

**JackPot ATM**

**OOPT Stage 2040 ver2**

**<Analyze>**

**Team 5**

Date

2018-05-09

201312259 백만일

201112052 방민석

201211383 조영래

---

# INDEX

Activity 2041. Design Real Use Cases

Activity 2042. Define Reports UI, and Storyboards

Activity 2043. Refine System Architecture

Activity 2044. Define Interaction Diagrams

Activity 2045. Define Design Class Diagrams

Activity 2046. Design Traceability Analysis

## Activity 2041. Design Real Use Cases

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	<p>(A):Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) User가 withdraw 버튼을 누른다.</li> <li>2.(S) 계좌 ID 입력 화면을 출력한다.</li> <li>3.(A) User는 계좌(inputID)를 입력한다.</li> <li>4.(S) Account.ID와 inputID를 비교 확인한다.</li> <li>5.(S) 출금 화면을 출력한다.</li> <li>6.(A) 출금 금액(inputMoney)을 입력한다.</li> <li>7.(S) 계좌 password 입력 화면을 출력한다.</li> <li>8.(A) 비밀번호(inputPW)를 입력한다.</li> <li>9.(S) Account.password와 inputPW가 일치하는지 확인한다.</li> <li>10.(S) Account.Balance와 inputMoney을 비교한다.</li> <li>11.(S) Account.Balance를 inputMoney만큼 감소시킨다.</li> <li>12.(S) ATM 잔액(TotalMoney)을 감소시킨다.</li> <li>13.(S) ATM 각 지폐별 잔고를 감소시킨다.</li> <li>14.(S) 거래후 사용자의 계좌 잔고(Account.Balance)를 출력한다.</li> </ol>
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: 계좌 잔고가 출금액보다 적을 경우 에러 메시지를 출력한다
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 deposit 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) 금액 입력 화면을 출력한다. 6.(A) 입금 금액(inputMoney)을 입력한다. 7.(S) Account.Balance를 inputMoney만큼 증가시킨다. 8.(S) ATM 잔액(TotalMoney)을 증가시킨다. 9.(S) ATM 각 지폐별 잔고를 증가시킨다. 10. (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

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.

<p>Typical Courses of Events</p>	<p>(A):Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) User가 remittance 버튼을 누른다.</li> <li>2.(S) 계좌 ID 입력 화면을 출력한다.</li> <li>3.(A) User는 계좌(inputID)를 입력한다.</li> <li>4.(S) Account.ID와 inputID를 비교 확인한다.</li> <li>5.(S) 송금받을 계좌 입력 화면을 출력한다.</li> <li>6.(A) User는 송금받을 계좌(inputID)를 입력한다.</li> <li>7.(S) Account.ID와 inputID를 비교한다.</li> <li>8.(A) User는 송금 금액(inputMoney)을 입력한다.</li> <li>9.(S) User의 잔고(Account.Balance)와 송금액(inputMoney)을 비교 확인한다.</li> <li>10.(A) User는 비밀번호(inputPW)를 입력한다.</li> <li>11.(S) Account.password와 inputPW가 일치하는지 확인한다.</li> <li>12.(S) 받는사람의 통장잔고(Account.Balance)를 증가시킨다.</li> <li>13.(S) User의 통장잔고(Account.Balance)를 감소시킨다.</li> <li>14.(S) 거래 후 계좌 잔액(Account.Balance)을 표시한다.</li> </ol>
<p>Alternative Courses of Events</p>	<p>N/A</p>
<p>Exceptional Courses of Events</p>	<p>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</p>

<p>Use Case</p>	<p>view_account_detail</p>
<p>Actor</p>	<p>User</p>

Purpose	사용자가 입력한 계좌의 잔액을 보여준다.
Overview	사용자가 계좌조회 버튼을 누르고 자신의 계좌를 입력한다. 계좌가 확인되면 입력된 계좌의 잔고를 표시한다.
Type	Primary and Essential
Cross Reference	System Function R.4
Pre-Requisites	ATM에 입력한 계좌가 은행에 계좌 정보가 있어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 view_account_detail 버튼을 누른다 2.(S) 계좌 입력 화면을 출력한다 3.(A) User는 계좌(inputID)를 입력한다 4.(S) Account.ID와 inputID를 비교 확인한다 5.(S) 계좌 password 입력 화면을 출력한다 6.(A) 비밀번호(inputPW)를 입력한다 4.(S) Account.password와 inputPW가 일치하는지 확인한다 5.(S) 확인이 되면 Account.Balance를 화면에 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

Use Case	Random_Jackpot
Actor	User
Purpose	출금 기능을 이용하는 임의의 사용자에게 추가로 5만원을 지급한다.

Overview	00시가 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	N/A
Typical Courses of Events	(A):Actor, (S):System 1.(S) ATM의 Timer가 00시가 되면 당첨되는 순서를 랜덤하게 선택한다. 2.(S) User가 돈을 출금했을때 해당 User가 해당 순서의 사용자인지 확인한다. 3.(S) 해당 User에게 돈을 5만원 더 추가로 출금해준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	manage_ATM
Actor	Manager
Purpose	관리자가 ATM의 잔고를 추가하거나 뺄 수 있다
Overview	ATM의 TotalMoney가 일정 수준 이하로 낮아지면 관리자가 TotalMoney를 추가하고 잔고가 일정 수준 이상으로 많아지면 TotalMoney를 뺀다
Type	Primary and Essential
Cross Reference	System Function R.5
Pre-Requisites	N/A

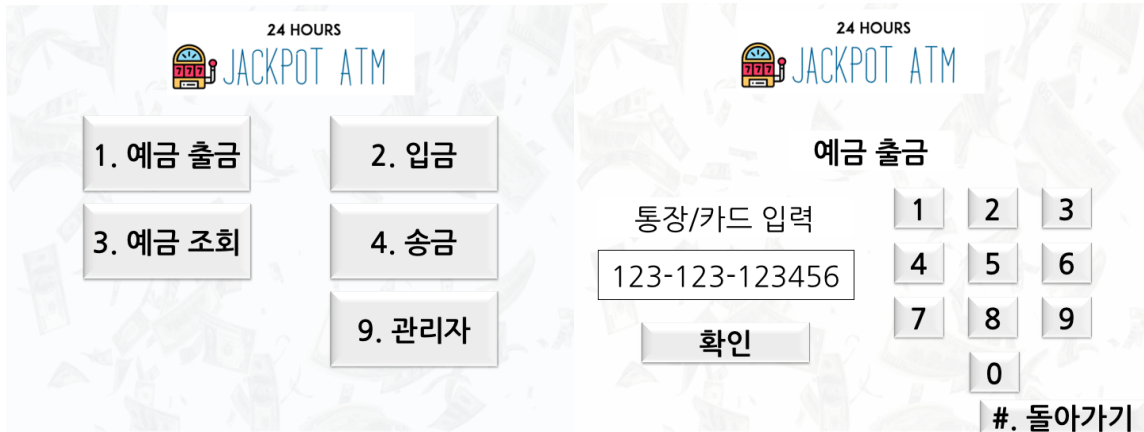


<p>Typical Courses of Events</p>	<p>(A):Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) Manager가 manage 버튼을 누른다.</li> <li>2.(S) 관리자 ID 입력화면을 출력한다.</li> <li>3.(A) Manager가 관리자 ID(inputID)를 입력한다</li> <li>4.(S) ManagerID와 inputID를 비교 확인한다</li> <li>5.(A) Manager가 관리자 Password(inputPW)를 입력한다</li> <li>6.(S) ManagerPW와 inputPW를 비교 확인한다</li> <li>5.(S) 확인이 끝나면 Manager가 화폐별 잔고에 접근할수 있도록 한다.</li> <li>6.(A) Manager가 화폐별 잔고를 추가하거나 뺀다</li> <li>7.(S) GUI_Interface.TotalMoney를 갱신한다</li> <li>8.(S) Count50000, Count10000, Count5000, Count1000 각 화폐별 금액을 갱신한다.</li> </ol>
<p>Alternative Courses of Events</p>	<p>N/A</p>
<p>Exceptional Courses of Events</p>	<p>E1.MID와 MPW가 잘못된 경우 에러 메시지를 출력한다</p>

## Activity 2042. Define Reports UI, and Storyboards

1. 초기 메뉴 화면

2. 예금 출금시 통장/카드번호 입력화면



3. 예금 출금시 비밀번호 입력화면 4. 예금 출금시 출금 금액 입력화면



5. 예금 출금시 거래내역 출력화면



6. 입금시 출력 화면



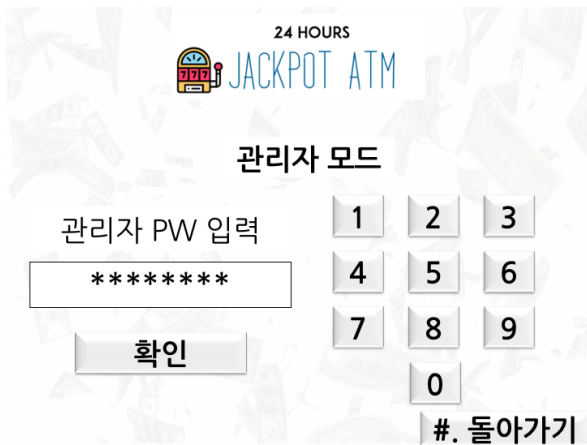
7. 예금조회시 계좌 잔고 출력화면



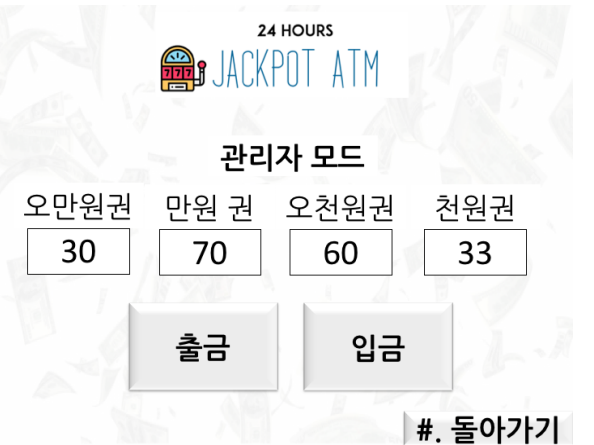
8. 관리자 모드 ID 입력화면



9. 관리자 모드 PW 입력화면



10. 관리자모드 진입 화면



11. 관리자 모드 출금시 지폐권 입력화면  
입력화면

12. 관리자 모드 입금시 지폐권  
입력화면

24 HOURS JACKPOT ATM

**관리자 모드(출금)**  
출금액 수량 입력

1,000	<input type="text" value="1"/>	<input type="button" value="1"/>	<input type="button" value="2"/>	<input type="button" value="3"/>
5,000	<input type="text" value="2"/>	<input type="button" value="4"/>	<input type="button" value="5"/>	<input type="button" value="6"/>
10,000	<input type="text" value="5"/>	<input type="button" value="7"/>	<input type="button" value="8"/>	<input type="button" value="9"/>
50,000	<input type="text" value="2"/>	<input type="button" value="0"/>	<input type="button" value="#. 돌아가기"/>	

24 HOURS JACKPOT ATM

**관리자 모드(입금)**  
입금액 수량 입력

1,000	<input type="text" value="1"/>	<input type="button" value="1"/>	<input type="button" value="2"/>	<input type="button" value="3"/>
5,000	<input type="text" value="2"/>	<input type="button" value="4"/>	<input type="button" value="5"/>	<input type="button" value="6"/>
10,000	<input type="text" value="5"/>	<input type="button" value="7"/>	<input type="button" value="8"/>	<input type="button" value="9"/>
50,000	<input type="text" value="2"/>	<input type="button" value="0"/>	<input type="button" value="#. 돌아가기"/>	

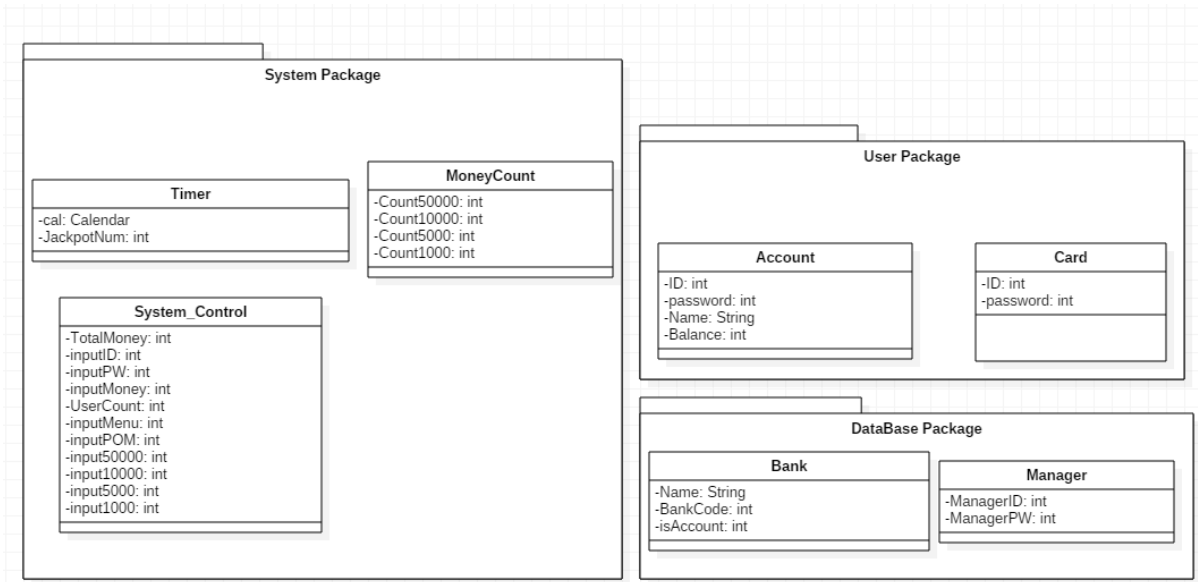
13. 잭팟ATM 당첨화면

24 HOURS JACKPOT ATM

**JACKPOT 777**

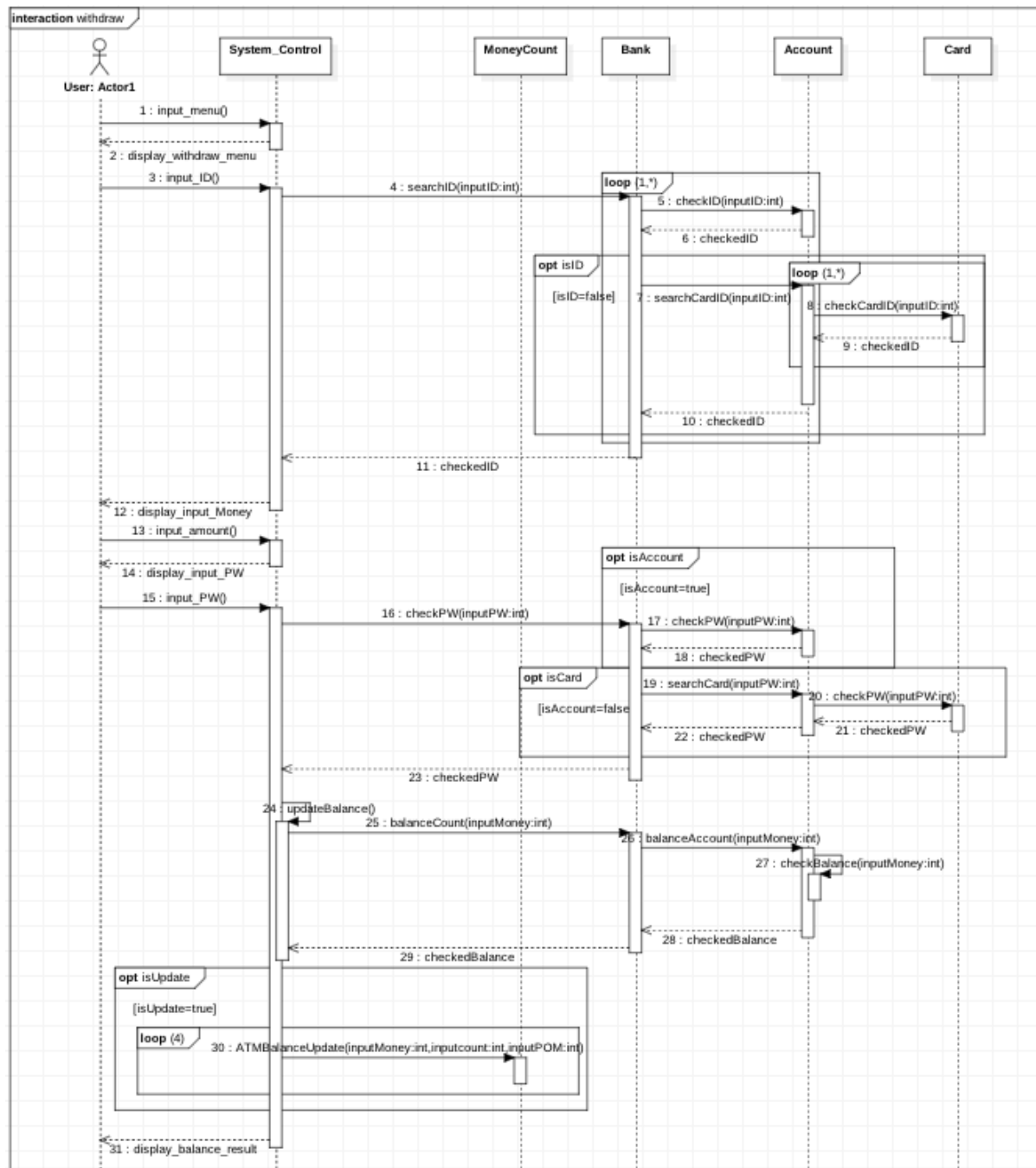
축하합니다. 오늘의 JACKPOT ATM 당첨 손님  
당첨금 5만원이 추가로 출금됩니다.

## Activity 2043. Refine System Architecture

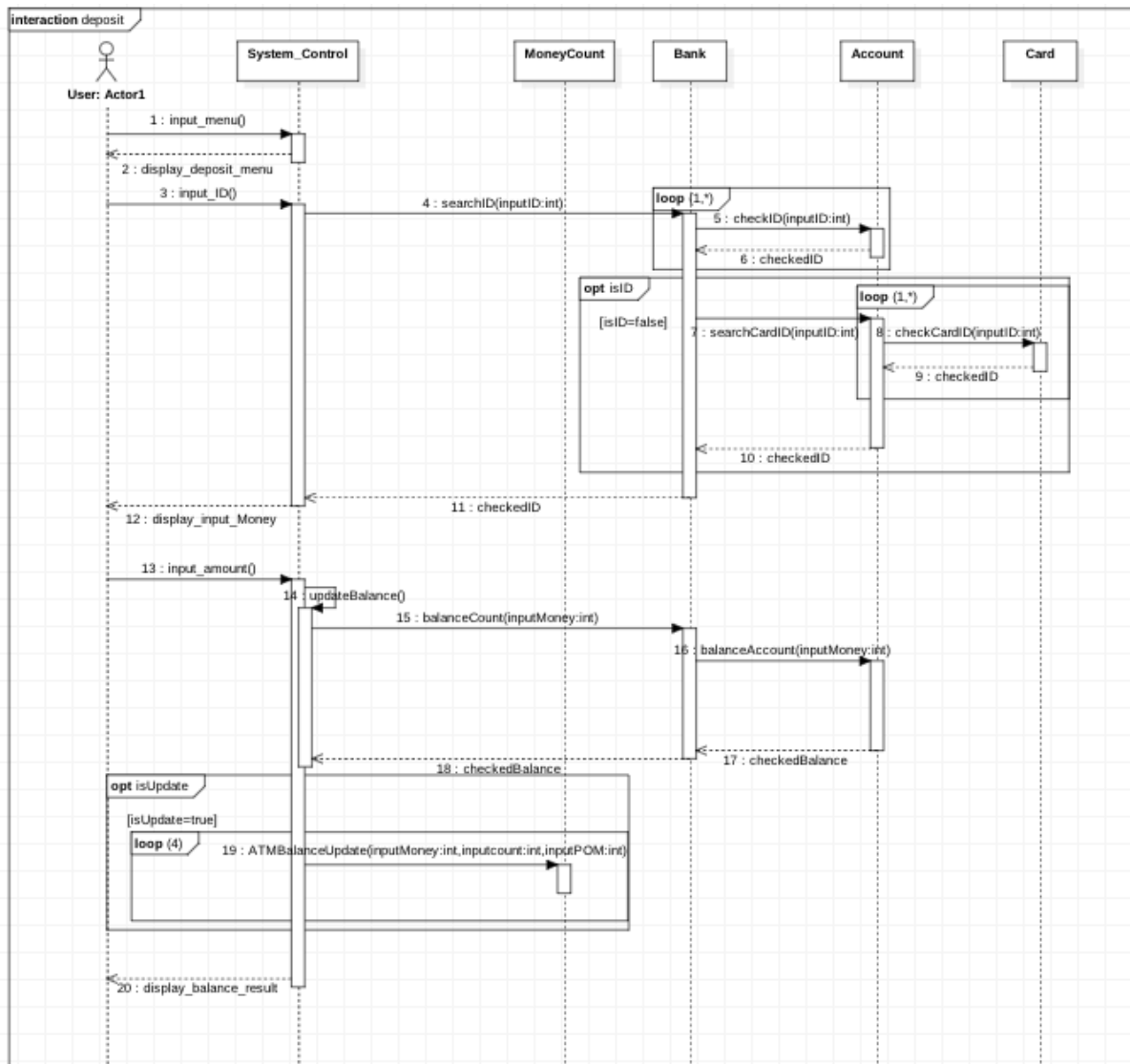


# Activity 2044. Define Interaction Diagrams

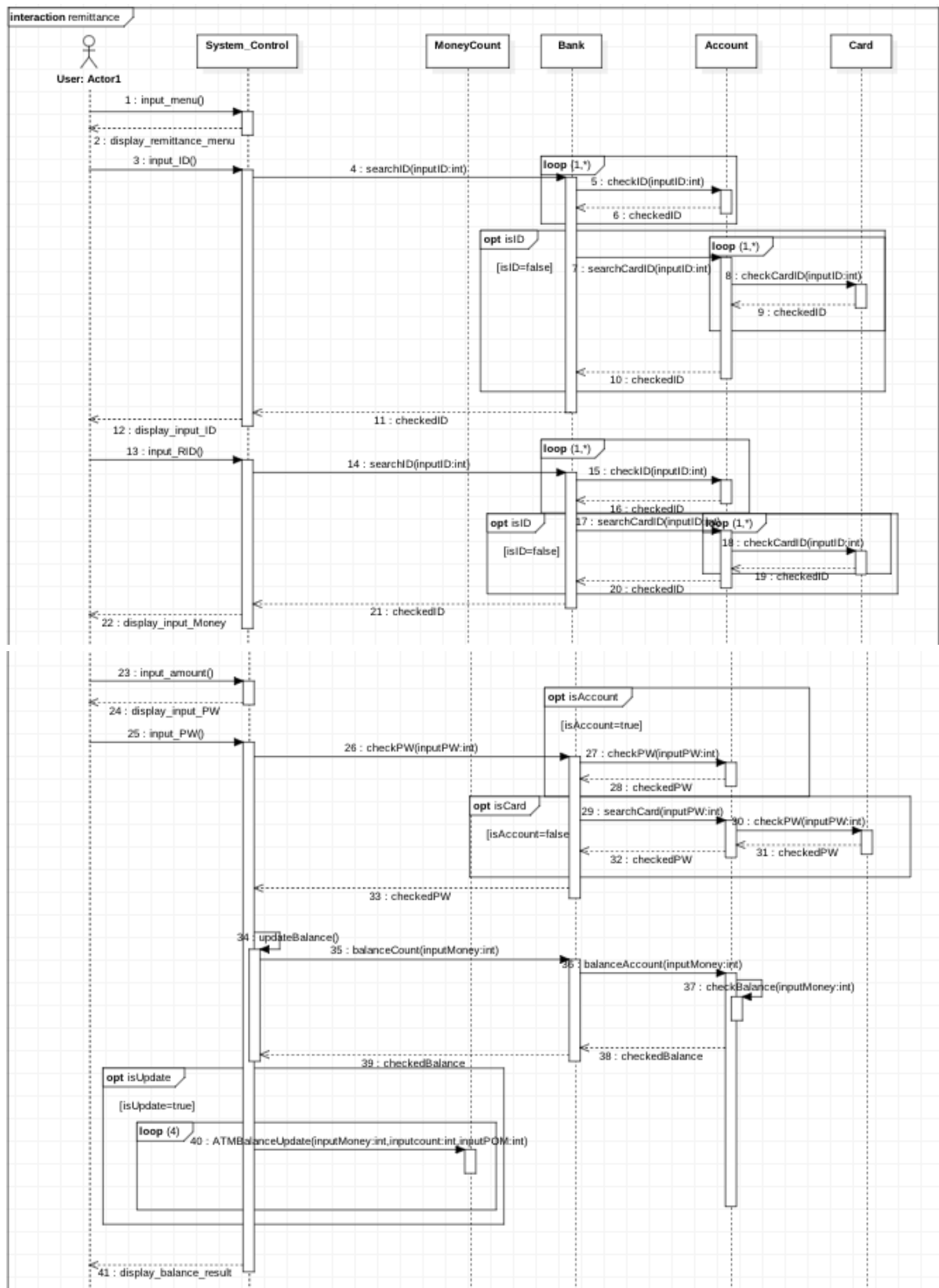
## 1.withdraw



## 2.deposit

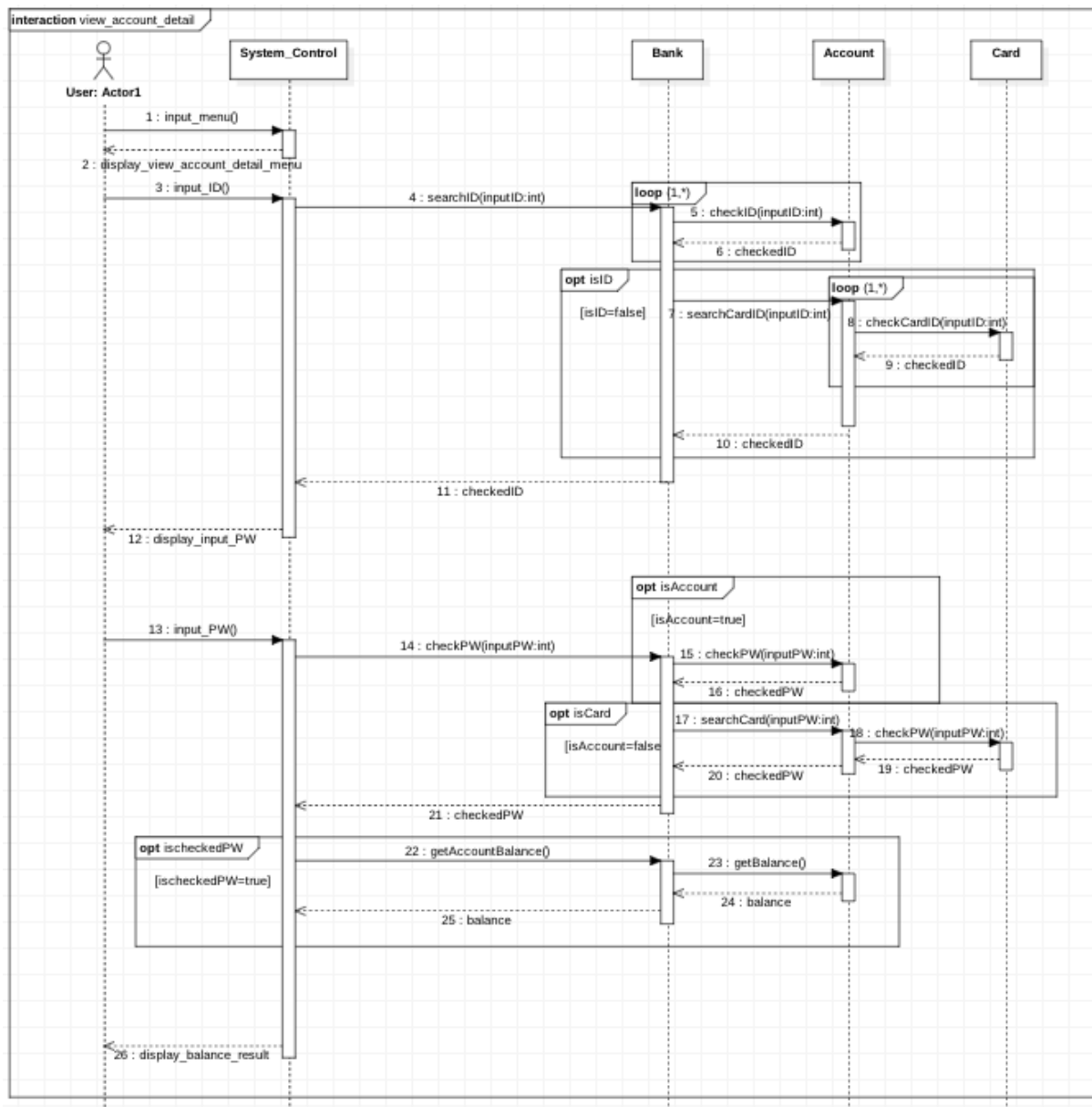


### 3.remittance

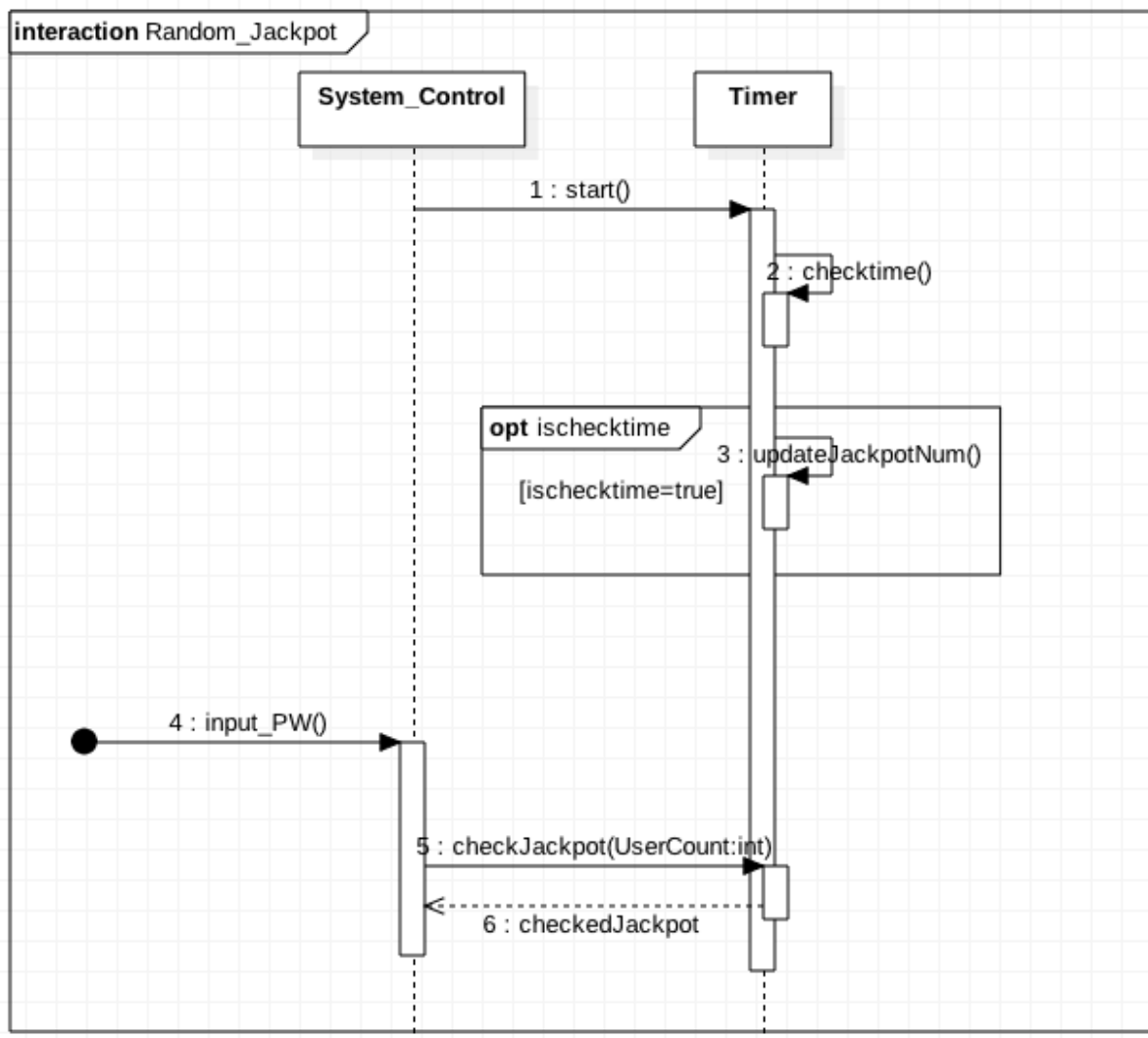




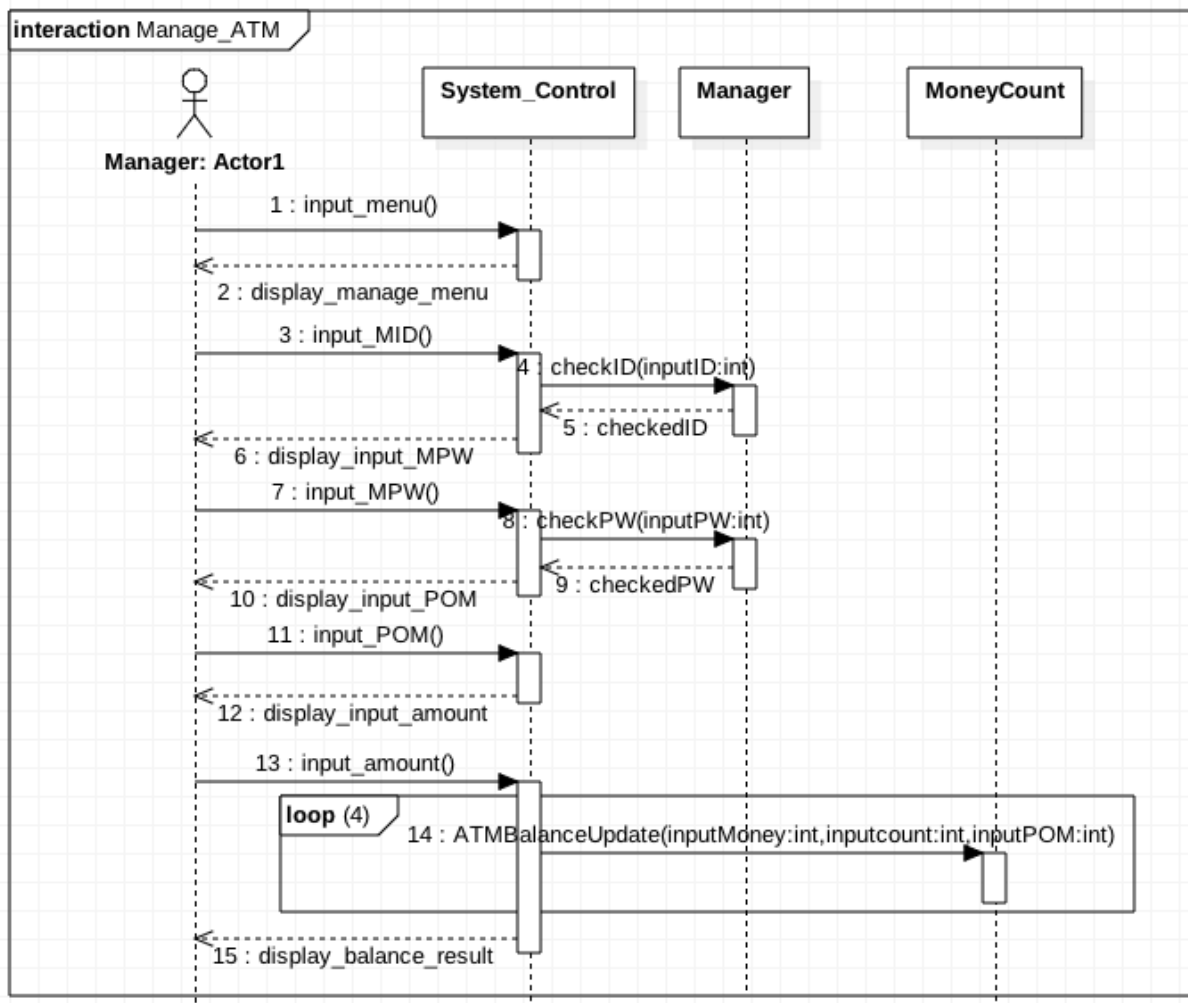
## 4.view\_account\_detail



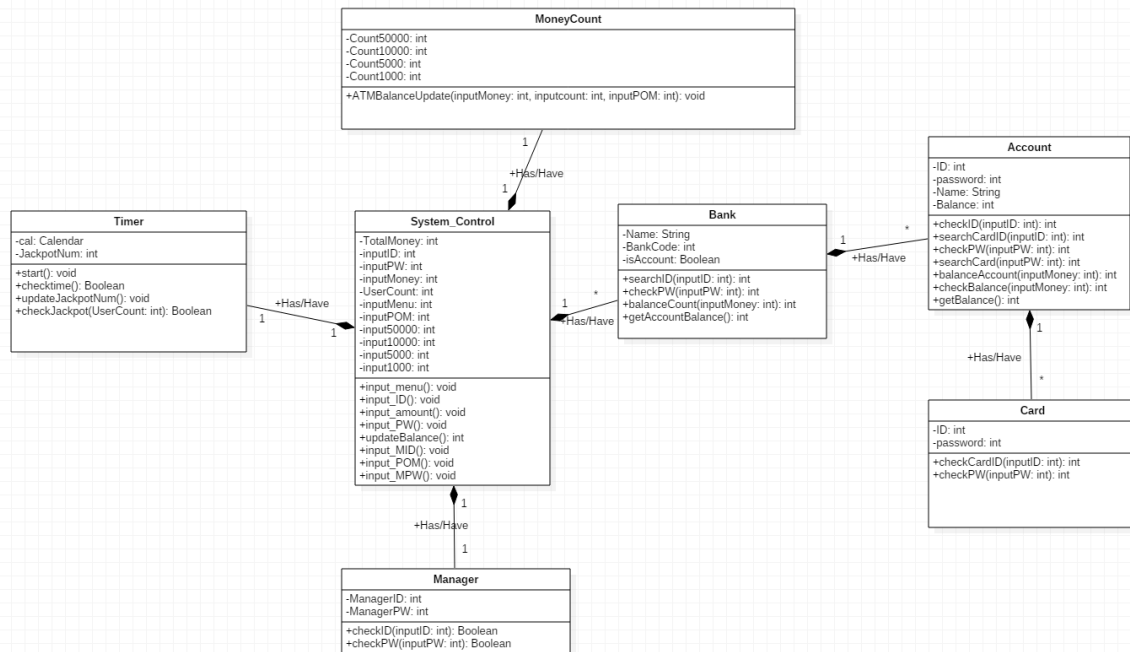
## 5.Random\_Jackpot



## 6.manage\_ATM



## Activity 2045. Define Design Class Diagrams



## Activity 2046. Design Traceability Analysis

