



Design Specification

TEAM #1

2016314958 강경룡

2014312269 이주연

2012314466 이현우

2016315339 이희관

2015314767 조준호

목차

1. Preface	5
1.1 Objective	5
1.2. Readership	5
1.3. Document Structure	5
A. Preface	5
B. Introduction	5
C. System Architecture	5
D. User Management System	5
E. AR System	6
F. Furniture Management System	6
G. Furniture Buy System	6
H. Protocol Design	6
I. Database Design	6
J. Development Environment	6
K. Index	6
L. Reference	6
2. Introduction	7
2.1. Objectives	7
2.2. Applied Diagram	7
A. Class Diagram	7
B. State Diagram	7
C. Sequence Diagram	7
D. ER Diagram	8
2.3. Applied Tools	8
A. Draw.io	8
B. Android Studio	8
C. Firebase	9
3. System Architecture	9

3.1. Objectives	9
3.2. System Organization	9
A. "가지마켓" System	9
B. User Management System	10
C. Furniture Management System	11
D. Furniture Buy System	12
E. AR System	13
4. User Management System	13
4.1. Objectives	13
4.2. Class Diagram	13
A. DB Handler	13
B. Sign Up	14
C. Log in	14
4.3. Sequence Diagram	15
4.4. State Diagram	16
5. AR System	16
5.1. Objectives	16
5.2. Class Diagram	17
A. DB Handler	17
B. Contents Information	17
5.3. Sequence Diagram	18
5.4. State Diagram	18
6. Furniture Management System	18
6.1. Objectives	18
6.2. Class Diagram	19
A. DB Handler	19
B. Data Handler	19
C. Account	19
D. Furniture	19
E. Wish List	20
6.3. Sequence Diagram	20
A. Register Furniture	20
B. Wish List	21
6.4. State Diagram	21

A. Register Furniture	21
B. Wish List	22
7. Furniture Buy System	22
7.1. Objectives	22
7.2. Class Diagram	22
A. Data Handler	23
B. Account	23
C. Furniture	23
D. Purchase	23
7.3. Sequence Diagram	24
A. Purchase	24
7.4. State Diagram	24
A. Purchase	24
8. Protocol Design	25
8.1. Objectives	25
8.2. Protocol Description	25
A. Sign Up Protocol	25
B. Sign In Protocol	26
C. Show Product List Protocol	26
D. Show Product Information Protocol	27
E. Purchase Protocol	27
F. Add Wishlist Protocol	28
G. Show User Information Protocol	28
H. Show Order Information Protocol	29
I. Show Wishlist Protocol	30
J. Upload Product Protocol	30
K. Change Information Protocol	31
L. Seller Registration Protocol	31
9. Database Design	32
9.1. Objectives	32
9.2. ER Diagram	33
A. Overall ER Diagram	33
B. Entity	34
C. Relationship	36

9.3. Tree	40
10. Development Environment	41
10.1. Objectives	41
10.2. Programming Language	41
10.3. IDE	42
11. Index	42
11.1. Table Index	42
11.2. Figure Index	43
11.3. Diagram Index	43

1. Preface

1.1. Objective

Preface에서는 본 문서의 구조와 각 부분의 역할들에 관하여 기술한다.

1.2. Readership

본 문서의 예상 독자는 가지마켓 시스템의 개발, 유지 보수와 관련된 모든 구성원이다.

1.3. Document Structure

본 Design Specification은 총 14개의 부분으로 구성되어 있다. Preface, Introduction, System Architecture, User Management System, AR System, Furniture Management System, Furniture Buy System, Protocol Design, Database Design, Development Environment, Index, Reference 로 이루어져 있으며 각 장의 내용은 아래와 같다.

A. Preface

Preface에서는 본 요구사항명세서의 구조와 각 부분의 역할에 대해 기술한다.

B. Introduction

Introduction에서는 본 프로젝트에서 시스템을 설계할 때 사용하는 모든 다이어그램 및 틀에 대해 서술한다.

C. System Architecture

해당 부분에서는 '가지마켓'의 시스템 구조를 서술한다. 그리고 전반적인 시스템 구조에 대한 설명을 추가하여 독자들의 원활한 이해를 돕는다.

D. User Management System

유저들의 로그인과 기록들 등의 데이터베이스를 관리하는 시스템이다. 유저의 분류에 따라 인터페이스 및 데이터 관리 분류 역시 이루어진다.

E. AR System

AR 시스템은 가구 구매시 AR을 통해 가구의 외관 및 배치를 짐작할 수 있게끔 돕는 시스템이다.

F. Furniture Management System

Furniture Management System 은 마켓에 가구를 등록하거나 위시리스트 추가 등의 과정에서 발생하는 데이터 처리를 다룬다.

G. Furniture Buy System

Furniture Buy System 은 가구 구매시 이루어지는 데이터 처리 및 과정에 대해서 서술한다.

H. Protocol Design

본 파트에서는 각 시스템 간의 커뮤니케이션에 필요한 프로토콜에 관하여 기술한다. 가마켓이 사용하는 데이터 포맷과 세부적인 설명 역시 포함된다.

I. Database Design

Database Design 파트에서는 요구사항명세서에 포함된 내용을 기반으로 Diagram을 작성하고 데이터베이스를 구현하는 과정을 다룬다.

J. Development Environment

Development Environment 부분에서는 개발환경에 관해 서술한다. 가마켓의 개발에서 사용되는 프로그램 언어와 기타 사항들에 관하여 설명한다.

K. Index

본 Design Specification에서 사용된 Table, Diagram에 대한 인덱스들이 포함된다.

L. Reference

본 Design Specification서 사용된 외부 자료들의 Reference가 포함된다.

2. Introduction

2.1 Objectives

Introduction 에서는 가지마켓 시스템 설계시 사용된 모든 다이어그램 및 툴에 대해 설명한다.

2.2. Applied Diagram

A. Class Diagram

클래스 다이어그램은 오브젝트 모델링 프로세스의 기본이라 할 수 있으며 시스템 구조를 모델링한다. 단일 혹은 복수의 클래스 다이어그램을 통해 시스템 컴포넌트를 모델링할 수 있다. 클래스 다이어그램은 시스템의 설계도라고 할 수 있고 오브젝트 모델링 뿐만 아니라 관계 표시, 각각의 룰에 대해 설명한다.

B. State Diagram

상태 머신은 개별 클래스 오브젝트, 사용 케이스, 전체 시스템의 스펙이다. 상태 머신을 작성시 접속하는 오브젝트는 상태 머신 소유가 된다. 상태 다이어그램은 상태 변경이 유발되는 조치와 상태가 전이되는 이벤트 및 상태 시퀀스의 시각화된 다이어그램이라고 할 수 있다. 동작 측면을 설명하는데 용이하다. 이를 통하여 중요한 동장이 있는 시스템과 클래스를 설명할 수 있으며 복잡한 상태 변경을 파악할 수 있다.

C. Sequence Diagram

시퀀스 다이어그램은 오브젝트 간 메시지 시퀀스를 설명하는 다이어그램이다. 또한 오브젝트 간의 제어구조도 표시할 수 있다. 시퀀스 다이어그램을 통하여 순서 혹은 절차적인 흐름을 시각적으로 용이하게 파악할 수 있다.

D. ER Diagram

ER 다이어그램이란 Entity-Relationship Diagram 의 약자이다. 데이터 베이스의 구조화된 데이터에 대한 표현이며 DB 저장 데이터 구조 및 조건들을 정의하고 모델링할 수 있다. ER 다이어그램은 개체와 관계로 구성된다. 개체란 현실세계의 객체로써 유무형의 대상으로 존재하며 구별될 수 있는 것을 의미하고 같은 속성의 개체들은 개체타입이라고 불린다. 관계는 개체 간에 존재하는 연관성이라고 할 수 있다. ER 다이어그램은 개체와 개체들 간의 관계를 시각화하여 용이하게 파악하는데 도움을 준다.

2.3 Applied Tools

A. Draw.io



draw.io

본 문서에는 다양한 다이어그램 작성을 위해 draw.io를 사용하였다. 이 툴은 다이어그램 작성이 매우 용이하게 되어있으며 무료 소프트웨어란 장점이 있고 작업 상황 공유가 가능하기도 하다.

B. Android Studio



Android Studio 는 안드로이드 플랫폼의 공식 개발 환경이다. 안드로이드 전용 툴이며 가장 기본적인 툴이라고 할 수 있으며 개발에 필수적이다.

C. Firebase



Firebase 는 DB를 구축하게 해줄 수 있는 모바일 앱 플랫폼으로써 안드로이드 스튜디오와의 호환성이 좋고 매우 간단하게 개발 단계를 줄여주기에 사용되었다.

3. System Architecture

3.1. Objective

System Architecture에는 팀에서 개발하고자 하는 시스템에 대해서 전반적으로 설명한다. “가지마켓”의 시스템을 Block Diagram을 통해 나타내어, 각 시스템의 연관 관계를 나타낸다. 또한 각 시스템의 구조에 대해서 System Architecture를 이용하여 간략히 설명한다.

3.2. System Organization

A. “가지마켓” System

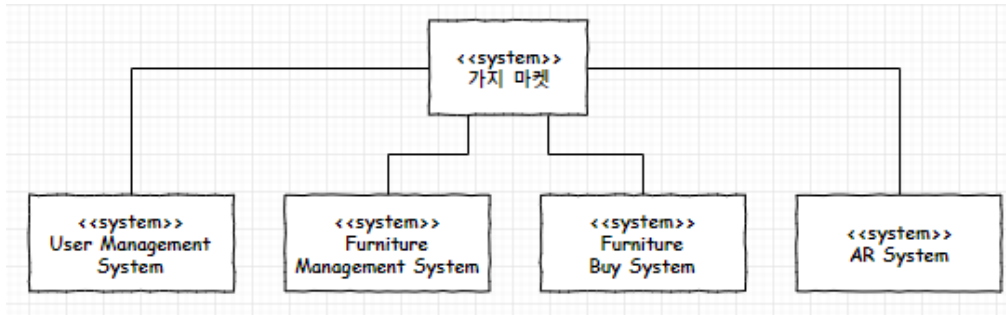


Diagram 1. Block Diagram of 가자마켓

“가자마켓”은 기본적인 인터넷 쇼핑의 흐름을 따른다. 따라서 기본적으로 User Management System이 존재하며, User의 분류가 일반 사용자인지, 판매자인지에 따라 Furniture Management System이 운영된다. Furniture Management System은 Checkup Management를 통해 등록된 가구를 등록할지 여부를 판단하는 시스템을 포함한다.. 일반 사용자의 경우에는 Furniture Buy System을 통해서 가구의 구매를 이루고 해당 시스템 내에서 Delivery Item Management 등을 포함한다.

또한 해당 프로젝트의 가장 핵심이 되는 AR System은 등록되어 있는 가구 중, AR System에 활용할 모델이 존재하는 경우 AR 기능을 통해 고객이 직접 가구를 살펴보고 인테리어와의 어울림 등을 직접 살펴볼 수 있도록 한다. 기본적인 인터넷 쇼핑의 흐름과 차별화되는 기능으로 작용할 것이다.

B. User Management System

User Management System은 유저의 회원가입을 통한 유저 정보 등록, 유저의 성격(일반 사용자, 판매자, 관리자)에 따른 Interface의 분류, 일반 사용자의 경우에는 Furniture Buy System 내부의 Delivery Item Management와 함께 유저의 My page를 관리하며, 판매자와 관리자는 Furniture Management System을 통해 가구의 승인 여부, 가구의 등록 요청 내용 확인 등으로 My page가 관리된다.

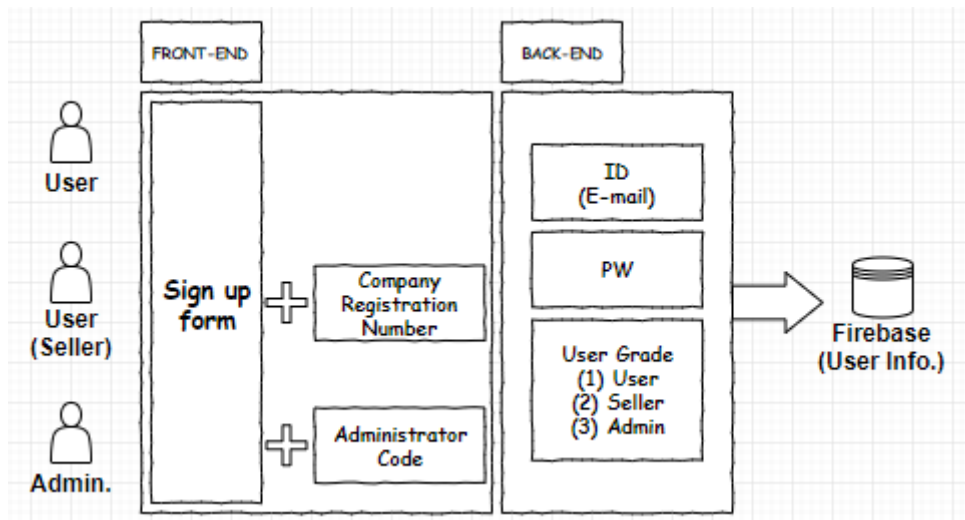


Diagram 2. User Management System Architecture (1)

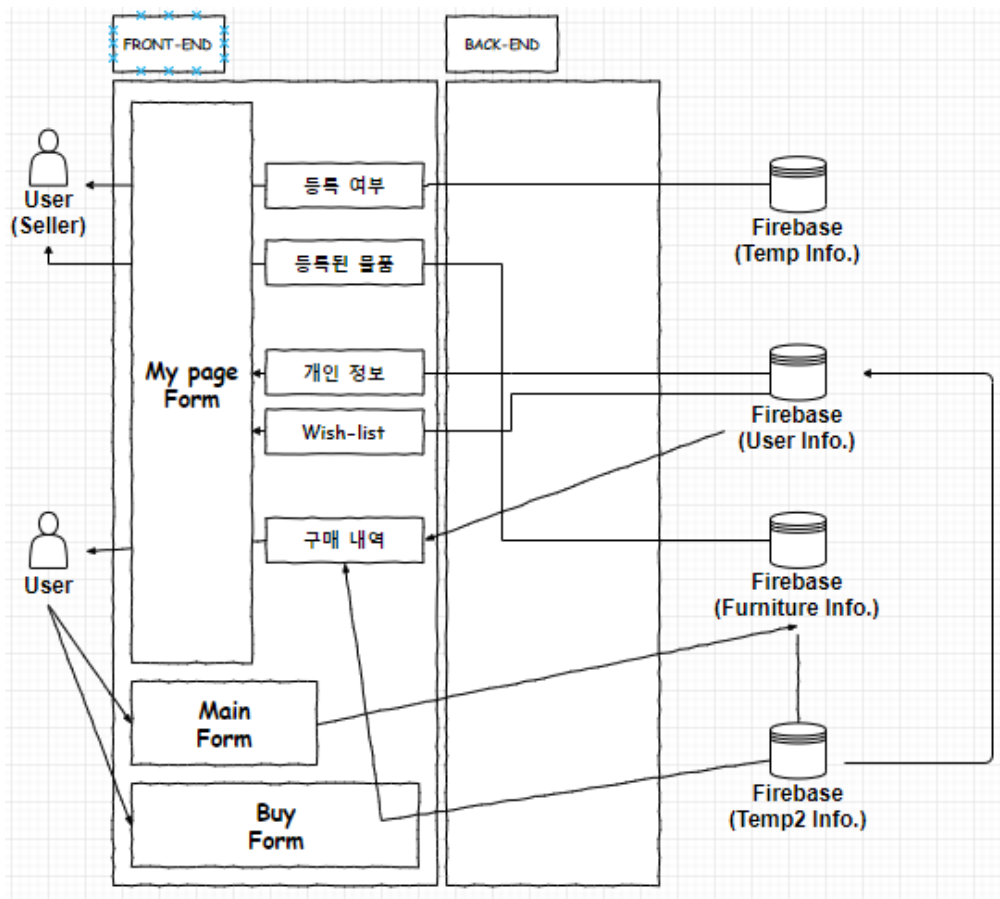


Diagram 3. User Management System Architecture (2)

C. Furniture Management System

Furniture Management System은 User의 분류가 관리자와 판매자인 경우에 적용되는 시스템이다. 기본적으로 판매자가 판매를 원하는 가구를 등록하고

관리자는 해당 가구가 판매에 적합한 지 여부를 판단하여 승인을 이룬다. 이러한 하부 시스템들이 Furniture Management System에 포함된다. 해당 내용은 각 유저들의 My page Interface에 영향을 준다.

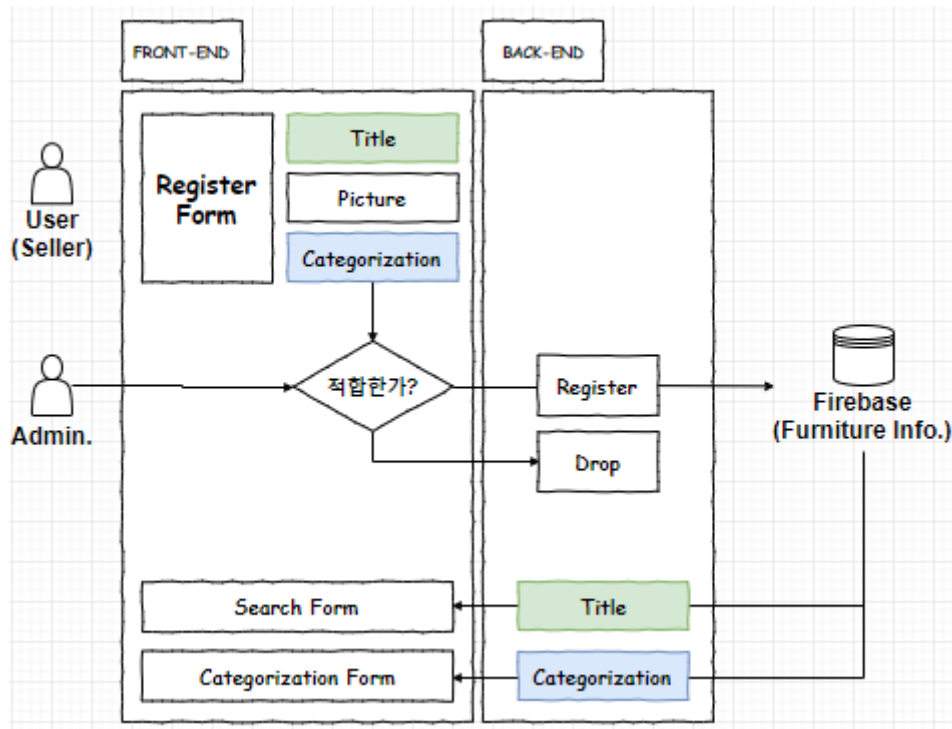


Diagram 4. Furniture Management System Architecture

D. Furniture Buy System

Furniture Buy System은 일반 사용자가 가구를 구매하는 과정에 대한 시스템이다. Delivery Item Management가 하부 시스템으로 포함되며 해당 내용은 일반 사용자의 My page에서 배송 정보를 판단하는 데 활용된다. 관리자와 판매자는 기본적으로 가구를 구매할 수 없다.

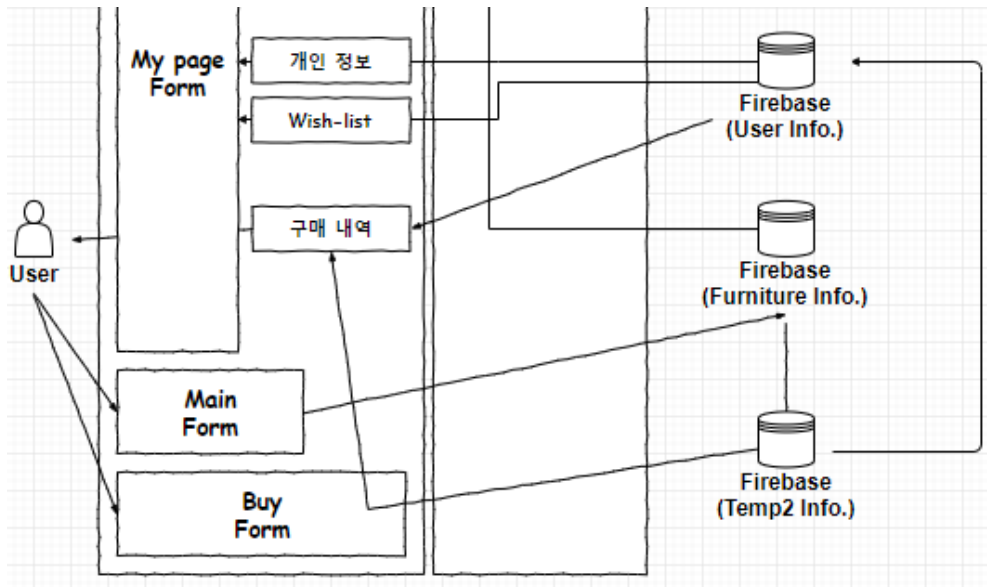


Diagram 5. Furniture Buy System Architecture

E. AR System

AR System은 관리자가 판매자의 가구를 승인한 이후부터 해당 가구를 모델링하여 모델링 자료를 관리한다. 해당 모델링 자료의 존재 여부에 따라 상품 상세 내용 페이지에서 "AR View"(가칭) 버튼을 통해 모델링을 불러와 ARCore를 통해 실제 AR 내용을 진행한다. AR 페이지에서 직접 구매와 연동하는 버튼을 추가하는 것도 고려중이다. (Optional)

4. User Management System

4.1. Objective

User Management System은 회원가입과 로그인/로그아웃에 따른 유저 정보 저장 및 변경에 대한 총체적인 관리가 이루어지는 시스템으로 기본적으로 유저의 분류와 각 유저 분류에 따른 Interface의 분류 또한 이루어진다. Interface의 분류는 My page에서 가장 두드러지게 일어나며 가구의 구매 버튼 활성화 여부도 해당 System의 유저 분류를 통해서 이루어진다.

4.2. Class Diagram

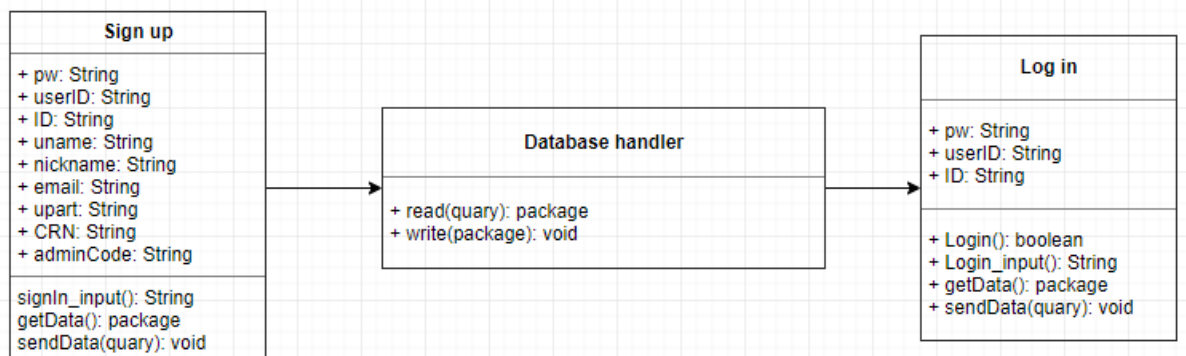


Diagram 6. User Management System Class Diagram

A. DB Handler

데이터베이스에 연결하여 테이블을 만들거나 업그레이드하여 데이터 저장, 수정, 검색, 삭제 등을 쉽게 할 수 있도록 도움을 주는 클래스

a. **Attributes:** 해당 사항 없음

b. **Methods**

- + package Read(query): 해당되는 DB에서 원하는 data를 읽어온다.
- + void Write(getData): 해당되는 DB에 data를 저장한다.

B. Sign Up

a. **Attributes**

- + ID: 유저가 등록한 아이디
- + userID: 시스템 내부에서 유저를 관리하기 위한 아이디
- + pw: 유저가 등록한 패스워드
- + uname: 유저의 이름
- + nickname: 유저의 닉네임
- + email: 유저의 이메일
- + upart: 유저의 분류 (일반 사용자, 판매자, 관리자)
- + CRN: 판매자의 경우 등록하는 사업자 등록번호
- + adminCode: 관리자의 경우 등록하는 관리자 번호

b. **Methods**

- + void sendData(query): 해당되는 DB에 데이터를 보낸다.
- + void getData(): 해당되는 data를 받아온다.
- + String signIn_input(String[]): 정해진 form에 회원가입에 필요한 정보들을 입력한다.

C. Log In

a. **Attributes**

- + ID: 유저가 등록한 아이디
- + userID: 시스템 내부에서 유저를 관리하기 위한 아이디
- + pw: 유저가 등록한 비밀번호

b. Methods

- + void sendData(query): 해당되는 DB에 데이터를 보낸다.
- + void getData(): 해당되는 data를 받아온다.
- + String Login_input(id, pw): 정해진 form에 id와 pw를 입력한다.
- + boolean Login(id, pw): 읽어온 data가 입력된 데이터와 일치하는 지 검사한다.

4.3. Sequence Diagram

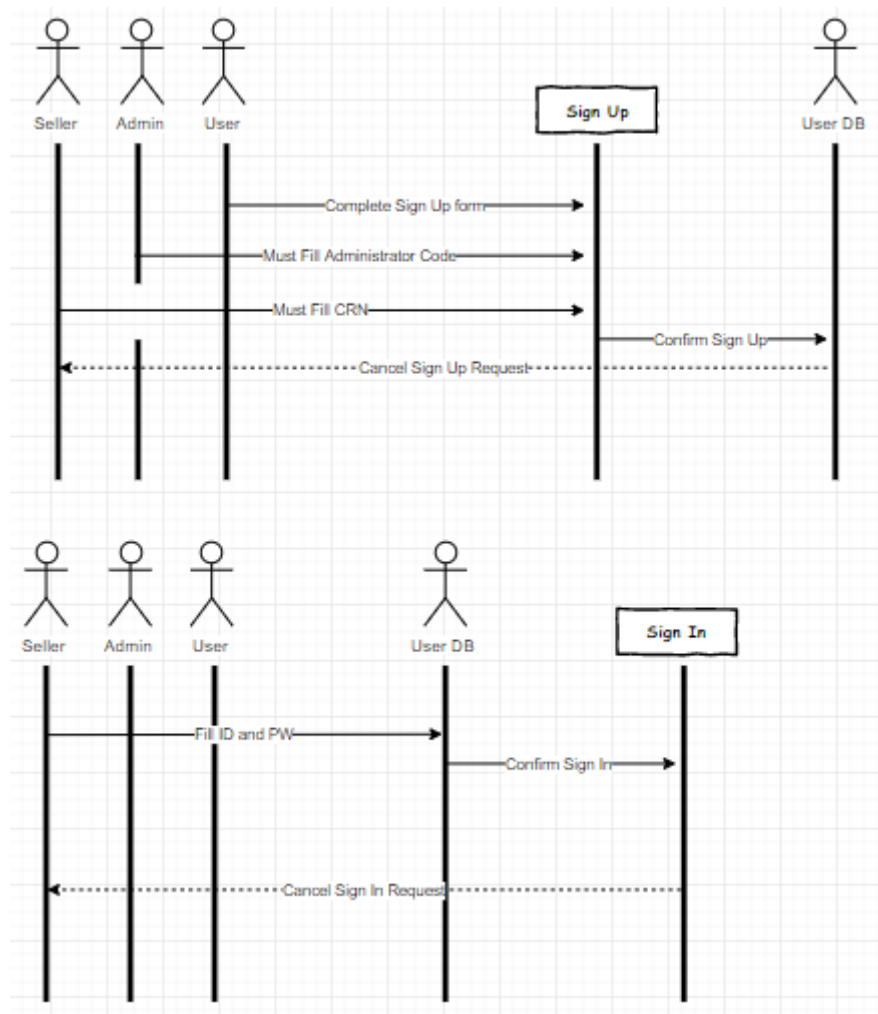


Diagram 7. User Management System Sequence Diagram

4.4. State Diagram

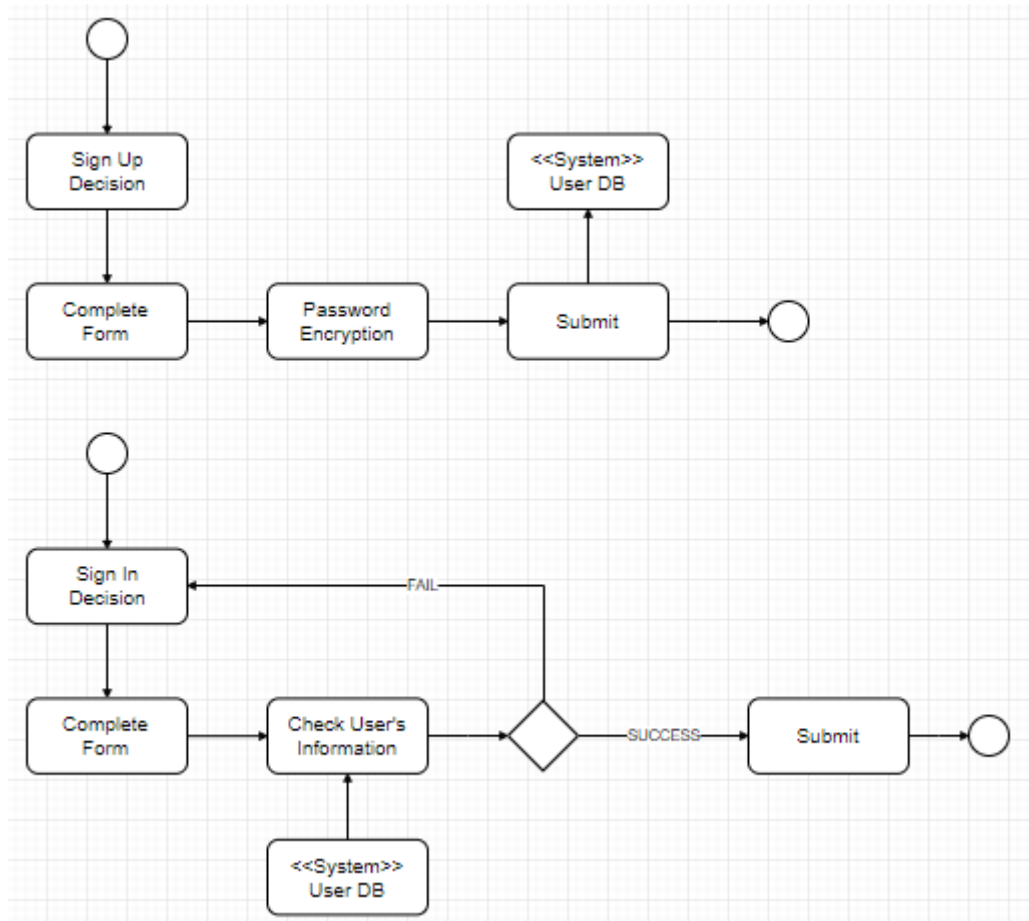


Diagram 8. User Management System State Diagram

5. AR 시스템

5.1 Objective

가치마켓에서 구매자가 LIST에서 가구 보기를 누른뒤 가구에 대한 정보가 있는 페이지에서 AR보기로 원하는 위치에 3D가구의 크기 및 방향을 시뮬레이션할수 있다.

5.2 Class Diagram

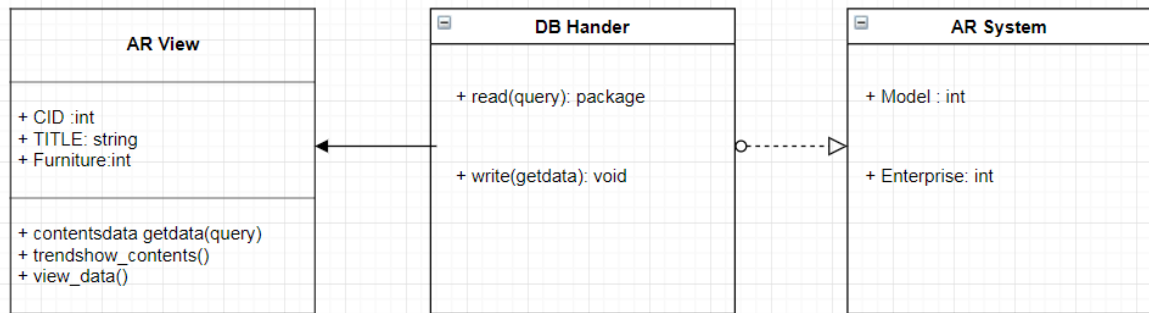


Diagram 9. Class Diagram of AR View System

A. DB Handler

a. Attributes

None

b. Methods

+ package read(query) 해당되는 DB에서 원하는 data를 읽어온다.

+ void write(getdata) 해당되는 DB에 data를 저장한다.

B. Contents Information

a. Attributes

+ int CID: contents의 번호

+ string TITLE: contents의 제목

+ int furniture: 가구 구현 개수

b. Methods

+ contentsdata getdata(query) contents의 정보를 읽어온다.

+ trendshow_contents() 각 user별로 DB에서 가구data를 가져와 제공한다.

+ view_data(data) user가 구현해본 내용을 DB에 저장

5.3. Sequence Diagram

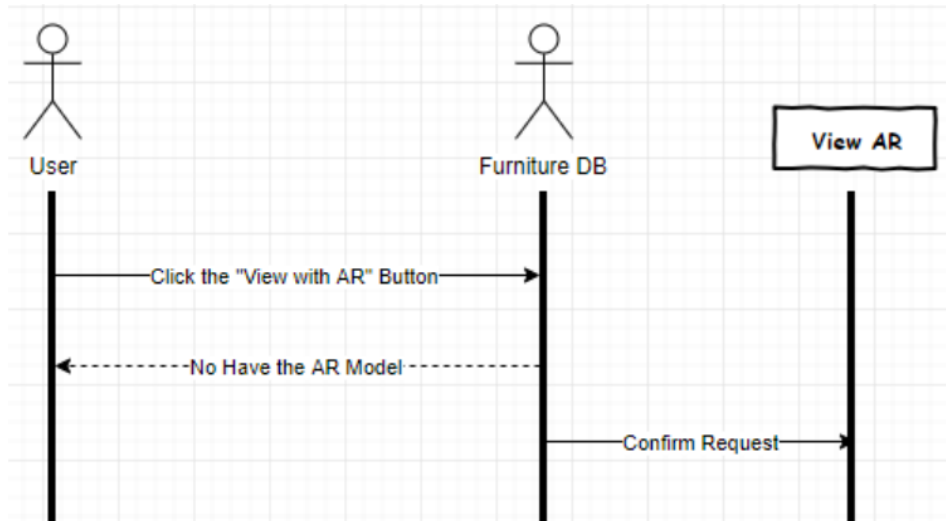


Diagram 10. Sequence Diagram of AR View

5.4. State Diagram

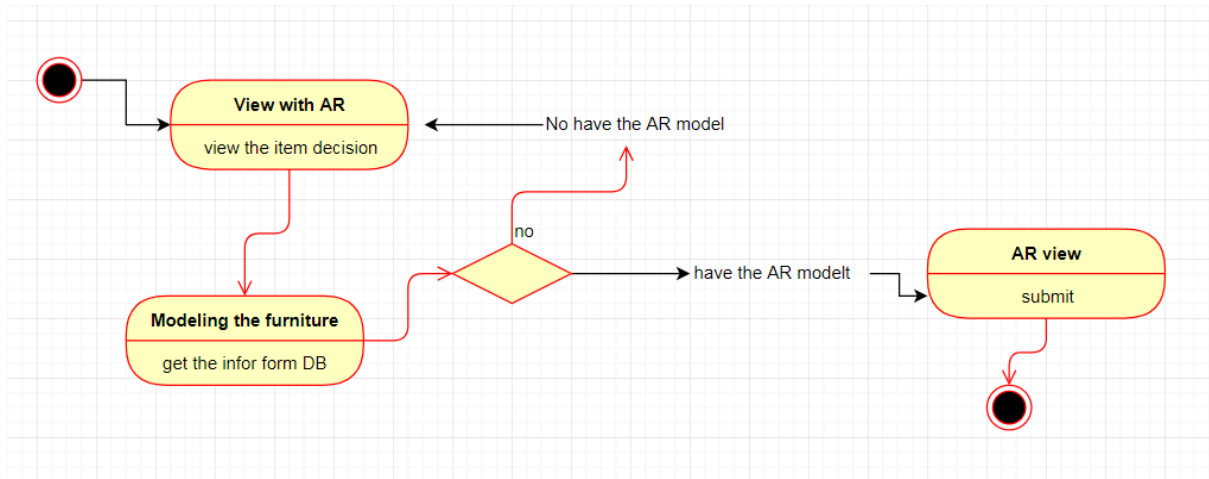


Diagram 11. State Diagram of AR View System

6. Furniture Management System

6.1. Objection

가지마켓에서 가구를 등록하거나 위시리스트에 추가하는 등의 과정에서 데이터를 처리하는 시스템에 관해 설명한다. 이를 Class Diagram, Sequence Diagram, State Diagram을 통해 상호작용을 표현하고 설명한다.

6.2. Class Diagram

A. DB Handler

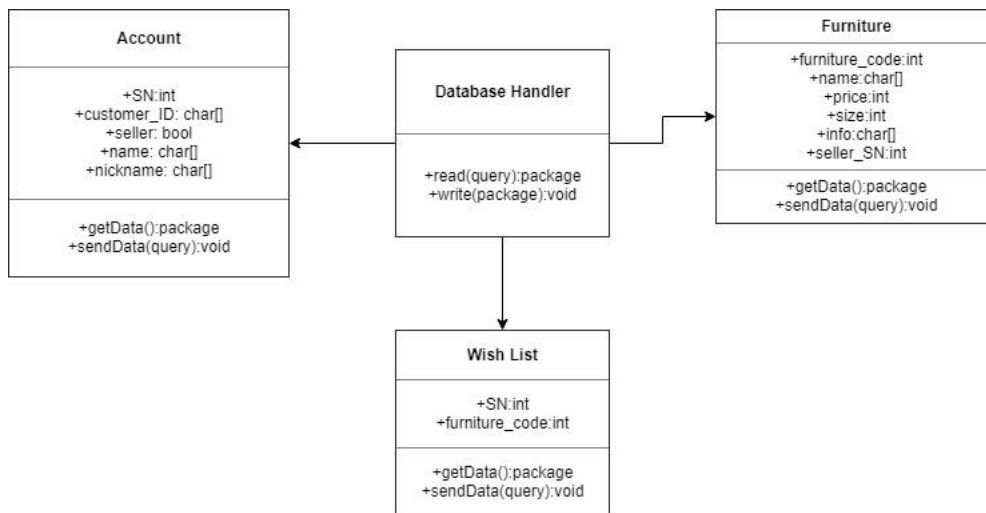


Diagram 12. Class Diagram for Subscribe System

B. Data Handler

a. Attributes

None.

b. Methods

- + `read(query): package` : 데이터베이스에서 데이터 읽어오기
- + `write(package): void` : 데이터베이스에 데이터 쓰기

C. Account

a. Attributes

- + `SN:int` : 사용자 고유번호
- + `customer_ID: char[]` : 사용자 아이디
- + `name: char[]` : 사용자 이름
- + `seller:bool` : 사용자가 판매자인지 아닌지 확인
- + `nickname:char[]`:사용자 별명

b. Methods

- + `getData():package`
- + `sendData(query):void`

D. Furniture

a. Attributes

- + `furniture_code:int` : 가구 코드

- + name:char[] : 가구 이름
- + price:int : 가구 가격
- + size:int : 가구 사이즈
- + info:char[] : 가구 정보
- + seller_SN:int : 판매자 고유번호

b. Methods

- + getData():package
- + sendData(query):void

E. Wish List

a. Attributes

- + SN:int : 사용자 고유번호
- + furniture_code:int : 가구 코드

b. Methods

- + getData():package
- + sendData(query):void

6.3. Sequence Diagram

A. Register Furniture

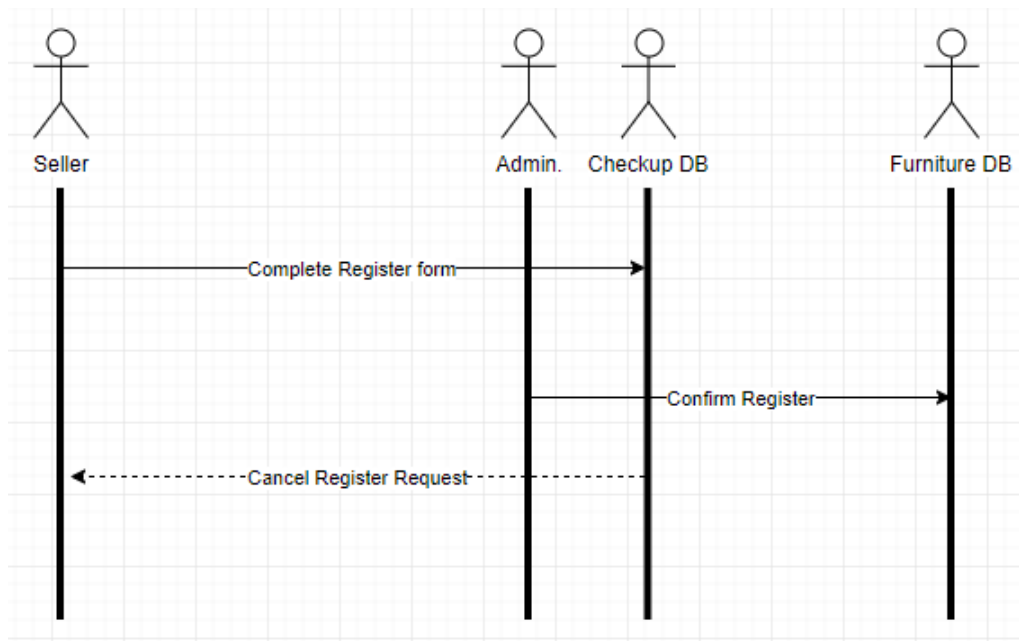


Diagram 13. Register Furniture Sequence Diagram for Furniture Management

B. Wish List

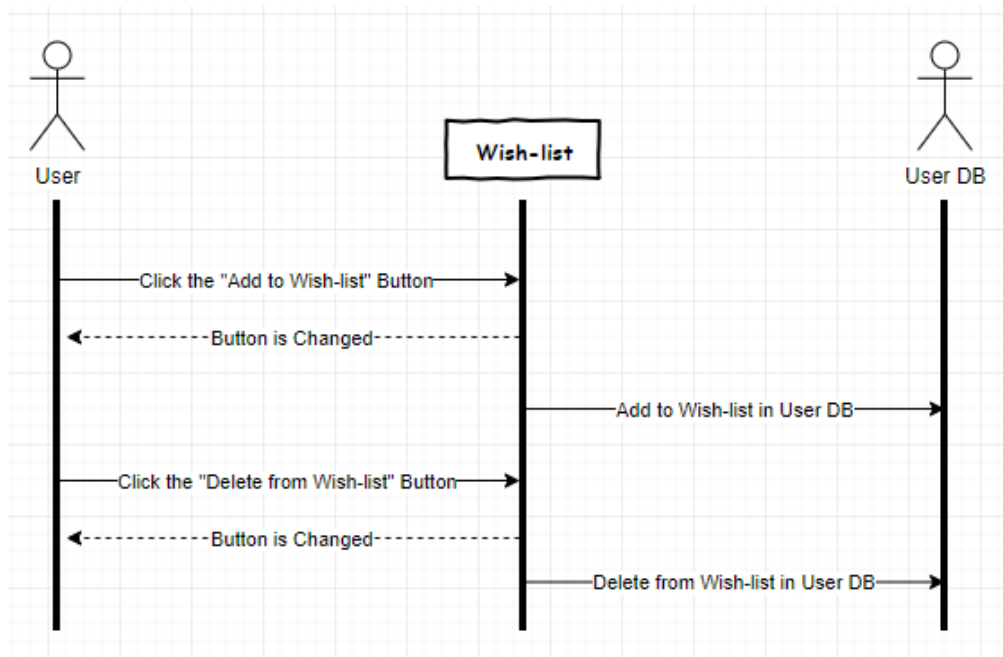


Diagram 14. Wish List Sequence Diagram for Furniture Management

6.4. State Diagram

A. Register Furniture



Diagram 15. Register Furniture State Diagram for Furniture Management

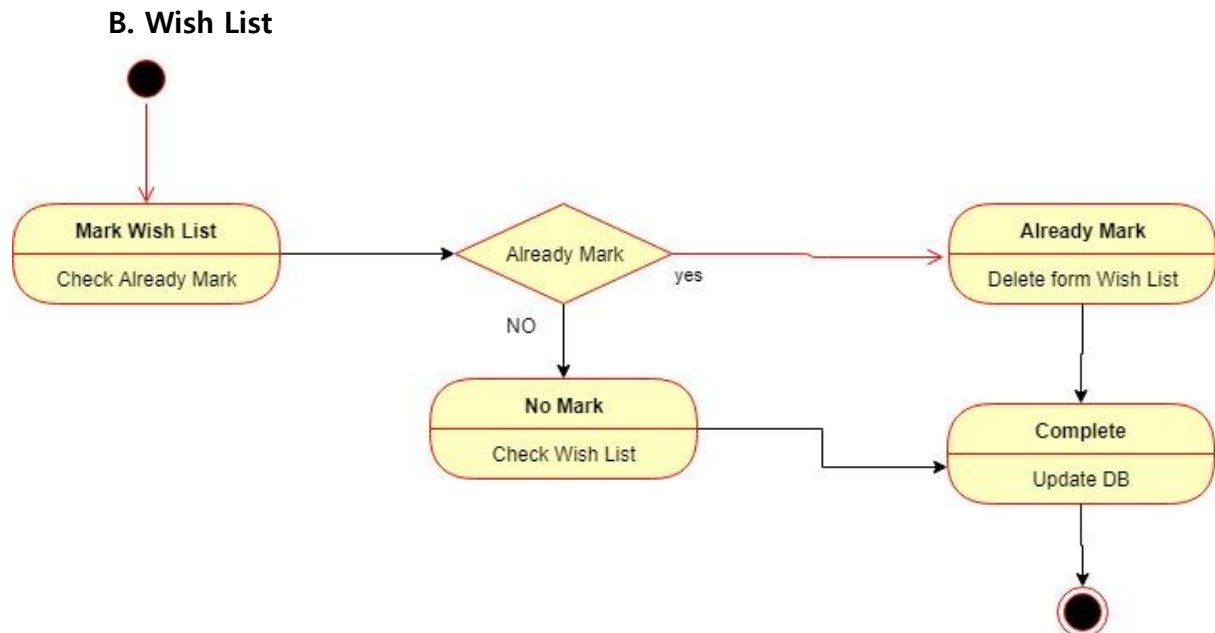


Diagram 16. Wish List State Diagram for Furniture Management

7. Furniture Purchase System

7.1. Objection

가마켓에서 가구를 구매하는 과정에서 데이터를 처리하는 시스템에 관해 설명한다. 이를 Class Diagram, Sequence Diagram, State Diagram을 통해 상호작용을 표현하고 설명한다.

7.2. Class Diagram

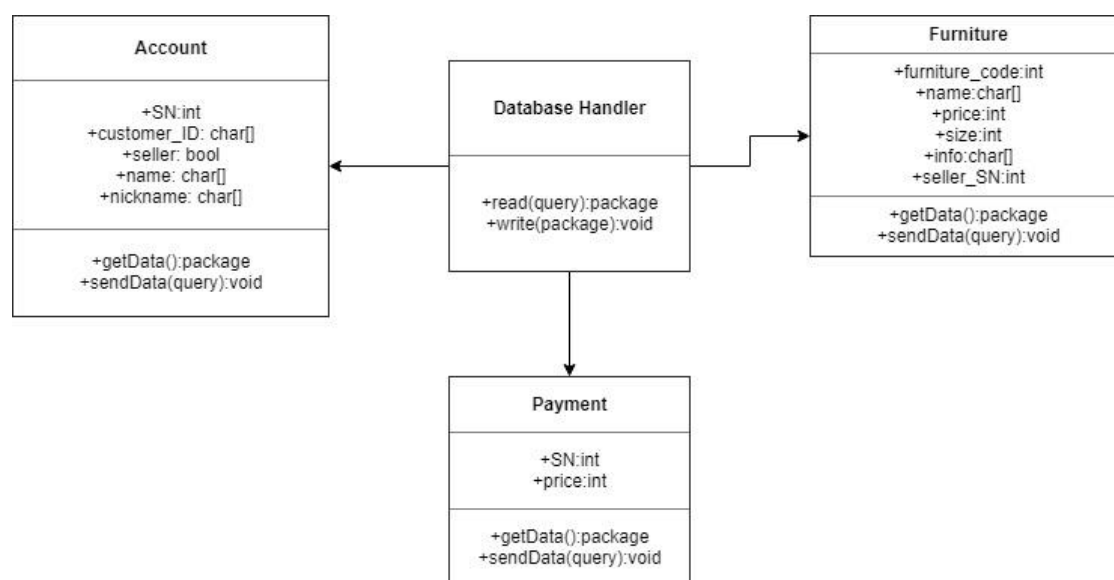


Diagram 17. Class Diagram for Furniture Purchase

A. Data Handler

a. Attributes

None.

b. Methods

- + read(query): package : 데이터베이스에서 데이터 읽어오기
- + write(package): void : 데이터베이스에 데이터 쓰기

B. Account

a. Attributes

- + SN:int : 사용자 고유번호
- + customer_ID: char[] : 사용자 아이디
- + name: char[] : 사용자 이름
- + seller:bool : 사용자가 판매자인지 아닌지 확인
- + nickname:char[]:사용자 별명

b. Methods

- + getData():package
- + sendData(query):void

C. Furniture

a. Attributes

- + furniture_code:int : 가구 코드
- + name:char[] : 가구 이름
- + price:int : 가구 가격
- + size:int : 가구 사이즈
- + info:char[] : 가구 정보
- + seller_SN:int : 판매자 고유번호

b. Methods

- + getData():package
- + sendData(query):void

D. Purchase

a. Attributes

- + SN:int : 사용자 고유번호

+ Price:int : 가구 코드

b. Methods

+ getData():package

+ sendData(query):void

7.3 Sequence Diagram

A. Purchase

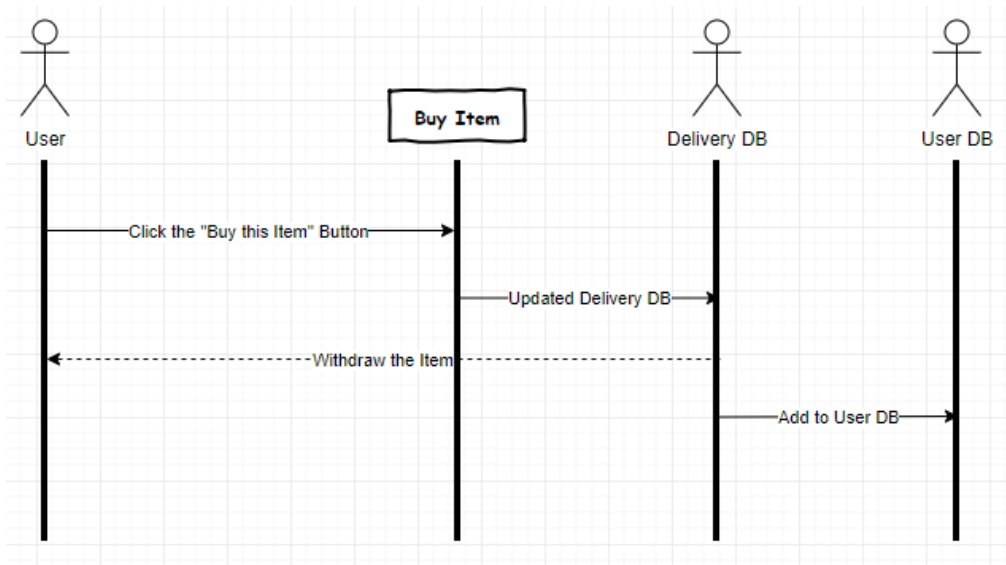


Diagram 18. Purchase Furniture Sequence Diagram for Furniture Management

7.4 State Diagram

A. Purchase

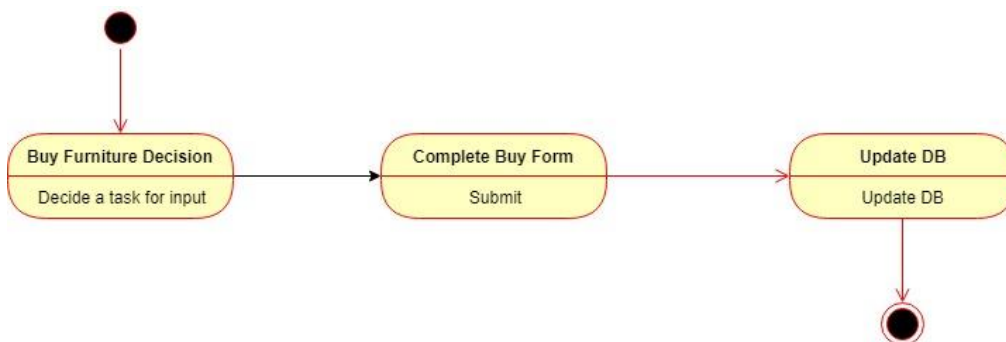


Diagram 19. State Diagram for Furniture Purchase

8. Protocol Design

8.1 Objective

Protocol Design에서는 Subsystem들이 상호작용하는 protocol에 대해 기술한다. 기본적으로 JSON이라는 Javascript Object Notation의 DATA-교환 형식을 이용하며, protocol에서 통신하는 메시지의 형식, 용도와 의미를 서술한다.

8.2 Protocol Description

8.2.1 Sign Up Protocol

A. User

- Request

Attribute	Value
user_id	사용자의 ID(E-mail 형식)
user_pw	사용자의 ID에 해당하는 password
user_name	사용자의 이름
user_addr	사용자의 주소
phone_number	사용자의 전화번호

- Response

Attribute	Value
reg_success	true / false

B. Seller

- Request

Attribute	Value
user_id	사용자의 ID(E-mail 형식)
user_pw	사용자의 ID에 해당하는 password

user_name	사용자의 이름
user_addr	사용자의 주소
user_phone_number	사용자의 전화번호
company_name	판매자의 상호명
company_number	판매자의 사업자등록번호
representative	대표자
company_addr	판매자의 주소

- **Response**

Attribute	Value
reg_success	true / false

8.2.2 Sign In Protocol

- **Request**

Attribute	Value
user_id	사용자의 ID
user_pw	사용자의 ID에 해당하는 Password

- **Response**

Attribute	Value
sin_success	true / false

8.2.3 Show Product List Protocol

- **Request** : Product 정보를 모두 조회한다.
- **Response** : 여러 개의 요약된 상품 정보로 이루어진다.

Attribute	Value
product_id	가구의 제품번호
product_name	가구의 이름
product_price	가구의 가격
product_size	가구의 규격

8.2.4 Show Product Information Protocol

- Request

Attribute	Value
product_id	가구의 제품번호

- Response

Attribute	Value
product_id	가구의 제품번호
product_name	가구의 이름
product_price	가구의 가격
product_size	가구의 규격
product_images	가구의 이미지
product_description	가구에 대한 설명
product_stock	가구의 재고
product_cate	가구의 분류

8.2.5 Purchase Protocol

- Request

Attribute	Value
product_id	가구의 제품번호

- Response

Attribute	Value
purchase_success	true / false

8.2.6 Add Wishlist Protocol

- Request

Attribute	Value
product_id	가구의 제품번호

- Response

Attribute	Value
wishlist_success	true / false

8.2.7 Show User Information Protocol

A. User

- Request

Attribute	Value
user_id	사용자의 ID

- Response

Attribute	Value
user_id	사용자의 ID
user_name	사용자의 이름

user_addr	사용자의 주소
user_phone_number	사용자의 전화번호

B. Seller

- Request

Attribute	Value
user_id	사용자의 ID

- Response

Attribute	Value
user_id	사용자의 ID
user_name	사용자의 이름
user_addr	사용자의 주소
user_phone_number	사용자의 전화번호
company_name	판매자의 상호명
company_number	판매자의 사업자등록번호
representative	대표자
company_addr	판매자의 주소

8.2.8 Show Order Information Protocol

- Request

Attribute	Value
user_id	사용자의 ID

- Response

Attribute	Value
product_id	가구의 제품번호
product_name	가구의 이름
product_price	가구의 가격
order_status	현재 주문 상태

8.2.9 Show Wishlist Protocol

- Request

Attribute	Value
user_id	사용자의 ID

- Response

Attribute	Value
product_id	가구의 제품번호
product_name	가구의 이름
product_price	가구의 가격
product_size	가구의 규격

8.2.10 Upload Product Protocol

- Request

Attribute	Value
product_id	가구의 제품번호
product_name	가구의 이름
product_price	가구의 가격

product_size	가구의 규격
product_images	가구의 이미지
product_description	가구에 대한 설명
product_stock	가구의 재고
product_cate	가구의 분류

- Response

Attribute	Value
upload_success	true / false

8.2.11 Change Information Protocol

- Request

Attribute	Value
user_pw	사용자의 비밀번호
user_name	사용자의 이름
user_addr	사용자의 주소
user_phone_number	사용자의 전화번호

- Response

Attribute	Value
change_info_success	true / false

8.2.12 Seller Registration Protocol

- Request

Attribute	Value
-----------	-------

company_name	판매자의 상호명
company_number	판매자의 사업자등록번호
representative	대표자
company_addr	판매자의 주소

- Response

Attribute	Value
seller_reg_success	true / false

9. Database Design

9.1 Objective

Database Design은 요구사항 명세서에서 기술된 요구사항을 바탕으로 수정을 거쳐 다시 요구사항을 재작성하였다. 재정립된 요구사항을 바탕으로 ER Diagram을 작성하여 Entity, Relationship에 대한 설명을 기술하고 이를 바탕으로 해당 데이터베이스의 Tree 구조에 대해서 기술한다.

9.2 ER Diagram

9.2.1 Overall ER Diagram

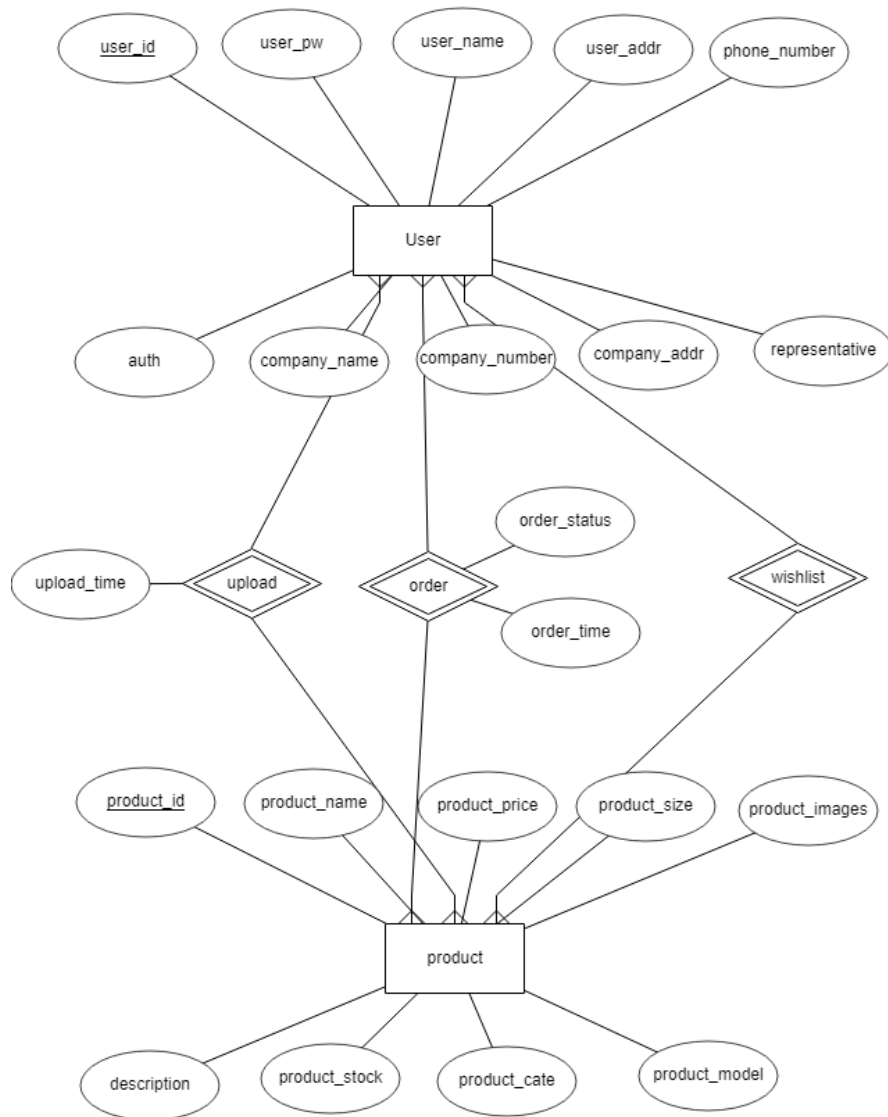


Diagram 20. Overall ER Diagram

해당 데이터베이스는 2개의 Entity(User, Product)와 3개의 Relationship(Upload, Order, Wishlist)로 구성된다.

9.2.2 Entity

A. User

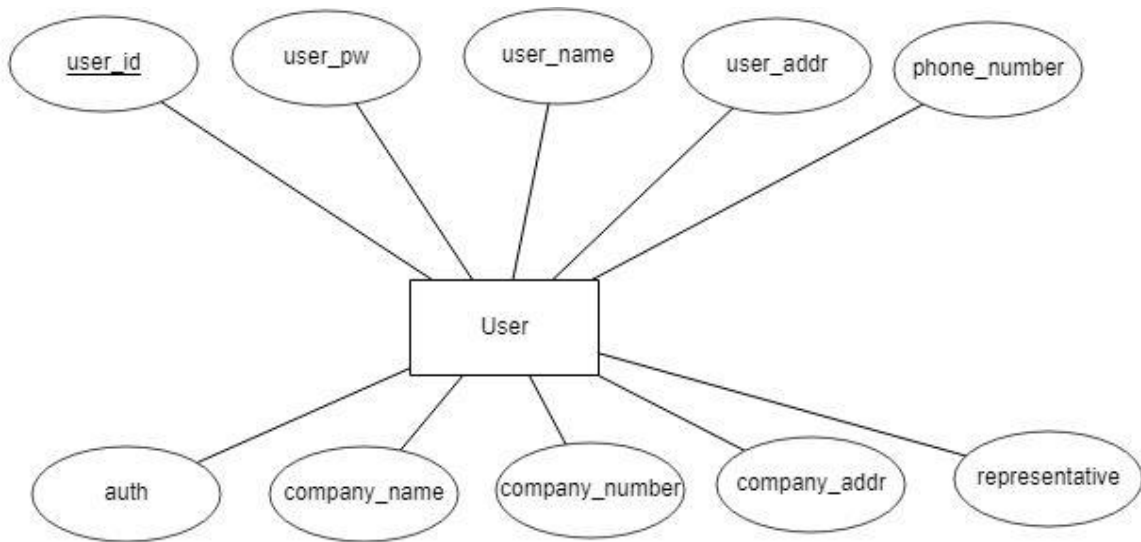


Diagram 21. ER Diagram - User

User는 회원 정보를 나타내는 개체이다. user_id, user_pw, user_name, user_addr, phone_number, auth, company_name, company_number, representative, company_addr의 속성들을 포함하고 있으며, 각 레코드를 구별하는 Key는 user_id이다.

해당 시스템의 Database의 경우 Key-Value 쌍을 기반으로 된 NoSQL을 사용하기 때문에 동적으로 데이터를 구성할 수 있다. 이에 따라 판매자, 사용자는 서로 다른 속성들을 포함할 수 있다.

일반 사용자의 경우 user_id, user_pw, user_name, user_addr, phone_number, auth 만을 포함하고 있으나 판매자의 경우 company_name, company_number, representative, company_addr 등의 판매자 정보를 추가적으로 포함하게 된다. 이외에도 Google 계정을 이용하는 사용자의 경우 user_pw는 포함하지 않는다.

Attribute	Description
user_id	사용자의 ID
user_pw	사용자의 비밀번호
user_name	사용자의 이름
user_addr	사용자의 주소
phone_number	사용자의 전화번호

auth	일반사용자와 판매자의 구분(0: 일반사용자, 1: 판매자)
company_name	판매자의 상호명
company_number	판매자의 사업자등록번호
representative	대표자
company_addr	판매자의 주소

B. Product

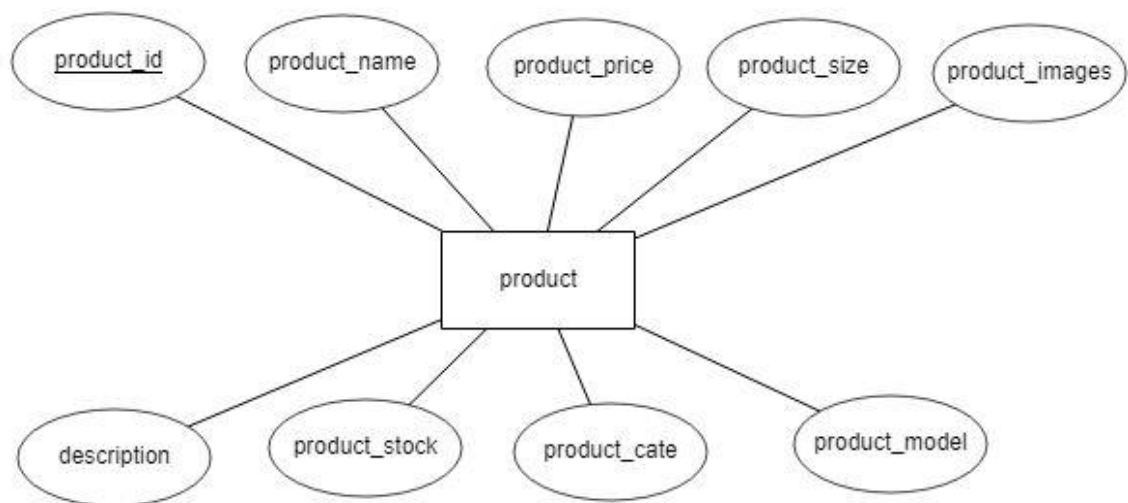


Diagram 22. ER Diagram – Product

Product는 시스템에 등록된 가구의 정보를 나타내는 개체이다. product_id, product_name, product_price, product_size, product_images, product_stock, product_cate, product_model 등의 속성들을 갖고 있으며, Key는 Product_id이다.

Attribute	Description
product_id	가구의 제품번호
product_name	가구의 이름
product_price	가구의 가격
product_size	가구의 규격

product_images	가구의 이미지
product_description	가구에 대한 설명
product_stock	가구의 재고
product_cate	가구의 분류
product_model	가구의 3D 모델

9.2.3 Relationship

A. Order

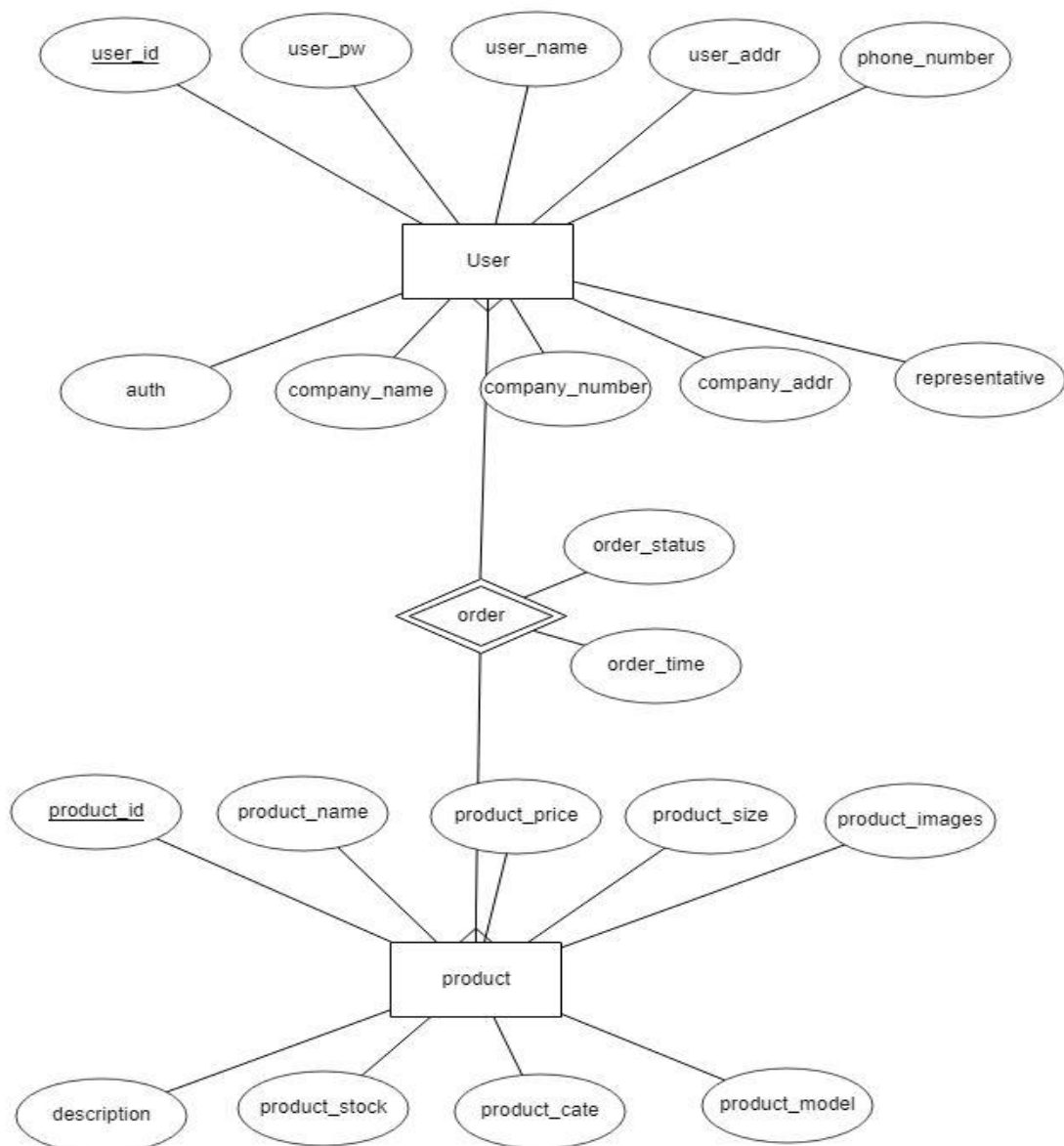


Diagram 23. ER Diagram - Order Relationship

Order Relationship은 User가 주문한 Product를 표현하는 관계이다. User는 여러 개의 Product를 구매할 수 있으며, 한 Product를 여러 User가 주문할 수 있으므로 다대다 관계를 형성한다. 또한 Order Relationship에는 order_status, order_time를 속성으로 포함해야한다.

Attribute	Description
order_status	Order의 상태(0: 준비중, 1: 배송중, 2: 판매완료)
order_time	사용자가 주문을 한 시각을 표시한다.

B. Wishlist

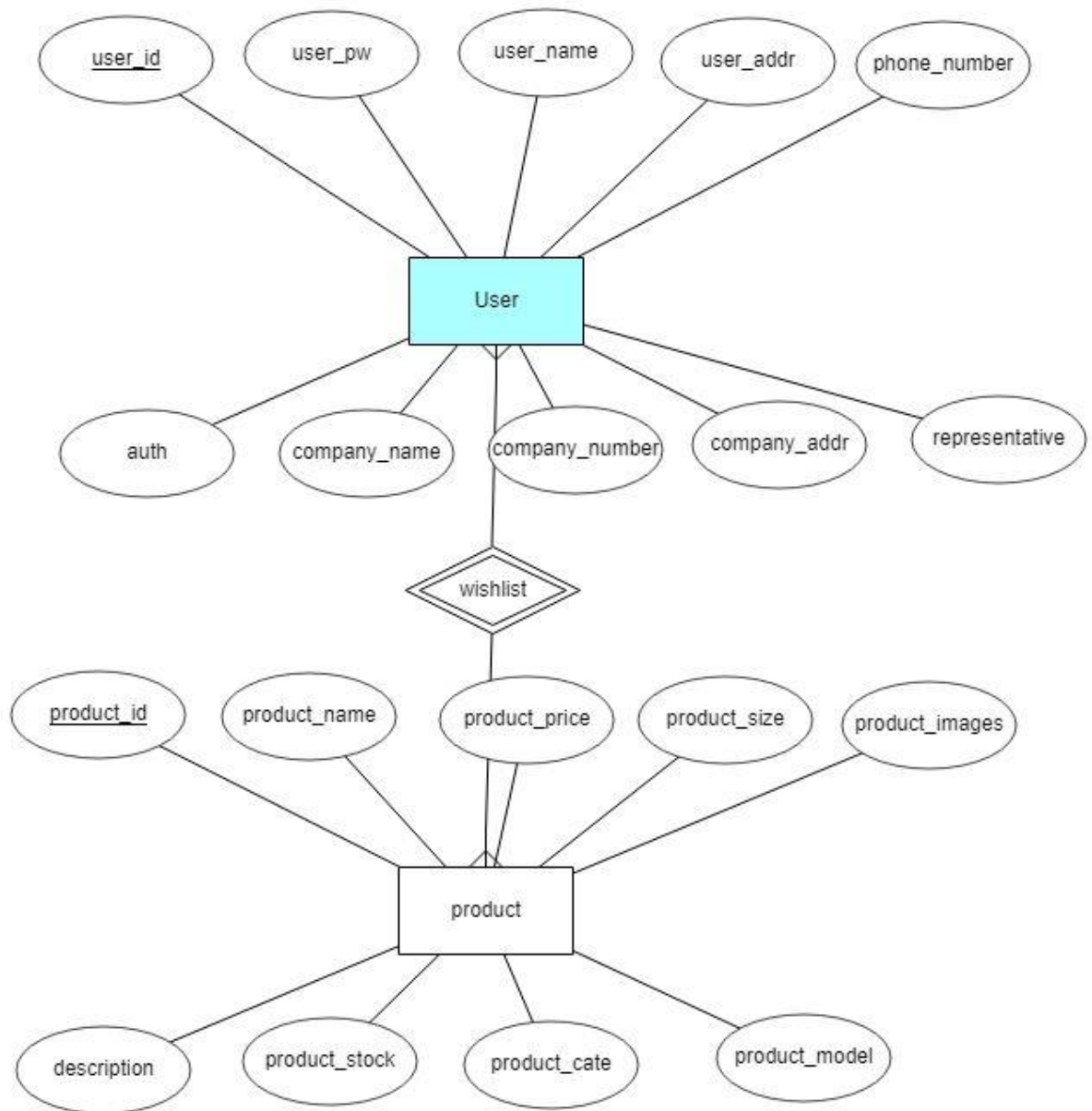


Diagram 24. ER Diagram - Wishlist Relationship

Wishlist Relationship은 User가 관심을 가지고 있는 Product를 보관하는 것을 표현하는 관계이다. User는 여러 개의 Product를 장바구니에 담을 수 있으며, 한 Product는 여러 User에 의해 장바구니에 보관되어질 수 있으므로 다대다 관계를 형성한다.

C. Upload

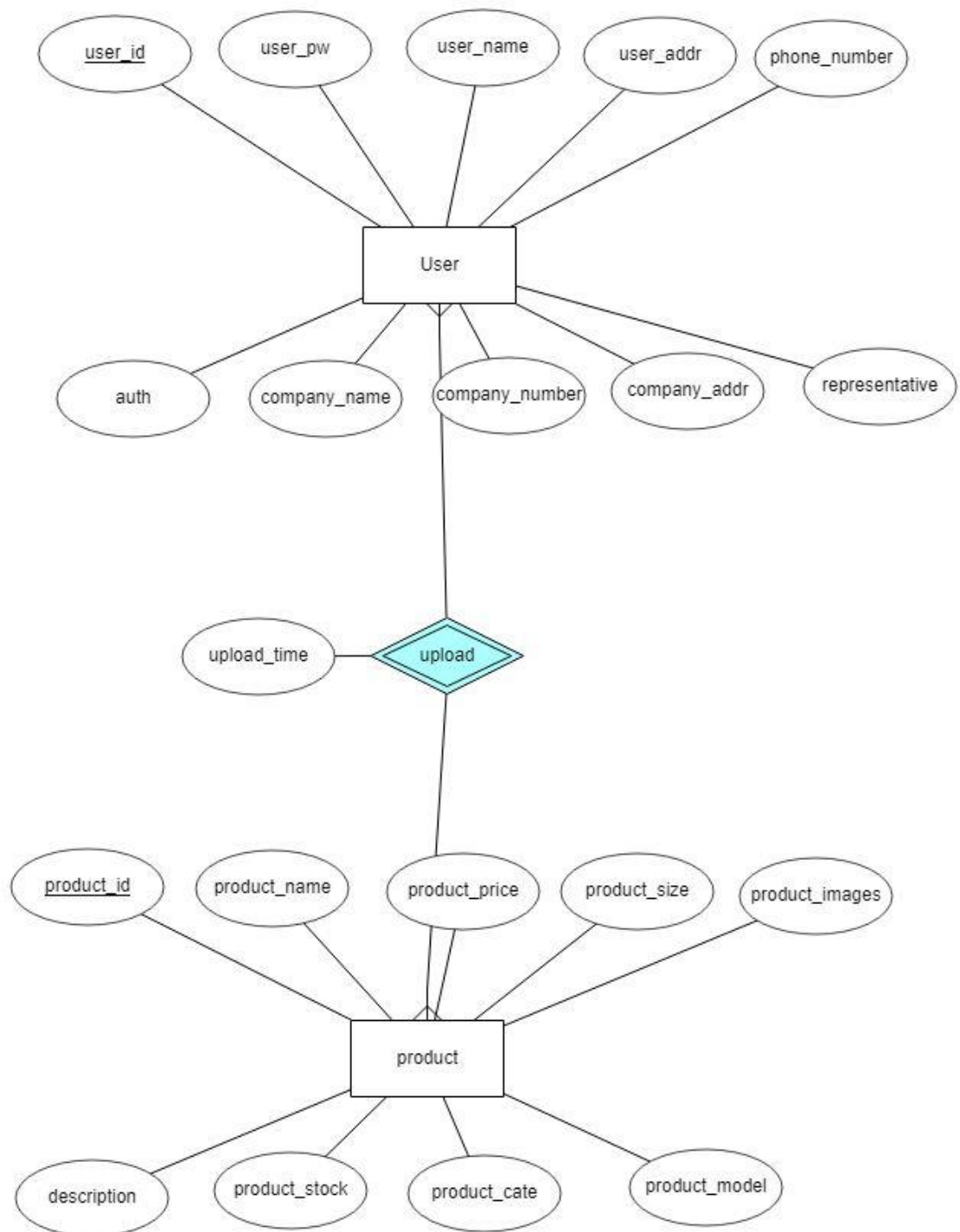


Diagram 25. ER Diagram - Upload Relationship

Upload Relationship은 판매자가 Product를 판매하기 위해 시스템에 Product를 Upload하는 관계를 표현한다. 판매자는 여러 Product를 Upload할 수 있으나, 하나의 Product는 한명의 판매자에 의해서만 업로드되어야 한다.

9.3 Tree

해당 시스템의 Database의 경우 Key-Value 쌍을 기반으로 된 NoSQL을 사용하기 때문에 다음과 같은 트리를 구성하게 된다.

<pre> { user: -- user_id1: ----- user_pw : ... ----- user_name : ... ----- user_addr : ... ----- phone_number : ... ----- auth : ... ----- company_name : ... ----- company_number : ... ----- representative : ... ----- company_addr : ... -- user_id2: ----- ... } { product: -- product_id1: ----- product_id : ... ----- product_name : ... ----- product_price : ... ----- product_size : ... ----- product_images : ... ----- product_stock : ... ----- product_cate : ... ----- product_model : ... -- product_id2: ----- ... </pre>	<pre> { order: -- product_id1: ----- user_id1 : ----- order_status : ----- order_time : ----- user_id2 : ----- order_status : ----- order_time : -- product_id2: ----- ... } { wishlist: -- product_id1: ----- user_id1 : true ----- user_id2 : true -- product_id2: ----- ... } { upload: -- product_id1: ----- user_id1 : ----- upload_time : ----- user_id2 </pre>
---	--

}	----- upload_time : }
---	-------------------------------

10. Developmental Environment

10. 1 Objectives

Development Environment 에서는 개발 환경에 대해 서술한다. 사용된 프로그래밍 언어를 구체적으로 서술한다. 또한 해당 언어와 해당 IDE 가 적합하다고 선정된 이유를 포함한다.

10.2 Programming Language



가치마켓 시스템 개발에 있어서의 개발 언어는 Java 를 사용한다. 이는 개발에 사용되는 IDE인 안드로이드 스튜디오의 기본 언어이기 때문이며 개발자들이 모두 익숙한 언어이기 때문에 선정되었다. 자바 언어는 가장 대표적인 언어 중 하나이기에 오류 탐색 및 개발 과정에서 정보를 구하기 쉽다는 장점도 고려되었다.

10.3 IDE



IDE 는 Integrated Development Environment 의 약자로서 통합 개발 환경이다. 위에 언급된바와같이 안드로이드 앱 개발이기에 Android Studio 가 사용되었다. Android Studio 는 안드로이드 플랫폼의 공식 개발 환경이다. 안드로이드 전용 툴이며 가장 기본적인 툴이라고 할 수 있으며 개발에 필수적이다.

11. Index

11.1. Table Index

<Table 1> User Request	24
<Table 2> User Response	24
<Table 3> Seller Request	24
<Table 4> Seller Response	25
<Table 5> Sign in Request	25
<Table 6> Sign in Response	25
<Table 7> Show Product List Response	26
<Table 8> Show Product Information Request	26
<Table 9> Show Product Information Response	26
<Table 10> Purchase Request	27
<Table 11> Purchase Response	27
<Table 12> Add Wish List Request	27
<Table 13> Add Wish List Response	27
<Table 14> Show User Information User Request	27
<Table 15> Show User Information User Response	27
<Table 16> Show User Information Seller Request	28

<Table 17> Show User Information Seller Response	28
<Table 18> Show Order Information Request	28
<Table 19> Show Order Information Response	29
<Table 20> Show Wish List Request	29
<Table 21> Show Wish List Response	29
<Table 22> Upload Product Request	29
<Table 23> Upload Product Response	30
<Table 24> Change Information Request	30
<Table 25> Change Information Response	30
<Table 26> Seller Registration Request	30
<Table 27> Seller Registration Response	31
<Table 28> Attributes for ER Diagram - User	34
<Table 29> Attributes for ER Diagram - Product	35
<Table 30> Attributes for ER Diagram - Order Relationship	37

11.2. Figure Index

<Figure 1> Draw IO Figure	8
<Figure 2> Android Studio Figure	8
<Figure 3> Firebase Figure	9
<Figure 4> Java Figure	41
<Figure 5> Android Studio Figure	42

11.3. Diagram Index

<Diagram 1> Block Diagram of 가지마켓	9
<Diagram 2> User management system architecture (1)	10
<Diagram 3> User management system architecture (2)	11
<Diagram 4> Furniture management system architecture	12
<Diagram 5> Furniture Buy system architecture	12
<Diagram 6> User management system class diagram	13
<Diagram 7> User management system sequence diagram	15
<Diagram 8> User management system state architecture	16
<Diagram 9> Class diagram of AR View system	17
<Diagram 10> Sequence Diagram of AR View	18
<Diagram 11> State Diagram of AR View System	18

<Diagram 12> Class Diagram for Subscribe System	19
<Diagram 13> Register Furniture Sequence Diagram for Furniture Management	20
<Diagram 14> Wish List Sequence Diagram for Furniture Management	21
<Diagram 15> Register Furniture State Diagram for Furniture Management	21
<Diagram 16> Wish List State Diagram for Furniture Management	22
<Diagram 17> Class Diagram for Furniture Purchase	22
<Diagram 18> Purchase Furniture Sequence Diagram for Furniture Management	24
<Diagram 19> State Diagram for Furniture Purchase	24
<Diagram 20> Overall ER Diagram	32
<Diagram 21> ER Diagram - User	33
<Diagram 22> ER Diagram - Product	35
<Diagram 23> ER Diagram - Order Relationship	36
<Diagram 24> ER Diagram - Wishlist Relationship	38
<Diagram 25> ER Diagram - Upload Relationship	39