

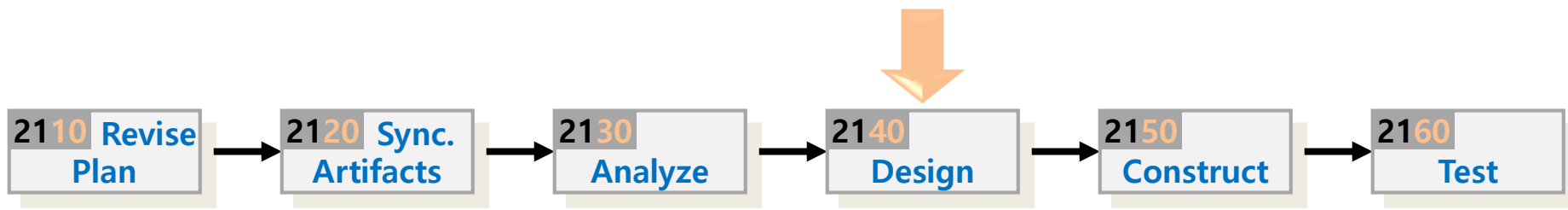
# Software Modeling & Analysis

OOPT

- Case Study : Printer System

Lecturer: JUNBEOM YOO  
jbyoo@konkuk.ac.kr

# Phase 2040. Design



# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Login
<b>Actor</b>	User, Manager
<b>Purpose</b>	User와 manager가 시스템에 접속하기 위해 로그인 할 수 있도록 한다.
<b>Overview</b>	ID/PW 를 입력 받아 계정과 비밀번호가 일치하는 경우 user or manager로 로그인 한다. 일치하는 계정이 없는 경우 로그인 되지 않는다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 1.1,
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 화면에 ID, PW 를 입력 한다. 2. (A) : 로그인 버튼을 누른다. 3. (S) : Account의 User instance들의 id, pw와 비교 한다. 4. (S) : 일치하는 instance가 존재하는 경우 user로 로그인 한다. 5. (S) : Account의 Manager instance들의 id, pw와 비교 한다. 6. (S) : 일치하는 instance가 존재하는 경우 manager 로 로그인 한다. 7. (S) : 현재 로그인된 계정을 기록 한다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	E1. 일치하는 계정이 없는 경우 접속되지 않는다.

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Logout
<b>Actor</b>	User, Manager
<b>Purpose</b>	접속된 User or Manager의 접속을 종료한다.
<b>Overview</b>	Logout 버튼을 누르면 접속된 user or manager의 접속을 종료하고 초기화면으로 돌아간다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 1.1
<b>Pre-Requisites</b>	Login이 선행 되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 로그아웃 버튼을 누른다. 2. (S) : ManagementSystem의 현재 로그인 정보를 초기화 한다. 3. (S) : 초기화면으로 돌아간다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	N/A

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Make Account
<b>Actor</b>	User
<b>Purpose</b>	새로운 사용자 계정을 생성한다.
<b>Overview</b>	생성할 계정의 id/pw를 입력 후 생성 버튼을 눌러 새로운 사용자 계정을 생성 할 수 있다. 새로 생성한 계정은 동일한 id가 없는 경우에만 생성 된다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 1.2
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) : 계정 생성 버튼을 누른다.</li> <li>2. (S) : 계정생성 화면으로 전환 한다.</li> <li>3. (A) : 화면에 id, pw를 입력 한다.</li> <li>4. (A) : 생성 요청 버튼을 누른다.</li> <li>5. (S) : Account의 user instance의 id를 확인 하여 기존 사용자와 비교 한다.</li> <li>6. (S) : 일치하는 id가 없는 경우 user instance를 생성하여 사용자 계정을 생성 하고 초기 정보를 입력 한다.</li> </ol>
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	E1. 동일한 id를 가진 계정이 이미 존재하는 경우 생성하지 않는다.

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Identify Balance
<b>Actor</b>	User
<b>Purpose</b>	사용자 계정에 남아있는 잔액을 확인 한다.
<b>Overview</b>	사용자의 잔액 확인 요청 시 남아있는 잔액을 확인하여 출력 한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 1.3
<b>Pre-Requisites</b>	사용자로 login이 선행 되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 잔액 확인 버튼을 누른다. 2. (S) : 현재 로그인된 user instance에서 balance값을 출력 한다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Recharge Balance
<b>Actor</b>	User
<b>Purpose</b>	해당 사용자 계정에 잔액을 충전 한다.
<b>Overview</b>	입력 받은 금액 만큼 해당 사용자 계정에 잔액을 충전 한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 1.4
<b>Pre-Requisites</b>	사용자로 login이 선행 되어야 한다.
<b>Typical Courses of Events</b>	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) : 화면에 금액을 입력 한다.</li> <li>2. (A) : 잔액 충전 버튼을 누른다.</li> <li>3. (S) : 입력 받은 금액 만큼 해당 user instance.balance 에 증가 시킨다.</li> </ol>
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	E1. 음수 값은 처리하지 않고 오류 메시지를 보낸다.

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Request Print
<b>Actor</b>	User
<b>Purpose</b>	사용자의 요청을 받아 인쇄를 진행 한다.
<b>Overview</b>	사용자의 요청을 받아 입력된 매수 만큼 인쇄를 진행 한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 2.1, R 2.2
<b>Pre-Requisites</b>	사용자로 login이 선행 되어야 한다.
<b>Typical Courses of Events</b>	<p>(A) : Actor, (S) : System</p> <ol style="list-style-type: none"> <li>1. (A) : 화면에 필요 인쇄 매수를 입력 한다.</li> <li>2. (A) : 인쇄 버튼을 누른다.</li> <li>3. (S) : ManagementSystem에서 매수에 따라 요금을 계산 한다.</li> <li>4. (S) : check balance use case를 통해 잔액을 비교 한다.</li> <li>5. (S) : 잔액이 부족하면 ERROR 메시지 후 종료 한다.</li> <li>6. (S) : 인쇄 매수와 printer의 paper를 비교하여 부족할 경우 종료 한다.</li> <li>7. (S) : printer 의 paper를 매수 만큼 감소 시킨다.</li> <li>8. (S) : allProfit 을 요금 만큼 증가 시킨다.</li> <li>9. (S) : user instance.balance를 요금만큼 감소 시킨다.</li> </ol>
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	E 2. 매수가 음수일 경우 진행하지 않는다.



# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Check Balance
<b>Actor</b>	Event-based
<b>Purpose</b>	인쇄 진행 시 잔액과 필요 요금을 확인 및 비교 한다.
<b>Overview</b>	사용자의 요청에 의해 인쇄 진행 시 시스템의 요청을 받아 필요 요금과 잔액을 체크한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 2.1, R 2.2 Use Cases: Request Print
<b>Pre-Requisites</b>	시스템에 인쇄 요청 후에 동작 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (S - A) : 사용자의 인쇄 요청을 받은 시스템이 잔액 확인을 요청한다. 2. (S) : user instance 의 balance와 필요 요금을 비교한 결과를 보낸다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Identify Paper
<b>Actor</b>	Manager
<b>Purpose</b>	시스템 관리자가 프린터에 남은 용지를 확인 할 수 있다.
<b>Overview</b>	시스템 관리자의 요청에 따라 프린터에 남은 용지 잔량을 확인한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 3.1
<b>Pre-Requisites</b>	Manager로 로그인 되어 있어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 용지 확인 버튼을 누른다. 2. (S) : printer 의 paper 값을 출력 한다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Recharge Paper
<b>Actor</b>	Manager
<b>Purpose</b>	시스템 관리자의 요청에 따라 시스템의 용지 잔량을 충전 한다.
<b>Overview</b>	시스템 관리자가 요청한 매수 만큼 시스템의 용지 잔량을 충전 한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 3.2
<b>Pre-Requisites</b>	Manager로 로그인 되어 있어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 화면에 충전할 매수를 입력 한다. 2. (A) : 용지 충전 버튼을 누른다. 3. (S) : 입력된 용지 수 만큼 printer의 paper값을 증가 시킨다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	E 1. 입력 받은 매수가 없거나 음수인 경우 충전이 되지 않는다.

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Identify User
<b>Actor</b>	Manager
<b>Purpose</b>	관리자가 사용자들의 정보를 확인 한다.
<b>Overview</b>	관리자의 요청에 따라 사용자들의 id/pw, 잔액 정보를 확인한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 3.3
<b>Pre-Requisites</b>	Manager로 로그인 되어 있어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 사용자 확인 버튼을 누른다. 2. (S) : Account 의 모든 user instance 정보를 출력 한다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	E1. user instance 에 아무 정보가 없는 경우 출력하지 않는다.

# Activity 2041. Design Real Use Cases

<b>Use Case</b>	Identify Money
<b>Actor</b>	Manager
<b>Purpose</b>	총 수익을 확인 한다.
<b>Overview</b>	관리자의 요청에 따라 지금까지의 총 수익을 확인 한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R 3.4
<b>Pre-Requisites</b>	Manager로 로그인 되어 있어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 수익 확인 버튼을 누른다. 2. (S) : allProfit 값을 출력 한다.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	

# Activity 2042.

## Define Reports, UI, and Storyboards

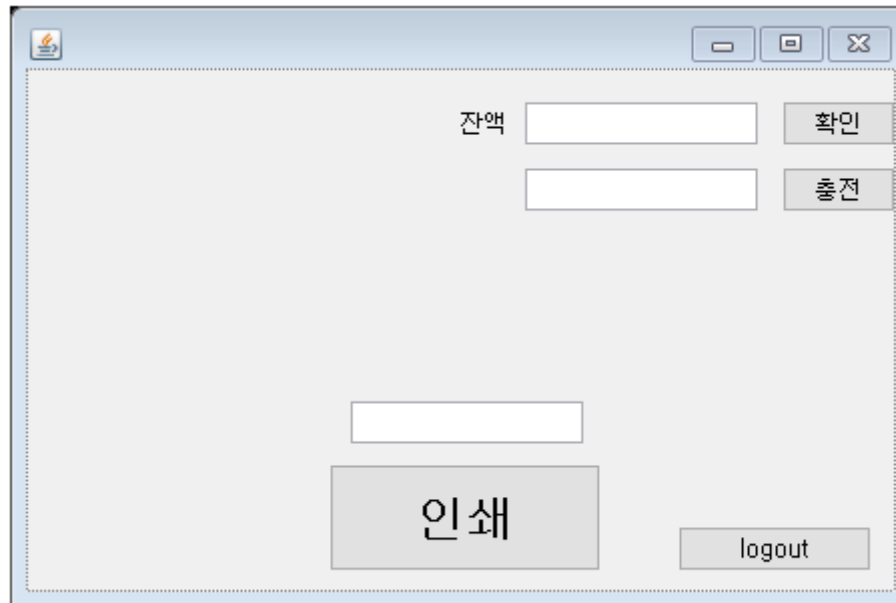
- 로그인 초기 화면 예시

The image shows a windowed application with a light blue title bar. Inside the window, there is a login form. The form consists of two text input fields. The first field is labeled 'ID' and the second is labeled 'PW'. Below the input fields are two buttons: 'Login' and '계정생성' (Account Creation). The window also features standard window controls (minimize, maximize, close) in the top right corner.

# Activity 2042.

## Define Reports, UI, and Storyboards

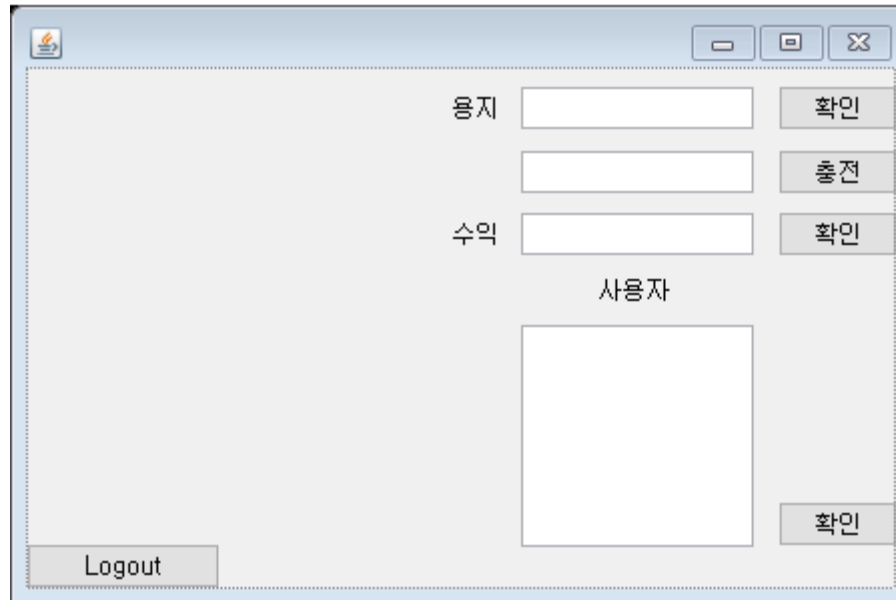
- 사용자 화면 예시



# Activity 2042.

## Define Reports, UI, and Storyboards

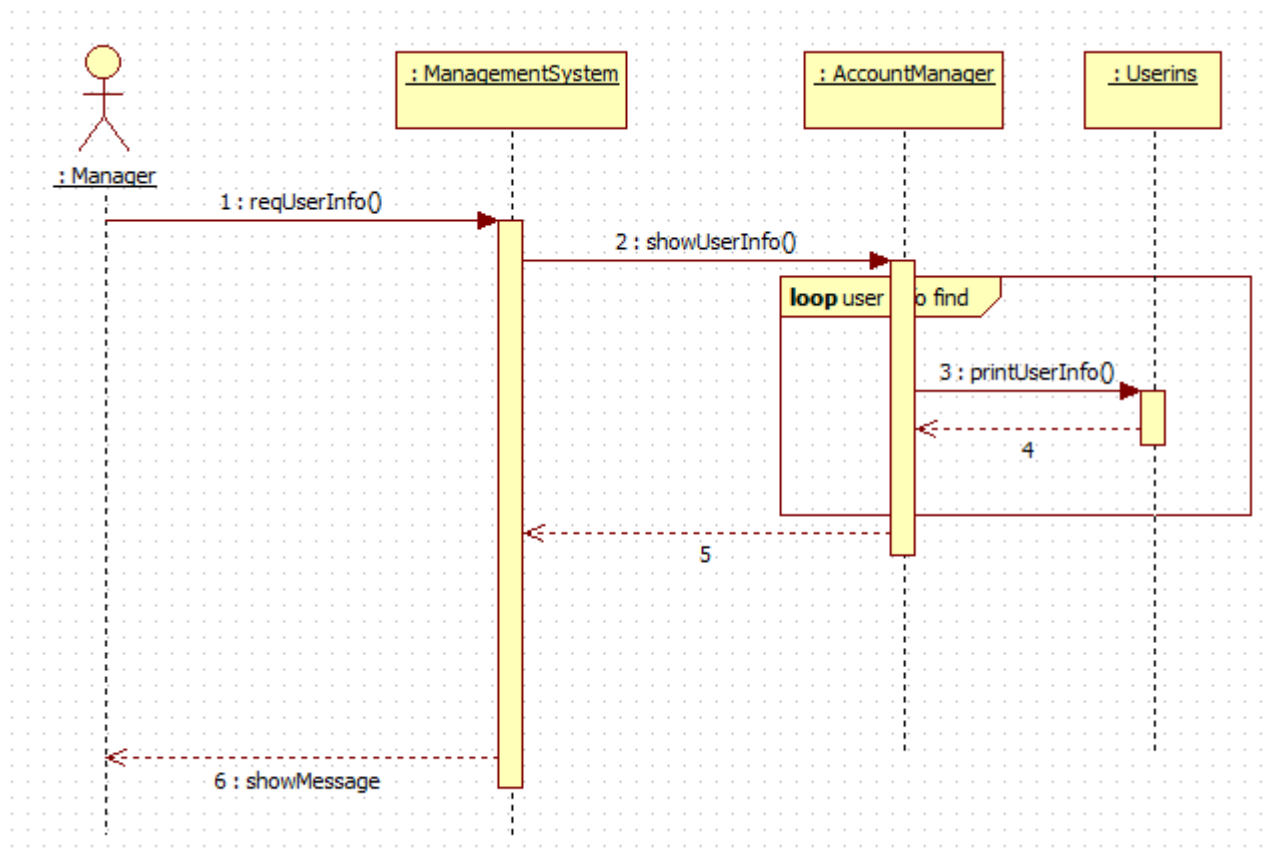
- 매니저 관리 화면 예시





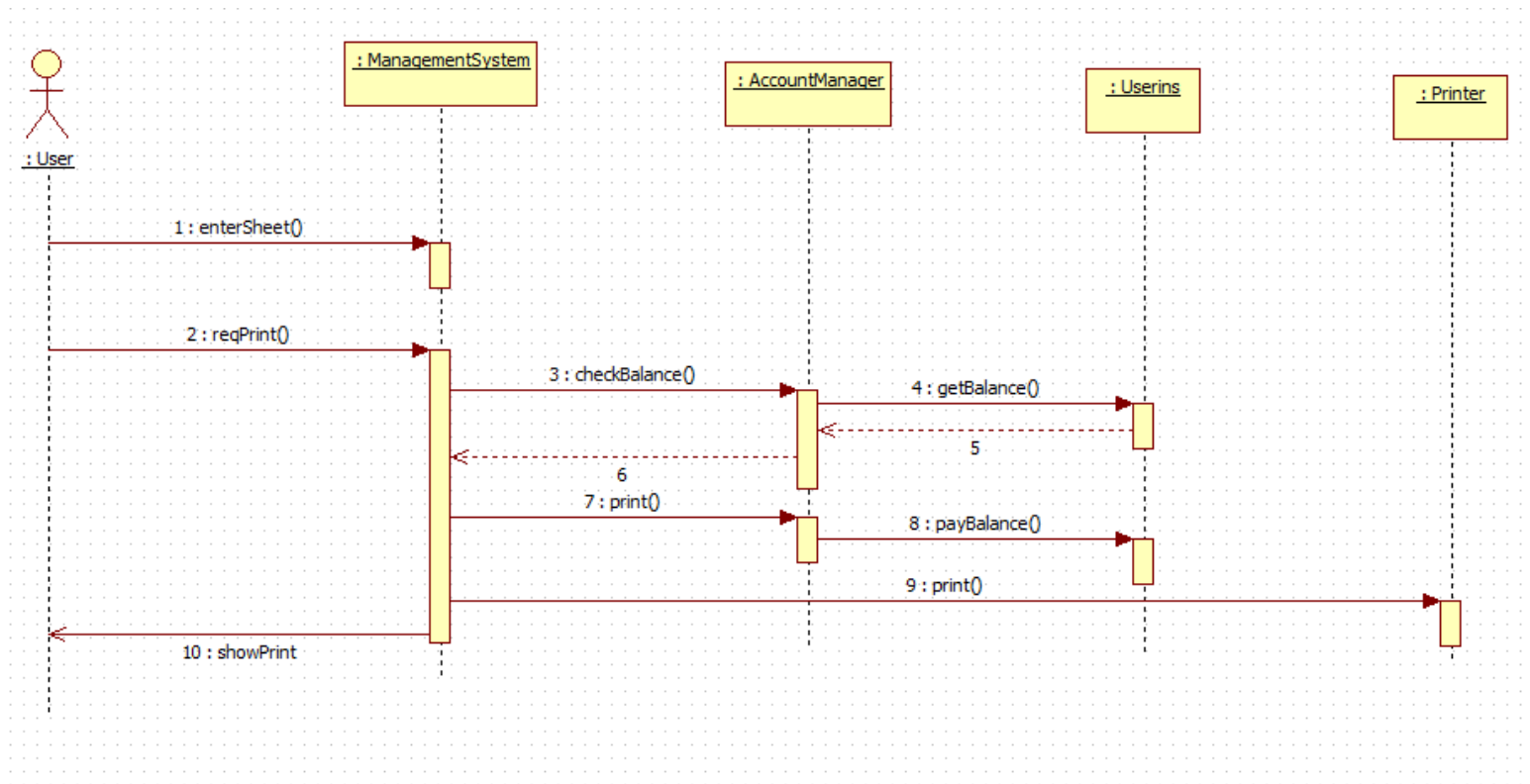
# Activity 2044. Define Interaction Diagrams

- Sequence diagram으로 작성
  - Identify User use case example

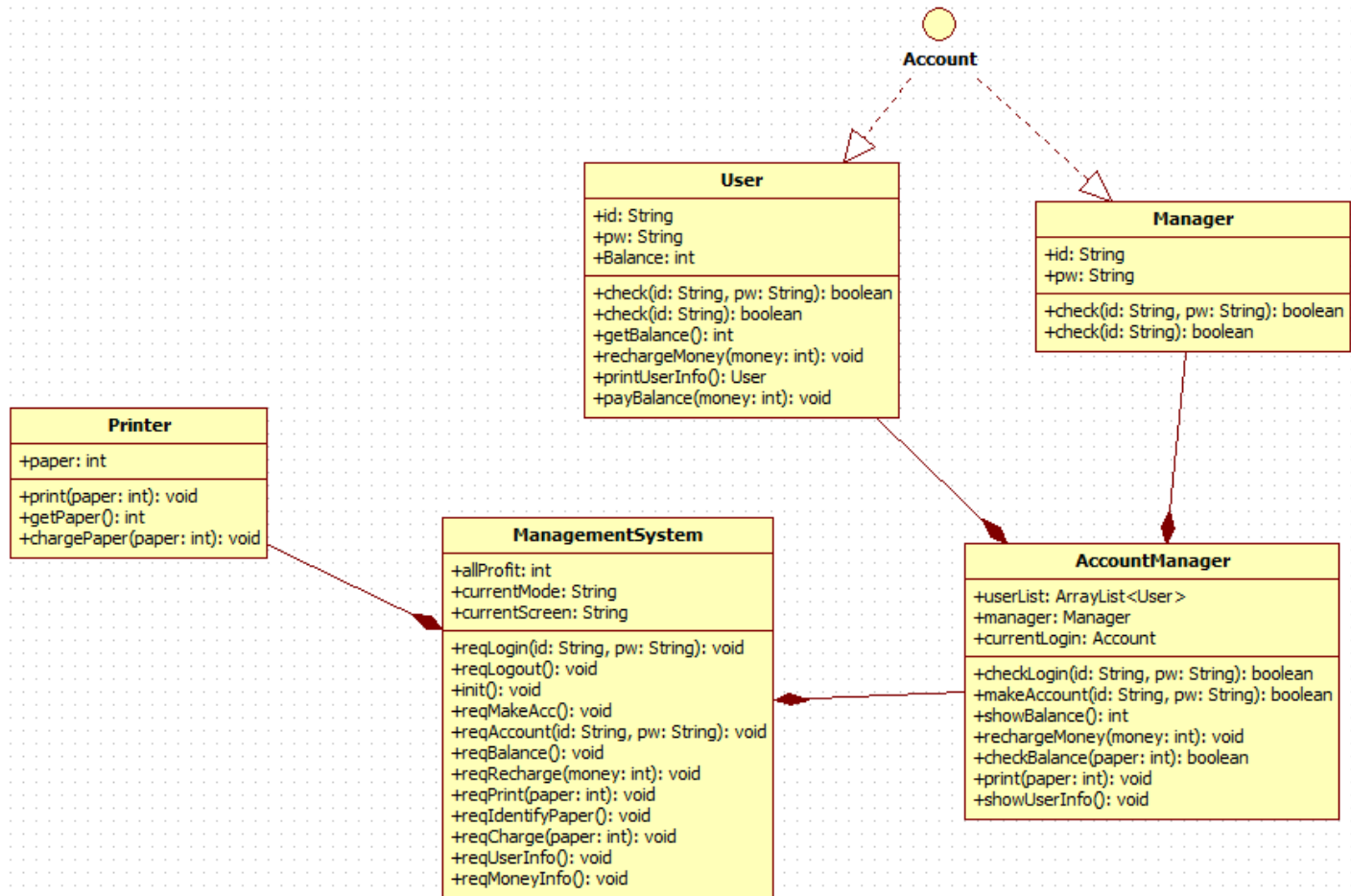


# Activity 2044. Define Interaction Diagrams

- Request Print use case

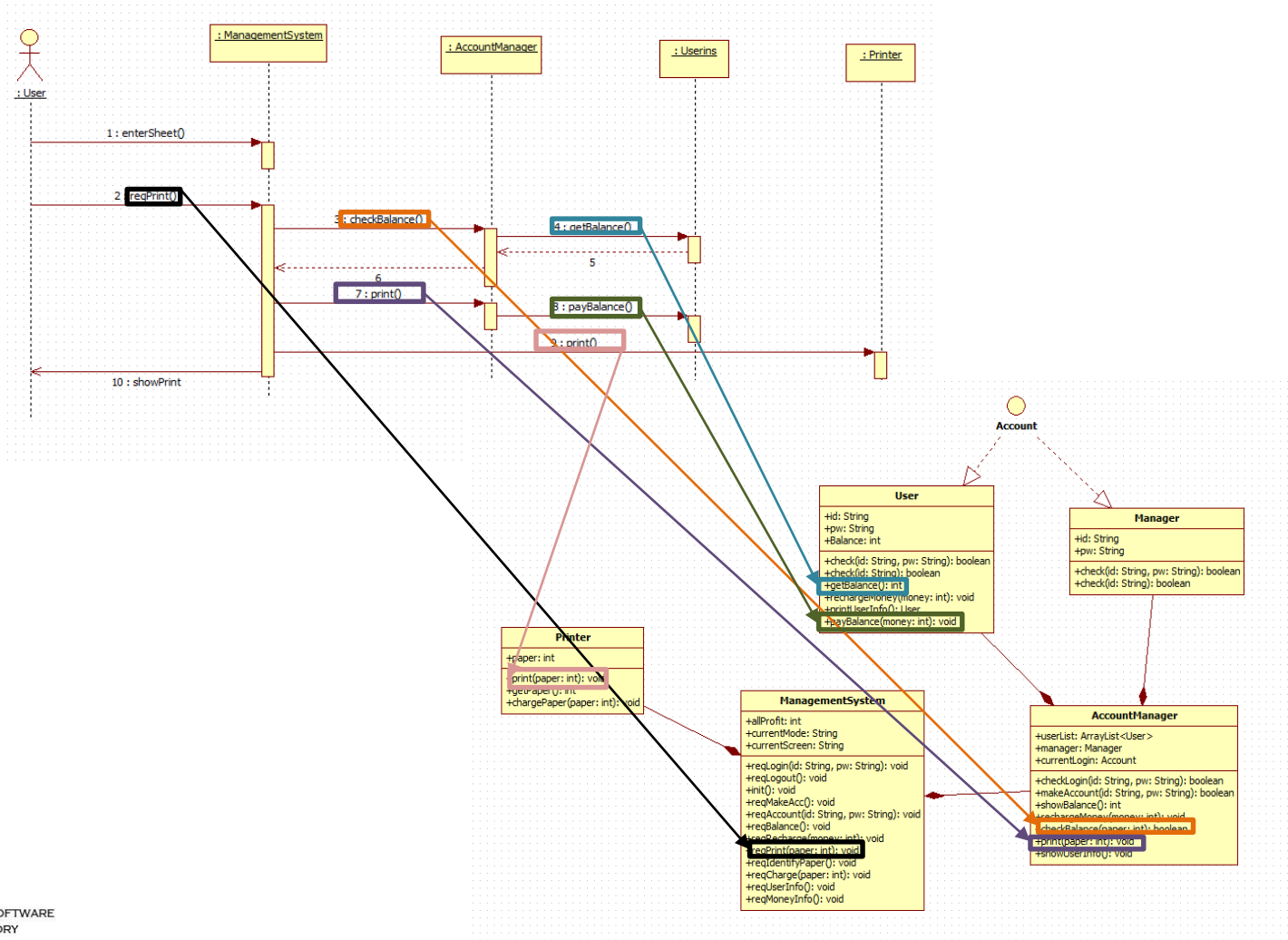


# Activity 2045. Define Design Class Diagrams



# Activity 2046. Design Traceability Analysis

- Tracing operations in interaction diagram with class diagram



# Activity 2046. Design Traceability Analysis

- Tracing operations in interaction diagram with class diagram

Operation in sequence diagram	Operation in interaction diagram	Method	Class	
1: enterInfo	enterInfo()	reqLogin(id:String, pw:String):void	ManagementSystem	
2: reqLogin	reqLogin	reqLogout():void		
3: reqLogout	checkLogin(id, pw)	init():void		
4: reqMakeAcc	check(id, pw)	reqMakeAcc():void		
5: enterAccInfo	check(id, pw)	reqAccount(id:String, pw:String):void		
6: reqAccount	saveCurrentLogin	reqBalance():void		
7: reqBalance	showMessage	reqRecharge(money:int):void		
8: enterFee	reqLogout()	reqPrint(paper:int):void		
9: reqRecharge	init()	reqIdentifyPaper():void		
10: enterSheet	showMessage()	reqCharge(paper:int):void		
11: reqPrint	reqMakeAcc()	reqUserInfo():void		
12: reqPaperIdentify	show()	reqMoneyInfo():void		
13: enterPaperNum	enterAccInfo()	checkLogin(id:String, pw:String):boolean		AccountManager
14: reqCharge	reqAccount()	makeAccount(id:String, pw:String):boolean		
15: reqUserInfo	makeAcc()	showBalance():int		
16: reqMoneyInfo	check()	rechargeMoney(money:int):void		
	check()	checkBalance(paper:int):boolean		
	showMessage()	print(paper:int):void		
	reqBalance()	showUserInfo():void		
	showBalance()	check(id:String, pw:String):boolean		
	getBalance()	check(id:String):boolean		
	showMessage()	getBalance():int		
	enterFee()	rechargeMoney(money:int):void	User	
	reqRecharge()	printUserInfo():User		
	rechargeMoney()	payBalance(money:int):void		
	rechargeMoney()	check(id:String, pw:String):boolean	Manager	
	showMessage()	check(id:String):boolean		
	enterSheet()	print(paper:int):void	Printer	
	reqPrint()	getPaper():int		
	checkBalance()	chargePaper(paper:int):void	GUIMain	
	getBalance()	text input		
	print()	screen output		
	payBalance()			
	print()			
	showPrint()			