

# Define Operation Contracts

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Name	Deposit()
Responsibility	해당 계좌에서 입금을 한다.
Type	System
Cross Reference	R.1,10,11,12
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.2,10,11,12
Exception	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 작은 금액을 받은 상태
Post-Condition	N/A

Name	Transfer()
Responsibility	해당 계좌에서 이체 계좌로 금액을 이체한다.
Type	Actor
Cross Reference	R.3,10,11,12
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

# Define Operation Contracts

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Name	Check Balance()
Responsibility	해당 계좌에서 잔액을 조회한다.
Type	Actor
Cross Reference	R.5,10,11,12
Exception	계좌 또는 이체 계좌가 없을 시 에러 메시지 출력
Output	N/A
Pre-Condition	올바른 계좌정보를 받은 상태
Post-Condition	N/A

Name	Language()
Responsibility	사용자 선택한 언어로 인터페이스를 변경
Type	Actor
Cross Reference	R.4
Exception	잘못된 언어를 선택할 시 오류 메시지 출력
Output	해당 언어에 해당하는 UI
Pre-Condition	한글/영어 중 한 개를 선택한다.
Post-Condition	N/A

Name	SizeUp()
Responsibility	인터페이스의 글씨 크기를 크게 바꿔준다.
Type	Actor
Cross Reference	R.6
Exception	N/A
Output	글씨의 사이즈가 커진 UI
Pre-Condition	N/A
Post-Condition	N/A

# Define Operation Contracts

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Name	RobberyReport()
Responsibility	도난 카드나 계좌의 번호를 저장하고 신고한다.
Type	Actor
Cross Reference	R.7,10,11,12
Exception	도난신고된 카드가 미리 있을 경우는 등록을 하지 않고 이미 등록되어있다는 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.8,10,11,12
Exception	ATM의 관리자 모드 로그인에 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.9,10,11,12
Exception	ATM의 관리자 모드 로그인에 실패할 경우 예외 발생
Output	N/A
Pre-Condition	올바른 비밀번호를 받은 상태
Post-Condition	N/A

# Define Operation Contracts

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

Name	Load()
Responsibility	Database로부터 원하는 정보를 검색해서 불러온다.
Type	System
Cross Reference	R.10
Exception	원하는 정보가 DB에 없을 시 예외 발생
Output	DB의 해당 정보
Pre-Condition	DB에 존재하는 정보여야 한다.
Post-Condition	N/A

Name	Store()
Responsibility	변경된 정보를 Database로 저장한다.
Type	System
Cross Reference	R.11
Exception	N/A
Output	N/A
Pre-Condition	DB에 미리 존재하는 정보형식이어야 한다.
Post-Condition	N/A

Name	Check()
Responsibility	DB로부터 불러온 정보를 요청된 계산을 해주는 기능
Type	System
Cross Reference	R.12
Exception	계산의 범위가 한도를 벗어나면 예외 발생
Output	계산된 결과 값
Pre-Condition	한도를 벗어나지 않은 범위 내의 계산만 가능
Post-Condition	N/A

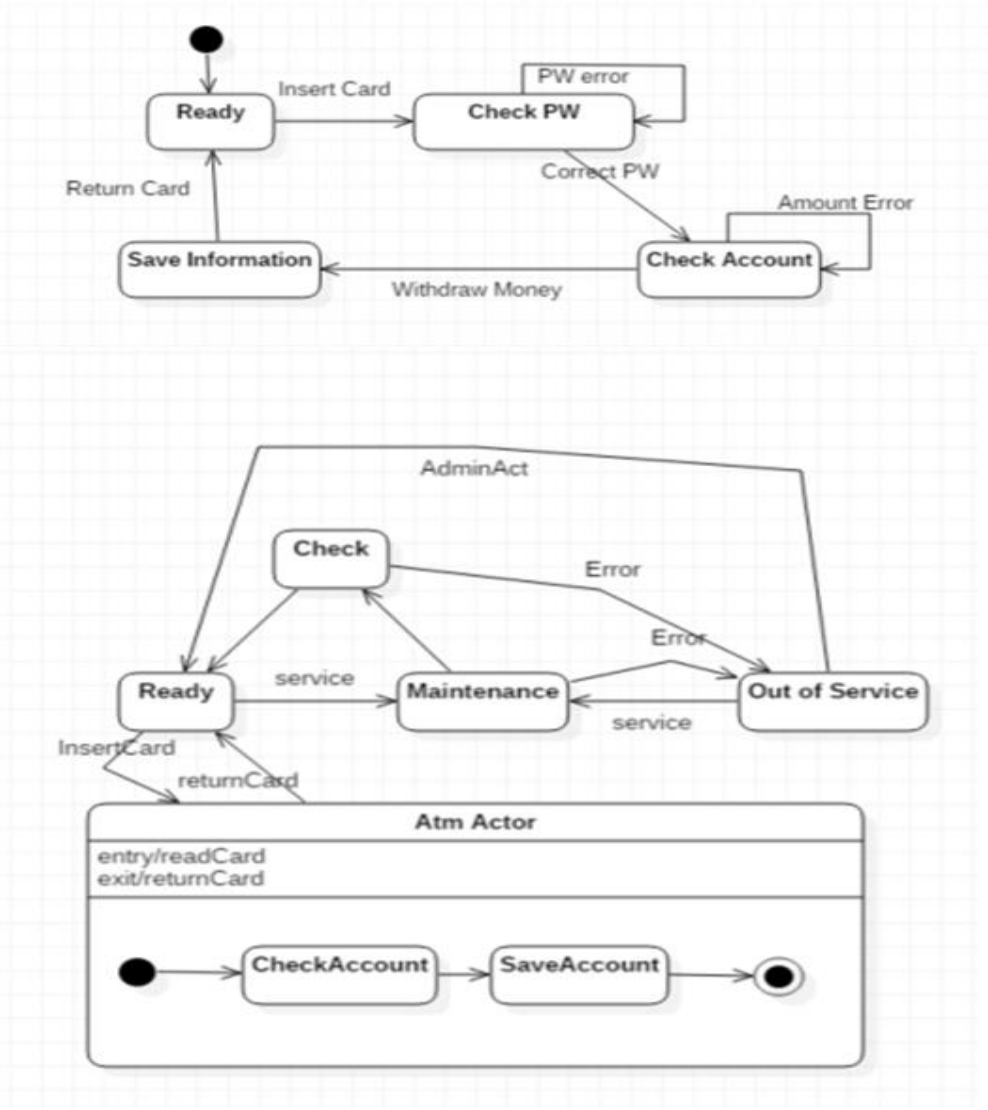
# Define State Diagrams

Stage 2037



# Define State Diagrams

01  
02  
03  
04  
05  
06  
07  
08  
09  
10



# Refine System Test Case

Stage 2038

# Refine System Test Case

01  
02  
03  
04  
05  
06  
07  
08  
09  
10

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	Size Up Test	글씨의 크기가 증가된 인터페이스로 잘 바뀌는 지 확인한다.	Size Up
6-1	Language Test	선택한 언어로 이루어진 인터페이스로 올바르게 바뀌는지 확인한다.	Language
7-1	Robbery Report Test	도난 신고 계좌 정보를 또 등록하려고 할 때 중복 메시지가 잘 나오는 지 확인한다.	Robbery Report
7-2	Robbery Report Test	도난 신고된 계좌의 번호가 DB에 제대로 들어갔는지 확인한다.	Robbery Report
8-1	Add Cash Test	ATM에 현금이 제대로 들어갔는지 확인한다.	Add Cash
8-2	Check Cash Test	ATM의 잔액정보가 올바르게 출력되는지 확인한다.	Check Cash

# Analyze Traceability Analysis

Stage 2039

## Analyze Traceability Analysis

01

02

03

04

05

06

07

08

09

10

System Function		Use Case		Operation
R1. Deposit	→	1. Deposit	→	1. Deposit()
R2. Withdraw	→	2. Withdraw	→	2. Withdraw()
R3. Transfer	→	3. Transfer	→	3. Transfer()
R4. Language	→	4. Language	→	4. Language()
R5. Check Balance	→	5. Check Balance	→	5. Check Balance()
R6. Size Up	→	6. Size Up	→	6. Size Up()
R7. Robbery Report	→	7. Robbery Report	→	7. Robbery Report()
R8. Add Cash	→	8. Add Cash	→	8. Add Cash()
R9. Check Cash	→	9. Check Cash	→	9. Check Cash()

Software Modeling

A NEW C.S.E. ATM

Thank you

201311299 | 이원오  
201311301 | 이재규  
201311309 | 전홍준