



# C++, The Convenient ATM

---

T2

201411262 김도현

201411271 박상우

201411312 장하나

201411316 정진호



# CONTENTS

1. Define Essential Use Cases
  2. Define Domain Model
  3. Refine Glossary
  4. Define System Sequence Diagram
  5. Define Operation Contracts
  6. Define State Diagrams
  7. Traceability Analysis
-



# 1. Define Use Case



# 1. Define Essential Use Cases

Use Case	1. Deposit
Actors	User
Purpose	지폐를 계좌에 입금한다.
Overview	User가 매체와 지폐를 입력할 경우, 해당 금액을 계좌에 입금한다. User가 매체의 입력 없이, 계좌번호와 지폐를 입력할 경우, 계좌의 소유자 확인 후 해당 금액을 계좌에 입금한다. 거래 결과를 출력한다.
Type	Primary
Cross References	Functions : R1.1, R5 Use Cases:
Pre-requisites	N/A
Typical Courses of Events	(A) : Actor, (S): System 1. (A) 계좌 정보를 입력한다. 2. (S) 계좌 정보의 유효성을 확인한다. (E1) 3. (A) 지폐를 입력한다. (E2) 4. (S) 해당 금액만큼 계좌에 입금한다. 5. (S) 거래결과를 출력한다.
Alternative Courses of Events	
Exceptional Courses of Events	E1 : 계좌 정보가 유효하지 않다면 진행되지 않는다. E2 : 입력한 지폐의 수량이 ATM이 수용할 수 있는 수량을 초과한다면 진행되지 않는다.

# 1. Define Essential Use Cases

Use Case	3. Withdraw
Actors	User
Purpose	지폐를 계좌에서 출금한다.
Overview	User가 매체 또는 계좌번호와 출금할 금액을 입력한다. 계좌번호를 입력했을 경우, 계좌의 소유자를 확인한다. 사용자 인증 후 해당 금액을 출금한다. 거래 결과를 출력한다.
Type	Primary
Cross References	Functions : R2.1, R5, R8 Use Cases:
Pre-requisites	N/A
Typical Courses of Events	(A) : Actor, (S): System <ol style="list-style-type: none"> <li>1. (A) 계좌 정보를 입력한다.</li> <li>2. (S) 계좌 정보의 유효성을 확인한다. (E1)</li> <li>3. (A) 계좌 비밀번호를 입력한다.</li> <li>4. (S) 계좌 비밀번호가 맞는지 확인한다. (E2)</li> <li>5. (A) 출금할 금액을 입력한다.(E3)(E4)</li> <li>6. (S) 해당 금액만큼 계좌에서 출금한다</li> <li>7. (S) 지폐를 출력한다.</li> <li>8. (S) 거래결과를 출력한다.</li> </ol>
Alternative Courses of Events	
Exceptional Courses of Events	E1 : 계좌 정보가 유효하지 않다면 진행되지 않는다. E2: 잘못된 비밀번호를 입력한다면 진행되지 않는다. E3 :입력한 지폐의 수량이 ATM의 현재 지폐 보유량을 초과한다면 진행되지 않는다. E4 :입력한 금액이 Actor의 계좌 잔액보다 크다면 진행되지 않는다.

# 1. Define Essential Use Cases

Use Case	5. Transfer Balance
Actors	User
Purpose	계좌의 잔액 중 일부를 다른 계좌로 이체한다.
Overview	User가 매체와 이체할 금액을 입력한다. User가 상대방의 계좌를 입력한다. 사용자 인증 후 해당 금액을 이체한다. 거래 결과를 출력한다.
Type	Primary
Cross References	Functions : R3.1, R5, R8 Use Cases:
Pre-requisites	N/A
Typical Courses of Events	(A) : Actor, (S): System <ol style="list-style-type: none"> <li>1. (A) 계좌 정보를 입력한다.</li> <li>2. (S) 계좌 정보의 유효성을 확인한다. (E1)</li> <li>3. (A) 계좌 비밀번호를 입력한다.</li> <li>4. (S) 계좌 비밀번호가맞는지 확인한다. (E2)</li> <li>5. (A) 상대방의 계좌 정보를 입력한다.</li> <li>6. (S) 계좌 정보의 유효성을 확인한다.(E1)</li> <li>7. (A) 이체할 금액을 입력한다.(E3)</li> <li>8. (S) 해당 금액만큼 이체한다.</li> <li>9. (S) 거래결과를 출력한다.</li> </ol>
Alternative Courses of Events	
Exceptional Courses of Events	E1 : 계좌 정보가 유효하지 않다면 진행되지 않는다. E2 : 잘못된 비밀번호를 입력한다면 진행되지 않는다. E3 : 입력한 금액이 Actor의 계좌 잔액보다 크다면 진행되지 않는다.

# 1. Define Essential Use Cases

Use Case	6. Split Pay
Actors	User
Purpose	기준이 되는 계좌로 여러 사용자가 이체한다.
Overview	대표 User가 매체와 금액, 인원수를 입력한다. 금액을 인원수로 나누어 인당 이체금액을 계산한다. 다른 User들은 순서대로 대표 User에게 계산된 금액을 이체한다. 다른 User들이 모두 거래를 마친 후, 대표 User의 거래 결과를 출력한다.
Type	Primary
Cross References	Functions : R3.2, R5, R8 Use Cases:
Pre-requisites	N/A
Typical Courses of Events	(A) : 대표 Actor (O) : 다른 Actors (S) : System 1. (A) 계좌 정보를 입력한다. 2. (S) 계좌 정보의 유효성을 확인한다. (E1) 3. (A) 총 금액과 인원수를 입력한다. 4. (S) 인당 이체할 금액을 계산한다. 5. (O) 계좌 정보를 입력한다. 6. (S) 계좌 정보의 유효성을 확인한다. (E1) 7. (O) 계좌 비밀번호를 입력한다. 8. (S) 비밀번호가 맞는지 확인한다. (E2) 9. (S) 인당 이체할 금액을 대표 Actor에게 이체한다. (E3) 10. (S) 거래결과를 출력한다. 11. (O) 입력된 인원 수 만큼 5~9를 반복한다. 12. (S) 전체 거래결과를 출력한다.
Alternative Courses of Events	
Exceptional Courses of Events	E1 : 계좌 정보가 유효하지 않다면 진행되지 않는다. E2 : 잘못된 비밀번호를 입력한다면 진행되지 않는다. E3 : 입력한 금액이 Actor의 계좌 잔액보다 크다면 진행되지 않는다.

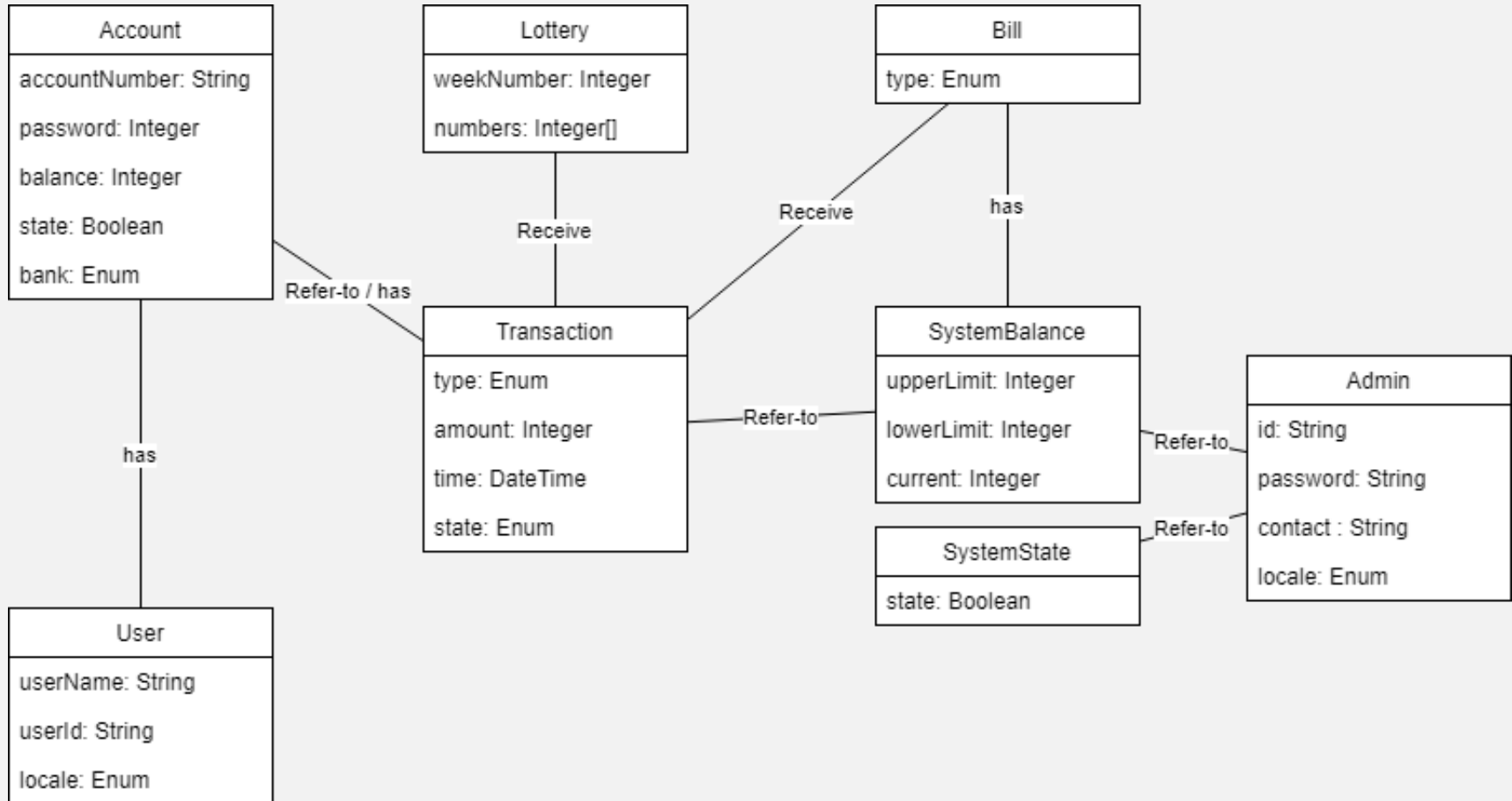


## 2. Define Domain Model





# 2. Define Domain Model





## 3. Refine Glossary



# 3. Refine Glossary

Glossary	Category	Description
User	Class	사용자
Admin	Class	관리자
Account	Class	계좌
Transaction	Class	거래
SystemBalance	Class	ATM의 지폐 보유량
Lottery	Class	로또 용지
Bill	Class	지폐
SystemState	Class	사용자 기능 상태
Account.accountNumber	Attribute	계좌의 계좌번호
Account.password	Attribute	계좌의 비밀번호
Account.balance	Attribute	계좌의 잔액
Account.state	Attribute	계좌의 상태
Account.bank	Attribute	계좌의 소속 은행

# 3. Refine Glossary

Glossary	Category	Description
User.userName	Attribute	사용자의 이름
User.userId	Attribute	사용자의 개인식별번호
User.locale	Attribute	사용자의 선택 언어
Lottery.weekNumber	Attribute	로또 추첨 회차
Lottery.numbers	Attribute	로또 응모 번호
Transaction.type	Attribute	거래 종류
Transaction.amount	Attribute	거래 금액
Transaction.time	Attribute	거래 발생 시각
Transaction.state	Attribute	거래 상태
Bill.type	Attribute	지폐 종류
SystemBalance.upperLimit	Attribute	ATM 지폐 보유량의 상한
SystemBalance.lowerLimit	Attribute	ATM 지폐 보유량의 하한
SystemBalance.current	Attribute	현재 ATM 지폐 보유량

# 3. Refine Glossary

<b>Glossary</b>	<b>Category</b>	<b>Description</b>
SystemState.state	Attribute	ATM 사용자 기능 상태
Admin.id	Attribute	관리자의 식별번호
Admin.password	Attribute	관리자의 비밀번호
Admin.contact	Attribute	관리자의 연락처
Admin.locale	Attribute	관리자의 선택 언어

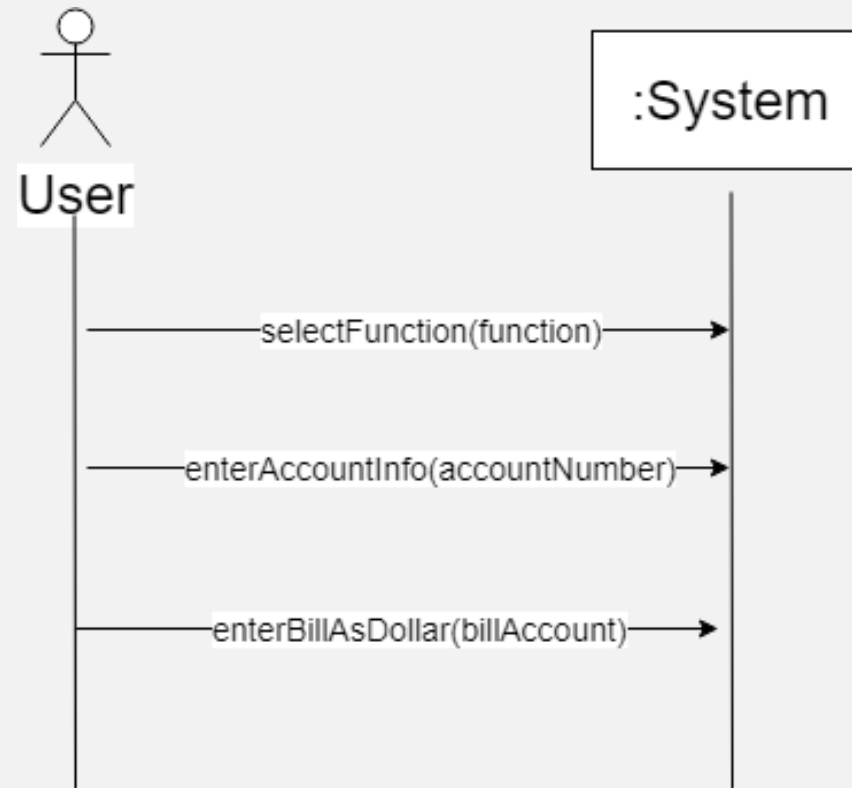


## 4. Define System Sequence Diagram



# 4. Define System Sequence Diagram

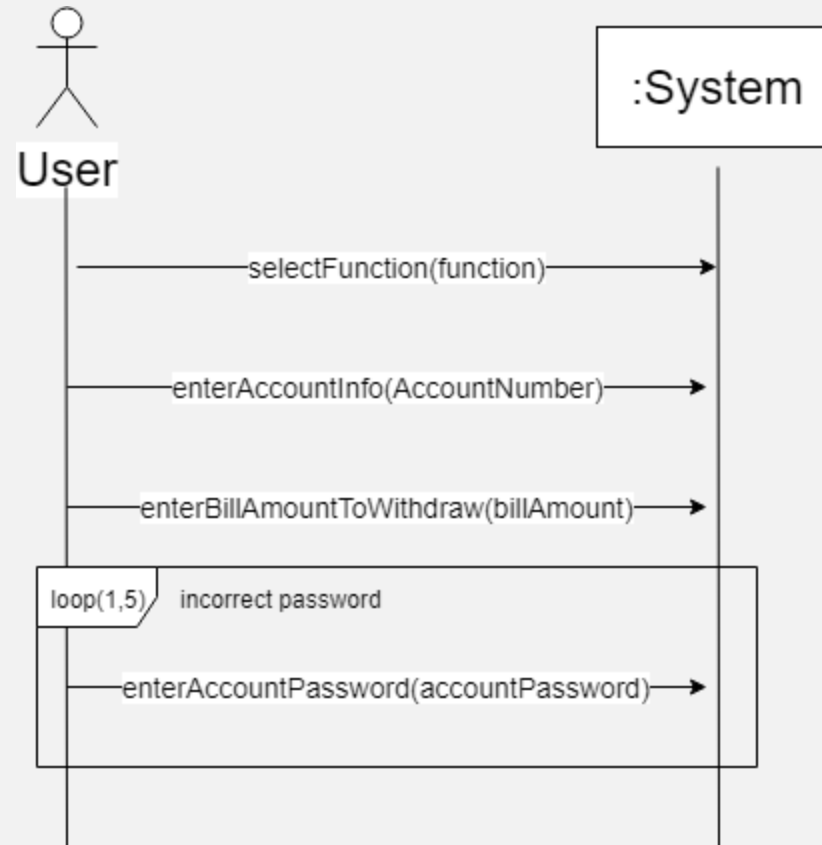
- Deposit



# 4. Define System Sequence Diagram



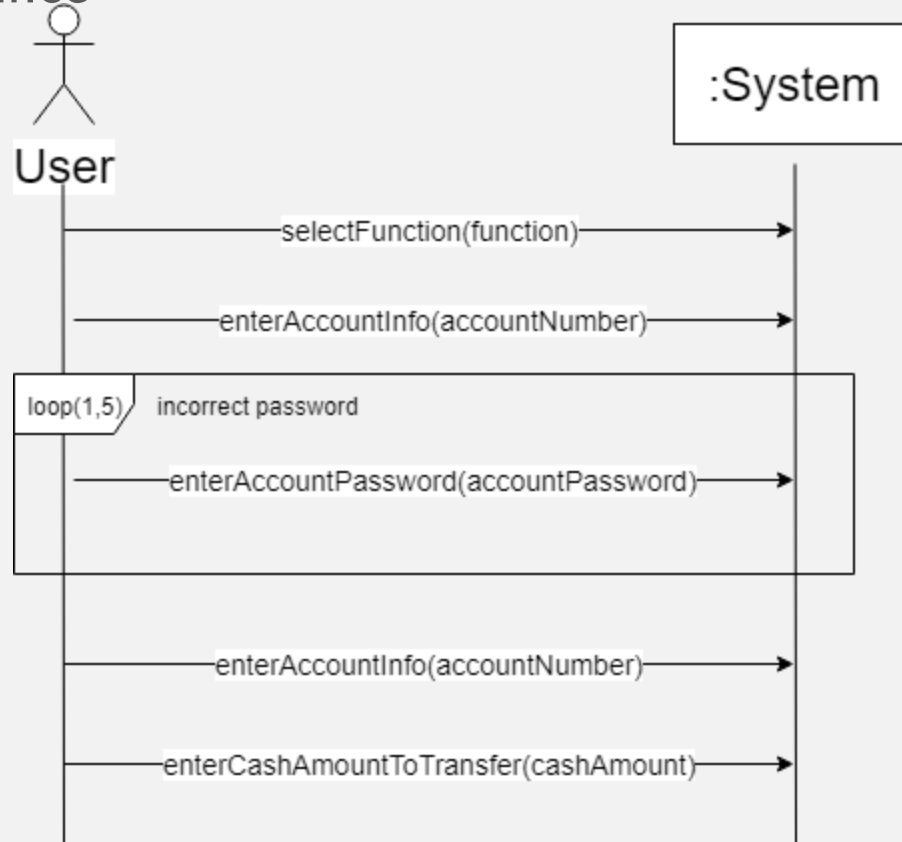
- Withdraw





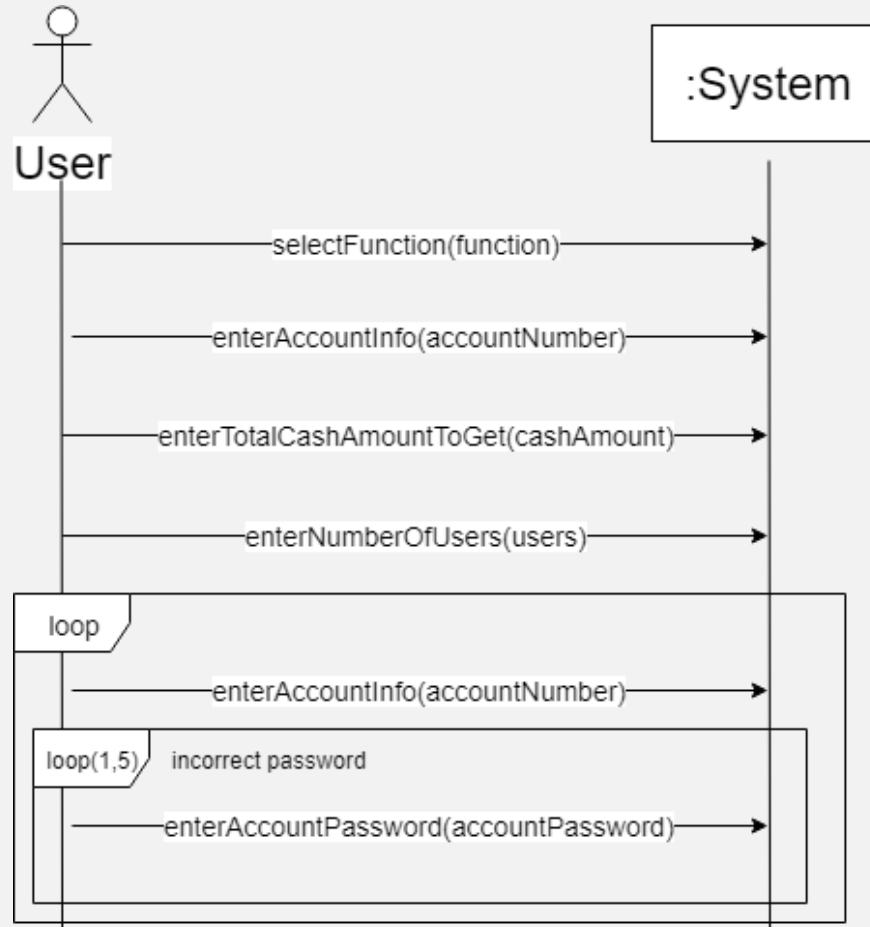
# 4. Define System Sequence Diagram

- Transfer Balance



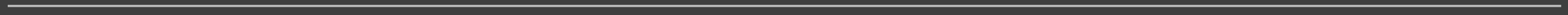
# 4. Define System Sequence Diagram

- Split pay





# 5. Define Operation Contracts



## 5. Define Operation Contracts

<b>Name</b>	selectFunction(function: Enum)
<b>Responsibilities</b>	초기 화면에서 기능을 선택한다.
<b>Type</b>	System
<b>Cross References</b>	System functions: R1.1, R1.2, R2.1, R2.2, R3.1, R3.2, R4.1, R4.2, R6, R7, R9, R10, R11, R12.1, R12.2 Use Case: 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16, 17, 18
<b>Notes</b>	
<b>Exceptions</b>	None
<b>Output</b>	사용자가 선택한 화면으로 넘어간다.
<b>Pre-conditions</b>	ATM이 사용자 기능 제한 상태가 아니다
<b>Post-conditions</b>	선택한 기능이 거래라면 Transaction을 생성한다. Transaction이 생성되었다면 type을 선택된 거래로 설정한다.

---

## 5. Define Operation Contracts

<b>Name</b>	enterAccountInfo(accountNumber: String)
<b>Responsibilities</b>	사용자에게서 계좌 정보를 입력받고, 확인한다.
<b>Type</b>	System
<b>Cross References</b>	System functions: R1.1, R1.2, R2.1, R2.2, R3.1, R3.2, R4.1, R4.2, R7 Use Case: 1, 2, 3, 4, 5, 6, 7, 8, 11
<b>Notes</b>	
<b>Exceptions</b>	계좌정보가 유효하지 않다면 진행되지 않는다.
<b>Output</b>	다음 화면으로 넘어간다.
<b>Pre-conditions</b>	기능이 선택되어 있다.
<b>Post-conditions</b>	계좌의 유효성이 확인된다. 선택한 기능이 거래일 경우 해당 Account와 Transaction을 연결한다.

---

## 5. Define Operation Contracts

<b>Name</b>	enterAccountPassword(accountPassword: Integer)
<b>Responsibilities</b>	비밀번호를 입력 받고, 계좌의 비밀번호와 비교한다.
<b>Type</b>	System
<b>Cross References</b>	System functions: R2.1, R2.2, R3.1, R3.2, R4.1, R4.2 Use Case: 3, 4, 5, 6, 7, 8
<b>Notes</b>	
<b>Exceptions</b>	5회 이상 잘못된 비밀번호를 입력한다면 매체를 반환하고 초기 화면으로 돌아간다. 해당 계좌의 거래를 제한한다.
<b>Output</b>	다음화면으로 넘어간다.
<b>Pre-conditions</b>	진행중인 거래가 존재한다.
<b>Post-conditions</b>	None

---

## 5. Define Operation Contracts

<b>Name</b>	authorizeAdmin(adminId : Integer, adminPw : String)
<b>Responsibilities</b>	아이디와 패스워드를 입력받아 관리자 권한을 승인받는다.
<b>Type</b>	System
<b>Cross References</b>	System functions:R10, R11, R12.1, R12.2 Use Case: 14, 15, 16, 17, 18
<b>Notes</b>	
<b>Exceptions</b>	입력받은 계정 정보가 유효하지 않다면 초기화면으로 돌아간다.
<b>Output</b>	None
<b>Pre-conditions</b>	None
<b>Post-conditions</b>	관리자 권한이 승인된다.

---



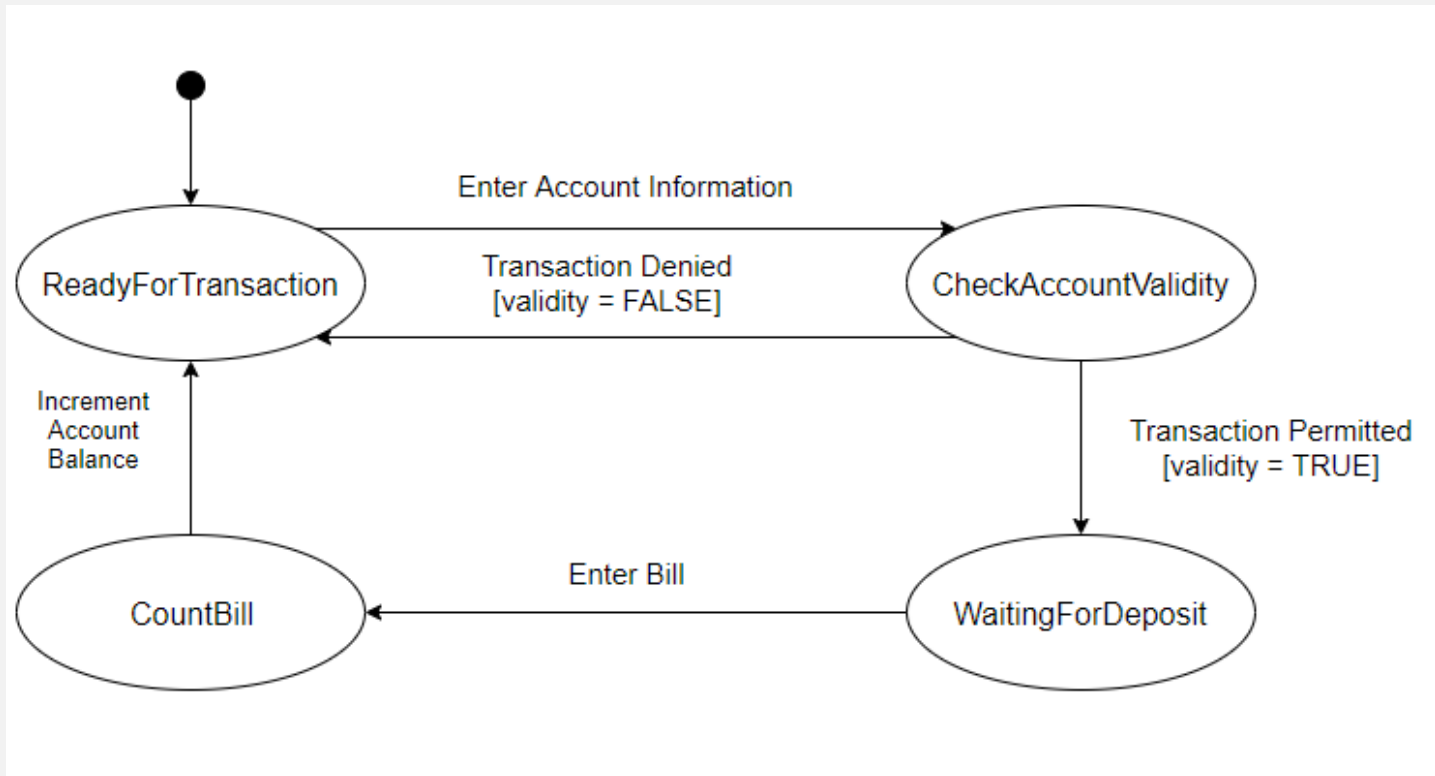
## 6. Define State Diagrams





# 6. Define State Diagrams

< State Diagram for "Deposit" >





# 7. Traceability Analysis



# 7. Traceability Analysis

	Requirements		Use Cases		Operations
R1.1	Deposit	→	1. Deposit	→	selectFunction
R1.2	Deposit Foreign Cash	→	2. Deposit Foreign Cash	→	enterAccountInfo
R2.1	Withdraw	→	3. Withdraw	→	enterBill
R2.2	Withdraw Foreign Cash	→	4. Withdraw Foreign Cash	→	enterBillAsDollar
R3.1	Transfer Balance	→	5. Transfer Balance	→	enterBillAmountToWithdraw
R3.2	Split Pay	→	6. Split Pay	→	enterAccountPassword
R4.1	Query Transaction List	→	7. Query Transaction List	→	enterBillAmountToWithdrawAsDollar
R4.2	Query Balance	→	8. Query Balance	→	enterCashAmountToTransfer
R5	Calculate Fee	→	9. Calculate Fee	→	enterTotalCashAmountToGet
R6	Report Lost Card	→	10. Report Lost Card	→	enterPeriodToQuery
R7	Get Lottery Prize	→	11. Get Lottery Prize	→	selectCard
R8	Freeze Account	→	12. Freeze Account	→	proceedRenewCard
R9	Change Locale	→	13. Change Locale	→	enterLottery
R10	Manage Admin	→	14. Add Admin	→	changeLocale
R11	Toggle ATM State	→	15. Remove Admin	→	enterAdminInfo
R12.1	Change ATM Balance	→	16. Toggle ATM State	→	authorizeAdmin
R12.2	Query ATM Balance	→	17. Change ATM Balance	→	removeAdmin
		→	18. Query ATM Balance	→	toggleATMState
					setNewATMBalance
					queryATMBalance
					enterNumberOfUsers
					getUserId



# Q & A

