

Software Modeling

# A NEW C.S.E. ATM

Stage 2040

201311299 | 이원오  
201311301 | 이재규  
201311309 | 전홍준

A NEW C.S.E. ATM

# INDEX

- Design Real Use Cases
- Define Reports, UI, and Storyboards
- Refine System Architecture
- Define Interaction Diagrams
- Define Design Class Diagrams
- Define Traceability Analysis

# Design Real Use Cases

Stage 2041

01

02

03

04

05

06

07

08

09

10

Use Case	1. Deposit
Actor	User
Purpose	사용자가 원하는 계좌에 입금을 한다.
Overview	User가 입금버튼을 누른다.
Type	Primary and Real
Cross Reference	Functional Requirements : R.1,10,11,12
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 Deposit버튼을 누른다.</li> <li>(S) User에게 Deposit UI를 띄워준다.</li> <li>(A) 입금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다.</li> <li>(S) 시스템에서 해당 계좌의 비밀번호를 확인한다.</li> <li>(S) 금액 입력창으로 UI 변경</li> <li>(A) User가 입금할 금액을 입력한다.</li> <li>(S) User가 입력한 금액을 해당 계좌나 카드에 추가한다.</li> <li>(S) 변경 내용을 저장한다.</li> <li>(S) 변경 내용을 User에게 보여준다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보를 입력하거나 입금금액이 일정 금액을 초과하면 알림을 띄운다.

01

02

03

04

05

06

07

08

09

10

Use Case	1. Transfer
Actor	User
Purpose	사용자가 원하는 계좌로 자신의 계좌에서 돈을 이체한다.
Overview	User가 계좌이체 버튼을 누른다.
Type	Primary and Real
Cross Reference	Functional Requirements : R.3,10,11,12
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 Transfer버튼을 누른다.</li> <li>(S) User에게 Transfer UI를 띄워준다.</li> <li>(A) 출금할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다.</li> <li>(S) 시스템에서 해당 계좌의 비밀번호를 확인한다.</li> <li>(S) 이체 계좌정보 및 금액 입력창으로 UI 변경</li> <li>(A) User가 이체할 계좌정보와 이체할 금액을 입력한다.</li> <li>(S) User가 입력한 금액을 이체 계좌나 카드에 더해주고 해당 계좌에서는 그 금액만큼 빼준다.</li> <li>(S) 변경 내용을 저장한다.</li> <li>(S) 변경 내용을 User에게 보여준다.</li> </ol>
Alternative Courses of Events	
Exceptional Courses of Events	<ol style="list-style-type: none"> <li>(S) 잘못된 계좌정보 또는 비밀번호를 입력하거나 이체금액이 한계 금액이나 계좌 잔액을 초과하면 알림을 띄운다.</li> </ol>

01

02

03

04

05

06

07

08

09

10

Use Case	1. Language
Actor	User
Purpose	사용자가 원하는 언어로 인터페이스를 바꾼다.
Overview	User가 Language 버튼을 누른다.
Type	Primary and Real
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

01

02

03

04

05

06

07

08

09

10

Use Case	1. Check Balance
Actor	User
Purpose	사용자가 원하는 계좌의 잔액을 보여준다.
Overview	User가 잔액확인 버튼을 누른다.
Type	Primary and Real
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>1. (A) User가 프로그램(UI)에서 Check Balance버튼을 누른다.</li> <li>2. (S) User에게 Check Balance UI를 띄워준다.</li> <li>3. (A) 확인할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다.</li> <li>4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다.</li> <li>5. (S) 해당계좌나 카드의 잔액을 화면에 띄운다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	1. (S) 잘못된 계좌정보 또는 비밀번호를 입력하면 알림을 띄운다.

01

02

03

04

05

06

07

08

09

10

Use Case	1. Robbery Report
Actor	User
Purpose	사용자가 도난 신고를 한다.
Overview	User가 도난 신고 버튼을 누른다.
Type	Primary and Real
Cross Reference	Functional Requirements : R.7,10,11,12
Pre-Requisites	N/A
Typical courses of events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) User가 프로그램(UI)에서 도난 신고버튼을 누른다.</li> <li>2. (S) User에게 도난신고 UI를 띄워준다.</li> <li>3. (A) 도난 신고할 계좌(or 카드번호)와 비밀번호를 프로그램에 입력한다.</li> <li>4. (S) 시스템에서 해당 계좌의 비밀번호를 확인한다.</li> <li>5. (S) 계좌정보(카드)를 도난 상태로 변경.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	<ol style="list-style-type: none"> <li>1. (S) 이미 있는 도난 카드의 번호를 입력했을 시에는 화면에 알림을 띄운다.</li> </ol>



01

02

03

04

05

06

07

08

09

10

Use Case	1. Add Cash
Actor	Admin
Purpose	관리자가 ATM에 현금을 추가한다.
Overview	관리자가 비밀번호를 입력해서 현금을 추가한다.
Type	Primary and Real
Cross Reference	Functional Requirements : R.8,10,11,12
Pre-Requisites	Actor가 Admin이어야한다.
Typical courses of events	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>(A) 관리자가 Admin버튼을 누른다.</li> <li>(S) 관리자 확인 UI로 바꿔준다.</li> <li>(A) 관리자 ID/PW를 입력한다.</li> <li>(S) 관리자가 입력한 ID/PW가 올바르게 입력된지 확인한다.</li> <li>(S) 관리자 UI로 바꿔준다.</li> <li>(A) Add Cash 버튼을 누른다.</li> <li>(S) Add Cash UI로 바꾼다.</li> <li>(A) 관리자가 추가할 현금 액수를 입력한다.</li> <li>(S) 입력한 액수만큼 잔액에 더해준다.</li> <li>(S) 변경내용을 저장한다.</li> </ol>
Alternative Courses of Events	
Exceptional Courses of Events	1. (S) 관리자 모드 정보를 잘못 입력하면 접근이 되지 않는다.

# Define Reports, UI, and Storyboards

Stage 2042

01

02

03

04

05

06

07

08

09

10



01

02

03

04

05

06

07

08

09

10

The image shows a red rectangular form titled "입금" (Deposit) in white text at the top center. Below the title, there are three rows of input fields. Each row consists of a label on the left and a white rectangular input box on the right. The labels are "은행선택" (Bank Selection), "계좌번호" (Account Number), and "비밀번호" (Password), all in white text. Each label and input box is framed by two thin white horizontal lines above and below it. The input boxes are empty.

01

02

03

04

05

06

07

08

09

10



01

02

03

04

05

06

07

08

09

10

The image shows a red rectangular form with the title "입금" (Deposit) centered at the top. Below the title, there are two input fields. The first field is labeled "입금금액" (Deposit Amount) and the second field is labeled "잔액" (Balance). Each label is centered above its respective input field. The input fields are white rectangles with blue borders. The entire form is set against a solid red background.

01

02

03

04

05

06

07

08

09

10

입금

입금이 완료되었습니다.

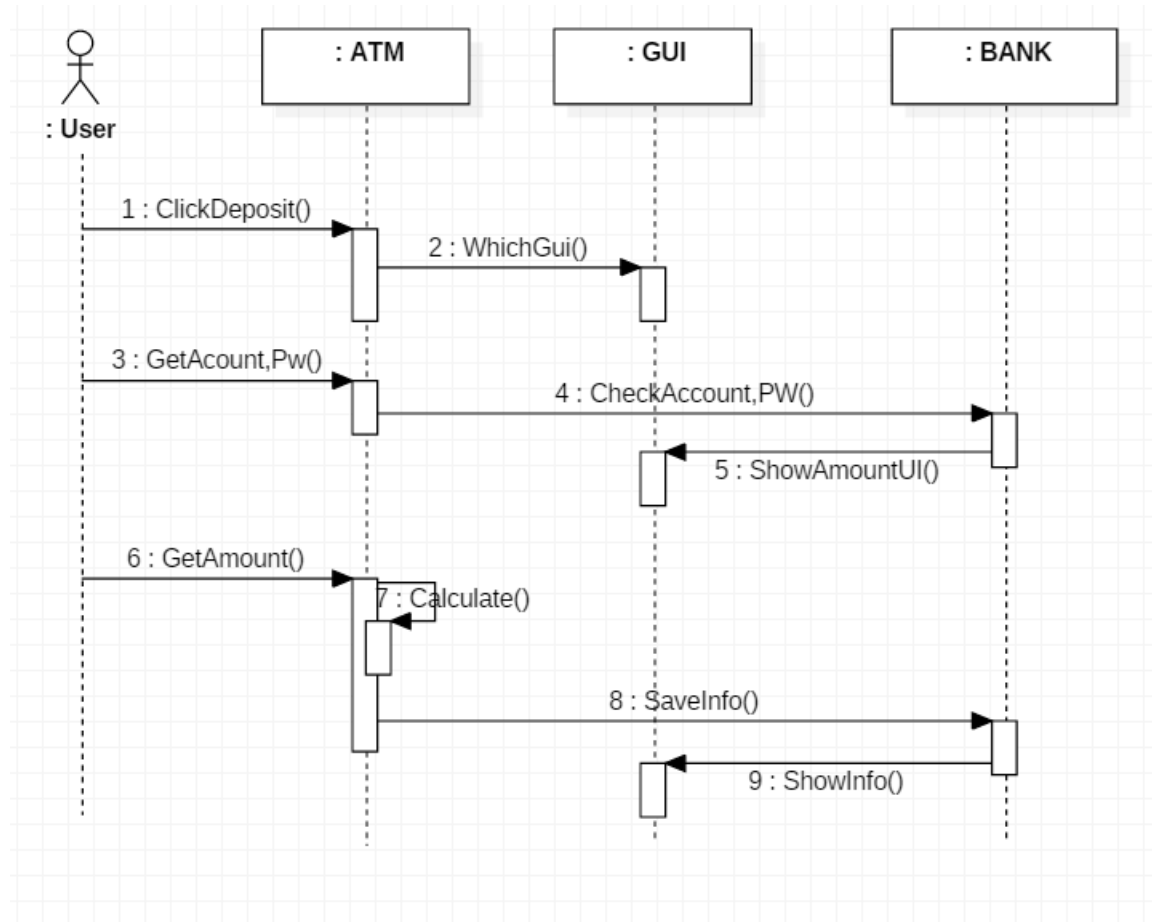
# Define Interaction Diagrams

Stage 2044



# Deposit

01  
02  
03  
04  
05  
06  
07  
08  
09  
10



# Language

01

02

03

04

05

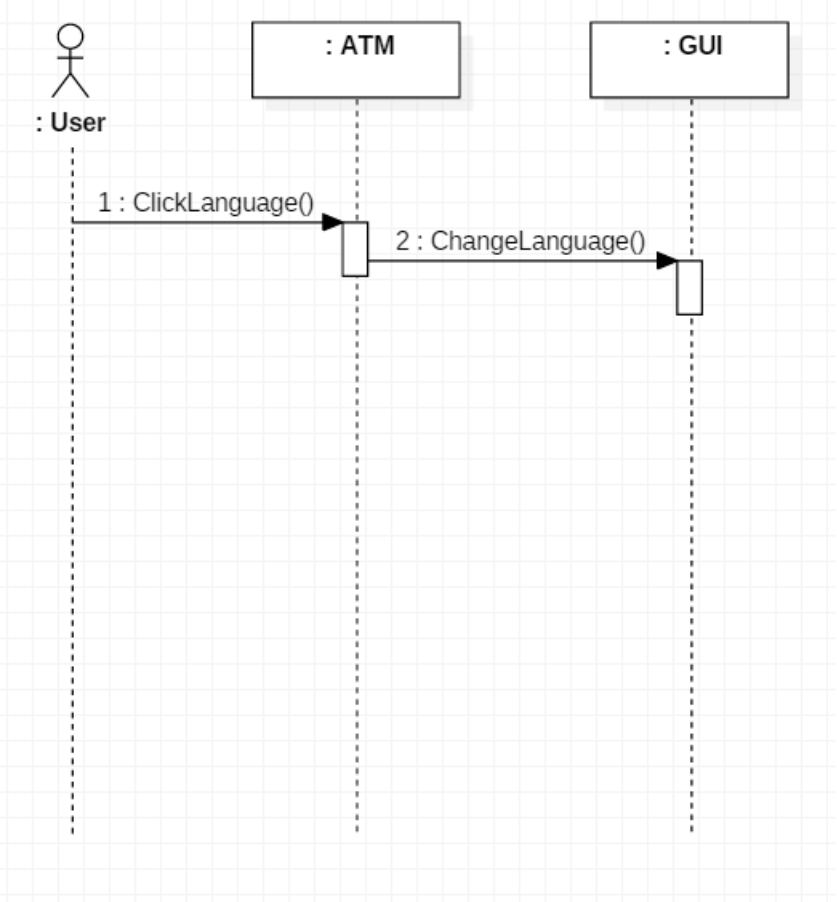
06

07

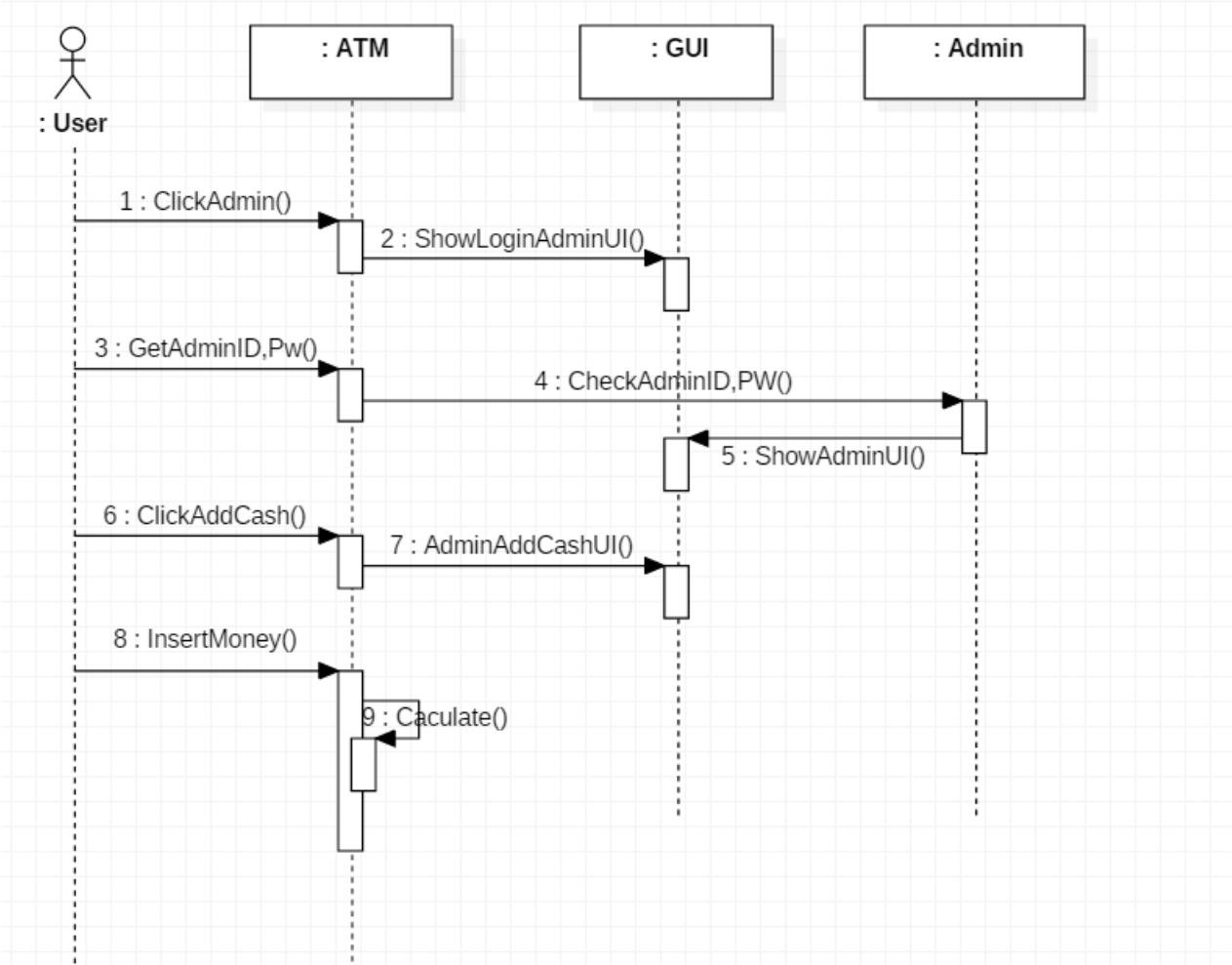
08

09

10



# AddCash



01

02

03

04

05

06

07

08

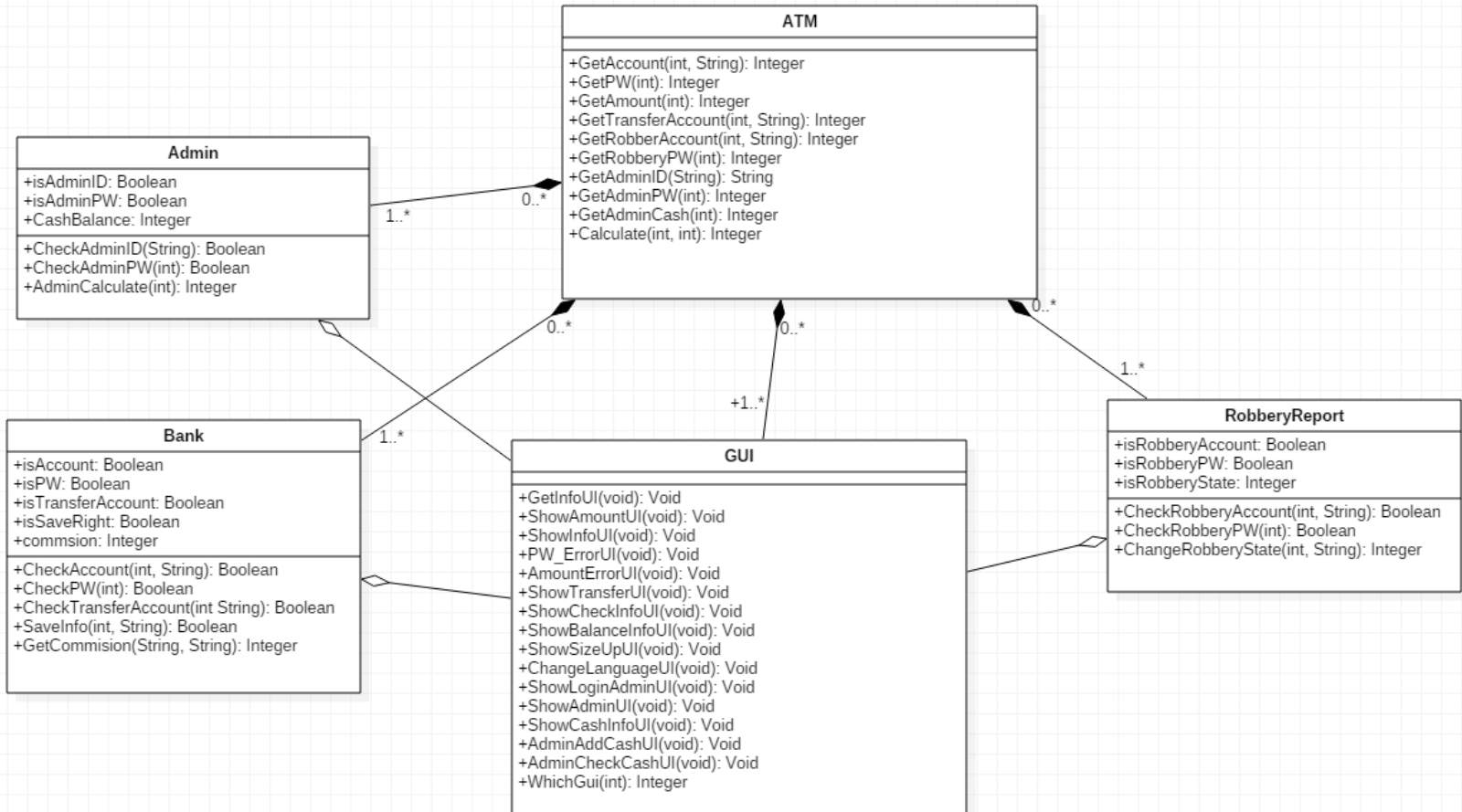
09

10

# Define Design Class Diagrams

Stage 2045

# Define Design Class Diagrams



01

02

03

04

05

06

07

08

09

10

# Design Traceability Analysis

Stage 2046

# Design Traceability Analysis

Use Cases	System Operation	Method	Class
1. Deposit	Deposit()	WhichGui(int): Integer	ATM
2. Withdraw	Withdraw()	GetAccount(int, String): Integer	
3. Transfer	Transfer()	GetPW(int): Integer	
4. Check Balance	CheckBalance()	GetAmount(int): Integer	
5. RobberyReport	RobberyReport()	GetTransferAccount(int, String): Integer	
6. SizeUp	SizeUp()	GetRobberyPW(int): Integer	
7. Language	Language()	GetAdminID(String): String	
8. Add Cash	AddCash()	GetAdminPW(int): Integer	
9. Check Cash	CheckCash()	GetAdminCash(int): Integer	
		Calculate(int, int): Integer	GUI
		GetInfoUI(void): Void	
		ShowAmountUI(void): Void	
		ShowInfoUI(void): Void	
		PW_ErrorUI(void): Void	
		AmountErrorUI(void): Void	
		ShowTransferUI(void): Void	
		ShowCheckInfoUI(void): Void	
		ShowBalanceInfoUI(void): Void	
		ShowSizeUpUI(void): Void	
		ShowEnglishUI(void): Void	
		ShowLoginAdminUI(void): Void	
		ShowAdminUI(void): Void	Admin
		ShowCashInfoUI(void): Void	
		AdminAddCashUI(void): Void	
		AdminCheckCashUI(void): Void	Bank
		CheckAdminID(String): Boolean	
		CheckAdminPW(int): Boolean	
		AdminCalculate(int): Integer	
		CheckAccount(int, String): Boolean	RobberyReport
		CheckPW(int): Boolean	
		CheckTransferAccount(int, String): Boolean	
		SaveInfo(int, String): Boolean	
		GetCommision(String, String): Integer	
		CheckRobberyAccount(int, String): Boolean	
		CheckRobberyPW(int): Boolean	
		ChangeRobberyState(int, String): Integer	

01

02

03

04

05

06

07

08

09

10

Software Modeling

A NEW C.S.E. ATM

Thank you

201311299 | 이원오  
201311301 | 이재규  
201311309 | 전홍준