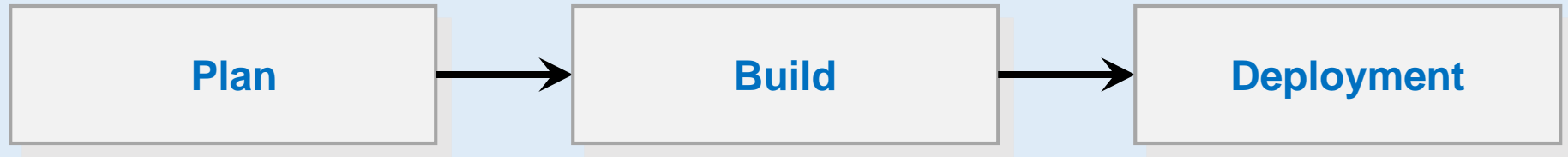


# 조별활동 참고

이번에는 슬라이드로 먼저 보여드립니다.  
지난 차수의 예제는 '정답'이 아니므로 과정을 참고해보세요.

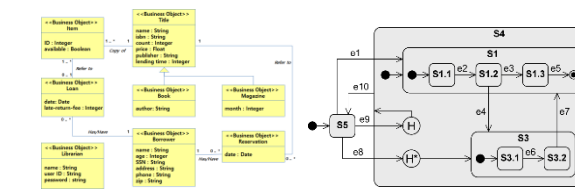
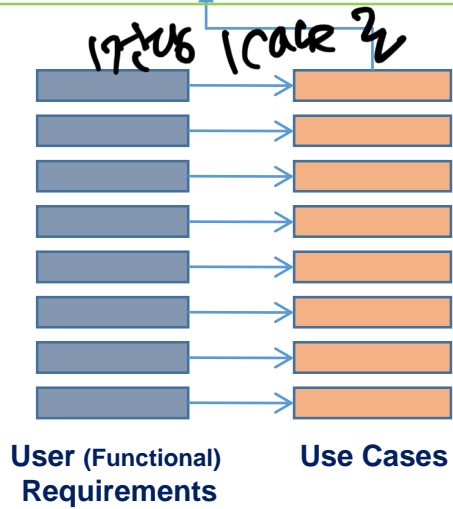
# **OOPT**

## **(Object-Oriented Process with Traceability)**



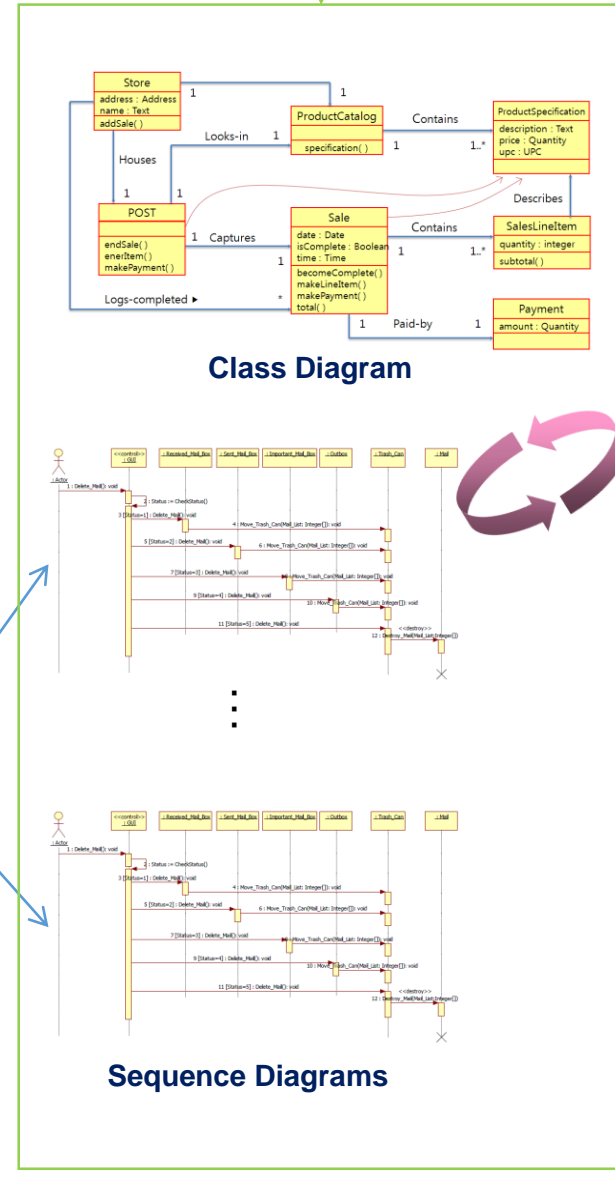
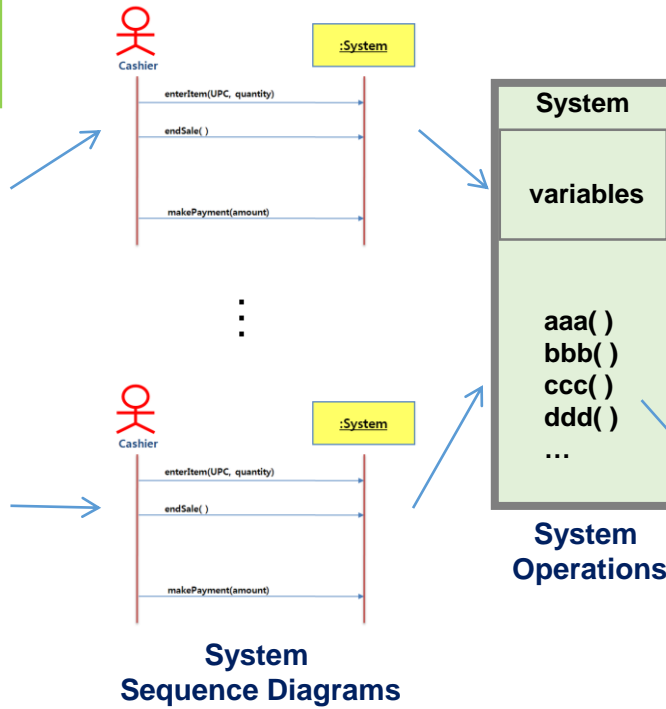
# OOPT of OOAD

Use Case Section	Comment
Use Case Name	Start with a verb.
Scope	The system under design.
Level	"user-goal" or "subfunction"
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, and worth telling the reader?
Success Guarantee	What must be true on successful completion, and worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.



## Domain Model

## State Diagram



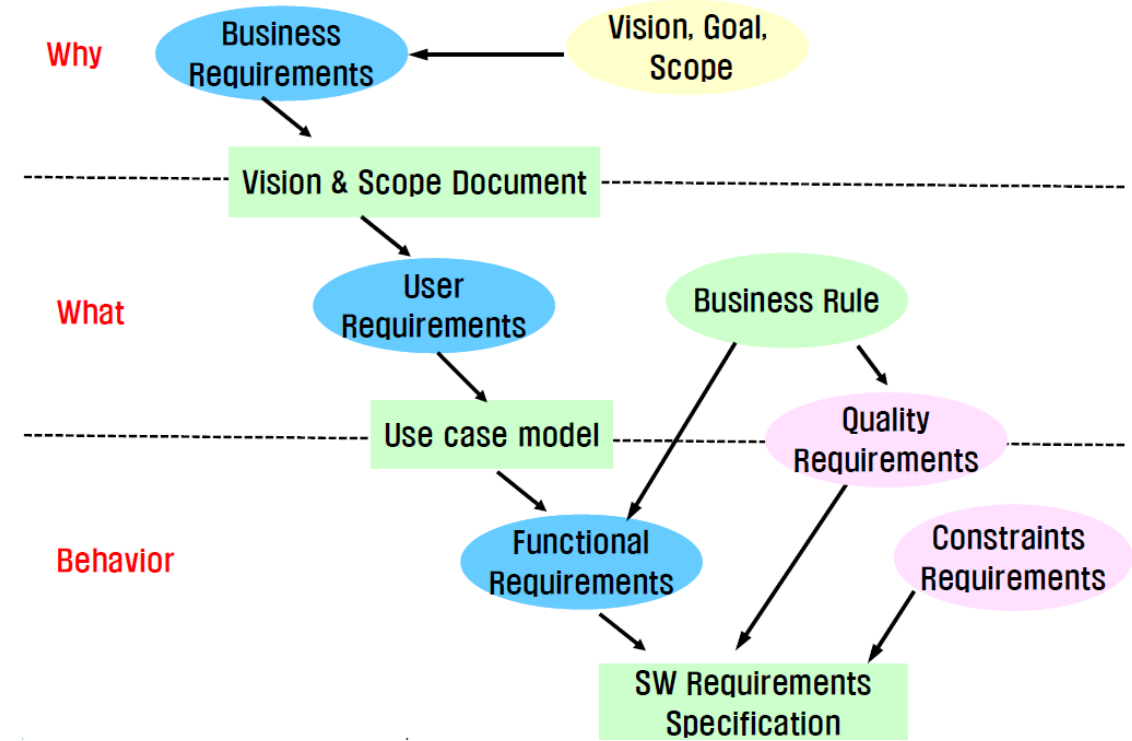
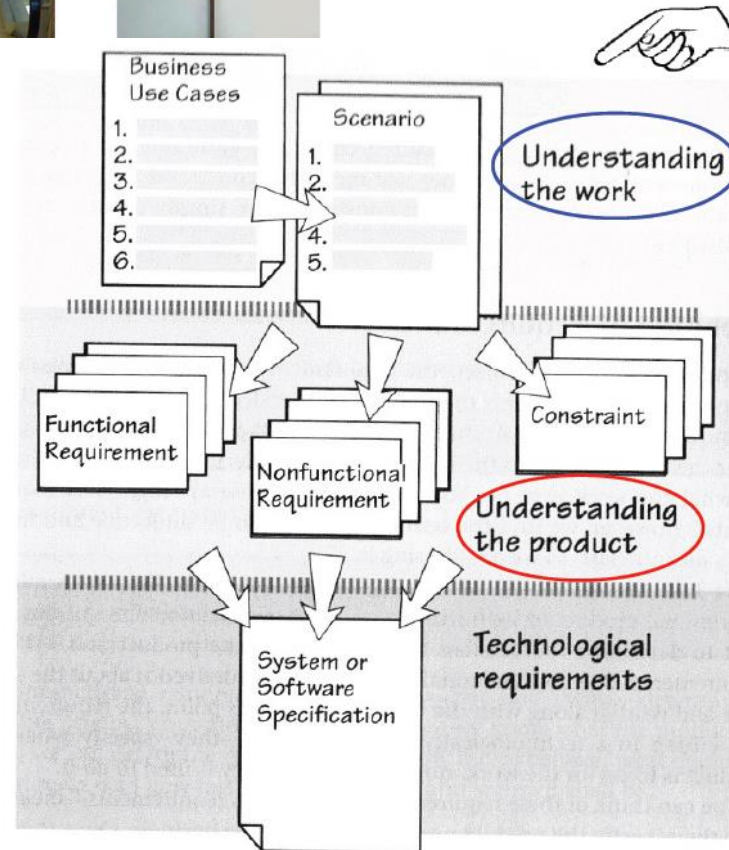
## Stage 1000. Plan

## Stage 2030. Analyze

## Stage 2040. Design



- ❑ **Business requirements**
  - Business Use Cases
- ❑ **User requirements**
  - Functional Requirements
  - Non-functional Requirements
- ❑ **System requirements**
  - System Specification
  - Sub-system specification
  - Software Requirements Specification (SRS)



<https://docs.google.com/spreadsheets/d/1bhWzDDp6eanD0kA8UIA7GSuvN-vMCXueOmTT9uR6vZo/edit#gid=0>

<https://docs.google.com/spreadsheets/d/15WvmLTtCMsiltb7INti-OHaJMGitl7VlwEwjCvcVvUA/edit#gid=629863390>

# 주제 정하고 요구사항 나열하기

## [7조] 모바일 폰을 이용한 자동차 공유시스템



손진오 1:1 채팅

2022.06.13. 17:14 조회 68

시스템 주제 : 모바일 폰을 이용한 자동차 공유시스템

### [Actor]

- 시스템, 자동차키 블루투스 모듈, 차량 임대인, 차량 임차인

### [요구사항]

- 스마트폰 어플리케이션으로 차량 키를 공유한다.
- 스마트폰 어플리케이션으로 차량 키를 대여 요청한다.
- 스마트폰 어플리케이션에 있는 대여한 차량 키를 반납한다.
- 다른 사용자의 스마트폰에 있는 차량 키를 회수한다.
- 차량 임차인은 과거에 사용했던 차량키 목록을 요청한다.
- 차량 임차인은 과거에 사용했던 차량키를 재요청한다.
- 차량은 키를 가지고 있는 사람이 다가오면 문을 열어준다.
- 차량은 키를 가지고 있는 사람이 멀어지면 문을 잠근다.
- 차량은 키를 가지고 있는 사람이 계속해서 차 안에 머무르고 있는지 체크한다.
- 사용자는 여러 대의 차량 키를 소유할 수 있다.
- 차량 임대인은 최대 5명에게 차량 키를 대여할 수 있다.
- 고객은 차량정보와 키정보를 등록한다.
- 고객은 원하는 차량을 종류, 사이즈 등으로 차량키를 검색한다.
- 시스템은 사용자가 원하는 정보를 기반으로 결과를 보여준다.
- 고객은 검색결과에서 원하는 차량 키를 요청한다.
- 시스템은 고객이 요청한 차량키의 대여 정보를 업데이트한다.
- 고객은 대여한 차량키를 반납한다.
- 고객은 자신이 등록한 차량키를 반납요청할 수 있다.

시스템 목적

지하철역 도서거래시스템

지하철역 사물함을 이용해 사용자 간 책을 거래할 수 있는 플랫폼을 제공한다. (당근마켓st)

Actor

시스템 : UI를 가진 책 중개 어플리케이션

판매자 : 책을 판매하려는 사람

구매자 : 책을 구매하려는 사람

시스템 기능

1. 판매자/구매자는 시스템에 회원가입하고 로그인 한다.
  - 1.1 시스템은 구매자의 로그인 정보를 확인하고 승인한다.
2. 판매자가 시스템에 본인이 소장중인 책을 등록한다.
  - 2.1 판매자가 책의 가격, 제목, 지하철 위치, 지하철 사물함 번호, 비밀번호를 시스템에 등록한다.
3. 구매자는 시스템을 통해 책 제목을 검색할 수 있다.
  - 3.1 시스템은 책의 위치와 가격 정보를 관리한다.
  - 3.2 구매자는 원하는 책제목을 시스템 검색창에 입력해서 찾는다.
  - 3.3 시스템은 검색된 책이 보관된 위치와 가격 정보를 목록으로 제공한다.
4. 구매자는 원하는 책을 시스템에서 결제를 한다.
  - 4.1 구매자는 시스템에 충전된 금액으로 결제한다.]
    - 4.1.1 구매자는 시스템에 금액을 충전할 수 있다.
    - 4.1.2 시스템은 충전된 금액을 구매자에게 반환해 줄 수 있다.
  - 4.2 충전금액이 충분하지 않다면 거래 불가 메시지를 보여준다.
5. 시스템은 결제된 대금을 판매자에게 전달한다.
  - 5.1 시스템은 해당 거래의 금액을 판매자의 계정으로 충전한다.
  - 5.2 판매자가 시스템에 본인 계정의 금액을 반환요청할 수 있다.
  - 5.3 시스템은 판매자에게 요청금액만큼 반환한다.

주어는 명확히,  
서술어는 능동태로  
문장은 간결히 작성합니다.

무엇을 어떻게 한다.

요구사항 단계 이식.

## [7조] 모바일 폰을 이용한 자동차 공유시스템



손진오 1:1 채팅

2022.06.13. 17:14 조회 68

시스템 주제 : 모바일 폰을 이용한 자동차 공유시스템

### [Actor]

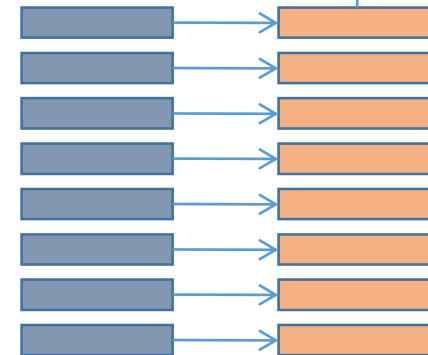
- 시스템, 자동차키 블루투스 모듈, 차량 임대인, 차량 임차인

### [요구사항]

- 스마트폰 어플리케이션으로 차량 키를 공유한다.
- 스마트폰 어플리케이션으로 차량 키를 대여 요청한다.
- 스마트폰 어플리케이션에 있는 대여한 차량 키를 반납한다.
- 다른 사용자의 스마트폰에 있는 차량 키를 회수한다.
- 차량 임차인은 과거에 사용했던 차량키 목록을 요청한다.
- 차량 임차인은 과거에 사용했던 차량키를 재요청한다.
- 차량은 키를 가지고 있는 사람이 다가오면 문을 열어준다.
- 차량은 키를 가지고 있는 사람이 멀어지면 문을 잠근다.
- 차량은 키를 가지고 있는 사람이 계속해서 차 안에 머무르고 있는지 체크한다.
- 사용자는 여러 대의 차량 키를 소유할 수 있다.
- 차량 임대인은 최대 5명에게 차량 키를 대여할 수 있다.
- 고객은 차량정보와 키정보를 등록한다.
- 고객은 원하는 차량을 종류, 사이즈 등으로 차량키를 검색한다.
- 시스템은 사용자가 원하는 정보를 기반으로 결과를 보여준다.
- 고객은 검색결과에서 원하는 차량 키를 요청한다.
- 시스템은 고객이 요청한 차량키의 대여 정보를 업데이트한다.
- 고객은 대여한 차량키를 반납한다.
- 고객은 자신이 등록한 차량키를 반납요청할 수 있다.



Use Case Section	Comment
Use Case Name	Start with a verb.
Scope	The system under design.
Level	"user-goal" or "subfunction"
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, and worth telling the reader?
Success Guarantee	What must be true on successful completion, and worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.



User (Functional)  
Requirements

Use Cases



# 요구사항 분석하기1 - case와 주요기능 도출

## [요구사항]

### [회원 가입]

차량 소유자 또는 대여자는 핸드폰 번호로 가입한다.

차량 소유자 또는 대여자는 회원탈퇴를 즉시 할 수 있다.

- 차량 소유자가 탈퇴하면, 해당 소유자가 공유한 디지털 차량 키들은 삭제된다.

### [차량 등록]

차량 소유자는 차량정보와 키정보를 등록한다.

차량 소유자는 최대 5명에게 디지털 차량키를 공유해줄 수 있다.

### [차량 삭제]

차량 소유자는 본인이 관리하는 디지털 차량키를 삭제할 수 있다.

### [차량 관리]

차량 소유자는 본인이 관리하는 디지털 차량키 목록을 볼 수 있다.

차량 대여자는 자신이 빌렸던 차량의 목록을 확인할 수 있다.

### [기본 기능 - 차량 잠금 및 해제]

권한이 있는 사람은 어플리케이션으로 직접 문을 잠그거나 열수있다.

차량은 권한이 있는 사람이 다가오면 문을 열어준다.

차량은 권한이 있는 사람이 멀어지면 문을 잠근다.

### [대여 / 회수 - 차량 소유자]

차량 소유자는 스마트폰 어플리케이션으로 차량 대여자에게 디지털 차량키를 공유한다.

- 차량 소유자는 요청이 오면 디지털 차량키의 사용권한을 수락/거절한다.
- 차량 소유자는 차량 대여자의 요청이 없이 디지털 차량키를 공유할 수 있다.
- 차량 소유자는 차량 대여자의 핸드폰 번호로 디지털 차량키를 공유할 수 있다.
- 차량 소유자는 차량 대여자의 디지털 차량키를 회수한다.

### [대여 / 반납 - 차량 대여자]

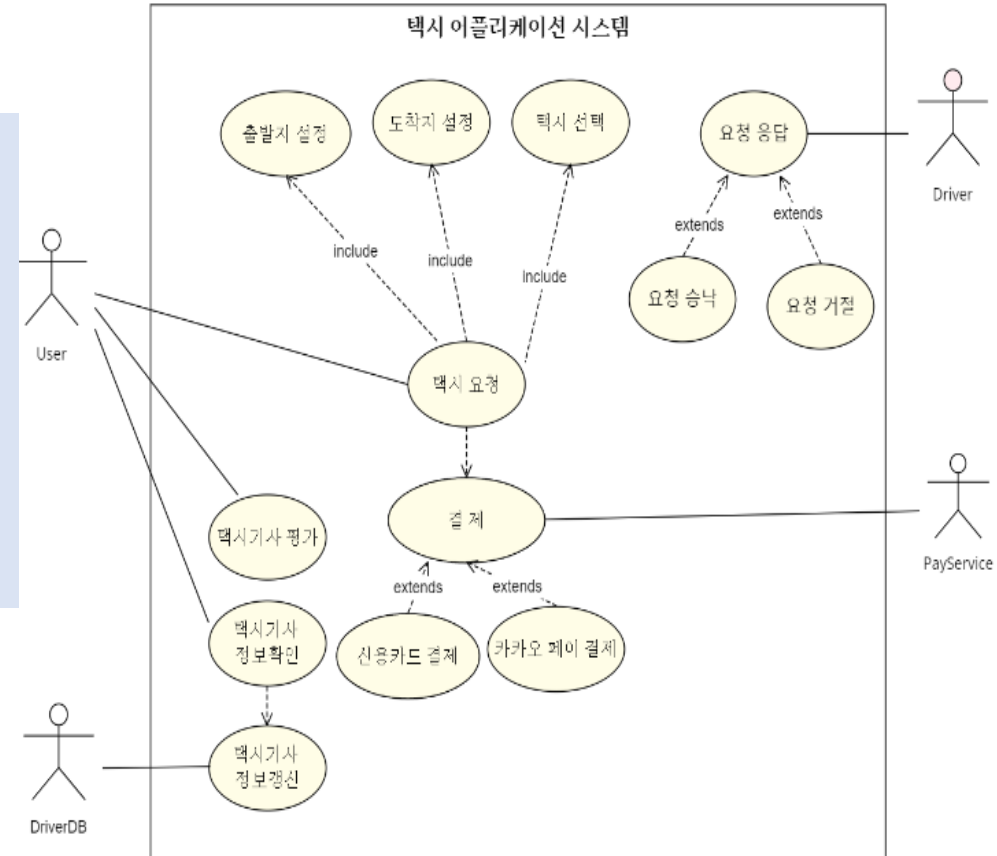
차량 대여자는 여러 대의 디지털 차량키를 소유할 수 있다.

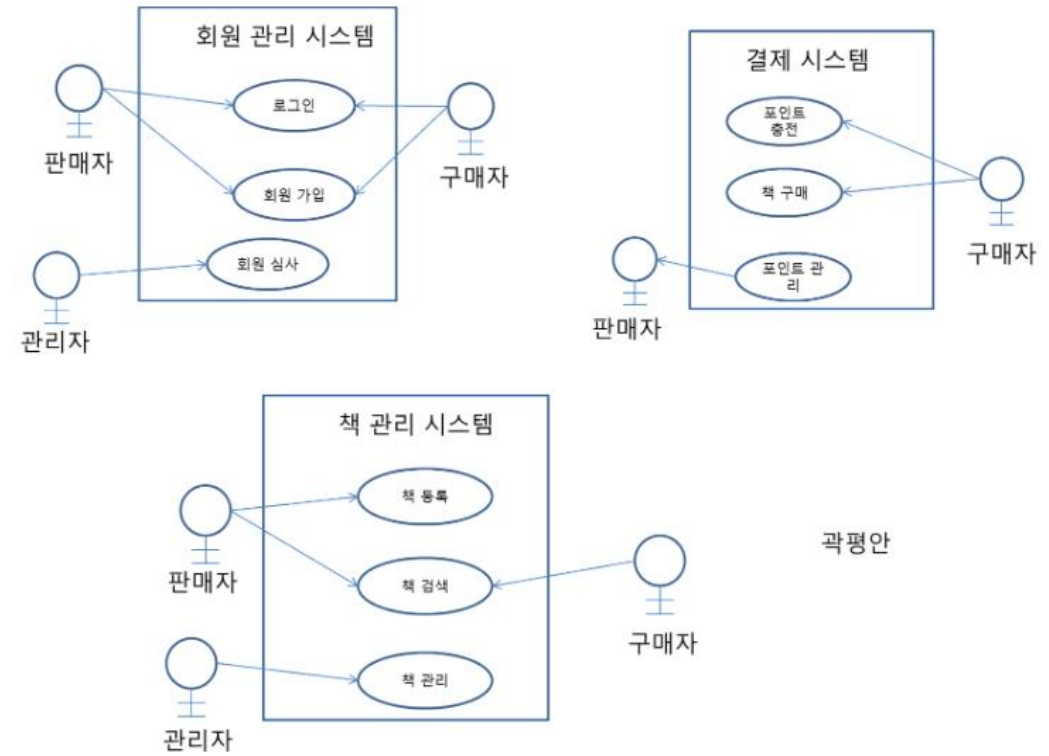
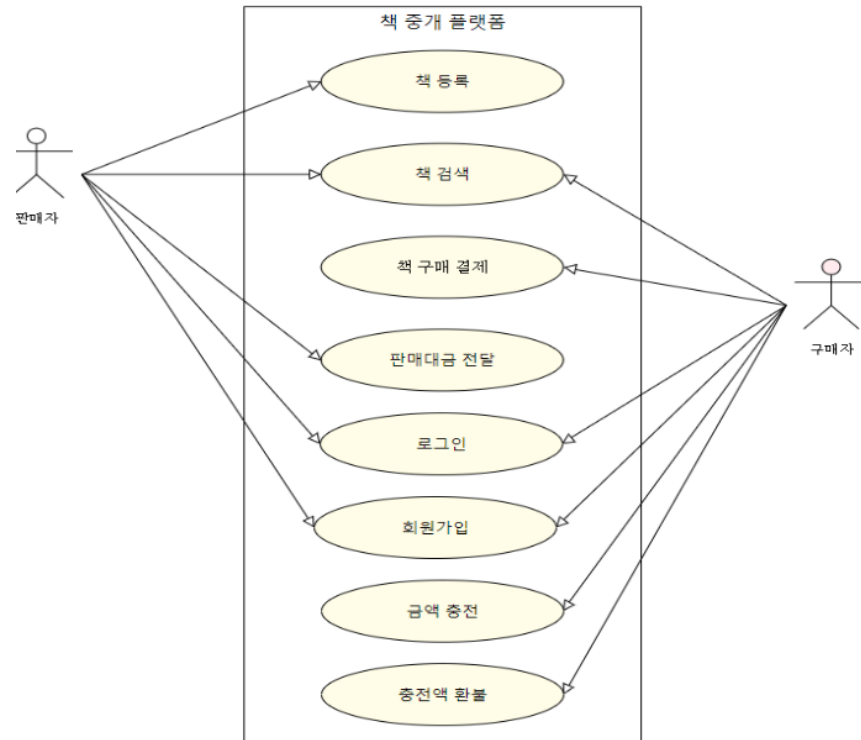
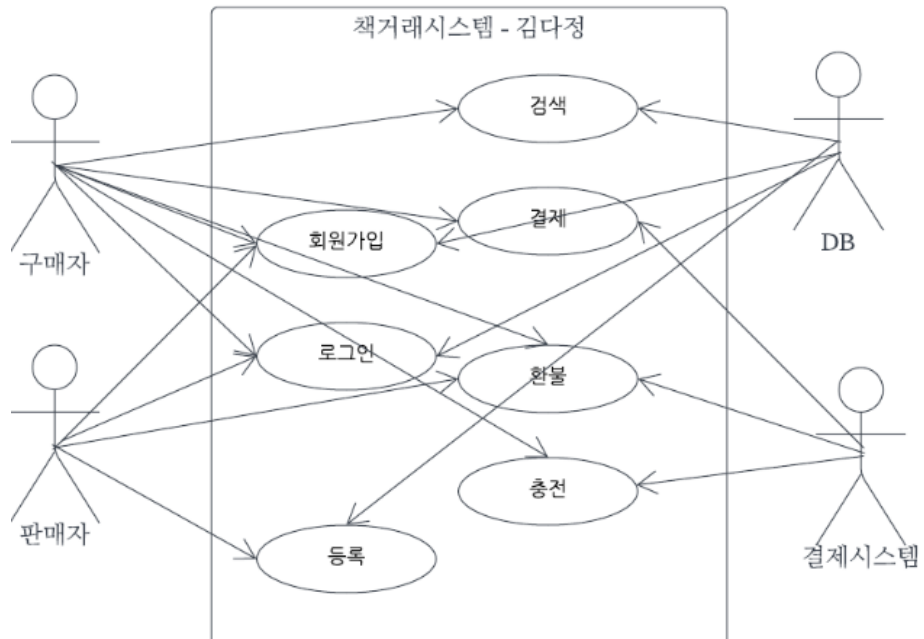
차량 대여자는 차량 키를 반납할 수 있다.

차량 대여자는 본인이 사용가능한 디지털 차량키 목록을 볼 수 있다.

차량 대여자는 이전에 빌렸던 이력이 있는 경우 차량 소유자에게 차량공유를 요청할 수 있다.

Use case를 도출하면서  
다이어그램을 그려볼 수도 있습니다.  
좌측표에서는 "회원가입", "차량등록",  
"차량삭제" 등이 case가 되며,  
가장 중요한 기능인  
"차량공유하기" 기능이 추가되어야  
합니다.





처음 그려본 그림으론 훌륭합니다!  
다음 반복을 통해 수정해가면 됩니다!

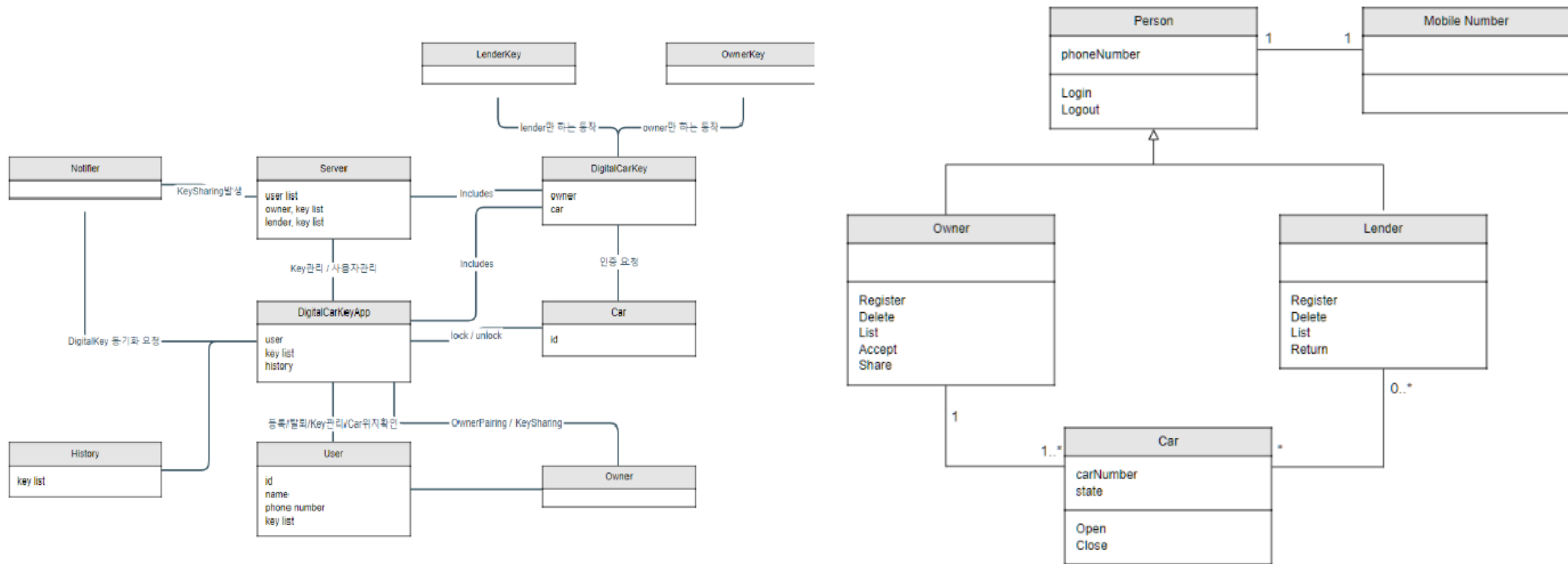


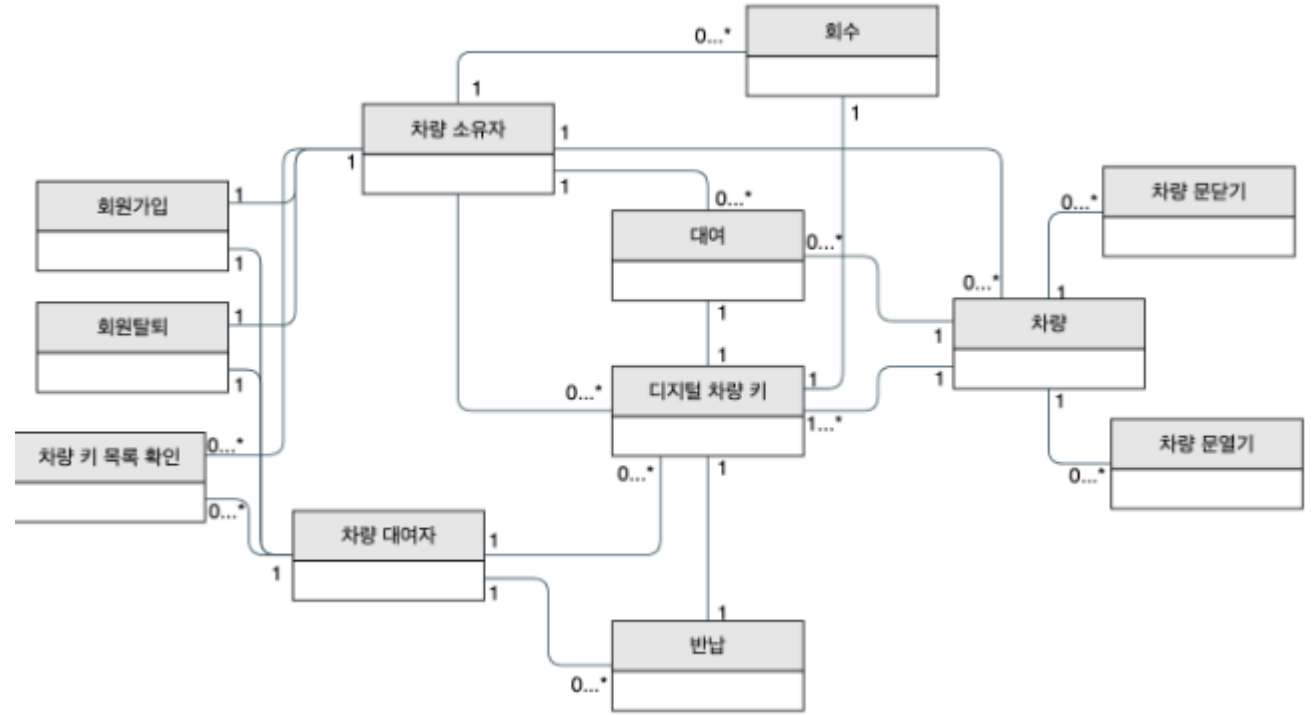
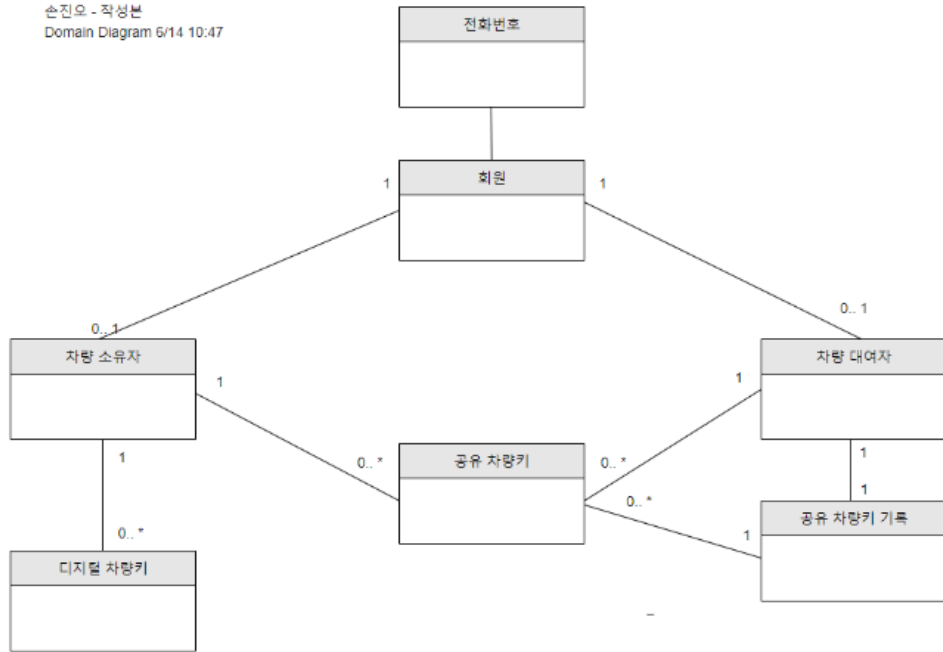
# 요구사항 분석하기2 :case 별 상세스펙 작성

Use Case Section	Comment
Use Case Name	Start with a verb.
Scope	The system under design.
Level	"user-goal" or "subfunction"
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, <i>and</i> worth telling the reader?
Success Guarantee	What must be true on successful completion, <i>and</i> worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.

Use Case Name	핸드폰 번호로 차량 공유해주기
Scope	차량 대여 Application
Level	user goal
Primary Actor	차량 소유자
Stakeholders and Interests	차량 소유자 : 차량 : 차량 소유자가 가지고 있는 차량으로 차량 대여자에게 빌려 줄 차량 차량 대여자 : 핸드폰 번호를 가지고 있고, 차량 소유자로부터 차량의 디지털 키를 공유 받을 사람. 핸드폰 번호 : 차량 소유자가 차량 대여자에게 핸드폰 번호를 입력하여 차량을 대여할 수 있도록 함. 차량 디지털 키 : 차량을 제어할 수 있는 키.
Preconditions	차량 소유자와 차량은 시스템에 등록되어있다.!
Success Guarantee	시스템에서 대여 처리 완료
Main Success Scenario	<ol style="list-style-type: none"> <li>1. 차량 소유자는 Application에 접속 및 로그인을 한다.</li> <li>2. 차량 소유자는 자신이 소유한 차량 중 대여하고자 하는 차량을 선택한다.</li> <li>3. 차량 소유자는 대여해주기 버튼을 클릭한다.</li> <li>4. 차량 대여자의 핸드폰 번호를 입력 후 확인 버튼을 누른다.</li> <li>5. 시스템은 차량 이용 권한을 정상적으로 전송했다는 메시지와 차량 대여자 정보를 차량 소유자에게 보여준다.</li> <li>6. 시스템은 차량별 대여자 목록을 업데이트하여 차량 소유자에게 보여준다.</li> </ol>
Extension	( 핸드폰 번호가 회원일 경우 ) 4a. 차량 디지털 키가 차량 대여자에게 전송된다. (핸드폰 번호가 회원이 아닐 경우) 4a. 차량 소유자에게 회원이 아니라는 알람을 띄운다.
Special Requirements	대여는 365일 24시간 가능해야 한다. 사용자들은 인터넷에 연결되어있는 상황이어야 한다.
Technology and Data Variations List	-
Frequency of Occurrence	항상 발생할 수 있음.
Miscellaneous	-

# 도메인 개념을 나타내볼까요?



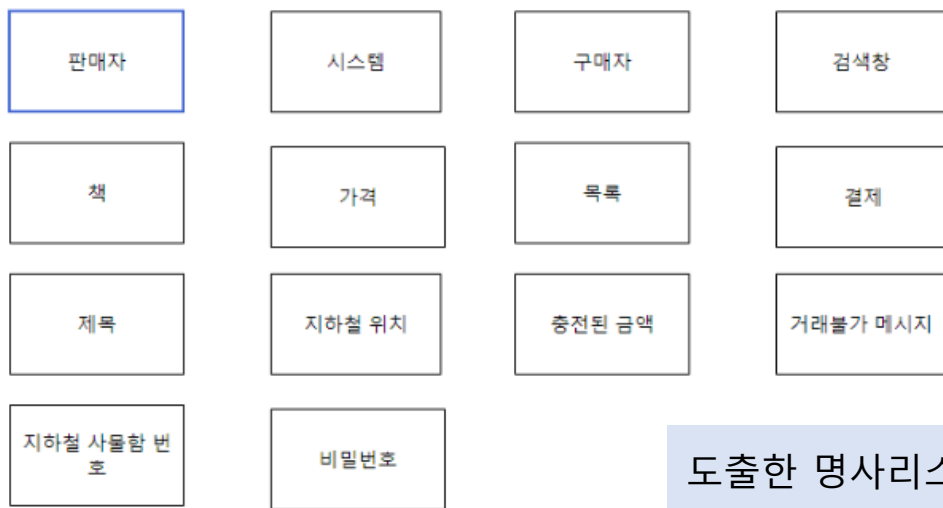


도메인모델은 클래스 다이어그램의 기초가 되는 모델이며, text의 명사들이 주 후보가 됩니다.

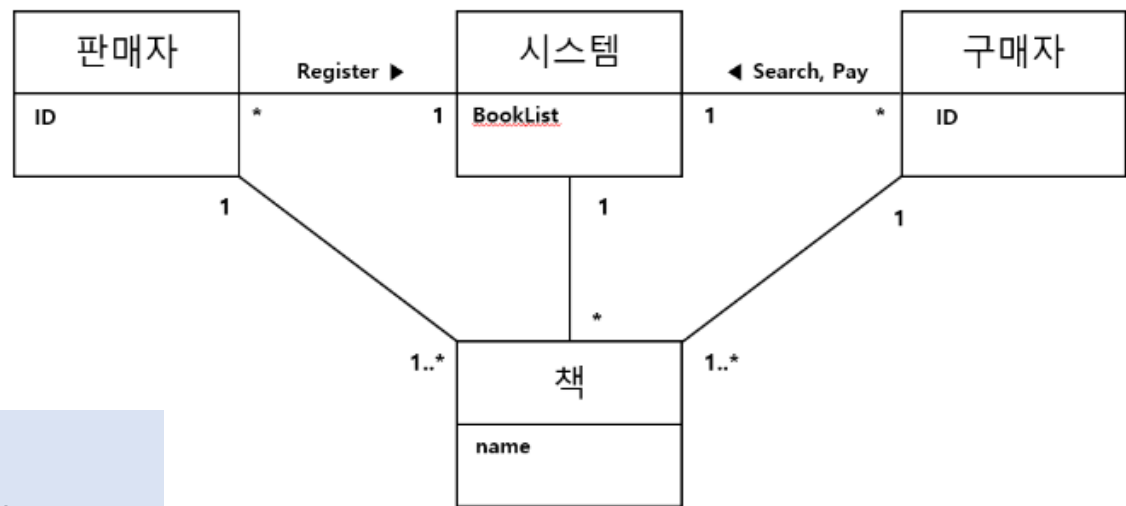
“차량문열기”는 기능으로서 메소드로 구현하는 것이 적합하므로 추후 분석을 통해 할당되어야 할 클래스를 찾아 배치합니다.

중요한 개념들이 모두 고려되었는지, 누락된 것은 없는지 관계정도만 표현해둡니다.

단, 클래스로서 존재하지 않아도 될 것들은 삭제합니다.



도출한 명사리스트  
사물함의 개념은 있어야 할까요?



#### [내부 토의 결과]

달랐던 점 :

사물함 <-> 위치

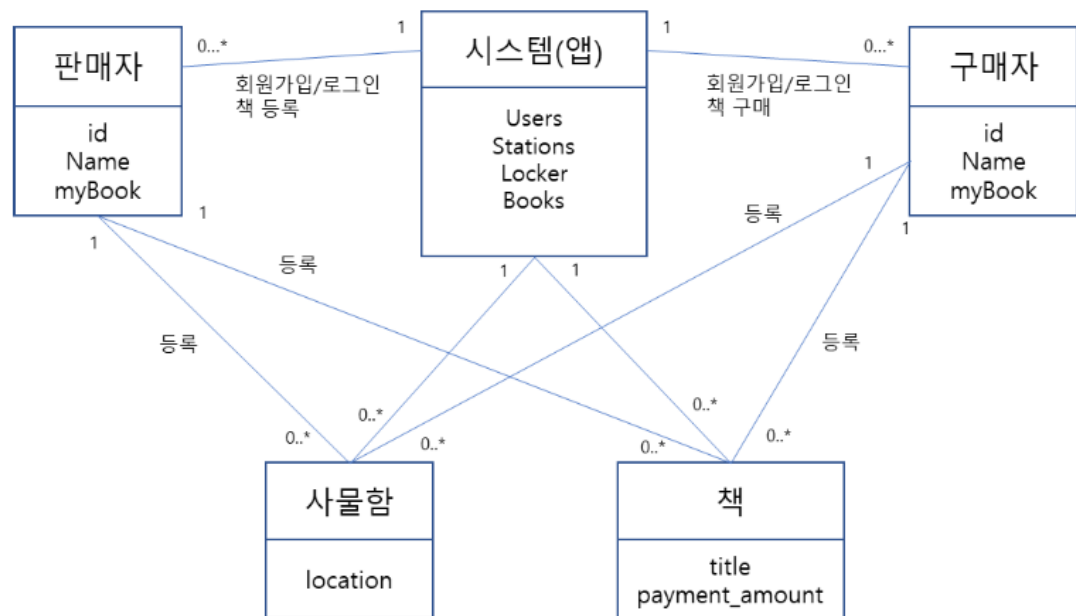
결제 도메인이 있는 사람 / 없는 사람

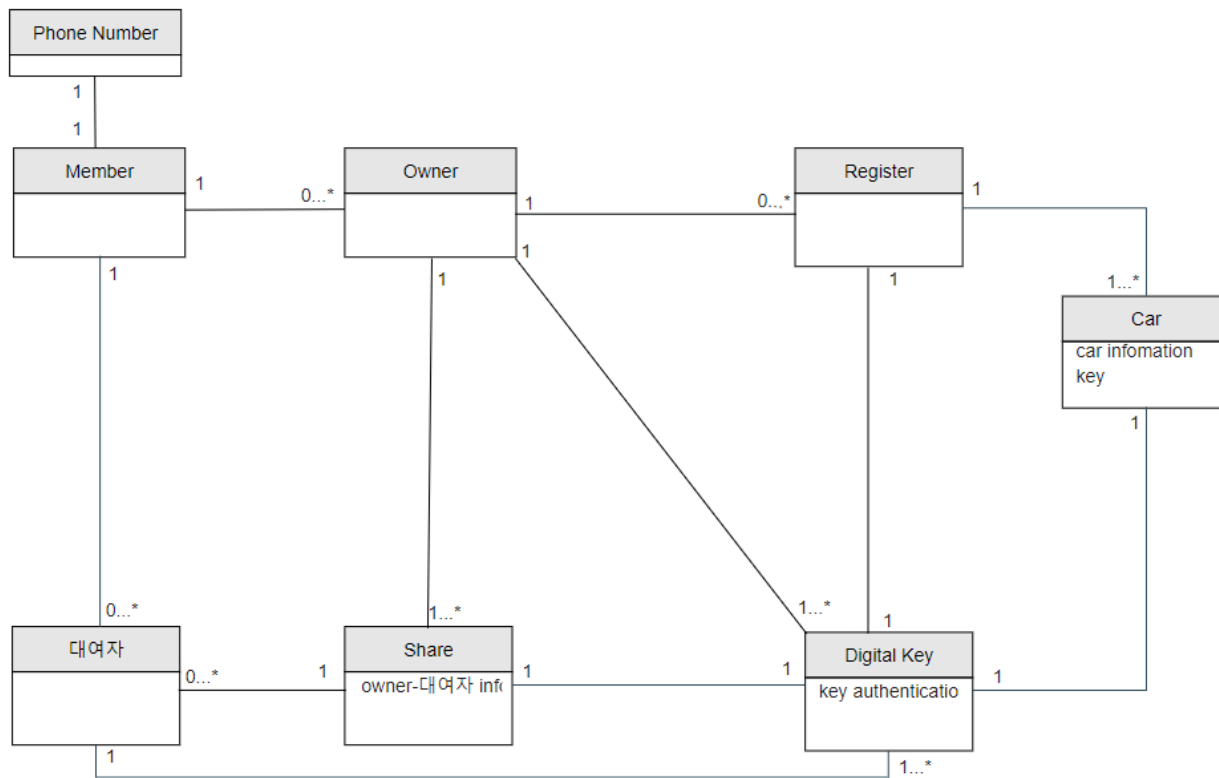
책에 위치 정보를 같이

개발 범위 내 외 를 구분

#### [교수님 FEEDBACK]

1. 시스템이라는 단어가 모호 => Traid Book, TB Server?
2. 관리자 추가 고려
3. 어느정도 복잡도 추가 필요
4. 디비 신용정보등은 외부 액터
5. 따라서 fully dressed 해야한다





앞서 개별적으로 그린 도메인모델을 잠시 함께 review한 결과 위 모델로 정리되어 가고 있습니다.  
물론 아직 수정할 부분은 더 보입니다.

\*\* 평소 바로 코딩에 이르는 멤버가 많아서, 각자 종류별로 하나이상의 그림을 , 많이 그려보기가 핵심이었습니다.  
또는 정적 모델링이 낯선 분들께서는 2인1조가 되어 서로 함께 그리는 방법을 권합니다

# Fully Dressedup Form

Use Case Name	핸드폰 번호로 차량 공유해주기
Scope	차량 대여 Application
Level	user goal
Primary Actor	차량 소유자
Stakeholders and Interests	차량 소유자 : 차량 : 차량 소유자가 가지고 있는 차량으로 차량 대여자에게 빌려 줄 차량 차량 대여자 : 핸드폰 번호를 가지고 있고, 차량 소유자로부터 차량의 디지털 키를 공유 받을 사람. 핸드폰 번호 : 차량 소유자가 차량 대여자에게 핸드폰 번호를 입력하여 차량을 대여할 수 있도록 함. 차량 디지털 키 : 차량을 제어할 수 있는 키.
Preconditions	차량 소유자와 차량은 시스템에 등록되어있다.
Success Guarantee	시스템에서 대여 처리 완료
Main Success Scenario	<ol style="list-style-type: none"> <li>1. 차량 소유자는 Application에 접속 및 로그인을 한다.</li> <li>2. 차량 소유자는 자신이 소유한 차량 중 대여하고자 하는 차량을 선택한다.</li> <li>3. 차량 소유자는 대여해주기 버튼을 클릭한다.</li> <li>4. 차량 대여자의 핸드폰 번호를 입력 후 확인 버튼을 누른다.</li> <li>5. 시스템은 차량 이용 권한을 정상적으로 전송했다는 메시지와 차량 대여자 정보를 차량 소유자에게 보여준다.</li> <li>6. 시스템은 차량별 대여자 목록을 업데이트하여 차량 소유자에게 보여준다.</li> </ol>
Extension	(핸드폰 번호가 회원일 경우) 4a. 차량 디지털 키가 차량 대여자에게 전송된다. (핸드폰 번호가 회원이 아닐 경우) 4a. 차량 소유자에게 회원이 아니라는 알람을 띄운다.
Special Requirements	대여는 365일 24시간 가능해야 한다. 사용자들은 인터넷에 연결되어있는 상황이어야 한다.
Technology and Data Variations List	-
Frequency of Occurrence	항상 발생할 수 있음.
Miscellaneous	-

Use Case Name	대여한 차량의 공유 취소하기
Scope	차량 대여 Application
Level	user goal
Primary Actor	차량 소유자
Stakeholders and Interests	차량 소유자 : 차량 대여자가 대여한 차량을 소유한 사람 차량 대여자 : 차량 소유자로 부터 차량을 공유받은 사람. 차량 : 차량 대여자가 차량 소유자로부터 공유받은 차량 차량 디지털 키 : 차량을 제어할 수 있는 키.
Preconditions	차량 소유와 차량은 시스템에 등록되어있다. 차량 대여자는 이전에 차량을 대여한 기록이 있다. 차량 대여자는 현재 해당 차량을 사용할 수 없는 상태 현재도 차량 소유자는 해당 차량을 소유하고 있다.
Success Guarantee	시스템에서 대여 처리 완료.
Main Success Scenario	<ol style="list-style-type: none"> <li>1. 차량 소유자는 Application에 접속 및 로그인을 한다.</li> <li>2. 차량 소유자는 자신이 공유한 차량 목록을 확인한다.</li> <li>3. 차량 소유자는 특정 차량 대여자를 선택 후 "공유 취소" 버튼을 클릭한다.</li> <li>4. 시스템은 차량 공유가 정상적으로 취소 됐다는 메시지와 차량 대여자 정보를 차량 소유자에게 보여준다.</li> <li>5. 시스템은 차량별 대여자 목록을 업데이트하여 차량 소유자에게 보여준다.</li> </ol>
Extension	(차량의 상태가 이용 중인 경우) 4a. 차량이 현재 이용중이므로 회수 예정이라는 알람을 띄운다.
Special Requirements	대여는 365일 24시간 가능해야 한다. 사용자들은 인터넷에 연결되어있는 상황이어야 한다.
Technology and Data Variations List	
Frequency of Occurrence	
Miscellaneous	

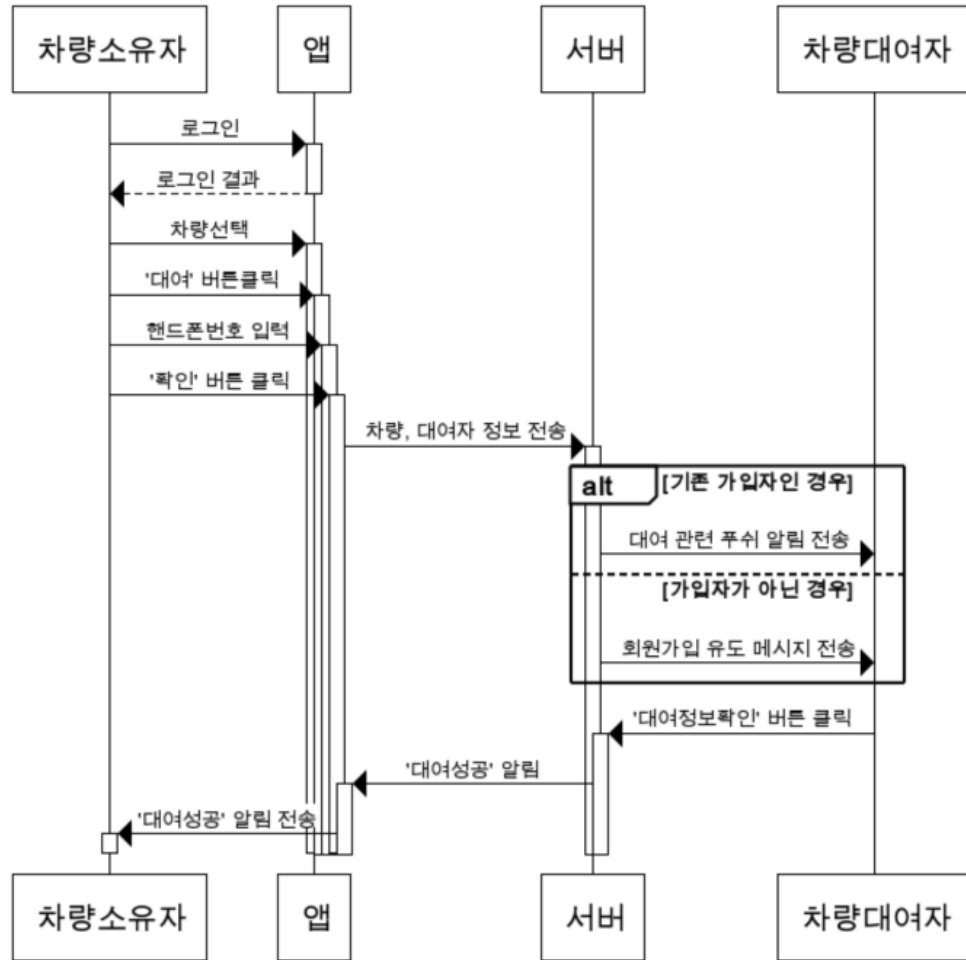
몇 건의 case에 대해서는 아주 상세한 정보들을 파악 해봅니다.

이 과정을 통해 해당시스템의 주요기능에 대한 이해도를 높일 수 있으며, 각 독립적인 case는 하나의 시퀀스 프로세스가 될 수 있습니다.

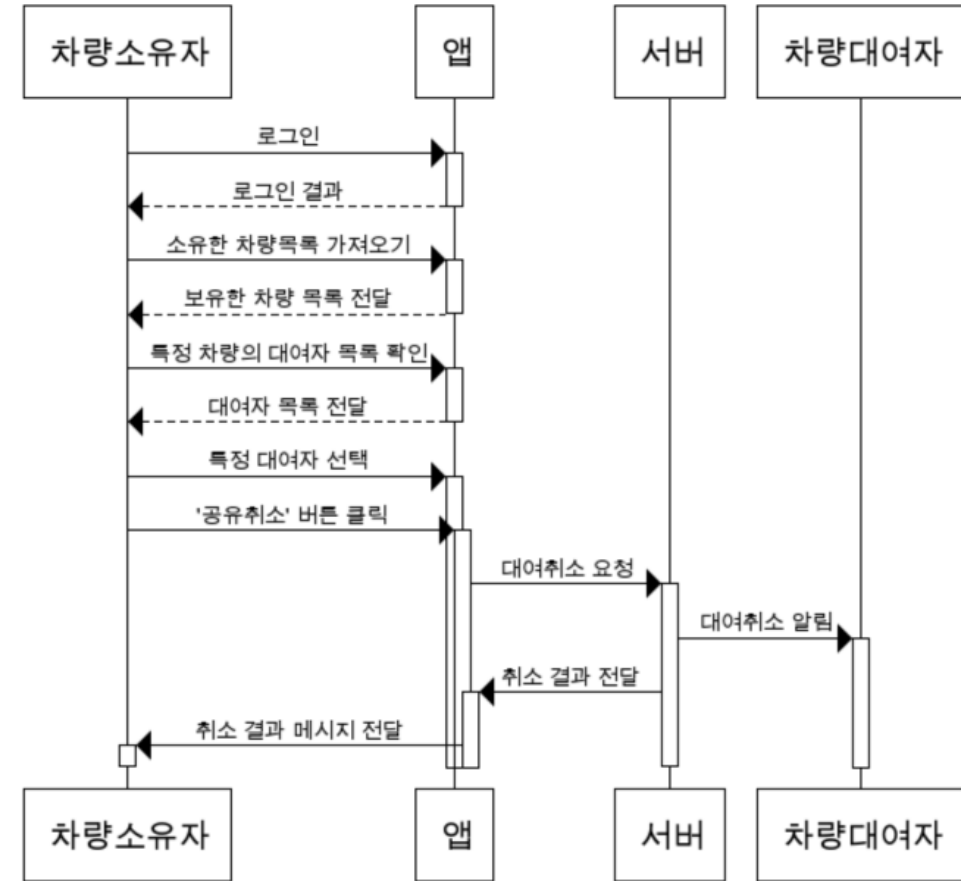


# 주요기능은 시퀀스 다이어그램으로 그려봅니다

핸드폰 번호로 차량 공유해주기

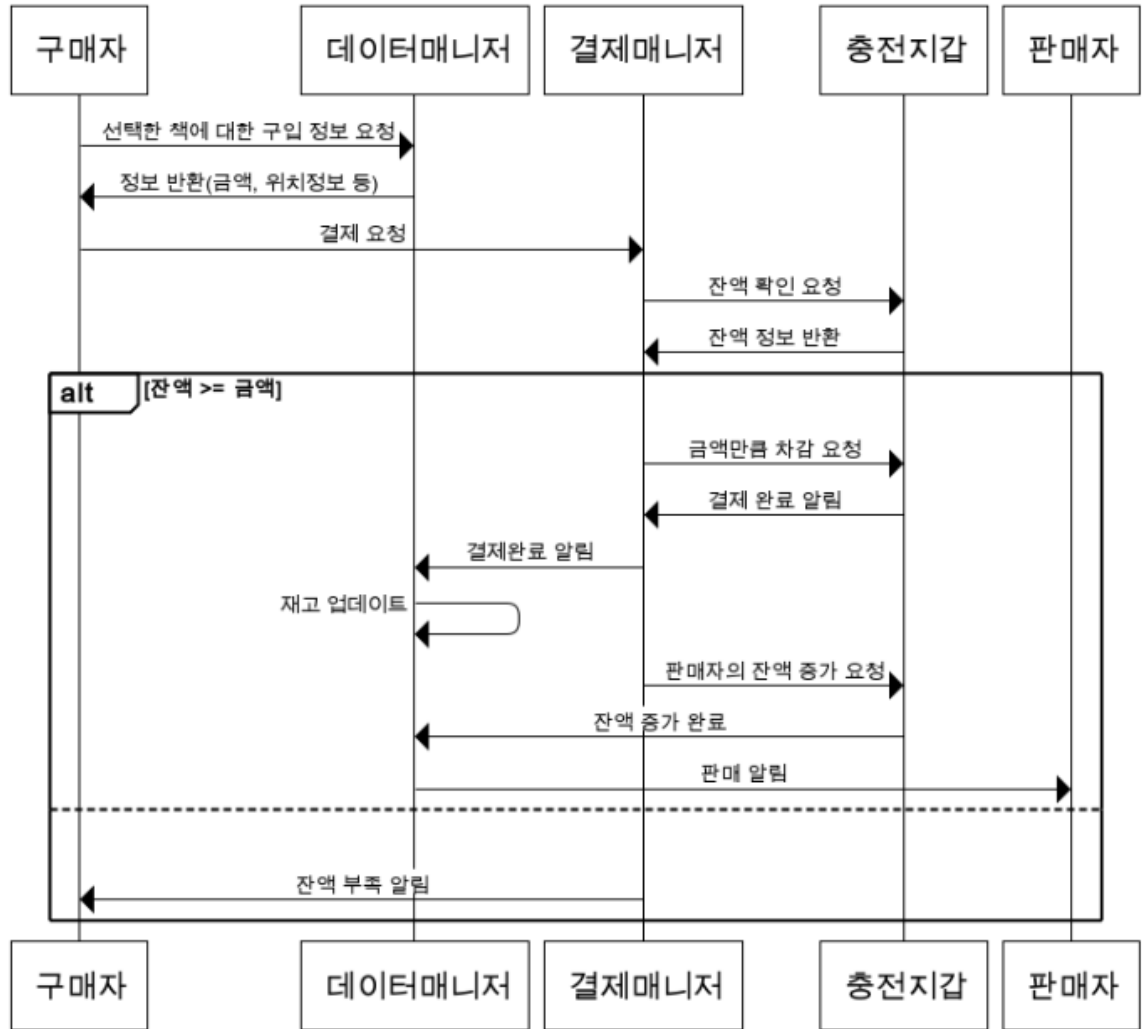


차량 공유 취소



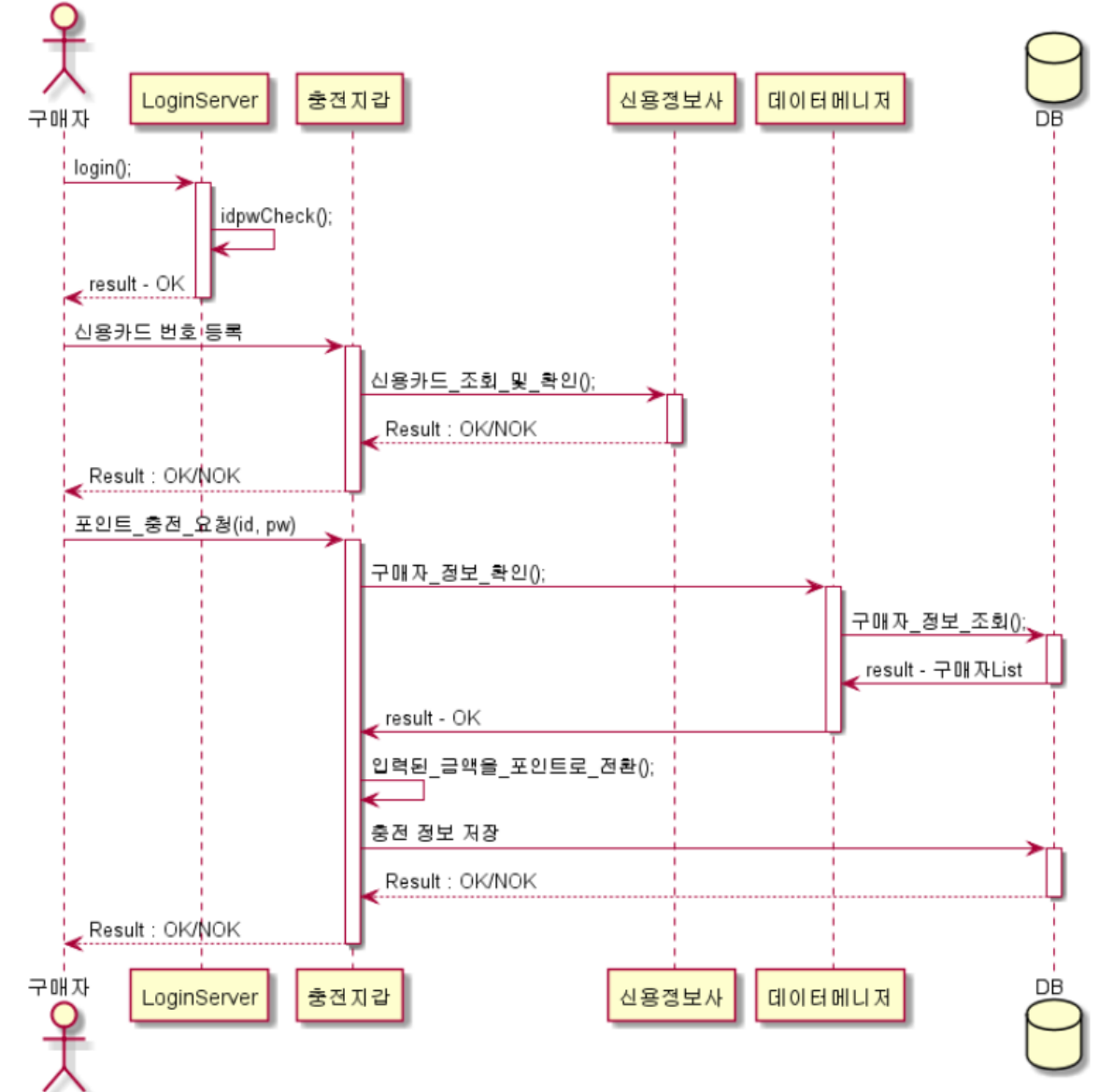
시퀀스다이어그램을 그릴 때는 코딩하는 마음으로 시작하면서! ....발전시킵니다.  
실제로 이 그림을 보면서 메소드를 완성해갑니다.

## 결제하기



www.websequencediagrams.com

## 곽평안 : 금액충전



같은 프로젝트일 경우에는 표기방법과 스타일을 일치시켜도 좋습니다.

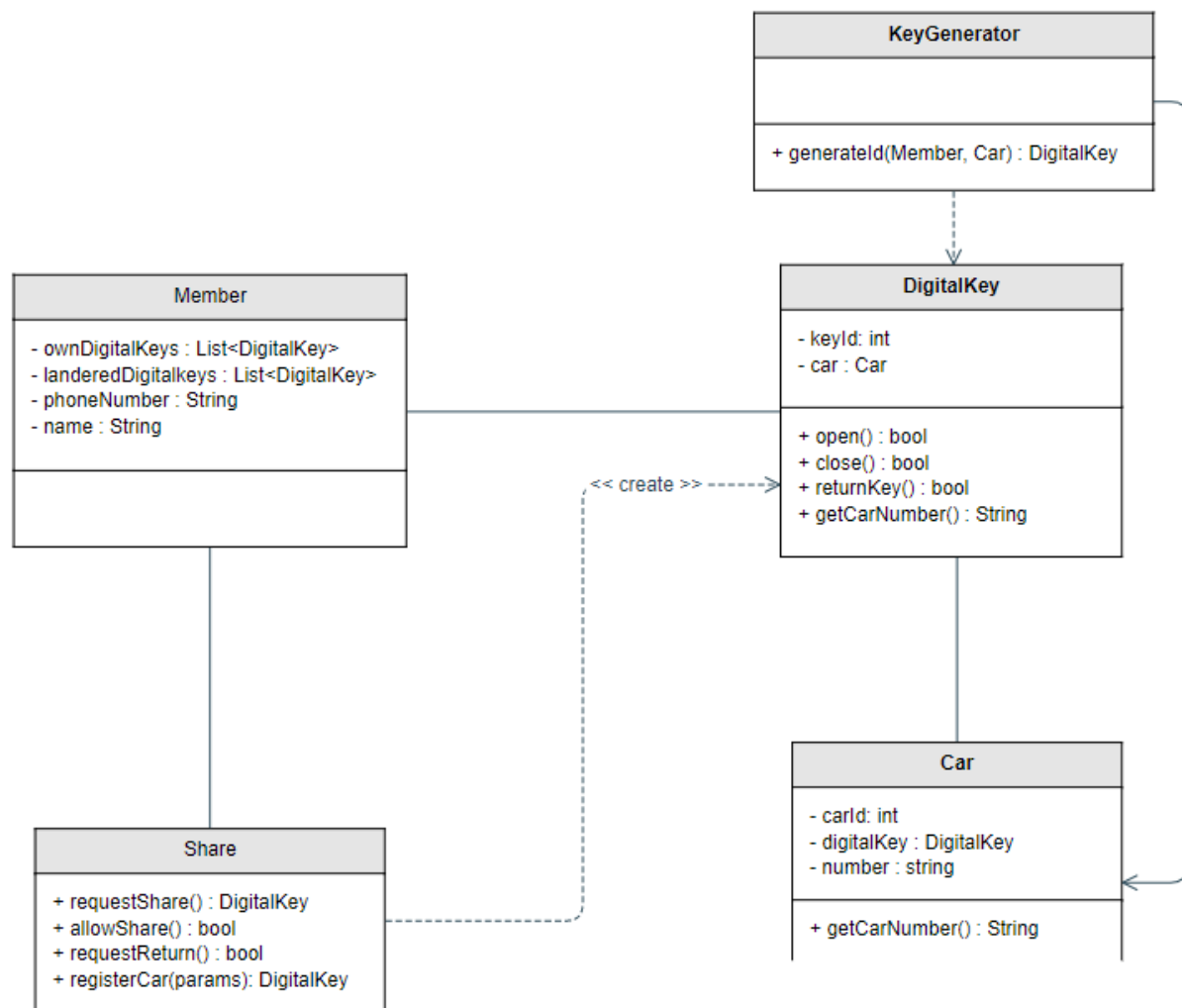


[대표적인 오류]

전체 시퀀스를 한 페이지에  
그리지 않습니다.

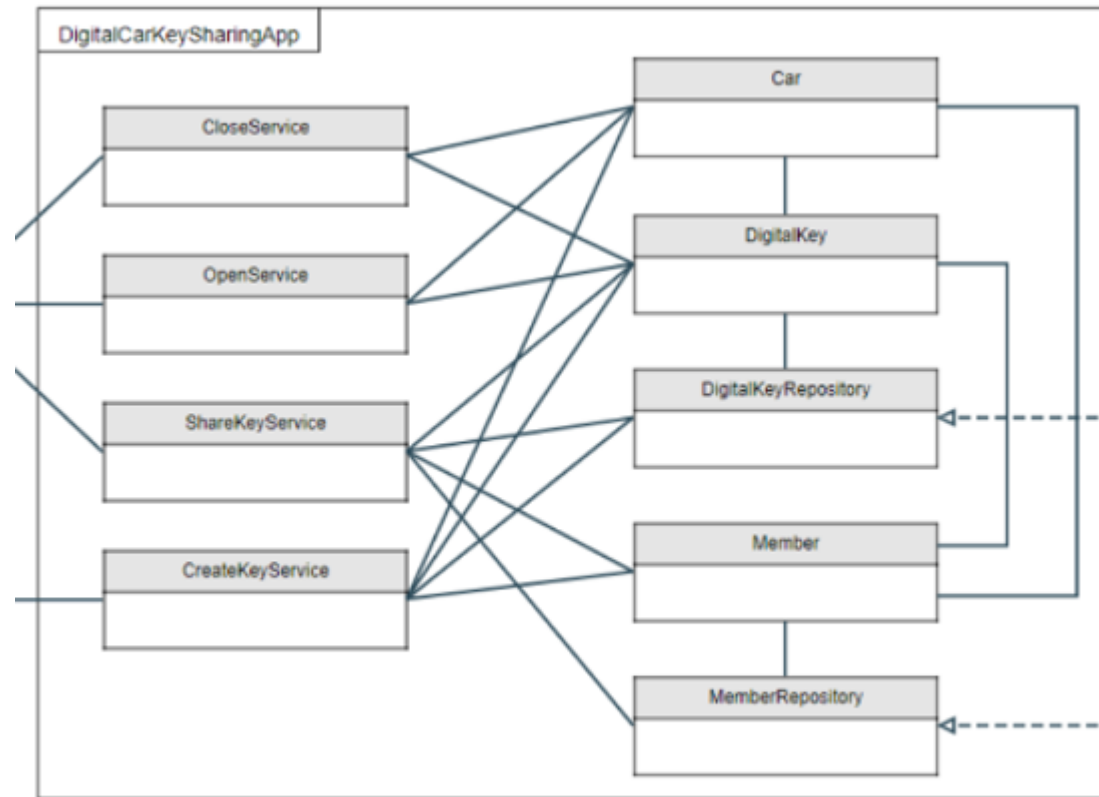
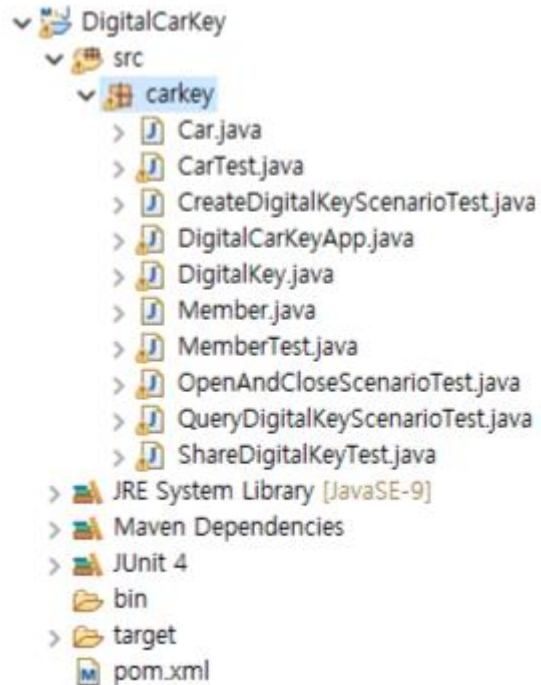
Case에서 도출된 정보로 단위  
메소드(함수)를 구현한다는  
생각으로, 별도로 그리는 것  
이 좋습니다.

# 도메인모델을 바탕으로 클래스들의 윤곽을 잡아봅시다.

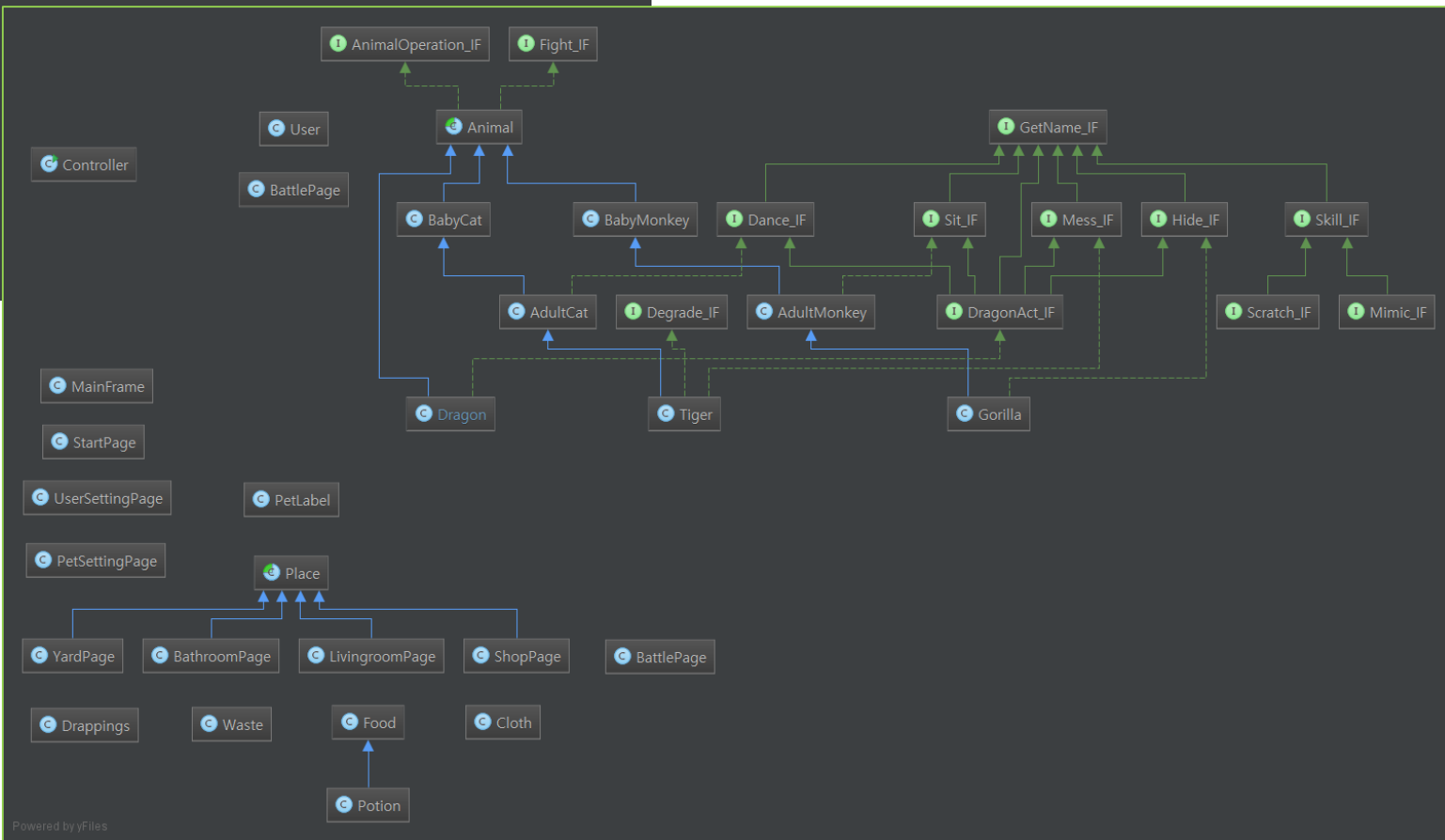
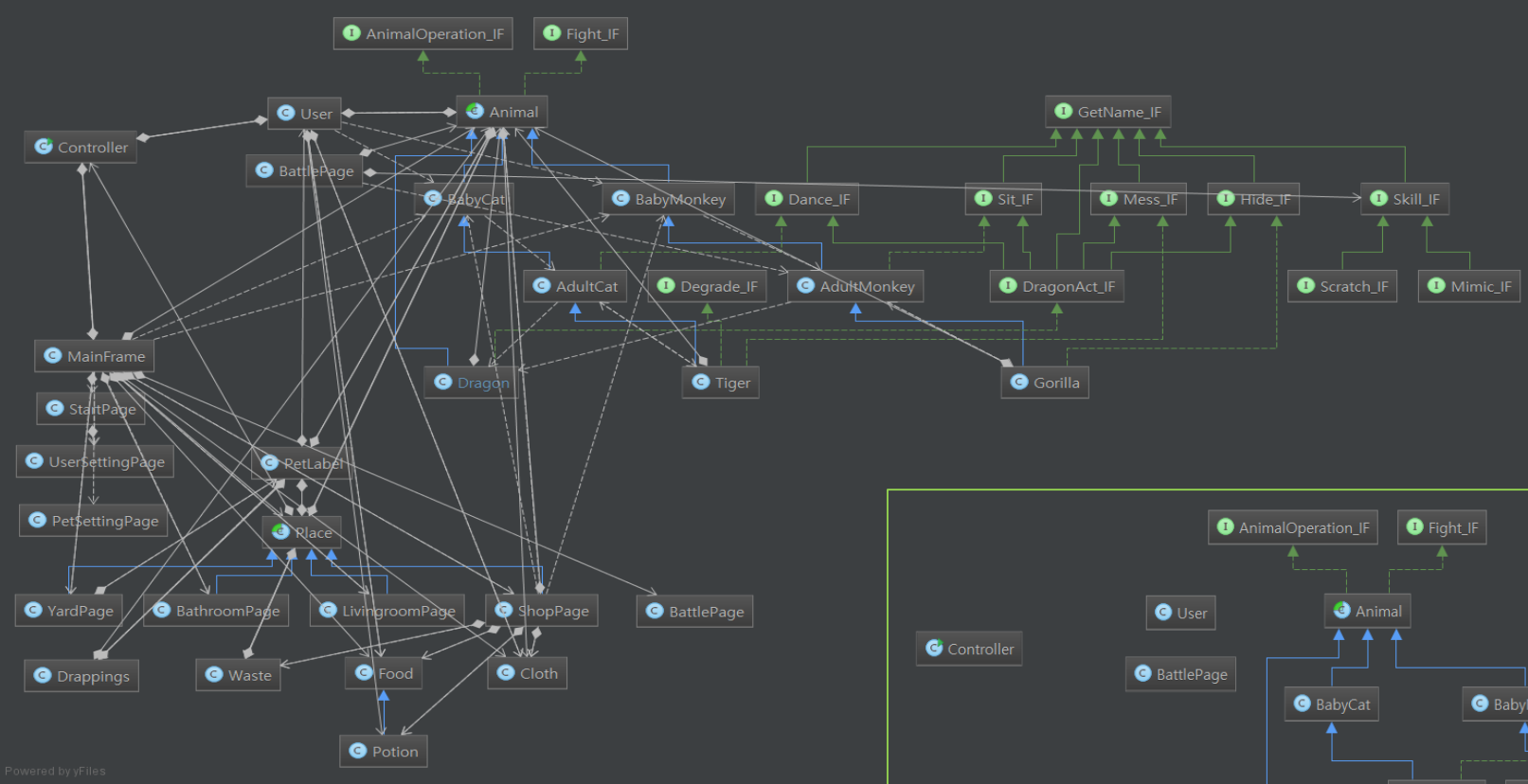


하나씩 하나씩 속성을 채우고  
메소드를 배치해 보아요

# 이렇게 분석설계를 하면 정말 나아질까요?



비로소 코딩을 시작하고 보니 보이는 것들이 생겼습니다.  
아주 중요한 개념이 누락되었어요. 또는 부수적으로 필요한 클래스들이 보입니다!  
여기까지 한바퀴 돌았습니다.



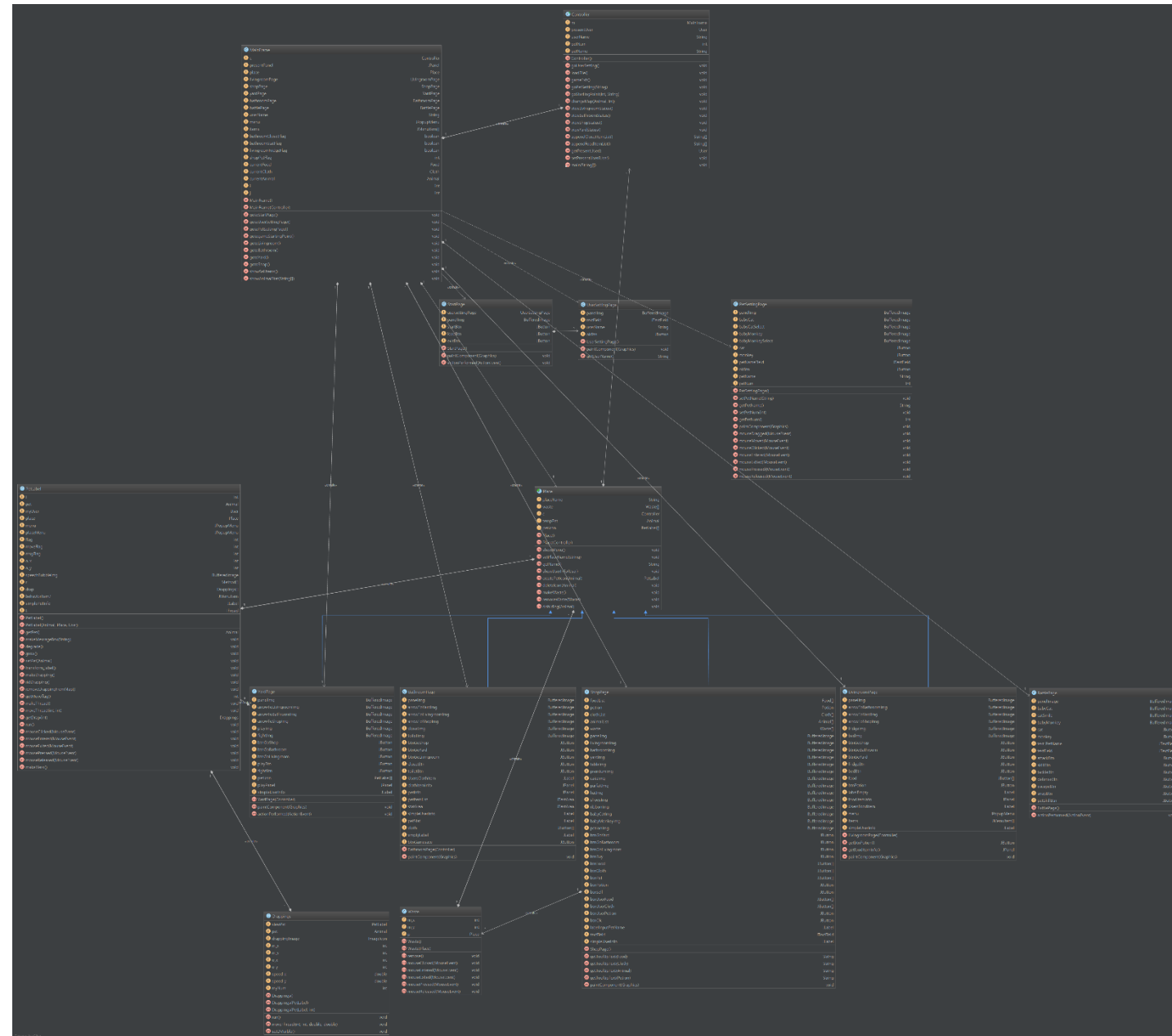
속도를 내어 개발을 하다보면  
어느덧 의존성사례들이 발생합니다.  
처음부터 고려해주는 것이 필요합니다.



속도를 내어 개발을 하다보면  
클래스들의 개수는 순식간에 늘어납니다.

최소한

Model / Control / View 클래스는 분리하고  
추상모델과 세부모델을 단계적으로 그려나갑니다.



# 뒤돌아보면서 확인하고 검토하는 과정이 필요합니다.

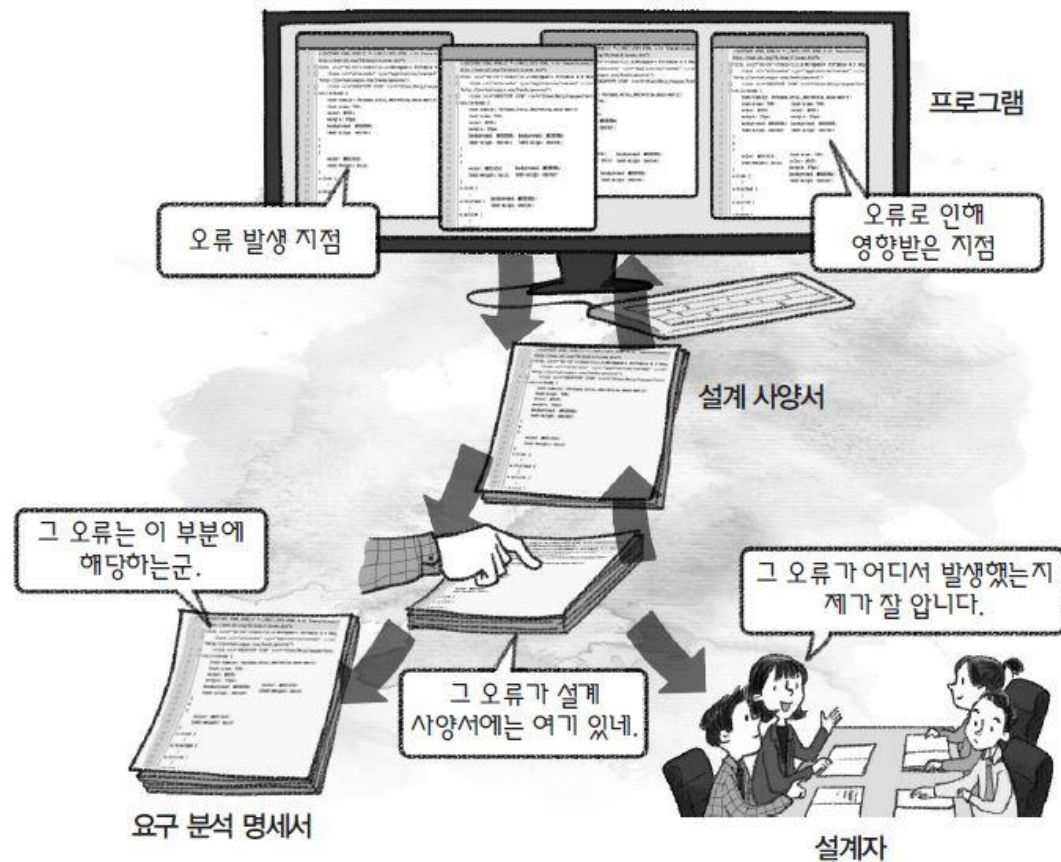
점진적이고 기민한 동작을 하는 것 같지만?



# 요구사항이 수시로 변경되는 일이 많다면

## Traceability

Requirement	use case name	operation in sequence diagram	Class
사용자는 시스템에 로그인을 한다.	login	login()	BookingSystem
사용자는 숙박리스트를 검색한다.	search	search()	BookingSystem
시스템은 사용자가 요청한 숙박리스트를 보여준다.	search	query()	BookingSystem
사용자는 원하는 숙박시설을 선택한다.	bookingRequest	select_hotel()	BookingSystem
시스템은 사용자가 선택한 숙박시설의 방 타입, 가격, 예약 가능 유무를 보여준다.	bookingRequest	select_room()	BookingSystem
사용자는 방타입, 인원수, 요청사항을 입력하여 예약을 신청한다.	bookingRequest	booking_request()	BookingSystem
시스템은 사용자의 예약 요청을 확정한다.	bookingApprove	booking_fail_notify()	BookingSystem
		notify_booking()	Hotel
		approve_booking()	BookingSystem
		sendSMS()	Notification
		sendEmail()	Notification



앞단계부터 하나씩 하나씩 기록을 시작하는 것으로 충분합니다.

초기단계에서 변경이 생길 때, 오픈소스코드의 스펙이 바뀌는 경우에도 수정해야할 지점을 쉽게 찾아낼 수 있어요.

진행과정에 대한 설명

1) 시행착오

- Glossary를 정리해 보는 것이 좋겠다고 교수님께서 말씀하셨는데, Glossary를 확실하게 정리하지 않고 팀원들간의 어느정도 합의를 이루는 수준으로 정리하고 넘어갔는데, 이후에도 개념에 대한 것들이 계속 발목을 잡고 토론을 어렵게 만드는 것을 경험해볼 수 있었습니다.
- 요구사항분석, 도메인모델에 대한 합의점을 도출, fully dressed up form 작성하는데에 시간을 많이 써서 그 이후 과정을 진행하는 데에 시간이 빠듯했던 것 같다.

2) 어려웠던 점

- 요구사항에 대한 이해는 같았지만, 구현방향에 대한 이해가 각자 달라서 합의하는 과정이 필요했다.
- Domain Diagram / Class Diagram을 같이 토의하며 그리는 과정이 처음이라 낯설었다.
- 시간적으로 여러번 반복하는게 불가능해 각 단계마다 토론하며 변경하는 것을 선택했는데, 역시나 시간이 부족했던 것이 어려웠고, 서로 다르게 이해하고 있는 개념을 통일 시키는 것이 어려웠던 것 같습니다.

3) 수월했던 점

- 관련경험을 해봤던 팀원이 있어서 조금 더 수월한 설계 및 구현이 진행될 수 있었다.
- 각자가 다양한 아이디어를 갖고 있어서 생각의 폭을 넓힐 수 있었다.
- 자신의 생각과 달라도 다수의 생각이 공통된다면, 적극 수용하는 자세로 임해서 전반적으로 합의점 도출이 수월했다.

2~3바퀴를 돌아보면서 진행과정을 돌이켜 보면서 과정을 정리해봅니다.

적용 소감

- 현업에서도 구현 전에 요구사항에 대한 충분한 분석과 정의하는 과정이 필요할 것 같다.
- 현업에 돌아가서, 이러한 과정을 거치지 않고 바로 개발에 들어간 프로젝트들을 다시 한번 살펴보면 감회가 새로울 것 같다.
- 충분히 논의하는 과정에서 더 좋은 도메인모델로 변경되는 과정을 경험해서 좋았다.
- 좋은 팀원들을 만나고 같이 협업하여서 즐거웠다.
- 다같이 협업을 한다는 것에 대해서는 좋았습니다, 다만 다소 급하게 들어가다보니 과제에 대해서 이해하는 시간과 보다 충분한 커뮤니케이션이 사전에 있었으면 좋았을 것 같습니다. 물론 그런 환경 때문에 반복작업을 통해서 산출물이 변경되는 것을 보기에는 매우 적합했다고 생각합니다.
- 팀원분들이 모두 자신의 과제처럼 잘 나서주셔서 최종 산출물이 나올 수 있었던 것 같습니다. AA 3차 7조 화이팅~!
- 팀원들과 이야기하면서 주제에 대한 요구사항을 분석하고 의견을 조율하는 과정이 재밌었습니다. 용어에 대한 정의가 덜 된 부분이 있었지만 주어진 시간이 좀 더 있었다면 반복하면서 정리될 수 있을 것 같습니다 :)

시행착오

- 공유문서 tool 결정하는데 어려움이 있었음
- 실제 사용해보지 않은 주제를 구상하다보니 우리가 임의로 결정해야 하는 부분이 있었음
- 시스템의 개발 범위를 정의하는데에 혼란이 있었음
- 각 Usecase 별로 Sequence Diagram을 그려보니 처음 회의시 나오지 않은 클래스, Actor 등이 등장하여 다시 회의를 하게 됨

우리는 왜 어려웠나/ 수월했나

- 조장님을 정해서 진행과정이 너무 매끄럽고 좋았습니다.
- 진행하다보니 주요 요구사항도 자주 변경되어서 어려웠음
- 구조를 복잡하게 만드는 조건들에 제약을 두어 문제를 회피할 수 있었음
- 요구사항 정리에 오랜시간이 걸린만큼 도메인 모델은 빠르게 작성됨.
- 우리도 다행이 있었음(fully dressedup, class diagram 최종)

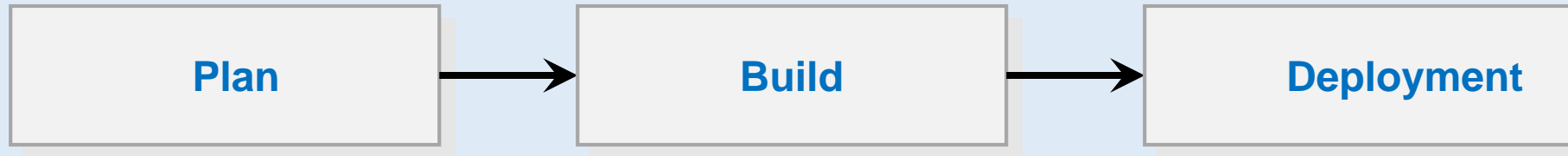
가장 적응이 빨랐던 단계, 고통스런 단계

- 고통스런 단계 : 첫 날 소회의실 처음 모인 순간 ..정적ㅋㅋㅋㅋㅋㅋㅋㅋ(공감..22)
- 적응이 빨랐던 단계 : 소음과 공존하기
- 적응이 빨랐던 단계 : 조장님을 따르기

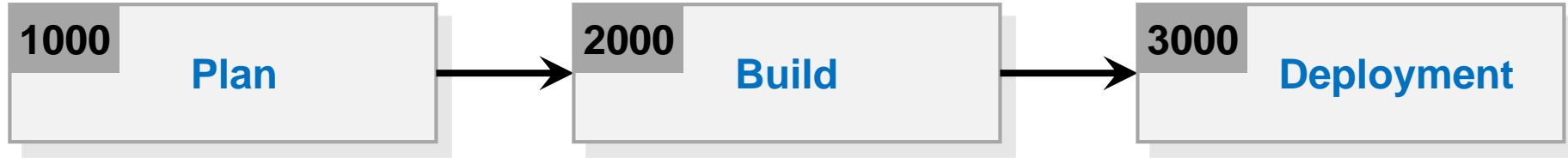
새로운 신규 프로젝트를 진행해야 할 때 이 프로세스를 참고해보세요

# OOPT

## (Object-Oriented Process with Traceability)



# 1. 3 Stages

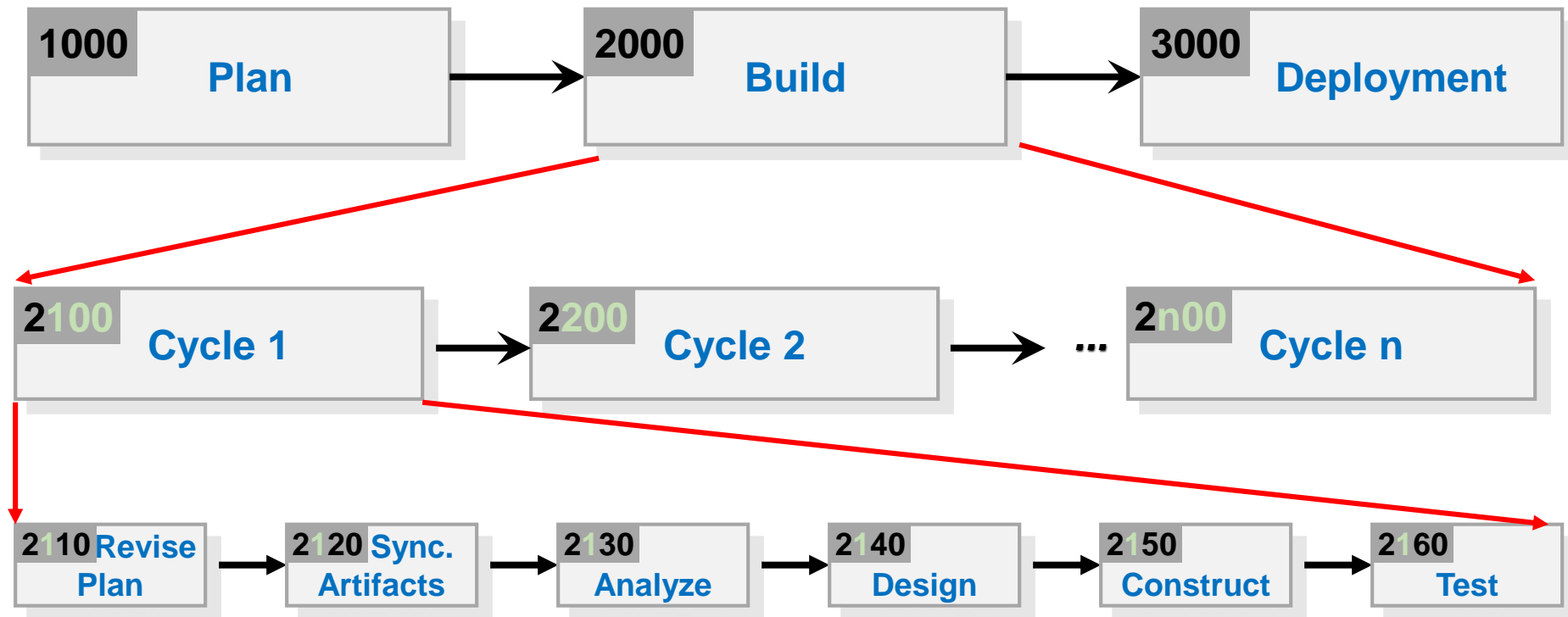


- Stage 1000 : Plan
  - Planning, defining requirements, building prototyping, etc
  - Corresponding to **Inception phases** in the RUP
- Stage 2000 : Build
  - **Elaboration** and **Construction** of the system
  - Corresponding to Elaboration/Construct phase in the RUP
- Stage 3000 : Deployment
  - Implementation of the system into use
  - Corresponding to **Transition** phase in the RUP

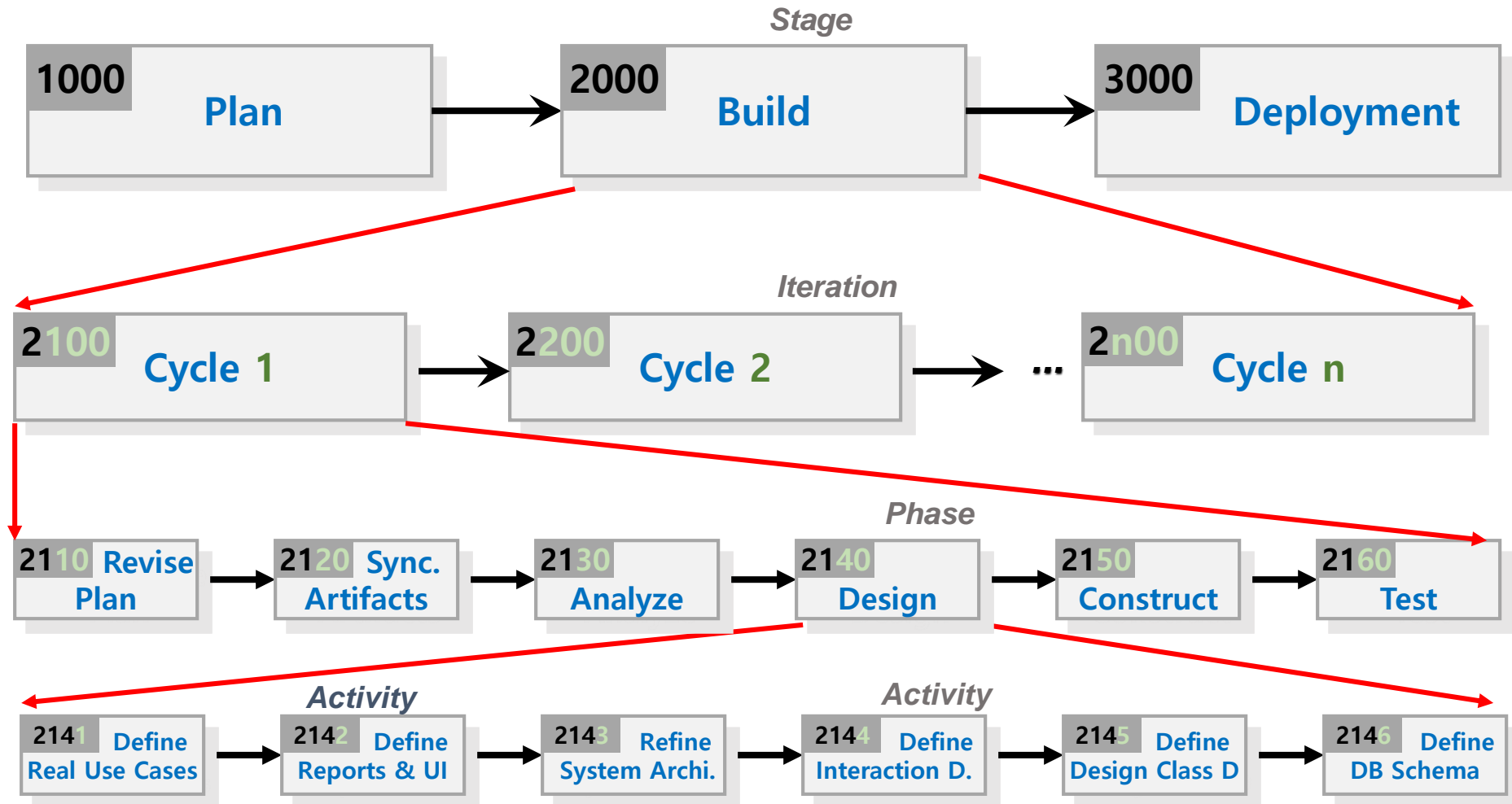


## 2. Iterative Development

- Multiple iterations in the Build stage



# Hierarchical Architecture



# Phase 2030. Analyze

a. if not yet done  
b. ongoing  
c. optional

