
Design Specification

Eat it!



제출일	2019. 05. 19	그룹	Group 2
과목	소프트웨어공학개론	교수	이은석 교수님
이름	노미래	학번	2015318622
이름	김영중	학번	2014315155
이름	문호	학번	2016313320
이름	이재봉	학번	2013311811
이름	장천하	학번	2016314955
이름	장호	학번	2014310022

Contents

1.	Preface.....	9
1.1	Objective.....	9
1.2	Readership	9
1.3	Document Structure	9
A.	Preface.....	9
B.	Introduction	9
C.	System Architecture	9
D.	User Management System.....	10
E.	Taste Analysis System.....	10
F.	Menu Recommendation System.....	10
G.	Searching System.....	10
H.	Item Display System	11
I.	Ordering System.....	11
J.	My Page System	11
K.	Protocol Design	11
L.	Database Design.....	11
M.	Testing Plan	12
N.	Development Environment.....	12
O.	Index.....	12
1.4	Version History.....	12
2.	Introduction	13
2.1	Objective.....	13

2.2	Applied Diagrams	13
A.	Block Diagram	13
B.	UML	13
C.	Class Diagram	14
D.	Sequence Diagram	14
E.	State Diagram	15
F.	ER Diagram	16
2.3	Applied Tools	17
2.4	Project Scope	17
3.	System Architecture	18
3.1	Objective	18
3.2	System Organization	18
A.	User Management System	20
B.	Taste Analysis System	20
C.	Menu Recommendation System	21
D.	Searching System	22
E.	Item Display System	22
F.	Ordering System	23
G.	My Page System	24
4.	User Management System	25
4.1	Objective	25
4.2	Class Diagram	25
A.	Database Handler	25
B.	Evaluation DB	26
C.	User DB	26

4.3	Sequence Diagram.....	27
A.	Signup System.....	27
B.	Login System.....	27
4.4	State Diagram.....	28
5.	Taste Analysis Sytem.....	29
5.1	Objective.....	29
5.2	Class Diagram.....	29
A.	Database Handler	30
B.	First Evaluation	30
C.	Order Record.....	30
D.	Analyze.....	31
5.3	Sequence Diagram.....	32
5.4	State Diagram.....	33
6.	Menu Recommendation System.....	34
6.1	Objective.....	34
6.2	Class Diagram.....	34
A.	Database Handler	35
B.	Recommendation	35
6.3	Sequence Diagram.....	36
6.4	State Diagram.....	37
7.	Searching System	38
7.1	Objective.....	38
7.2	Class Diagram.....	38
A.	Database Handler	39
B.	Product Search	39

C. Category Search	39
7.3 Sequence Diagram.....	40
7.4 State Diagram.....	41
8. Item Display System	42
8.1 Objective.....	42
8.2 Class Diagram.....	42
A. Database Handler	42
B. Recommendation	43
C. Menu DB	43
8.3 Sequence Diagram.....	44
8.4 State Diagram.....	44
9. Ordering System.....	45
9.1 Objective.....	45
9.2 Class Diagram.....	45
A. Database Handler	45
B. Cart	46
C. Payment.....	46
9.3 Sequence Diagram.....	47
9.4 State Diagram.....	48
10. My Page System	49
10.1 Objective.....	49
10.2 Class Diagram.....	49
A. Database Handler	49
B. Order.....	50
C. Evaluation.....	50

10.3	Sequence Diagram.....	51
A.	Order Management	51
B.	Order Evaluation.....	51
10.4	State Diagram.....	52
11.	Protocol Design.....	53
11.1	Objective.....	53
11.2	JSON	53
11.3	Protocol Description.....	54
A.	Overview.....	54
B.	Login Protocol	54
C.	Sign Up	55
D.	Taste Analysis System.....	55
E.	Recommend Menu View Protocol	56
F.	Menu Search Protocol	57
G.	Product Post Protocol	58
H.	Ingredient of Product View Protocol	59
12.	Database Design.....	60
12.1	Objective.....	60
12.2	ER Diagram	60
A.	Entity	61
B.	Relationship	64
12.3	Related Schema	67
A.	User.....	67
B.	Evaluation_Pd	67
C.	Evaluation.....	67

D. Taste Model	68
E. Product.....	68
F. Cart	69
G. Order.....	69
12.4 SQL DDL.....	70
A. User.....	70
B. Evaluation_Pd	70
C. Evaluation.....	71
D. Taste Model	71
E. Product.....	72
F. Cart	72
G. Order.....	73
13. Testing Plan.....	74
13.1 Objective.....	74
13.2 Testing Policy	74
A. Development Testing.....	74
B. Release Testing.....	74
C. User Testing	74
13.3 Test Case.....	75
A. User Management System.....	75
B. Taste Analysis System.....	76
C. Menu Recommendation System.....	76
D. Searching System	77
E. Item Display System	78
F. Ordering System.....	79

G. My Page System	81
14. Development Environment	83
14.1 Objectives	83
14.2 Front-End Environment	83
A. Android Studio	83
B. Java	84
14.3 Back-End Environment.....	84
A. Python and Pycharm	84
B. Django.....	85
14.4 APIs and Tools.....	85
A. TensorFlow.....	85
B. NAVER and Kakao Login API.....	86
C. Github.....	86
15. Index.....	87
15.1 Objective.....	87
15.2 Figure Index.....	87
15.3 Table Index.....	88
15.4 Diagram Index.....	89

1. Preface

1.1 Objective

Preface에서는 본 명세서의 예상 독자를 정의하고 문서의 구조와 각 부분의 역할에 대해 기술한다. 또한 문서 작성 과정에서의 각 version과 각각의 변경사항에 대해 서술한다.

1.2 Readership

본 문서의 독자는 시스템의 개발과 유지를 담당하는 사람들이라고 가정한다. 이는 시스템을 실제로 개발하는 소프트웨어 엔지니어, 시스템을 설계하는 설계자, 클라이언트 엔지니어, 고객 지원을 위한 서비스 팀 등 시스템 개발에 관련되는 모든 구성원들을 포함한다.

1.3 Document Structure

A. Preface

Preface에서는 본 명세서의 예상 독자를 정의하고 문서의 구조와 각 부분의 역할에 대해 기술한다. 또한 문서 작성 과정에서의 각 version과 각각의 변경사항에 대해 서술한다.

B. Introduction

Introduction에서는 본 명세서의 System Design을 위해 사용된 Diagram과 Tool을 소개하고 본 프로젝트의 범위에 대해 논한다.

C. System Architecture

System Architecture에서는 우리 팀에서 개발하고자 하는 시스템의 전반적인 구조에 대해 서술한다. Block Diagram을 통해 시스템의 전체적 구조와 각 Subsystem,

그들 간의 관계를 대략적으로 설명한다.

D. User Management System

User management System 은 사용자의 정보 또는 로그인 및 회원가입을 관리하는 시스템이다. Subsystem인 SignUp System을 통해 회원가입을 하고, Login System을 통해 UserDB에서 사용자 정보를 불러올 수도 있다. Class diagram, Sequence Diagram, state diagram을 통해 User Management System의 구조를 표현하고 설명한다.

E. Taste Analysis System

Taste Analysis System은 'eat it'의 주로 기능 중의 추천 기능을 실현하기 위해 기반으로 필요한 subsystem이다. 사용자의 처음에 평가와 어플을 사용하면서 구매 기록 등 정보를 분석 데이터로 machine learning을 통하여 사용자가 선호할 수 있는 요리, 식품, 입맛을 분석한다. 이 시스템들의 기능과 설계를 설명하고, Class Diagram, Sequence Diagram, State Diagram을 통해 구조를 표현한다.

F. Menu Recommendation System

Recommendation System은 'eat it'이 식품을 구매하는 어플이지만 사용자에게 더 좋고 편한 shopping 경험을 제공하기 위해 사용되는 메뉴 추천기능이다. 사용자가 뭘 먹는지 생각이 없을 경우에 사용자 개인 맞춤형 메뉴를 추천하고 사용자가 필요한 식품을 구매할 수 있다.

G. Searching System

Searching System은 User가 Product를 검색할 때 구동되는 시스템이다. 검색을 할 때, 상품 이름 등 키워드를 검색하는 경우와 카테고리에 따라 검색하는 경우로 나누어 결과값들이 상이하게 나올 수 있다. 따라서 그것을 case로 설정하여 Class Diagram, Sequence Diagram, State Diagram으로, Search System의 구조와 'Eat It'와의 상호 작용을 표현하고 설명한다.

H. Item Display System

Item Display System은 Menu Recommendation System과 Search System으로부터 받은 메뉴를 사용자에게 보여주는 시스템으로, 사용자가 메뉴를 보고 선택하여 Cart에 담을 수 있게끔 한다. 이러한 Item Display System에 대하여 Class Diagram, Sequence Diagram과 State Diagram을 통해 Item Display System의 구조를 표현하고 설명한다.

I. Ordering System

Ordering System은 구매를 원하는 고객과 관련된 시스템으로, 추천된 메뉴나 쿼리를 입력하여 검색된 메뉴를 Cart로 옮겨 담아 사용자가 구매를 할 수 있게끔 한다. 이러한 Ordering System에 대하여 Class diagram, Sequence diagram과 State Diagram을 통해 Ordering System의 구조를 표현하고 설명한다.

J. My Page System

My Page System은 사용자의 주문을 관리하거나 주문한 상품을 평가하는 subsystem이다. Class diagram, Sequence diagram, State Diagram을 통해 My page System의 구조를 표현하고 설명한다.

K. Protocol Design

Protocol Design에서는 'Eat It'에서 각 System간의 communication에 있어서 필요한 protocol에 대해서 기술하고자 한다. Protocol에 필요한 기본적인 data format에는 XML, JSON 등이 있는데 'Eat It'에서는 JSON을 사용한다. 따라서 이에 대한 설명과 각 Protocol에서 사용되는 Attribute와 Value는 무엇인지 자세히 작성된다.

L. Database Design

Database Design은 요구사항 명세서에서 기술한 데이터베이스 요구사항을 바탕으로 작성하였다. 요구사항을 바탕으로 ER Diagram을 작성하고, 이를 통해 Relational Schema와 SQL DDL을 작성한다.

M. Testing Plan

시스템이 의도한 방향으로 실행되고 시스템 내부의 결함을 찾기 위해 testing을 한다. 이를 위해 설치단계에 미리 계획한다. 따라서 Testing Plan에서는 Testing Policy와 여러 Test Case에 대해 기술한다.

N. Development Environment

Development Environment에서는 Programming Language, IDE와 같이 시스템을 개발하기 위해 필요한 Front-end, Back-end의 개발 환경과 외부 API(Application Programming Interface)에 대하여 서술한다.

O. Index

Index에서는 본 문서에서 사용된 그림과 표, 다이어그램에 대한 인덱스를 철자 순서대로 기술한다.

1.4 Version History

Version	Modified Date	Explanation
0.1	2019. 05. 13	설계 명세서 15장에 대한 문서 목차 작성
1.0	2019. 05. 14	1. Preface 작성
2.0	2019. 05. 14	3. System Architecture 작성
3.0	2019. 05. 17	4, 7장을 제외한 항목 일차 작성 완료
4.0	2019. 05. 18	1차 통합, 각 Subsystem의 State Diagram 추가
4.5	2019. 05. 18	양식 통합 수정, 각 단락 내용 추가
5.0	2019. 05. 19	최종 수정, 통합

[Table 1] Version History

2. Introduction

2.1 Objective

Introduction에서는 본 명세서의 System Design을 위해 사용된 Diagram과 Tool을 소개하고 본 프로젝트의 범위에 대해 논한다.

2.2 Applied Diagrams

A. Block Diagram

본 설계 명세서에는 시스템과 각 Subsystem의 전반적인 구조를 표현하기 위해 간단한 비공식 Block Diagram을 사용한다. 이는 특정 System, 혹은 Subsystem의 Entity와 Database, 그들 간의 Interaction과 교환하는 정보를 간략하게 기술한다.

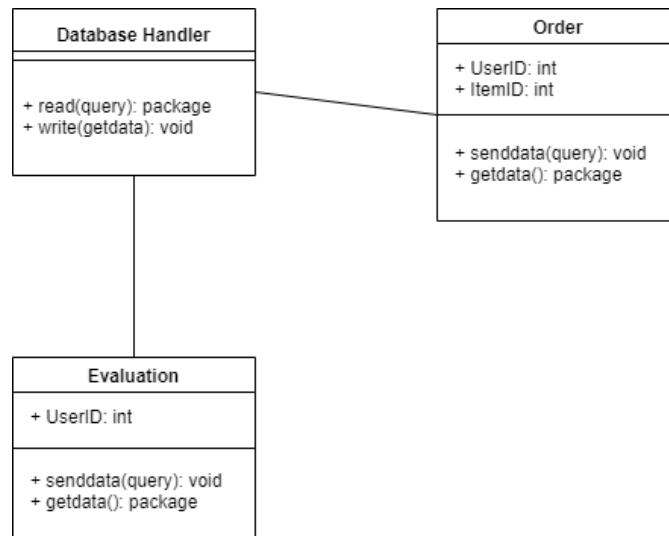
B. UML



[Figure 1] Logo of UML

UML(Unified Modeling Language)은 소프트웨어 공학에서 사용되는 표준화된 범용 modeling 언어로, 현업에서 많이 사용되는 여러 Diagram들을 통합하여 만들어졌다. UML은 Object-Oriented Design을 수행할 때 시스템의 구조를 표준화된 시각적 Model로 표현하여 시스템의 Stakeholder들의 의사소통을 원활히 하기 위해 사용된다. 본 설계 명세서에는 UML의 Class Diagram, Sequence Diagram, State Diagram을 사용하여 개발할 시스템의 구조를 시각화 한다.

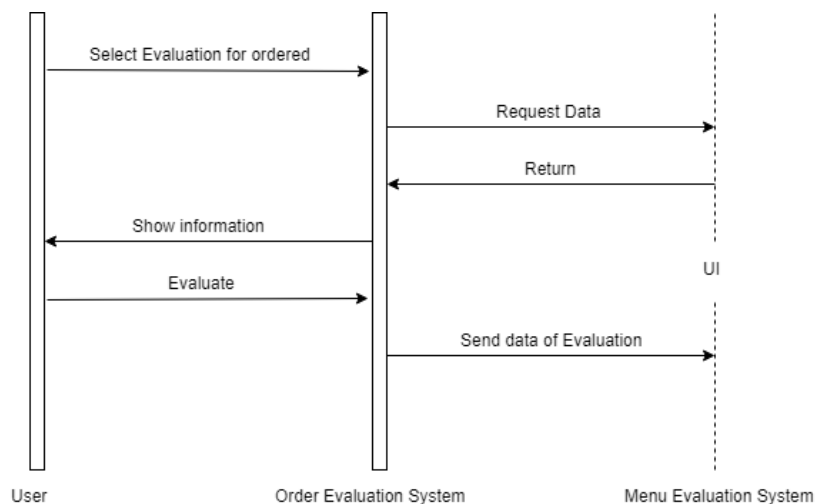
C. Class Diagram



[Figure 2] Example of Class Diagram

본 설계 명세서에서는 Class Diagram을 이용하여 시스템의 object들을 class로 표현한다. Class Diagram은 OOD(Object-oriented Design)의 관점에서 시스템의 정적 구조를 표현하는 UML Diagram으로, 시스템의 class들과 그들 사이의 관계인 Association을 나타낸다. 여기서 Class란 System Object의 한 종류에 대한 보편적인 정의이며, Association이란 Class들 사이의 관계를 나타내는 link를 의미한다.

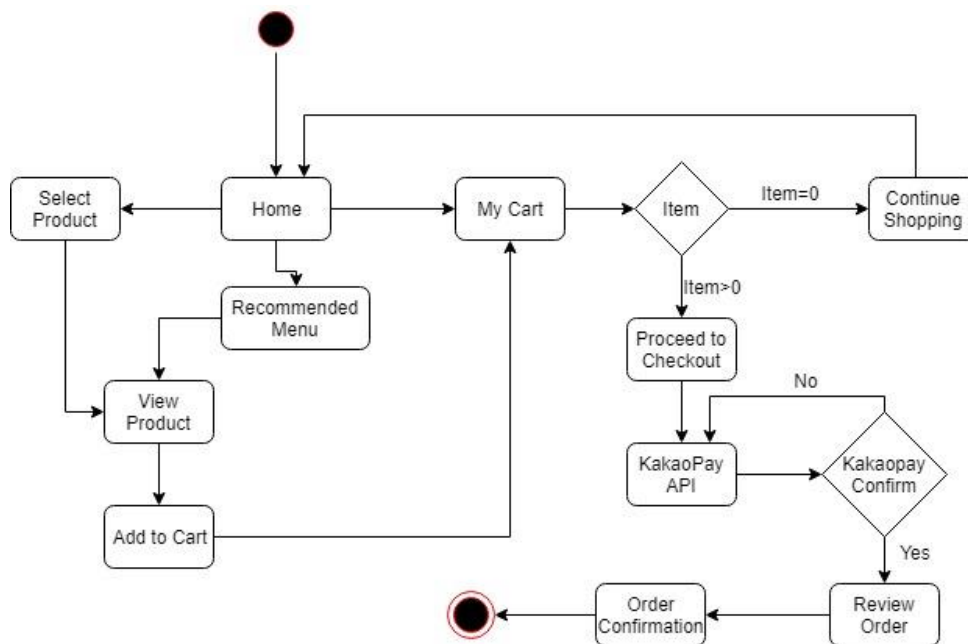
D. Sequence Diagram



[Figure 3] Example of Sequence Diagram

본 설계 명세서에서는 Sequence Diagram을 이용하여 시스템의 구성요소들 간의 상호작용을 표현한다. Sequence Diagram이란 시스템과 그의 Actor, 그리고 System Components간의 상호작용을 나타내는 UML Diagram이다. 이는 시스템의 특정 use case에서 일어나는 상호작용들의 sequence로 표현된다.

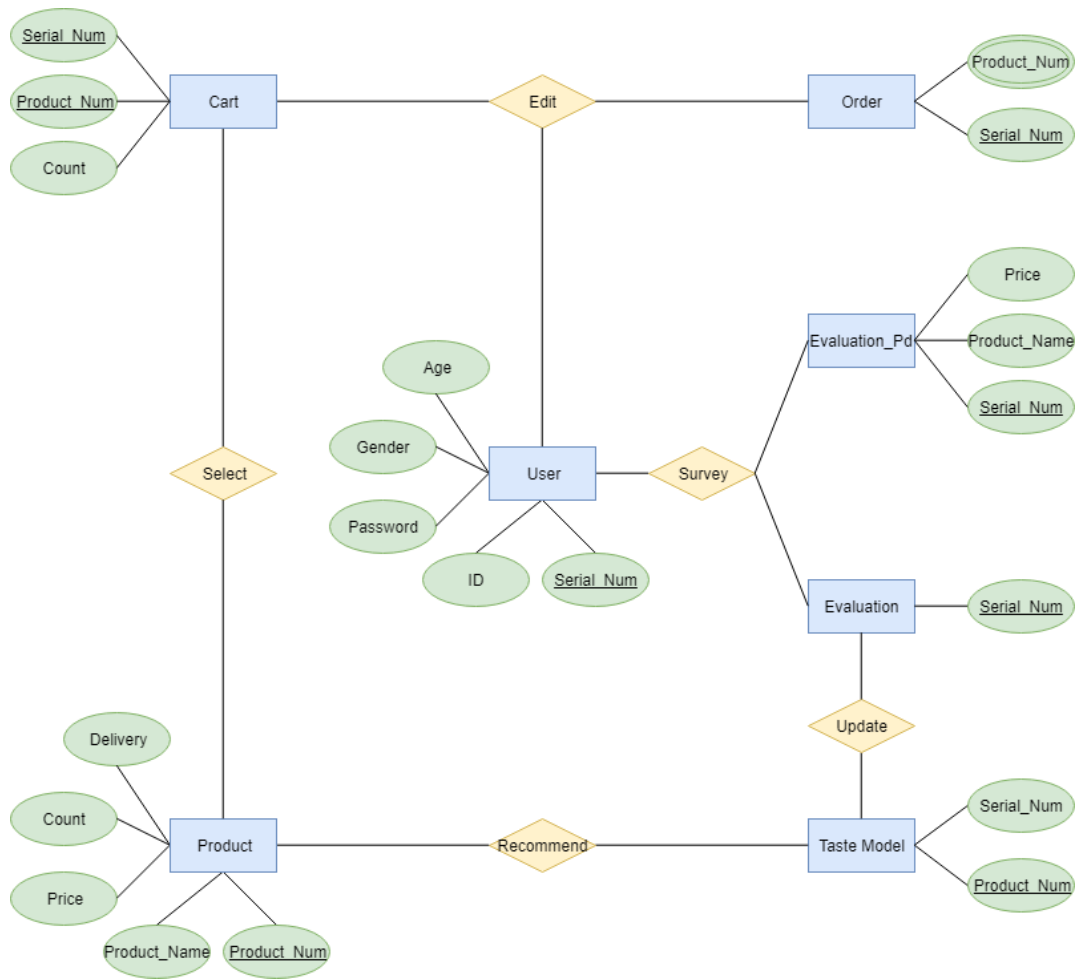
E. State Diagram



[Figure 4] Example of State Diagram

본 설계 명세서에서는 State Diagram을 이용하여 시스템의 내부, 외부에서 일어나는 Event들에 대한 시스템의 반응을 표현한다. State Diagram은 OOD의 관점에서 시스템의 동적 반응을 설명하기 위해 사용되는 Diagram으로, Class의 Event에 대한 시스템의 전체적인 반응, 즉 작동을 상세하게 기술하여 나타낸다. 본래의 상태에서 특정 Event가 발생하였을 때 시스템의 동작과 함께 클래스의 상태(State)가 변하게 되는데, 이 때의 동작과 상태의 변화를 기술한다.

F. ER Diagram



[Figure 5] Example of ER Diagram

Entity-Relationship(ER) Model은 Database Design Process중 데이터베이스를 Table과 Constraints로 나타내는 Relational Model 이전에 수행되는 단계로, 데이터베이스를 Entity와 Entity들의 Relationship으로 나타내는 Conceptual Model이다. ER Model에서 Entity는 Real World에서의 특정한 Object를 Database에 표현한 것이고, Attribute는 Entity를 나타내기 위해 사용되는 특성이며, Relationship은 다수의 Entity들을 구체적인 의미를 통해 연결하는 것이다. 본 문서의 Database Design에서 사용되는 ER Diagram은 위의 ER Model을 시각화 하기위해 사용하는 Diagram이다.

2.3 Applied Tools



[Figure 6] Logo of draw.io

본 설계 명세서의 모든 System Architecture와 Diagram들은 'draw.io'를 이용하여 작성했다. 'draw.io'는 온라인 환경이라면 누구나 다양한 목적의 Flowchart와 Diagram을 작성할 수 있는 Online Diagram Tool이다. 접근성이 뛰어나고 사용법이 간단하기 때문에 자체적인 Diagram Tool을 보유하지 않은 회사에서 많이 사용된다.

2.4 Project Scope

'Eat it' 시스템은 사용자들이 하나하나 상품을 선택해야 했던 기존의 식품 쇼핑 서비스들의 단점을 보완하여 온라인 식품 쇼핑 시스템에 사용자 취향 기반 상품 추천 시스템을 접목한 서비스이다. 따라서 식품 쇼핑 기능과 취향 기반 추천 기능이 가장 중요한 System Objective이며, 7개의 Subsystem들의 상호작용으로 이 기능들을 구현하도록 설계되었다. 따라서 본 프로젝트의 구현 범위는 다음과 같다.

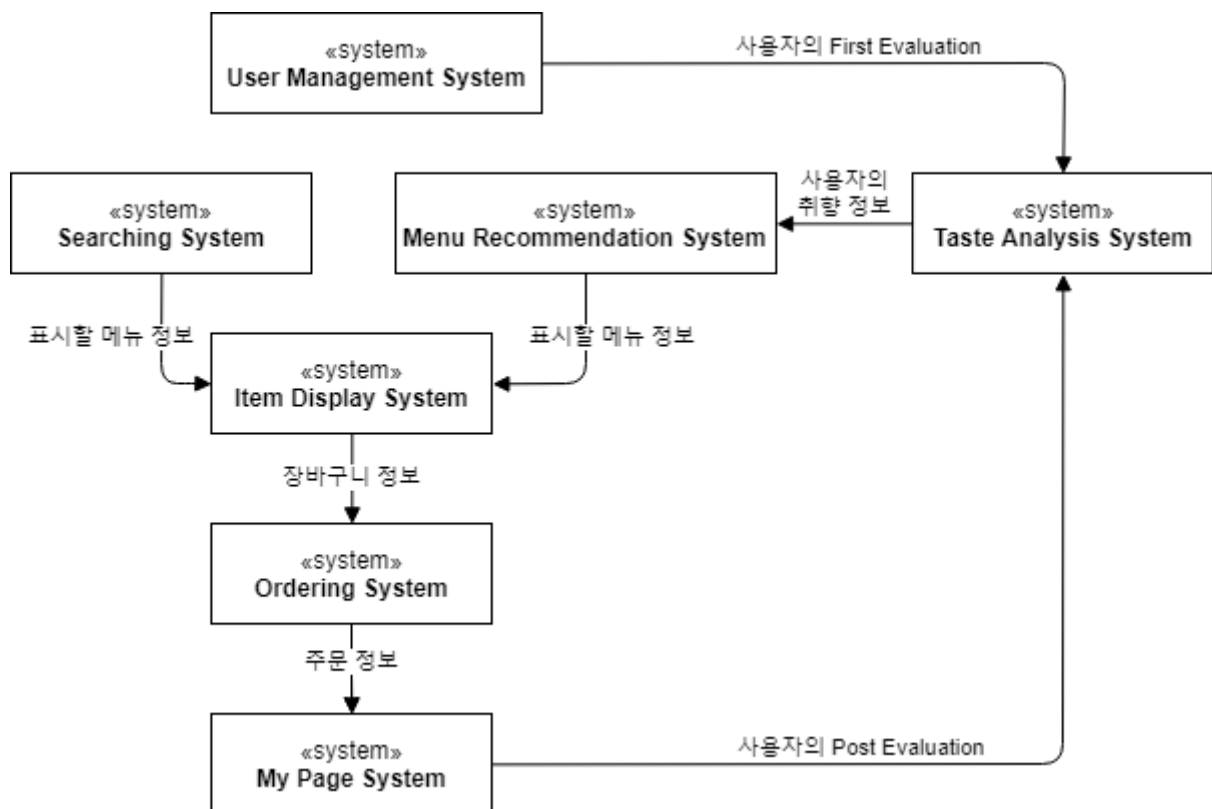
'User Management System'으로 사용자의 로그인 정보를 관리하며, 이는 회원가입과 로그인 기능을 담당한다. 'Taste Analysis System'으로 사용자의 음식 취향 모델을 생성, 학습한다. 여기서 생성된 취향 모델을 기반으로 'Menu Recommendation System'에서 적합한 메뉴를 제안한다. 또한 사용자는 'Searching System'을 이용하여 구매하고 싶은 특정 상품을 검색할 수도 있다. 'Menu Recommendation System'이나 'Searching System'의 Output으로 도출된 메뉴들의 목록을 'Item Display System'이 출력한다. 사용자가 마음에 드는 상품을 장바구니에 담으면 'Ordering System'을 통해 상품들을 주문할 수 있다. 마지막으로 'My Page System'으로 주문한 내역을 관리하거나, 주문한 상품을 평가하여 취향 모델을 업데이트한다.

3. System Architecture

3.1 Objective

System Architecture에서는 우리 팀에서 개발하고자 하는 시스템의 전반적인 구조에 대해 서술한다. Block Diagram을 통해 시스템의 전체적 구조와 각 Subsystem, 그들 간의 관계를 대략적으로 설명한다.

3.2 System Organization



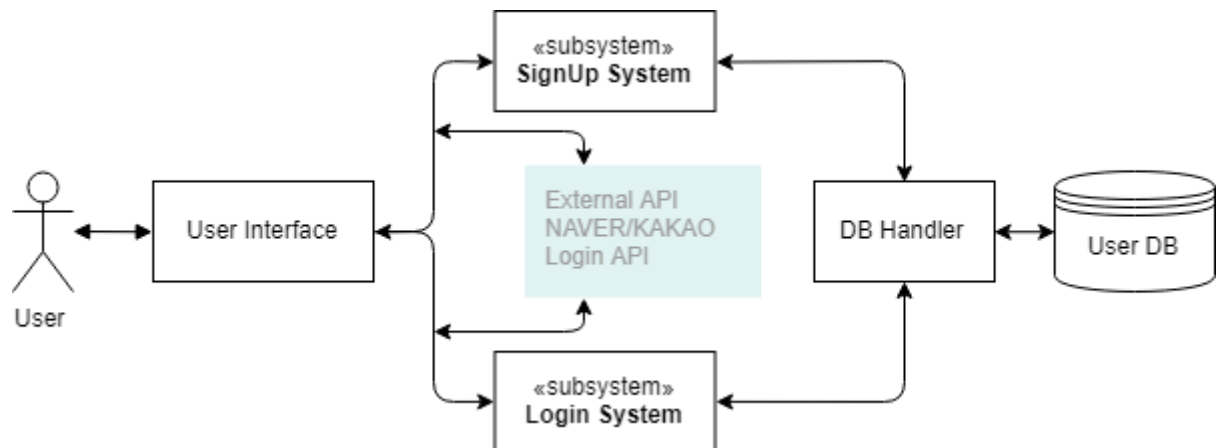
[Diagram 1] System Organization Block Diagram

"Eat it!" 서비스는 Client-Server Model을 사용하여 구현된다. 사용자는 Client의 User Interface와 상호작용하고, 이에 따른 입력을 각 Subsystem들이 처리하여 다시 Client에서 필요한 결과를 출력한다. Server는 Client에게서 사용자의 Menu Evaluation 정보를 얻어와 Deep Learning 학습을 통해 사용자의 취향 분석 Model

을 생성, 업데이트하며 Client에서 출력되어야 하는 정보를 Database에서 전달하는 역할을 한다.

위의 그림은 'Eat it!' 시스템의 전반적인 구조를 조직화한 Block Diagram이다. 메뉴 추천 기능의 핵심인 Server의 Taste Analysis System은 Client의 User Management System과 My Page System으로부터 사용자의 메뉴 평가 정보를 받아 Deep Learning 학습으로 사용자의 취향 분석 Model을 생성, 업데이트한다. Server의 Menu Recommendation System은 Taste Analysis System으로부터 취향 분석 Model을 받아 판매 가능한 상품들의 Database에서 사용자가 좋아할 만한 Item들을 선택해 Client의 Item Display System으로 전달한다. Item Display System은 사용자가 원하는 특정 메뉴를 검색하는 Searching System으로부터도 표시할 메뉴 정보를 받을 수 있다. Item Display System은 Ordering System에게 사용자가 장바구니에 담은 Item의 정보를 전달하고, Ordering System은 장바구니와 주문 과정을 거쳐 결제가 완료된 주문 정보를 My Page System에게 전달한다. My Page System은 사용자에게 주문 내역 조회와 주문한 Item의 평가 기능을 제공한다.

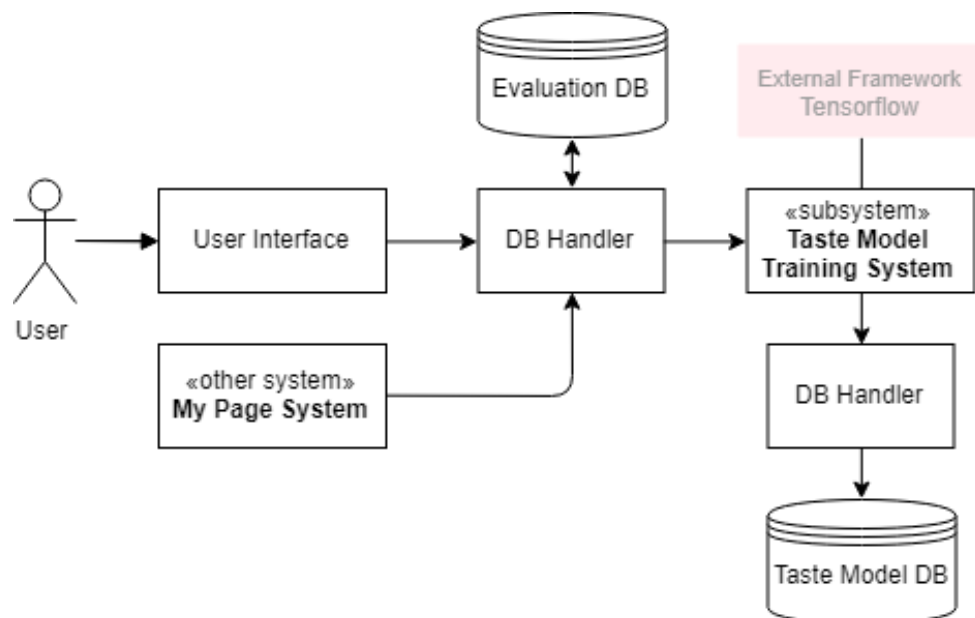
A. User Management System



[Diagram 2] User Management System Block Diagram

User Management System은 사용자 정보를 관리하는 시스템으로, 회원 가입을 수행하는 Signup과 로그인/로그아웃 기능을 제공하는 Login 총 2개의 Subsystem들로 구성된다. NAVER/KAKAO 아이디로 로그인 기능을 제공하므로 두 Subsystem은 NAVER Login API, KAKAO Login API와 연동된다.

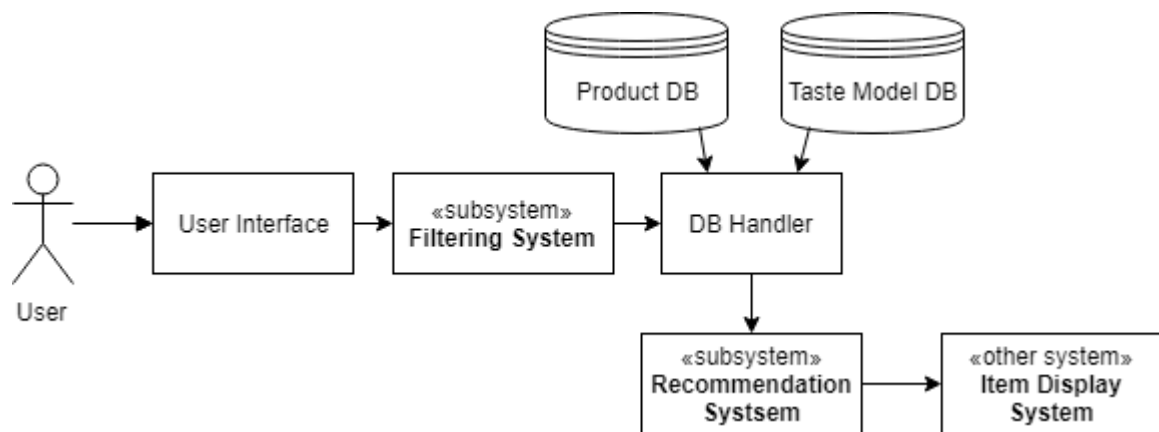
B. Taste Analysis System



[Diagram 3] Taste Analysis System Block Diagram

Taste Analysis System은 사용자의 메뉴 평가 정보를 통해 사용자의 취향 학습 Model을 생성, 업데이트 하는 시스템이다. 이는 Server단에서 동작하며 새 사용자가 가입 과정에서 평가한 first evaluation data 혹은 상품 구매 후 My Page System에서 평가한 Post evaluation data를 통해 Evaluation DB를 업데이트하고, 이 데이터베이스의 평가 정보들에 Deep-Learning을 적용하여 사용자의 취향 학습 Model을 생성/업데이트한다. Deep-Learning을 적용하기 위해 Tensorflow Framework를 사용한다.

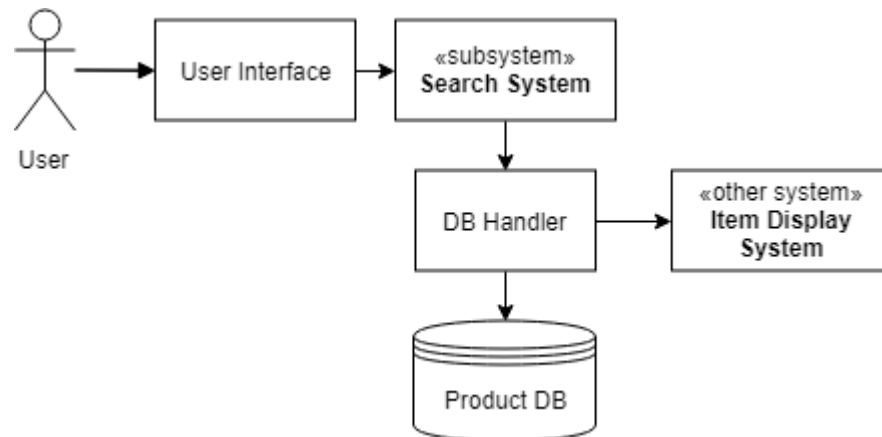
C. Menu Recommendation System



[Diagram 4] Menu Recommendation System Block Diagram

Menu Recommendation System은 판매하는 상품 중 사용자가 좋아할 상품의 정보를 골라 Item Display System에 전달하는 시스템이다. 이 시스템은 Filtering, Recommendation의 두가지 Subsystem으로 구성된다. 사용자가 상품에 필터링을 적용하면 Filtering System을 거쳐 Product DB에 접근한다. Recommendation System에서 Taste Model DB를 이용하여 Product DB의 entity들 중 사용자가 좋아할 entity들을 가져온다. 가져온 상품들의 목록을 Item Display System에 전달한다.

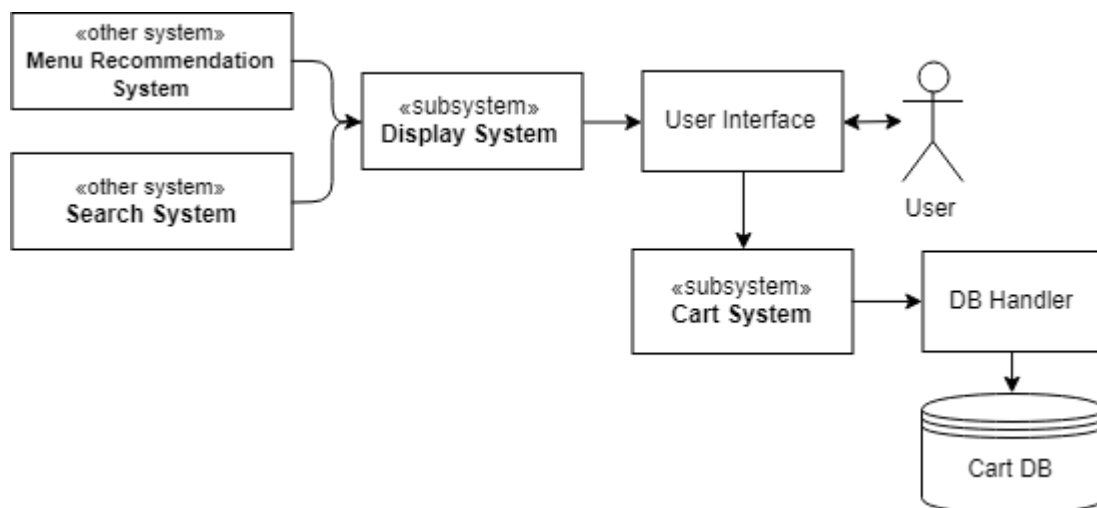
D. Searching System



[Diagram 5] Searching System Block Diagram

Searching System은 사용자가 검색 UI에 입력한 특정 상품을 검색하여 Item Display System에 전달하는 시스템이다. 이 시스템은 Search의 한 가지 Subsystem으로 구성된다. 사용자가 UI에 입력한 키워드를 받은 Search System이 Product DB에서 키워드와 일치하는 상품의 정보를 찾아 Item Display System에 전달한다.

E. Item Display System

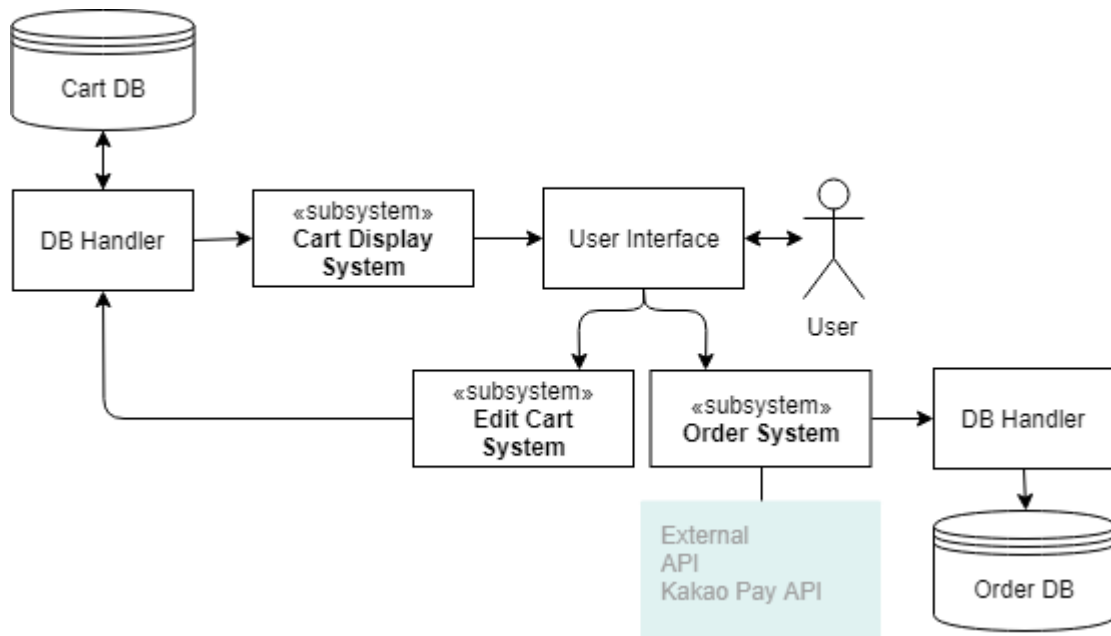


[Diagram 6] Item Display System Block Diagram

Item Display System은 상품들의 목록을 UI에 출력하고 사용자가 상품을 장바구니에 담으면 해당 상품의 정보를 Cart DB에 저장하는 시스템이다. 이 시스템은

Display와 Cart의 두 Subsystem으로 구성된다. Subsystem인 Display System 은 Menu Recommendation System 혹은 Search System에서 화면에 표시할 상품들의 목록을 받아 UI에 출력한다. 사용자가 특정 상품을 장바구니에 담으면 Cart System 이 사용자가 담은 상품의 정보를 Cart DB에 저장한다.

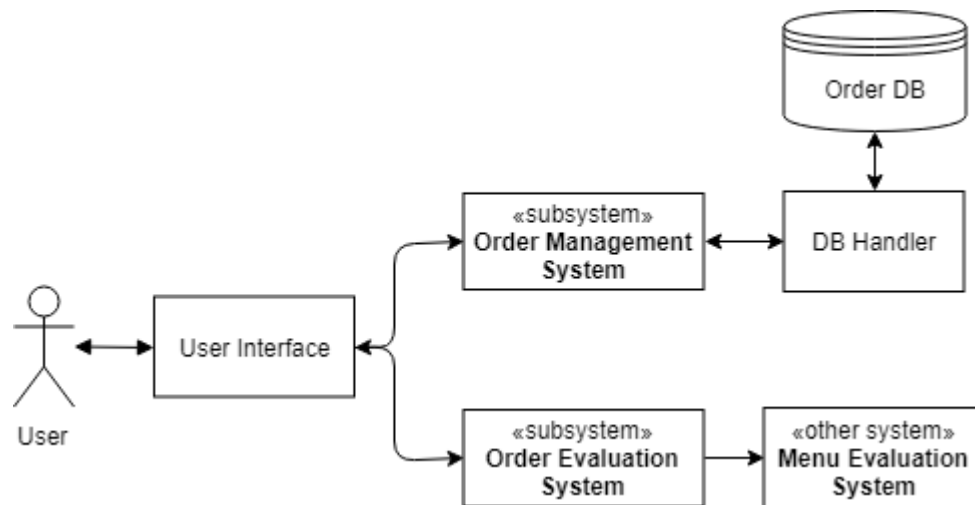
F. Ordering System



[Diagram 7] Ordering System Block Diagram

Ordering System은 사용자의 장바구니 정보를 UI에 출력하고 사용자의 선택에 따라 장바구니 정보를 수정하거나 주문을 진행하는 시스템이다. 이 시스템은 Cart Display, Edit Cart, Order의 세 가지 Subsystem으로 구성된다. 먼저 Cart Display System이 Cart DB에서 장바구니 정보를 읽어와 UI에 출력하고, 사용자가 장바구니의 정보를 수정하면 Edit Cart System이 Cart DB를 업데이트한다. 사용자가 주문을 결정하면 Order System을 통해 주문과 결제가 진행되고, Order DB를 업데이트한다. Order System은 Kakao Pay API와 연동되어 결제 기능을 제공한다.

G. My Page System



[Diagram 8] My Page System Block Diagram

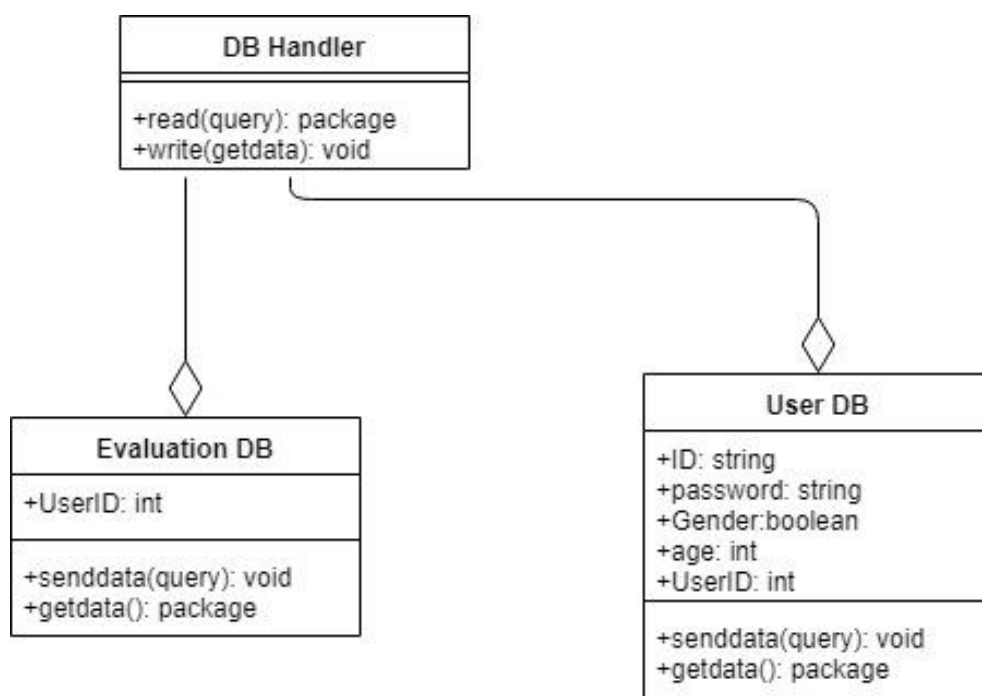
My Page System은 사용자의 주문을 관리하거나 주문한 상품을 평가하는 시스템이다. 이 시스템은 Order Management와 Order Evaluation의 두 가지 Subsystem으로 구성된다. 사용자가 UI를 통해 '주문 관리'를 선택하면 Order Management System이 Order DB에서 주문 정보를 읽어와 UI에 출력하고, 배송 전 상태일 시에는 사용자가 주문을 취소할 수 있다. 사용자가 UI를 통해 주문 메뉴 평가를 선택하면 Order Evaluation System이 UI로부터 사용자의 평가 정보를 얻어와 Menu Evaluation System에 전달한다.

4. User Management System

4.1 Objective

User management System 은 사용자의 정보 또는 로그인 및 회원가입을 관리하는 시스템이다. Subsystem인 SignUp System을 통해 회원가입을 하고, Login System을 통해 UserDB에서 사용자 정보를 불러올 수도 있다. Class diagram, Sequence Diagram, state diagram을 통해 User Management System의 구조를 표현하고 설명한다.

4.2 Class Diagram



[Diagram 9] Class Diagram of User Management System

A. Database Handler

A.1 Attributes

없음

A.2 Methods

+package read(query): DB에서 원하는 data를 가져온다.

+void write(package): DB에 data를 저장한다.

B. Evaluation DB

B.1 Attributes

+UserID: 사용자 고유 번호

B.2 Methods

+senddata(query):void : DB에 data를 보낸다.

+getdata():package : DB에서 data를 가져온다.

C. User DB

C.1 Attributes

+ID:string : 사용자 아이디

+password:string : 사용자 비밀번호

+gender:Boolean : 사용자의 성별, 여성 혹은 남성

+age:int : 사용자의 나이

+UserID: 사용자 고유 번호

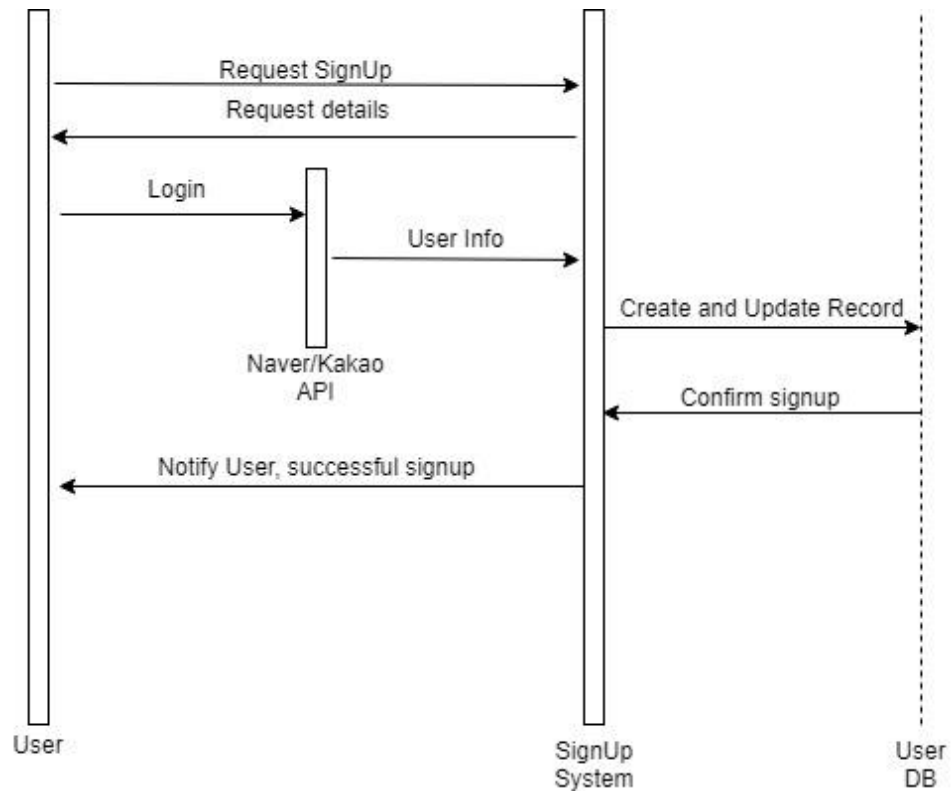
C.2 Methods

+senddata(query):void : DB에 data를 보낸다.

+getdata():package : DB에서 data를 가져온다.

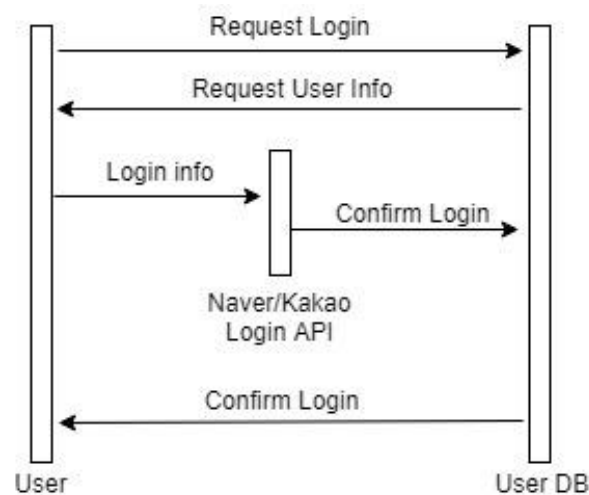
4.3 Sequence Diagram

A. Signup System



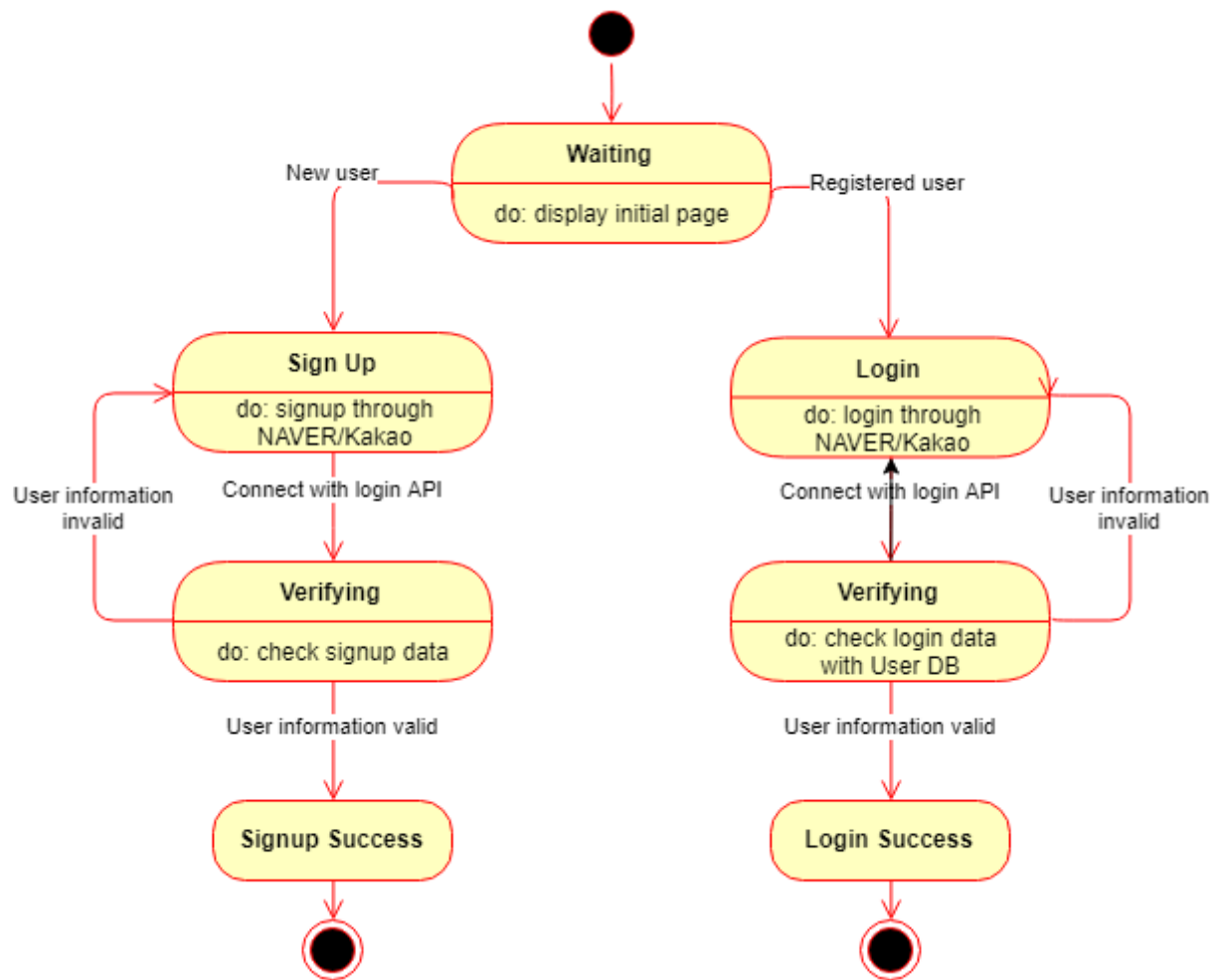
[Diagram 10] Sequence Diagram of Signup System

B. Login System



[Diagram 11] Sequence Diagram of Login System

4.4 State Diagram



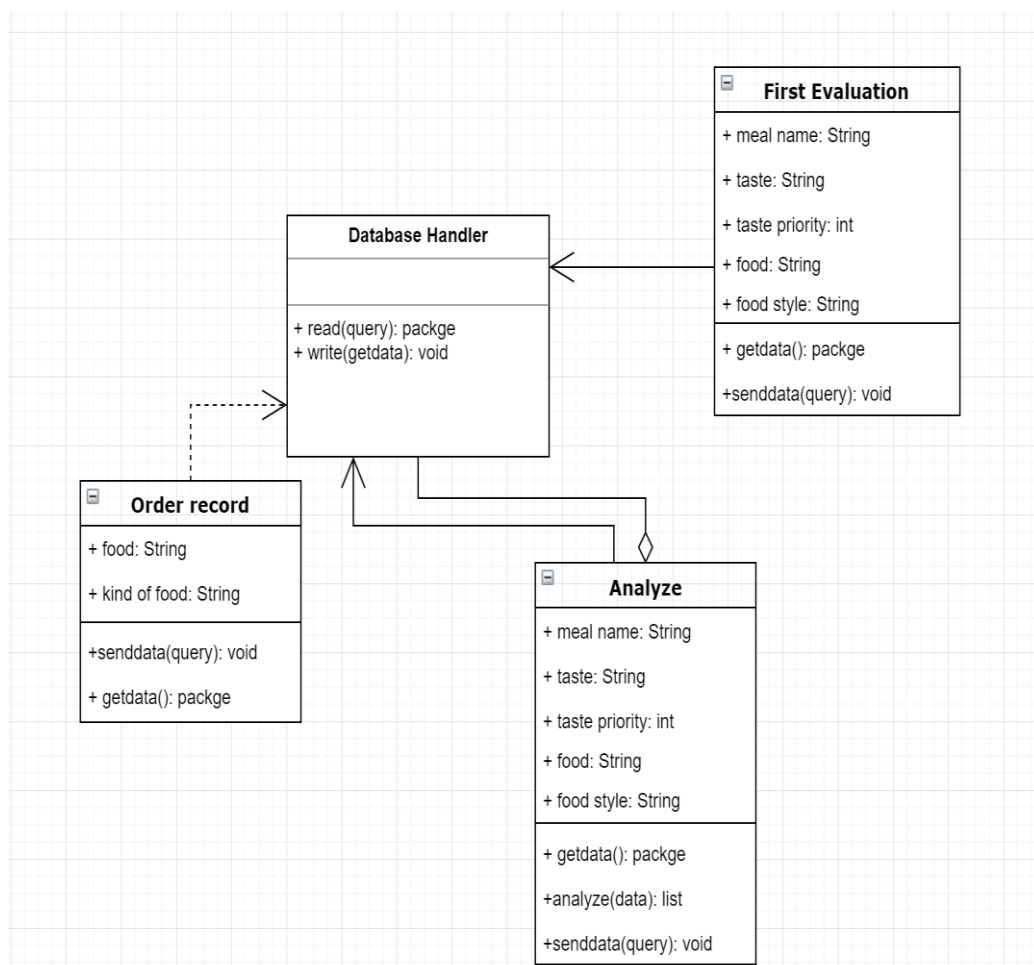
[Diagram 12] State Diagram of User Management System

5. Taste Analysis Sytem

5.1 Objective

Taste Analysis System은 'eat it'의 주요 기능 중의 추천 기능을 실현하기 위해 기반으로 필요한 subsystem이다. 사용자의 처음에 평가와 어플을 사용하면서 구매 기록 등 정보를 분석 데이터로 machine learning을 통하여 사용자가 선호할 수 있는 요리, 식품, 입맛을 분석한다. 이 시스템들의 기능과 설계를 설명하고, Class Diagram, Sequence Diagram, State Diagram을 통해 구조를 표현한다.

5.2 Class Diagram



[Diagram 13] Class Diagram of Taste Analysis System

A. Database Handler

A.1 Attributes

없음

A.2 Methods

- + package read (query): 해당되는 DB에서 원하는 데이터를 읽어온다.
- + void write(data): 해당되는 DB에 데이터를 저장한다.

B. First Evaluation

B.1 Attributes

- + meal name: 요리의 이름
- + taste: 사용자의 선호하는 맛
- + taste priority: 숫자 1~10로 선호도를 표시
- + food: 사용자가 좋아하는 식품
- + food style: 사용자가 좋아하는 식품 종류

B.2 Methods

- + getdata() : 해당되는 DB에서 정보를 가져온다.
- + senddata (data) : 해당되는 DB에 정보를 보내준다.

C. Order Record

C.1 Attributes

- + food: 식품 이름
- + kind of food: 식품 종류

C.2 Methods

- + getdata() : 해당되는 DB에서 정보를 가져온다.

- + senddata (data) : 해당되는 DB에 정보를 보내준다.

D. Analyze

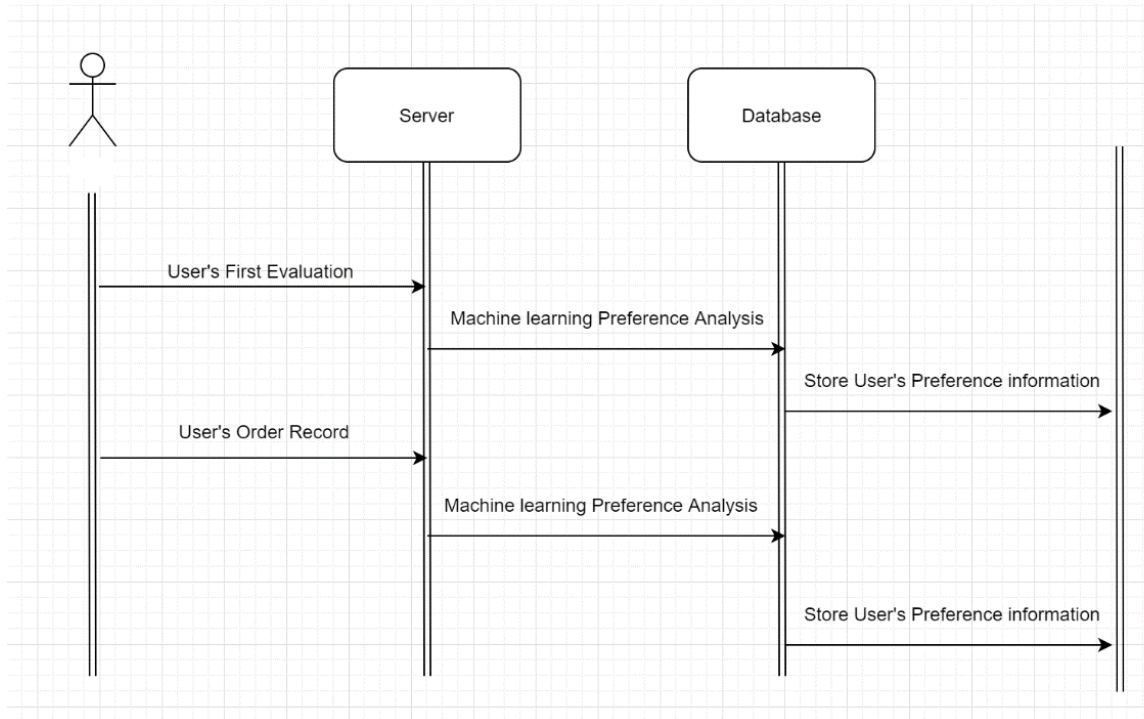
D.1 Attributes

- + meal name: 사용자가 좋아하는 요리의 이름
- + taste: 사용자의 선호하는 맛
- + taste priority: 숫자 1~10로 선호도를 표시
- + food: 사용자가 좋아하는 식품
- + food style: 사용자가 좋아하는 식품 종류

D.2 Methods

- + getdata() : 해당되는 DB에서 정보를 가져온다.
- + senddata (data) : 해당되는 DB에 정보를 보내준다.
- + analyze(data): 사용자가 선호하는 식품, 입맛, 요리 종류를 분석한다.

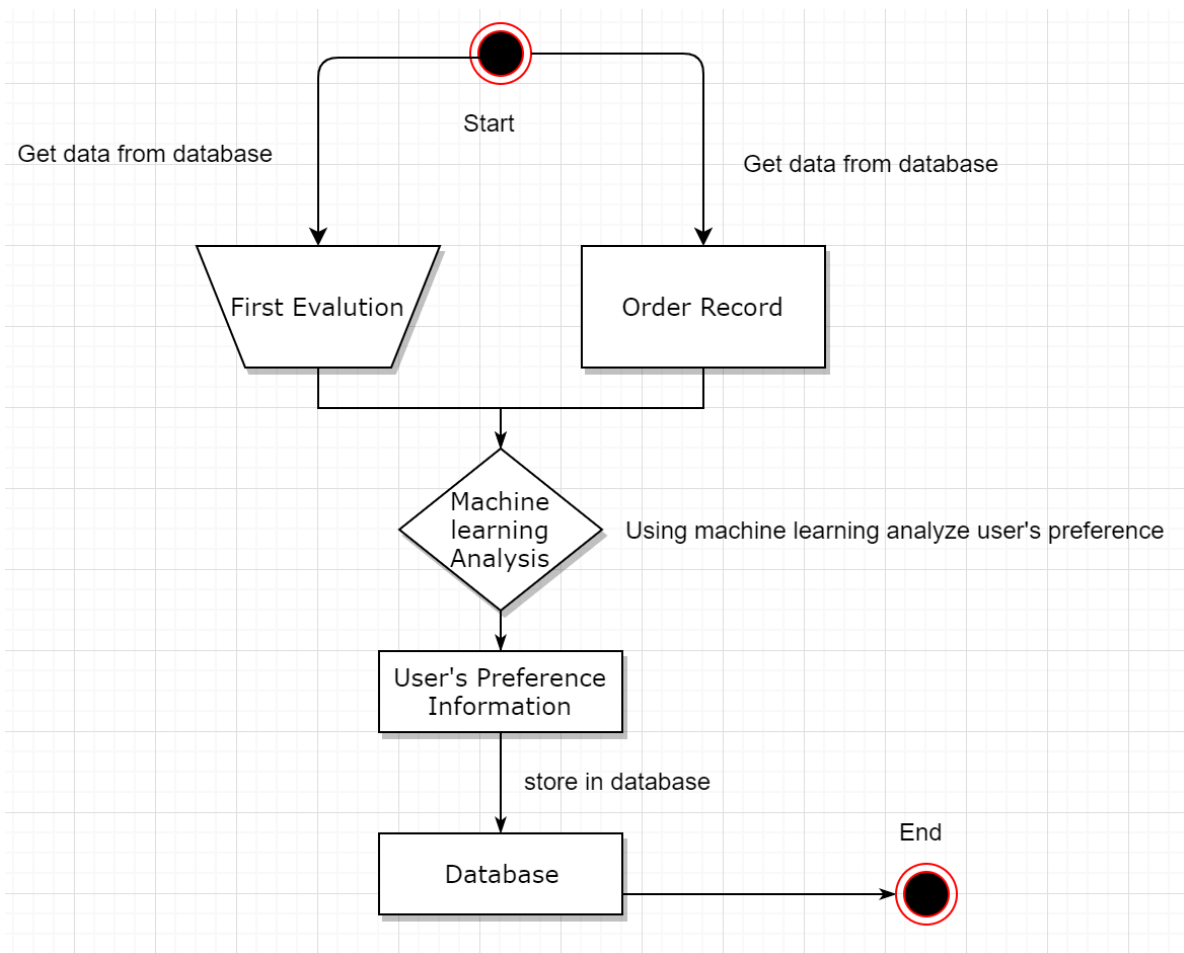
5.3 Sequence Diagram



[Diagram 14] Sequence Diagram of Taste Analysis System

사용자의 가입 후 초기 평가 데이터와 'eat it' 시스템에서의 구매 평가를 분석정보로 받고, Back-End에서 Deep Learning을 이용하여 사용자의 입맛, 좋아하는 식품, 식사습관 등을 포함하는 취향 모델을 생성, 업데이트하여 데이터베이스에 저장한다.

5.4 State Diagram



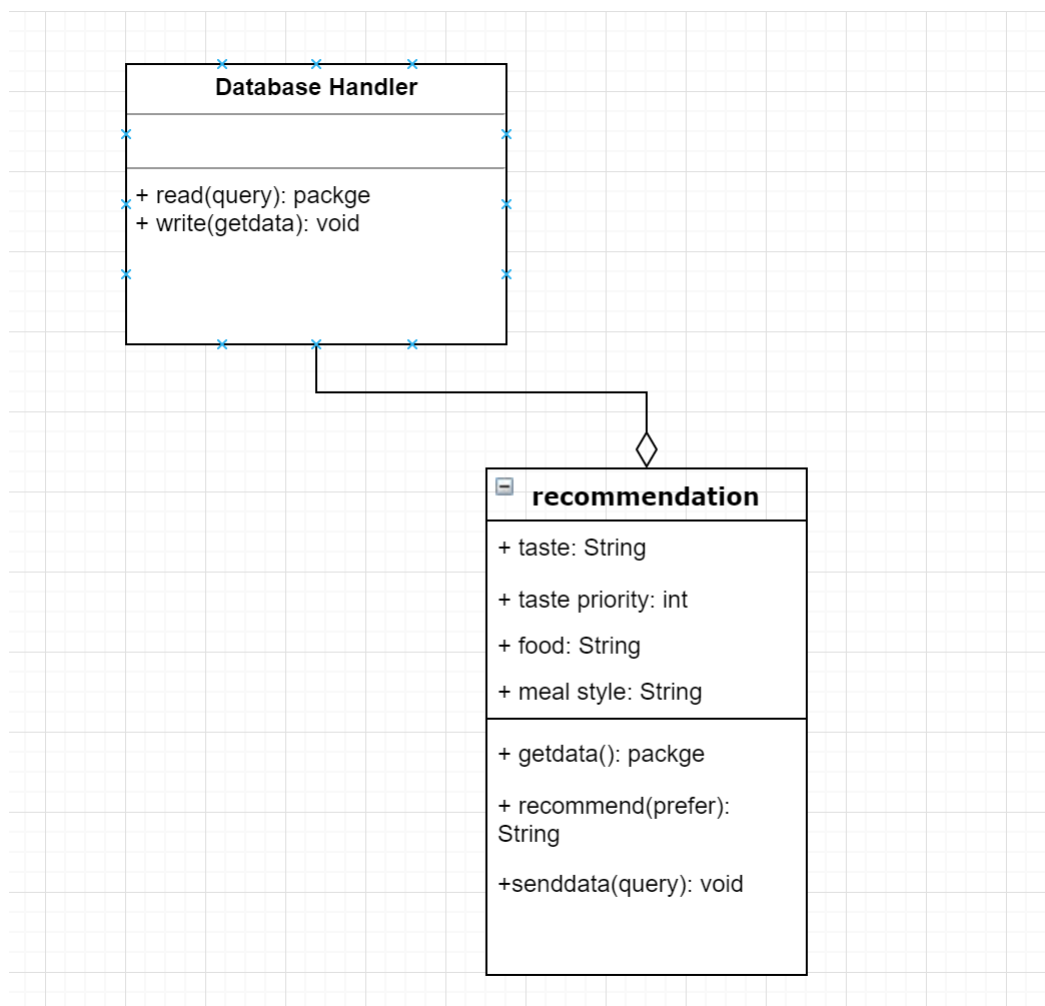
[Diagram 15] State Diagram of Taste Analysis System

6. Menu Recommendation System

6.1 Objective

Recommendation System은 'eat it'이 식품을 구매하는 어플이지만 사용자에게 더 좋고 편한 shopping 경험을 제공하기 위해 사용되는 메뉴 추천기능이다. 사용자가 뭘 먹는지 생각이 없을 경우에 사용자 개인 맞춤형 메뉴를 추천하고 사용자가 필요한 식품을 구매할 수 있다.

6.2 Class Diagram



[Diagram 16] Class Diagram of Menu Recommendation System

A. Database Handler

A.1 Attributes

없음

A.2 Methods

+ package read (query): 해당되는 DB에서 원하는 데이터를 읽어온다.

+ void write(data): 해당되는 DB에 데이터를 저장한다.

B. Recommendation

B.1 Attributes

+ taste: 사용자의 입맛

+ taste priority: 입맛 선호도 표시

+ food: 사용자가 좋아하는 식품

+ meal style: 사용자가 좋아하는 요리 스타일

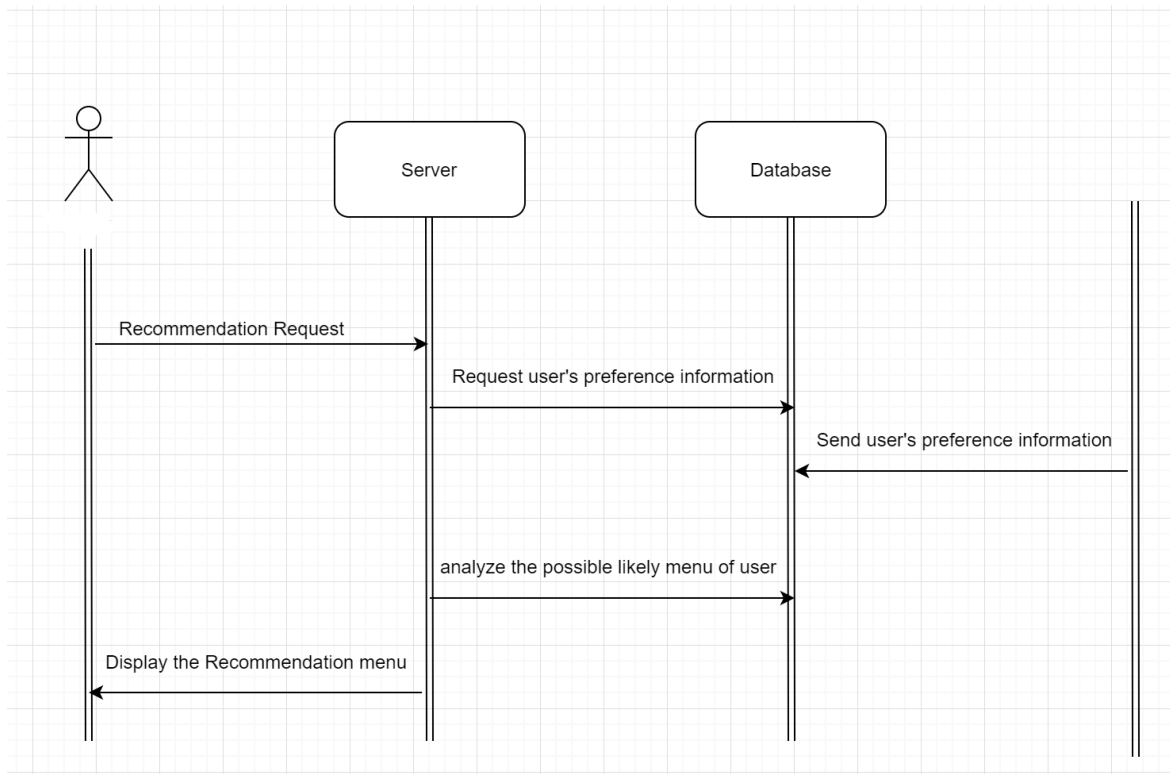
B.2 Methods

+ getdata() : 해당되는 DB에서 정보를 가져온다.

+ senddata (data) : 해당되는 DB에 정보를 보내준다.

+ recommendation: 사용자가 좋아할 수 있는 메뉴를 보여준다.

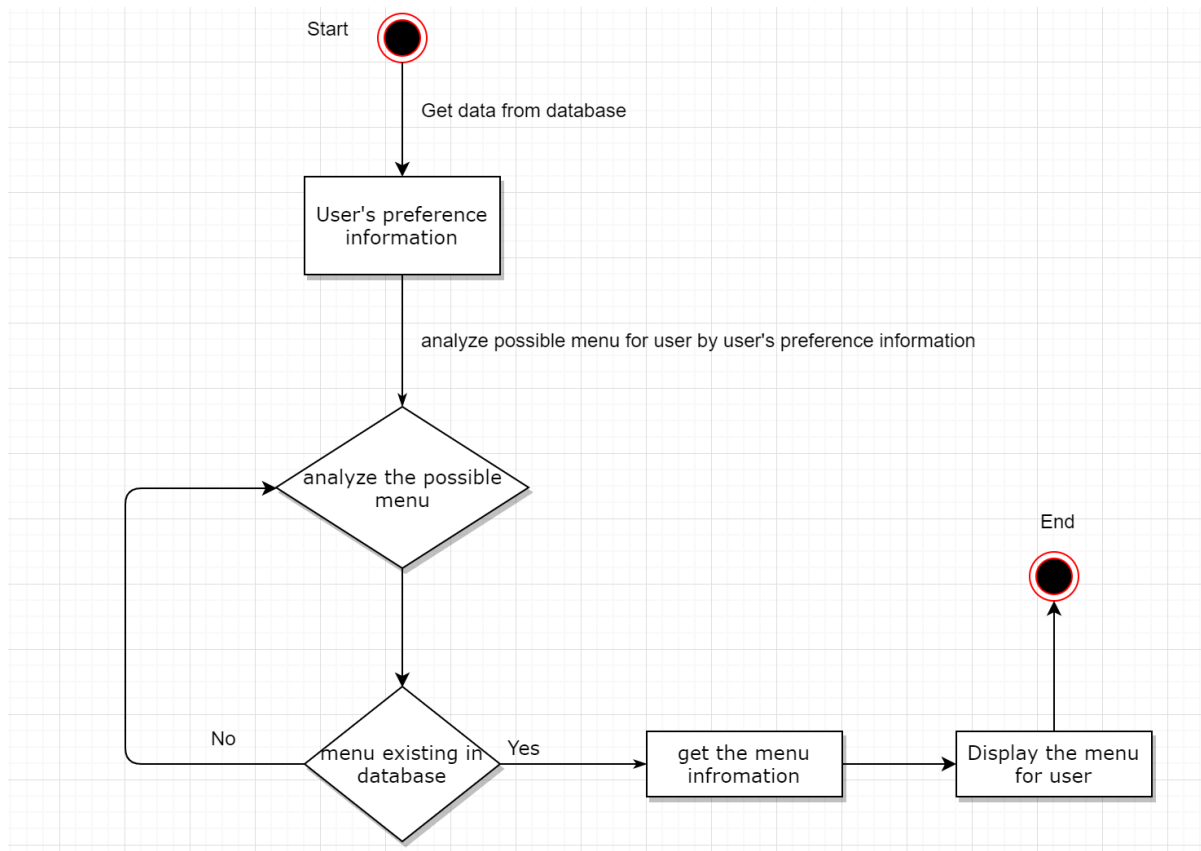
6.3 Sequence Diagram



[Diagram 17] Sequence Diagram of Menu Recommendation System

사용자가 추천요청을 하면 데이터베이스에 사용자의 취향정보를 가져오고, 해당 취향 모델을 이용하여 사용자가 좋아할 수 있는 메뉴를 데이터베이스에서 검색한 후 사용자에게 추천메뉴를 보여준다.

6.4 State Diagram



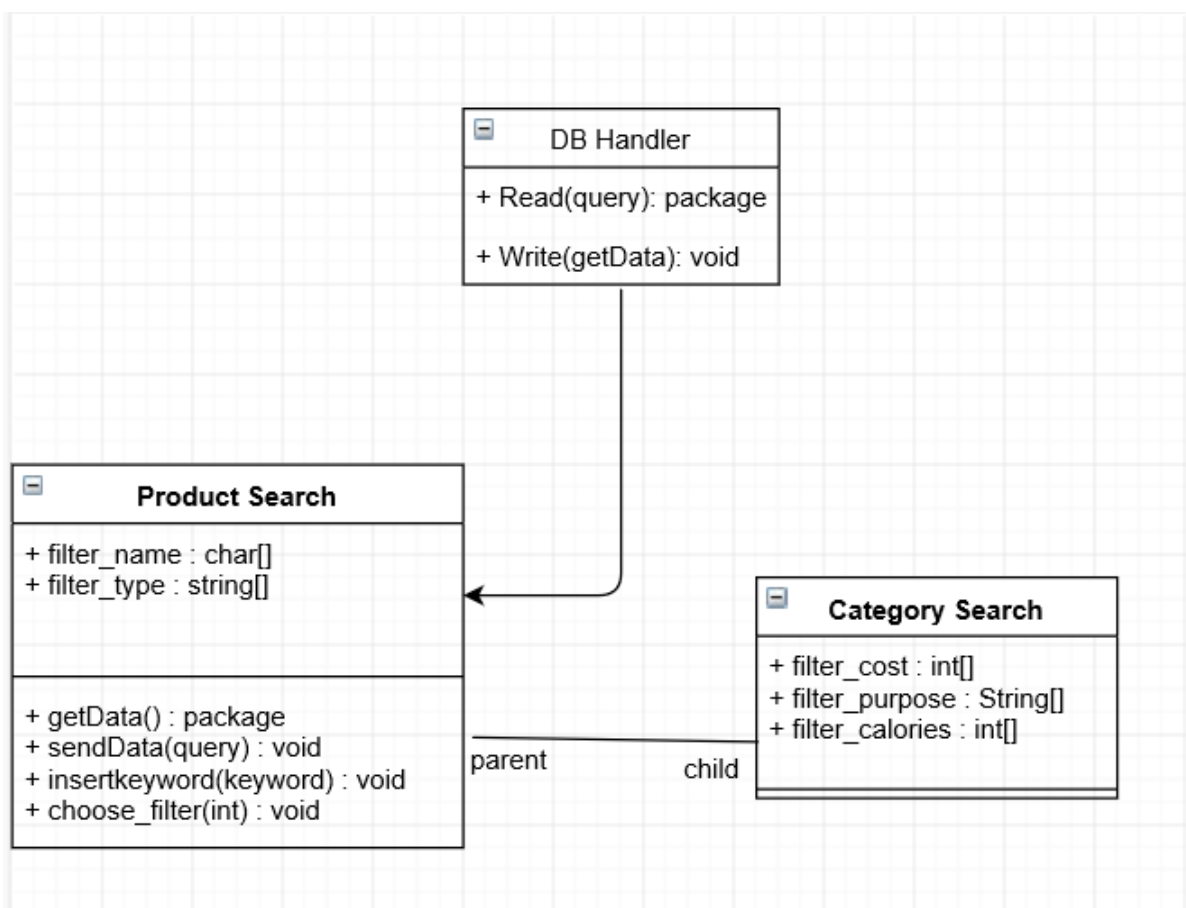
[Diagram 18] State Diagram of Menu Recommendation System

7. Searching System

7.1 Objective

Searching System은 User가 Product를 검색할 때 구동되는 시스템이다. 검색을 할 때, 상품 이름 등 키워드를 검색하는 경우와 카테고리에 따라 검색하는 경우로 나누어 결과값들이 상이하게 나올 수 있다. 따라서 그것을 case로 설정하여 Class Diagram, Sequence Diagram, State Diagram으로, Search System의 구조와 'Eat It'와의 상호 작용을 표현하고 설명한다.

7.2 Class Diagram



[Diagram 19] Class Diagram of Searching System

A. Database Handler

A.1 Attributes

없음

A.2 Methods

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

B. Product Search

B.1 Attributes

filter_name : char[] - 음식 이름 정보

filter_type : string[] - 음식의 맛 유형 정보

B.2 Methods

getData() : package - DB로부터 data를 받는다.

sendData(query) : void - DB로 data를 보낸다.

insertKeyword(keyword) : void - 검색 키워드를 입력받는다.

choose_filter(int) : void - filter 선택한다. (키워드 입력 혹은 카테고리 입력)

C. Category Search

C.1 Attributes

(부모 class로부터 상속)

filter_cost : int[] - 음식 가격 정보 (범위)

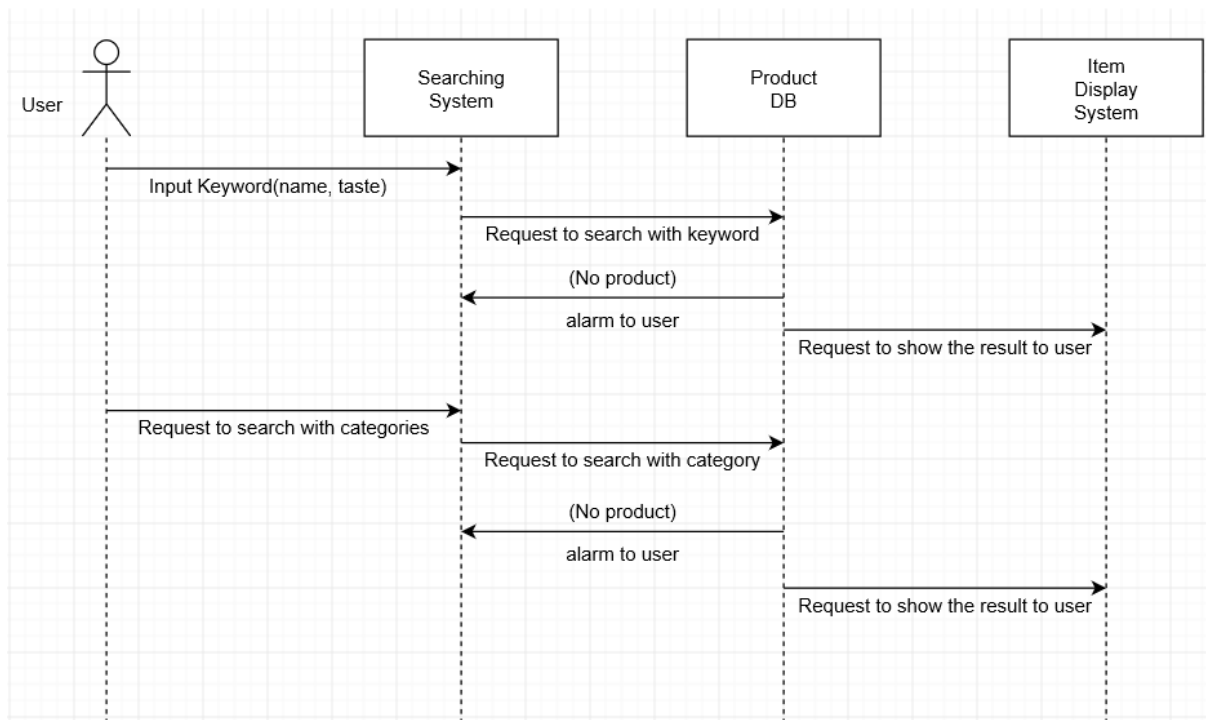
filter_purpose: string[] - 음식의 목적 정보

filter_calories : int[] - 음식 열량 정보 (범위)

C.2 Methods

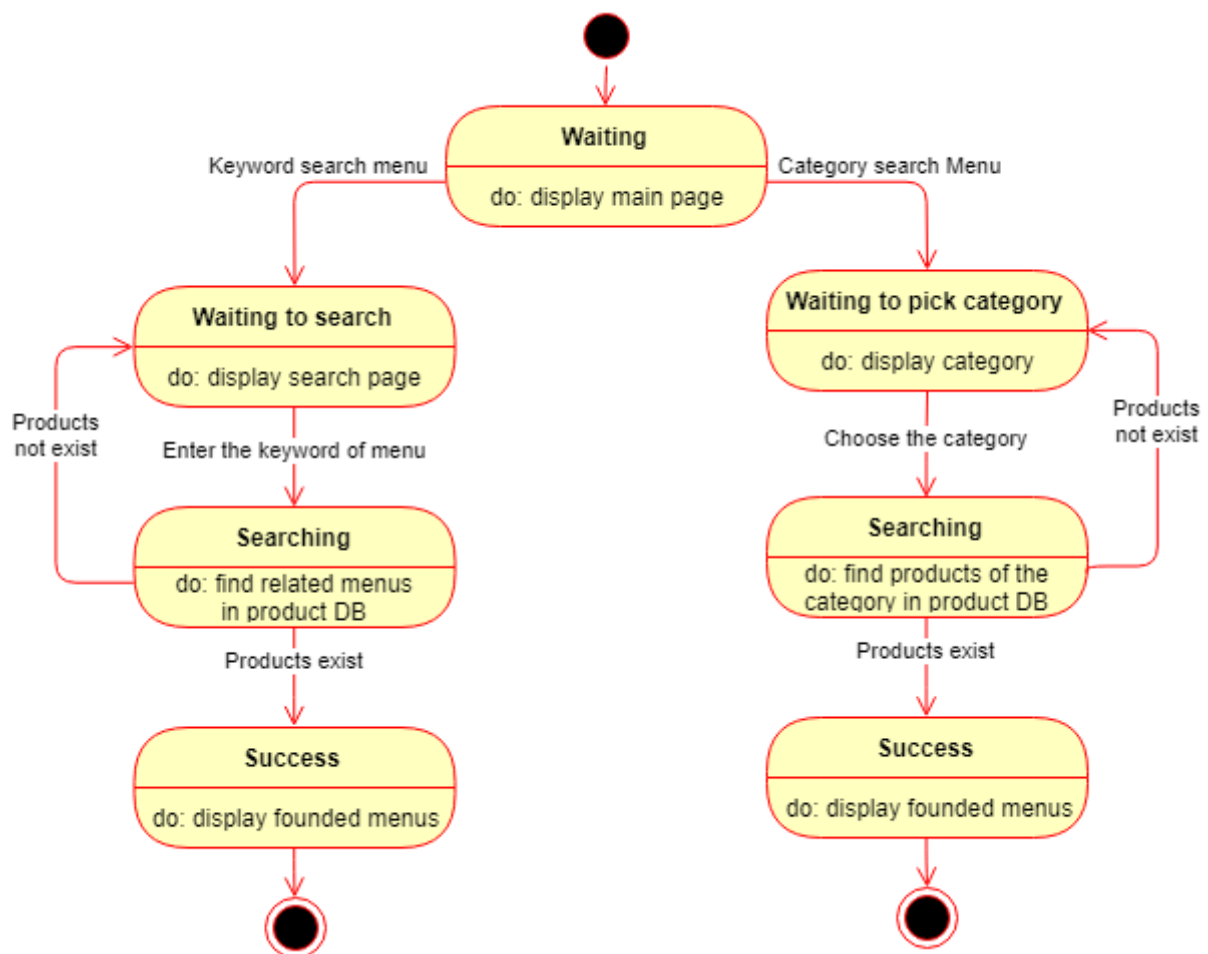
(부모 class로부터 상속)

7.3 Sequence Diagram



[Diagram 20] Sequence Diagram of Searching System

7.4 State Diagram



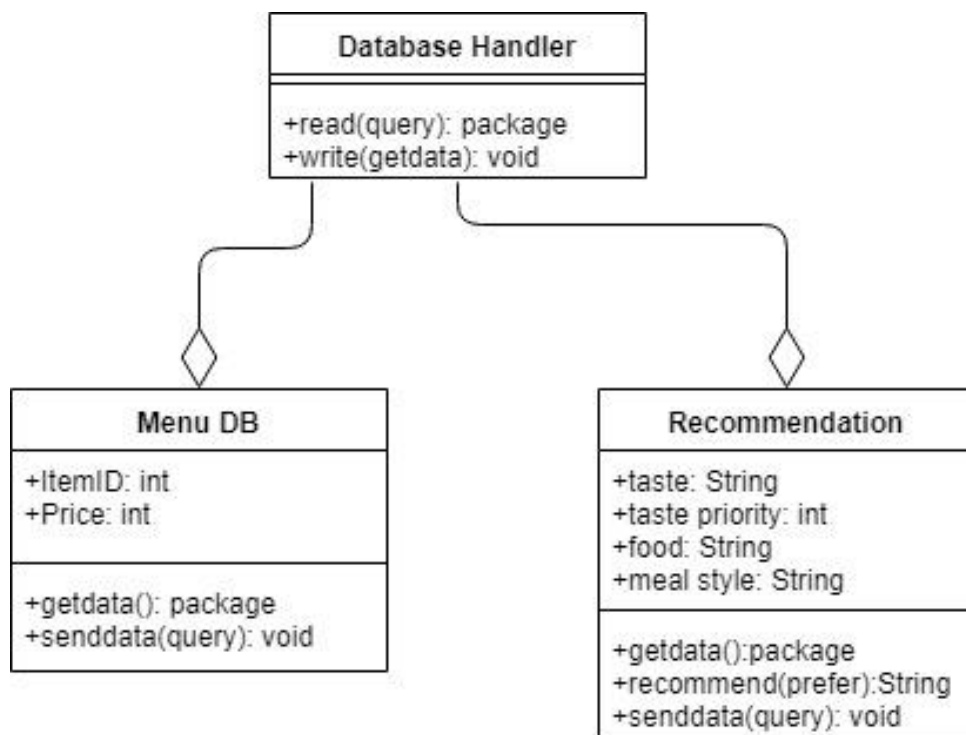
[Diagram 21] State Diagram of Searching System

8. Item Display System

8.1 Objective

Item Display System은 Menu Recommendation System과 Search System으로부터 받은 메뉴를 사용자에게 보여주는 시스템으로, 사용자가 메뉴를 보고 선택하여 Cart에 담을 수 있게끔 한다. 이러한 Item Display System에 대하여 Class Diagram, Sequence Diagram과 State Diagram을 통해 Item Display System의 구조를 표현하고 설명한다.

8.2 Class Diagram



[Diagram 22] Class Diagram of Item Display System

A. Database Handler

A.1 Attributes

없음

A.2 Methods

+read(query):package: 해당하는 DB에서 원하는 data를 가져옴

+write(package):void: 해당하는 DB에 data를 저장

B. Recommendation

B.1 Attributes

+taste:String : 맛 종류

+taste priority:int : 맛 우선순위

+food: string : 음식 이름

+meal style: String : 식사 종류

B.2 Methods

+getdata():package :

+recommend(prefer):String :

+senddata(query): void :

C. Menu DB

C.1 Attributes

+itemID: int : 메뉴 고유번호

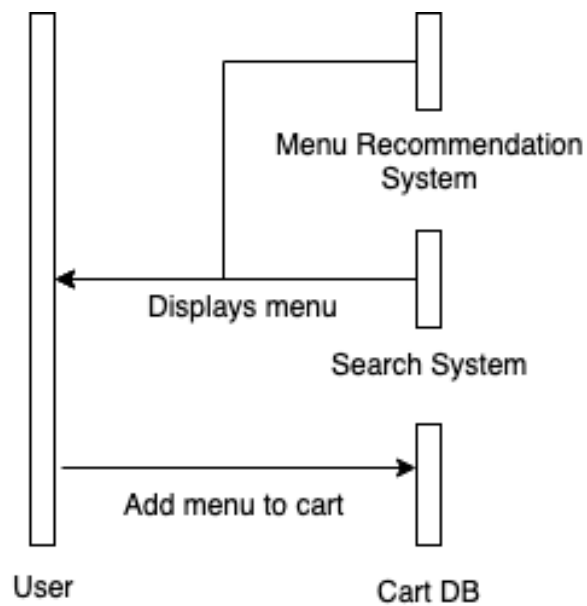
+Price: int : 가격

C.2 Methods

+read(query):package: 해당하는 DB에서 원하는 data를 가져옴

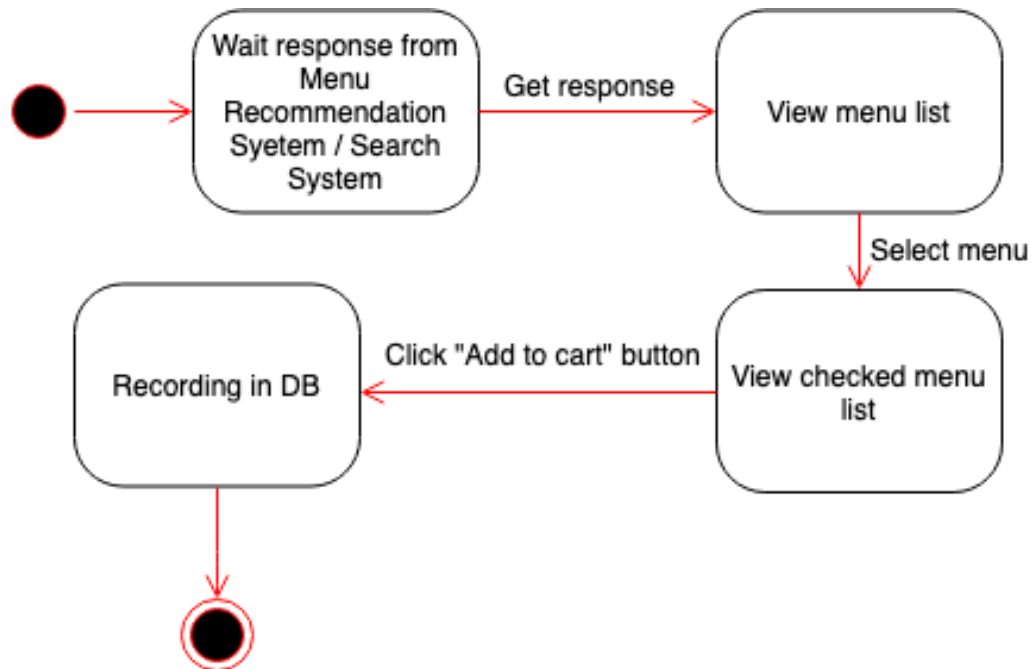
+write(package):void: 해당하는 DB에 data를 저장

8.3 Sequence Diagram



[Diagram 23] Sequence Diagram of Item Display System

8.4 State Diagram



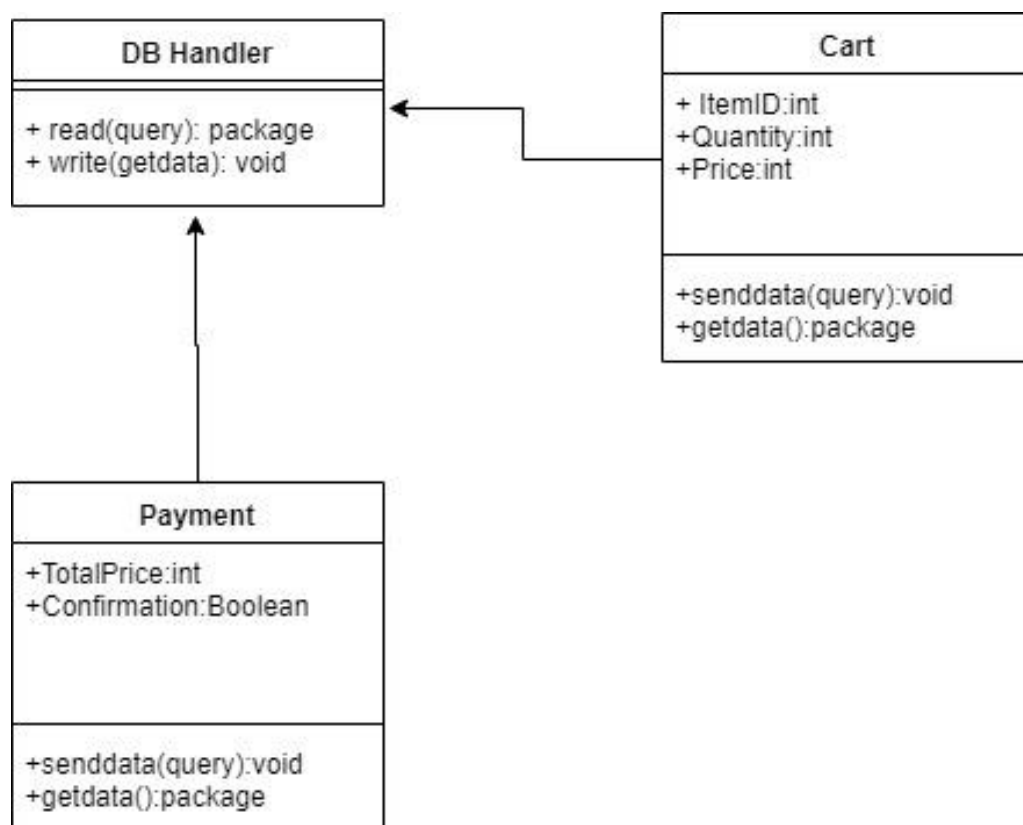
[Diagram 24] State Diagram of Item Display System

9. Ordering System

9.1 Objective

Ordering System은 구매를 원하는 고객과 관련된 시스템으로, 추천된 메뉴나 쿼리를 입력하여 검색된 메뉴를 Cart로 옮겨 담아 사용자가 구매를 할 수 있게끔 한다. 이러한 Ordering System에 대하여 Class diagram, Sequence diagram과 State Diagram을 통해 Ordering System의 구조를 표현하고 설명한다.

9.2 Class Diagram



[Diagram 25] Class Diagram of Ordering System

A. Database Handler

A.1 Attributes

없음

A.2 Methods

+read(query):package: 해당하는 DB에서 원하는 data를 가져옴

+write(package):void: 해당하는 DB에 data를 저장

B. Cart

B.1 Attributes

+ItemID:int: 메뉴 고유 번호

+Quantity:int: 주문 수량

+Price:int: 가격

B.2 Methods

+getdata():package: DB에서 data를 가져온다

+senddata():query: DB에 data를 보낸다

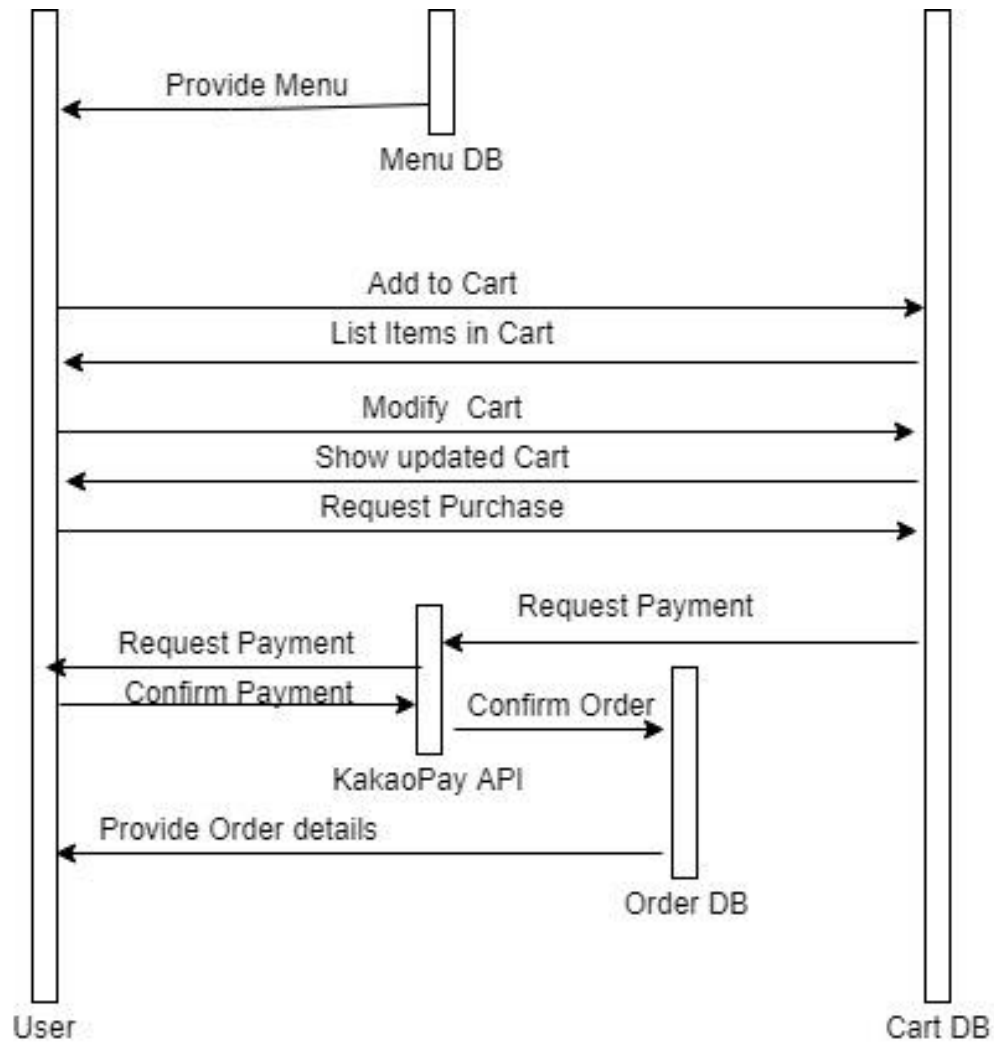
C. Payment

C.1 Attributes

+TotalPrice:int : 총 합산 금액

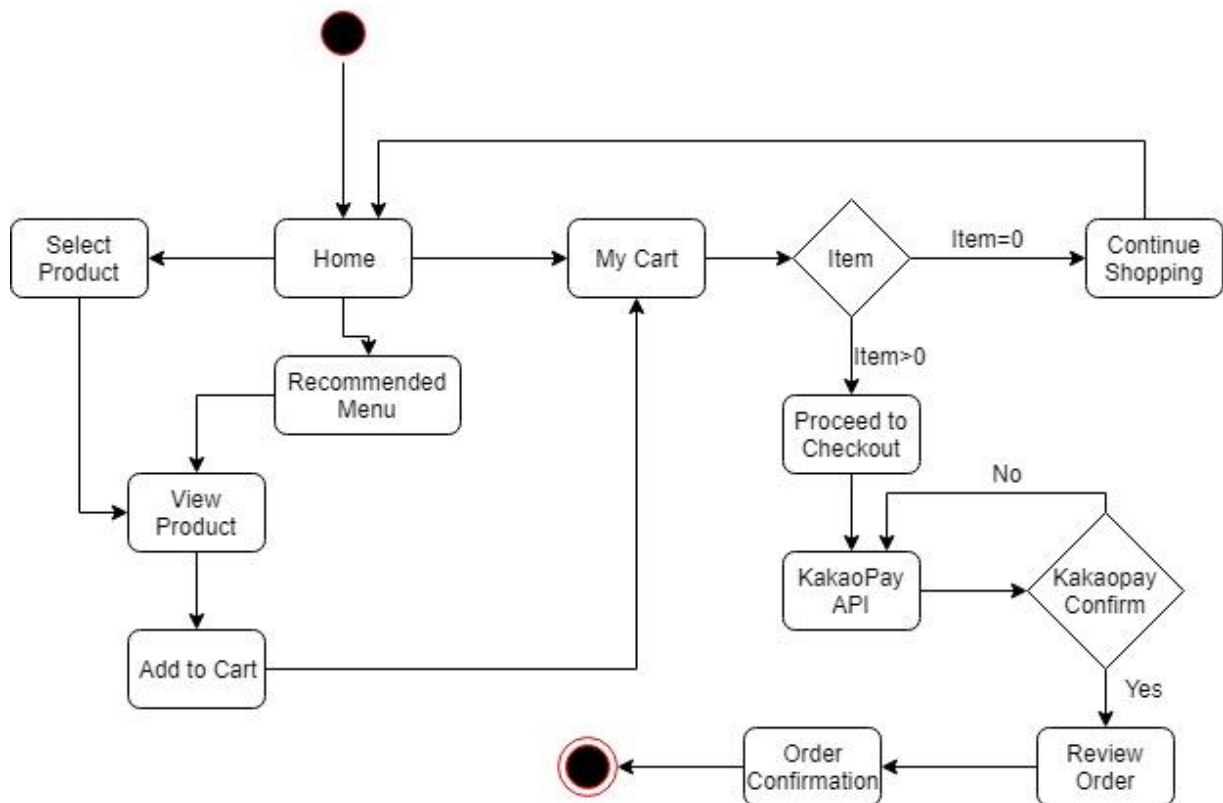
+Confirmation:Boolean : 결제 승인 혹은 거절

9.3 Sequence Diagram



[Diagram 26] Sequence Diagram of Ordering System

9.4 State Diagram



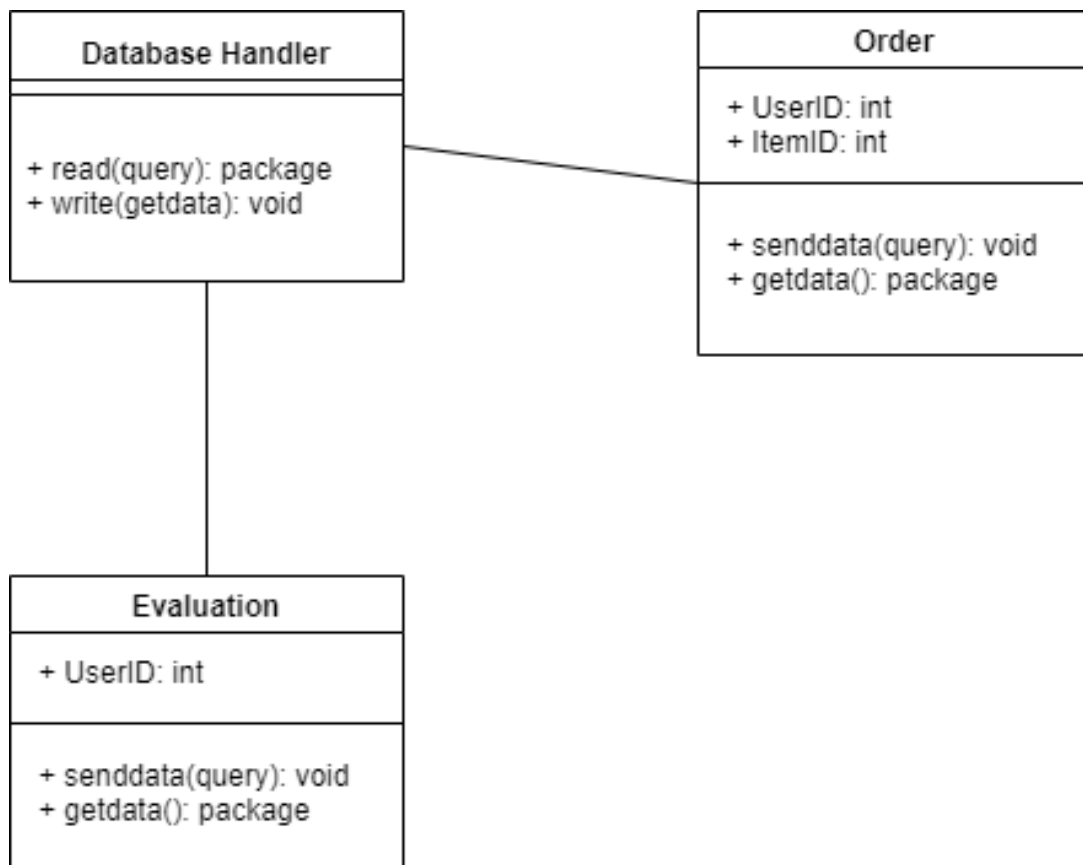
[Diagram 27] State Diagram of Ordering System

10. My Page System

10.1 Objective

My Page System은 사용자의 주문을 관리하거나 주문한 상품을 평가하는 subsystem이다. Class diagram, Sequence diagram, State Diagram을 통해 My page System의 구조를 표현하고 설명한다.

10.2 Class Diagram



[Diagram 28] Class Diagram of My Page System

A. Database Handler

A.1 Attributes

없음

A.2 Methods

+package read(query): DB에서 원하는 data를 가져온다.

+void write(package): DB에 data를 저장한다.

B. Order

B.1 Attributes

+UserID: 사용자 고유 번호

+ItemID: 메뉴 고유 번호

B.2 Methods

+package senddata(): DB에 data를 보낸다.

+void getdata(package): DB에서 data를 가져온다.

C. Evaluation

C.1 Attributes

+UserID: 사용자 고유 번호

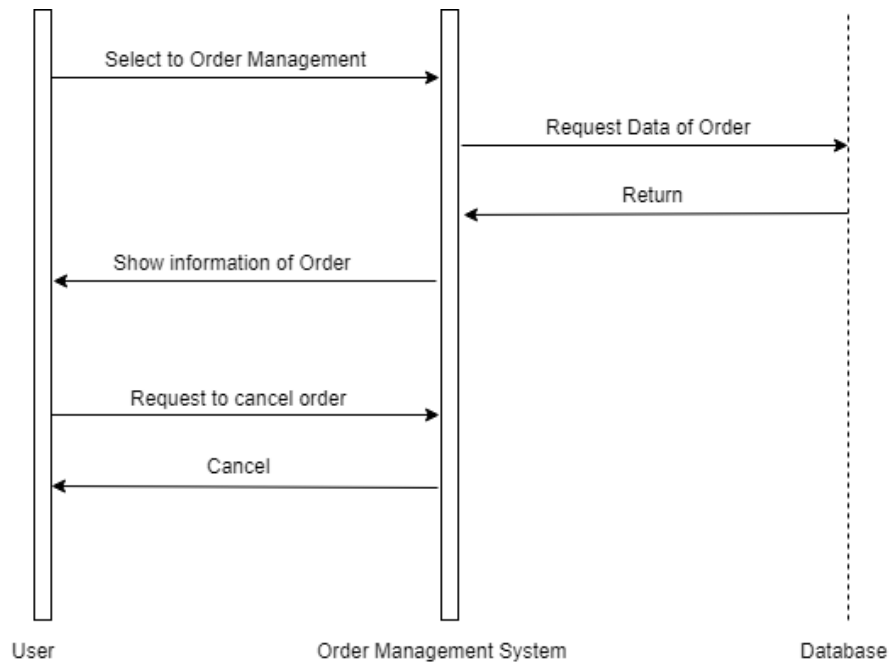
C.2 Methods

+void senddata(package): DB에 data를 보낸다.

+package getdata(): DB에서 data를 가져온다.

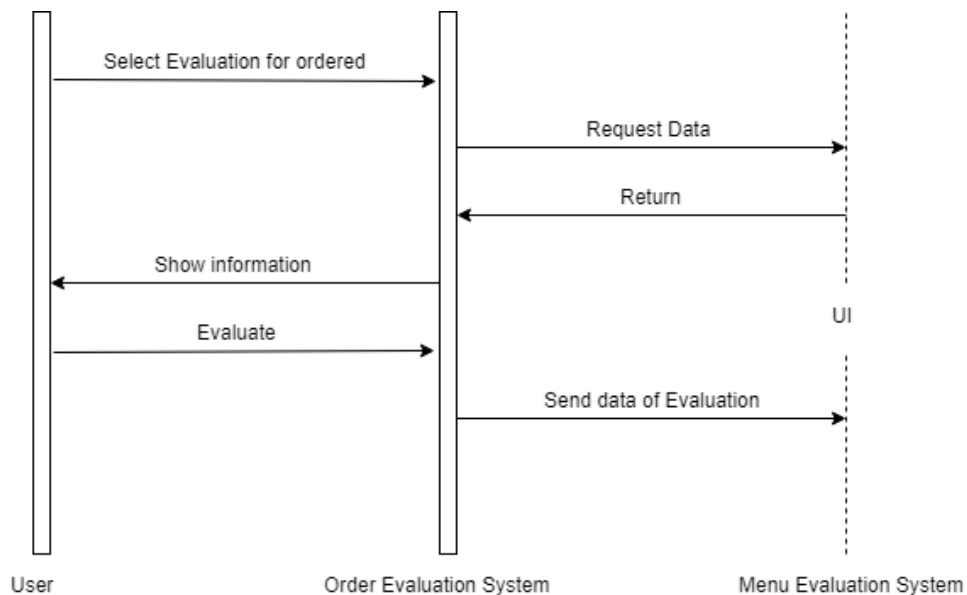
10.3 Sequence Diagram

A. Order Management



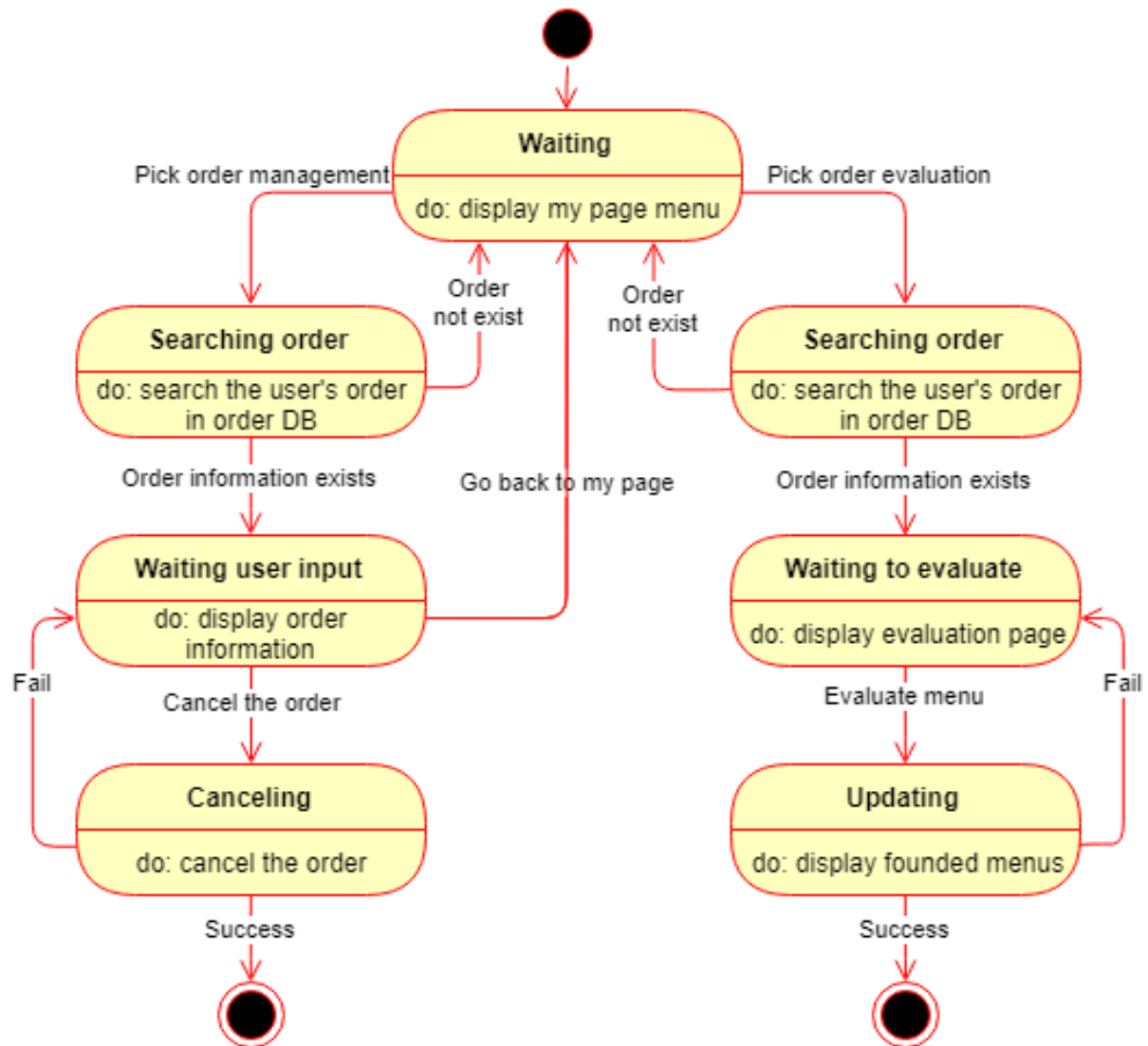
[Diagram 29] Sequence Diagram 1 of My Page System

B. Order Evaluation



[Diagram 30] Sequence Diagram 2 of My Page System

10.4 State Diagram



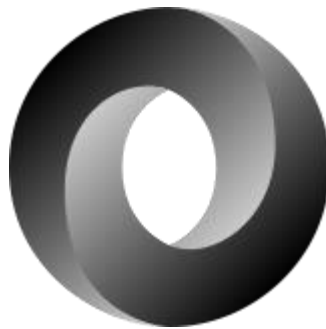
[Diagram 31] State Diagram of My Page System

11. Protocol Design

11.1 Objective

Protocol Design에서는 'Eat It'에서 각 System간의 communication에 있어서 필요한 protocol에 대해서 기술하고자 한다. Protocol에 필요한 기본적인 data format에는 XML, JSON 등이 있는데 'Eat It'에서는 JSON을 사용한다. 따라서 이에 대한 설명과 각 Protocol에서 사용되는 Attribute와 Value는 무엇인지 자세히 작성된다.

11.2 JSON



[Figure 7] Logo of JSON

JSON은 JavaScript Object Notation의 약어로서 XML과 더불어 대표적인 데이터 교환 방식이다. 자바스크립트에 기반하여 만들어진 데이터 표현형식이지만 프로그래밍언어나 플랫폼에 독립적인 특성을 갖고 있어, 다양한 언어에서 JSON을 활용할 수 있다. 또한 기능이 적어 파싱(Parsing)이 빠르다는 장점을 가지고 있다. 자료의 종류에 큰 제한이 없으며, 컴퓨터 프로그램의 변수 값을 표현하는 데 적합하다. Attribute와 value의 쌍으로 표현되며, 거의 대부분의 자료 형을 사용할 수 있다.

11.3 Protocol Description

A. Overview

본 항목에서는 'Eat It'에서 사용되는 메시지에 대해 JSON에 대응하는 Attribute와 Value로 작성하여 나타낸다. Server와 Client 사이에서 전송되는 메시지를 HTTP(Hyper Text Transfer Protocol)에서는 request와 response로 구분한다. request는 client에서 server에 전송하는 메시지, response는 서버가 client의 request에 반응하는 메시지를 나타낸다. 이하 표에 대해선 캡션을 생략한다.

B. Login Protocol

B.1 Request

Attribute	Value
E-mail	사용자의 이메일
Password	사용자의 비밀번호

[Table 2] Login Request

B.2 Response

Attribute	Value
Login_success	로그인 성공여부

[Table 3] Login Response

C. Sign Up

C.1 Request

Attribute	Value
E-mail	사용자의 이메일
Password	사용자의 비밀번호
Name	사용자의 이름
Nickname	사용자의 닉네임
Age	사용자의 나이
Sex	사용자의 성별
Address	사용자의 주소
Call-number	사용자의 전화번호

[Table 4] Signup Request

C.2 Response

Attribute	Value
Up_success	회원가입 성공여부

[Table 5] Signup Response

D. Taste Analysis System

D.1 Request

Attribute	Value
Food	취향 평가 대상 음식
Prefer	평가 대상 음식별 사용자의 선호도

[Table 6] Taste Analysis Request

D.2 Response

Attribute	Value
Eval_success	평가 성공 여부

[Table 7] Taste Analysis Response

E. Recommend Menu View Protocol

E.1 Request

Attribute	Value
User_SN	사용자의 고유번호

[Table 8] Recommend Menu View Request

E.2 Response

Attribute	Value
Food_SN	음식의 고유번호
Type	음식의 맛 유형
Name	음식의 이름
Ingredient	음식의 재료
Brand	재료의 브랜드
Price	재료의 가격
Stock	재료의 재고
Food_calories	음식의 열량
Ing_calories	재료의 열량
Rank	음식의 평점
Img_URL	재료 이미지 URL
Comment_URL	상세 정보 이미지 URL

[Table 9] Recommend Menu View Response

F. Menu Search Protocol

F.1 Request

Attribute	Value
Search_word	검색할 단어
Search_taste	검색할 음식의 유형(맛)
Search_name	검색할 음식의 이름
Search_calories	검색할 음식의 최대 열량

[Table 10] Menu Search Request

F.2 Response

Attribute	Value
Food_SN	음식의 고유번호
Type	음식의 맛 유형
Name	음식의 이름
Ingredient	음식의 재료
Brand	재료의 브랜드
Price	재료의 가격
Stock	재료의 재고
Food_calories	음식의 열량
Ing_calories	재료의 열량
Rank	음식의 평점
Img_URL	재료 이미지 URL
Comment_URL	상세 정보 이미지 URL

[Table 11] Menu Search Response

G. Product Post Protocol

G.1 Request

Attribute	Value
Food_SN	음식의 고유번호
Type	음식의 맛 유형
Name	음식의 이름
Ingredient	음식의 재료
Brand	재료의 브랜드
Price	재료의 가격
Stock	재료의 재고
Food_calories	음식의 열량
Ing_calories	재료의 열량
Img_URL	재료 이미지 URL
Comment_URL	상세 정보 이미지 URL

[Table 12] Product Post Request

G.2 Response

Attribute	Value
Post_success	음식 등록 성공여부

[Table 13] Product Post Response

H. Ingredient of Product View Protocol

H.1 Request

Attribute	Value
Food_SN	음식의 고유번호
Taste	음식의 맛

[Table 14] Ingredient of product View Request

H.2 Response

Attribute	Value
Ingredient	음식의 재료
Brand	재료의 브랜드
Price	재료의 가격
Stock	재료의 재고
Ing_calories	재료의 열량
Img_URL	재료 이미지 URL
Comment_URL	상세 정보 이미지 URL

[Table 15] Ingredient of product View Response

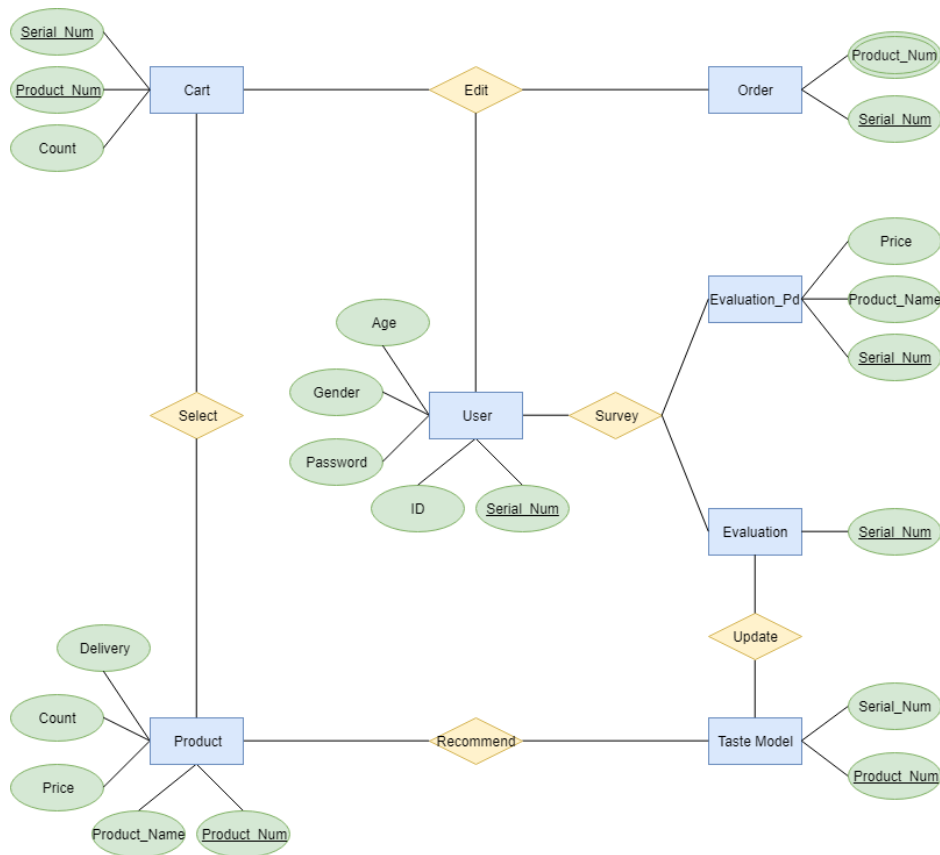
12. Database Design

12.1 Objective

Database Design은 요구사항 명세서에서 기술한 데이터베이스 요구사항을 바탕으로 작성하였다. 요구사항을 바탕으로 ER Diagram을 작성하고, 이를 통해 Relational Schema와 SQL DDL을 작성한다.

12.2 ER Diagram

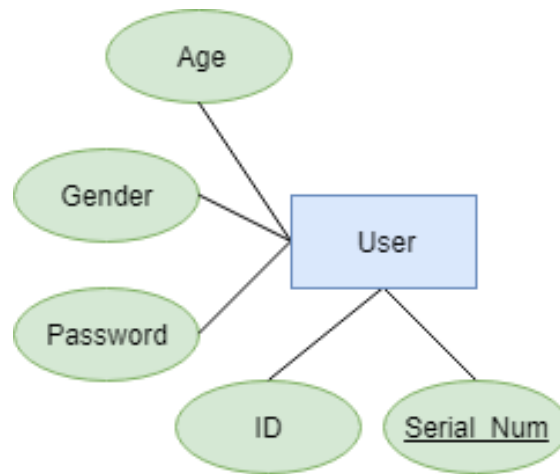
ER Diagram에 대한 설명은 다음과 같다. Entity는 직사각형으로 표현하며, Entity들 간의 관계를 나타내는 Relation은 마름모 모양으로 표현한다. 또한, Entity와 Relation은 Attribute를 가질 수 있는데, 이는 타원형으로 표현한다. 마지막으로 각 Entity의 Primary key는 아래에 밑줄을 치고, Multi-valued Attribute는 이중 타원형으로 표현한다.



[Diagram 32] Overall ER Diagram

A. Entity

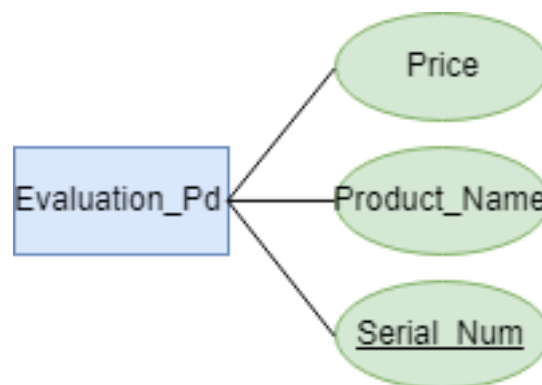
A.1 User



[Figure 8] User Entity

User는 회원의 정보를 나타낸다. Primary key는 Serial number(Serial_Num)이고, Serial_Num, Age, Gender, Password, ID의 속성을 가지고 있다.

A.2 Evaluation_Pd



[Figure 9] Evaluation_Pd Entity

Evaluation_Pd는 사용자가 처음 로그인 했을 때 진행하는 평가에 사용되는 상품 정보이다. Primary key는 Serial number(Serial_Num)이고, Serial_Num, Price, Product_Name의 속성을 가지고 있다.

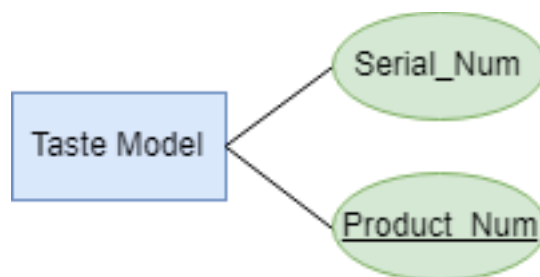
A.3 Evaluation



[Figure 10] Evaluation Entity

Evaluation은 사용자가 첫 로그인 시에 진행되는 평가의 데이터를 갖고 있다. Primary key는 Serial number(Serial_Num)이고, Serial_Num의 속성을 가지고 있다.

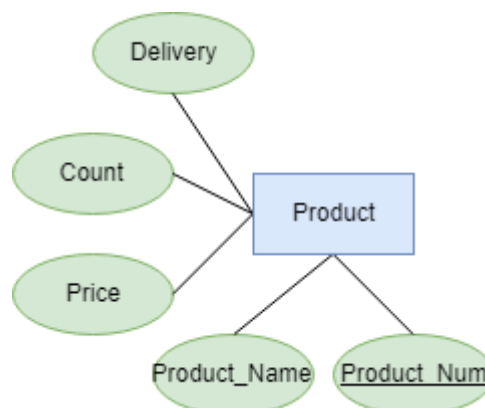
A.4 Taste Model



[Figure 11] Taste Model Entity

Taste Model은 사용자의 취향 데이터를 가지고 있다. Primary key는 Product number(Product_Num)이고, Product_Num, Serial_Num의 속성을 가지고 있다.

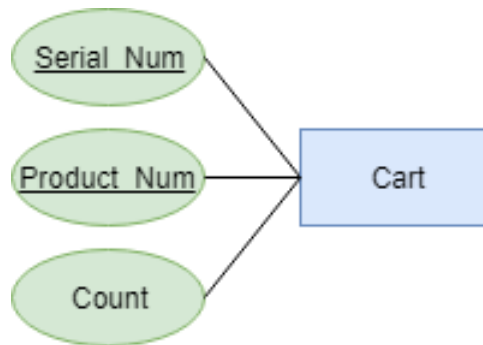
A.5 Product



[Figure 12] Product Entity

Product은 상품의 정보를 나타낸다. Primary key는 Product_Num이고, Product_Num, Product_Name, Price, Count, Delivery의 속성을 가지고 있다.

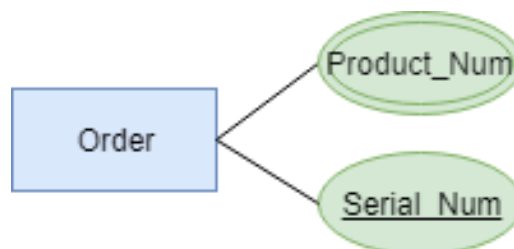
A.6 Cart



[Figure 13] Cart Entity

Cart는 사용자가 선택한 상품이 들어간 장바구니를 나타낸다. Primary key는 Serial number(Serial_Num)와 Product number(Product_Num)이고, Serial_Num, Product_Num, count의 속성을 가지고 있다.

A.7 Order

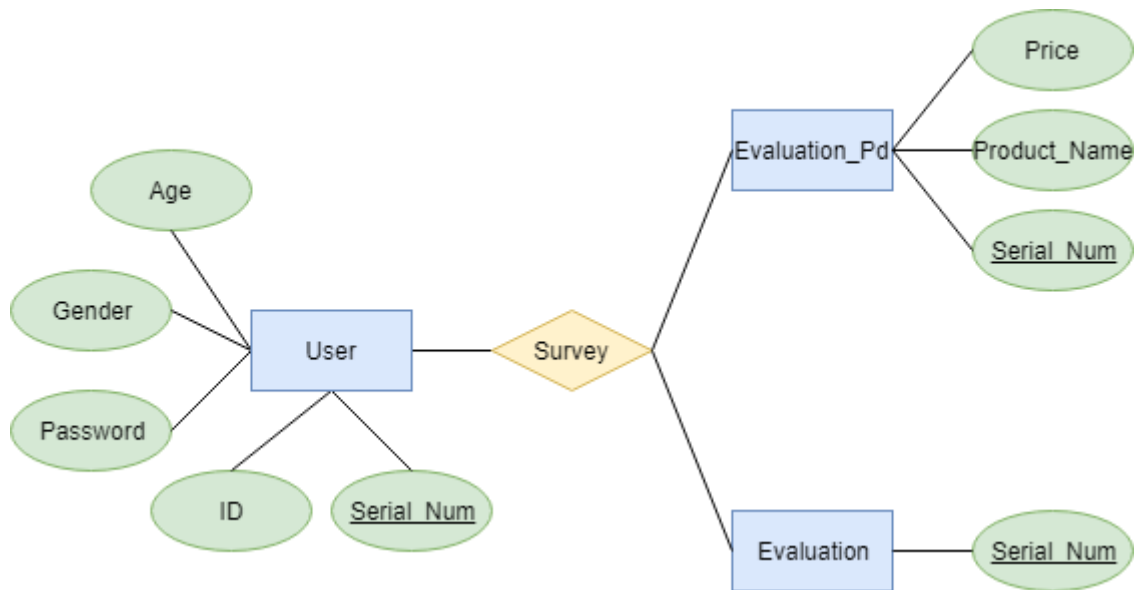


[Figure 14] Order Entity

Order는 User의 구매에 대해 나타낸다. Primary key는 Serial number(Serial_Num)이고, Serial_Num, Product_Num의 속성을 가지고 있다.

B. Relationship

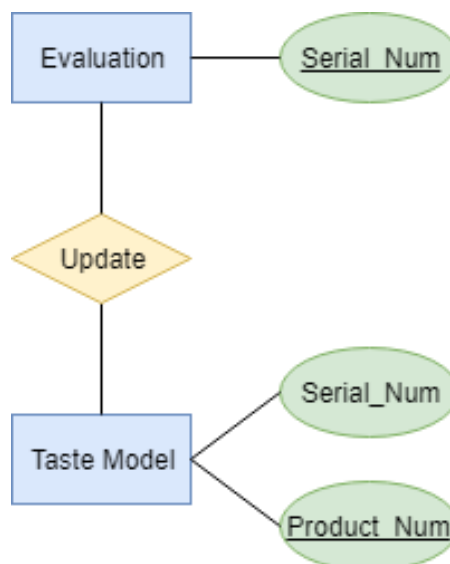
B.1 Survey



[Figure 15] Survey Relationship

Survey Relationship은 survey를 통해서 사용자의 취향 데이터를 얻는다. 한 명의 사용자는 하나의 취향을 가진다.

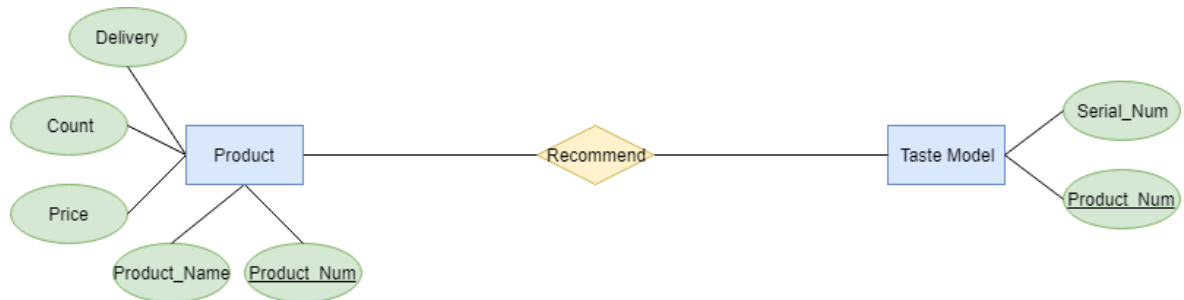
B.2 Update



[Figure 16] Update Relationship

Update relationship은 사용자의 취향을 토대로 taste model을 만들기 때문에, 하나의 취향이 하나의 모델을 만든다. 또한, 하나의 모델은 여러 취향에 담길 수도 있다.

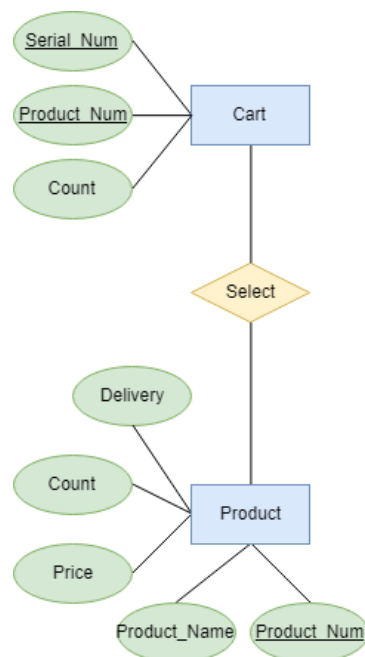
B.3 Recommend



[Figure 17] Recommend Relationship

Recommend relationship은 taste model을 바탕으로 사용자의 취향에 맞는 상품을 상품 DB 중에서 추천해주는 관계로, 하나의 모델이 여러 개의 상품을 추천 받을 수 있다.

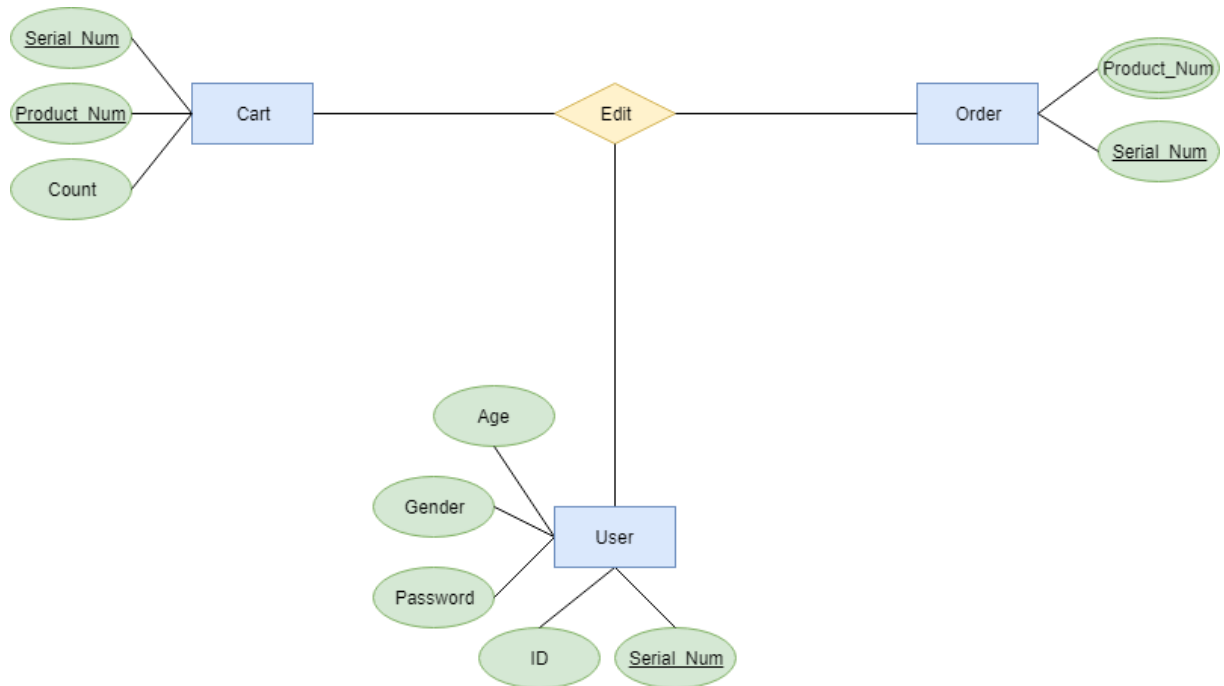
B.4 Select



[Figure 18] Select Relationship

Select relationship은 상품 중에서 선택 받은 것만이 카트에 담기게 되는 관계로, 하나의 상품이 카트에 담길 수도 있고, 또 여러 개의 상품이 카트에 담길 수 있다.

B.5 Edit



[Figure 19] Edit Relationship

Edit relationship은 사용자가 카트를 편집할 수 있으며, 사용자의 선택에 따라 주문을 정할 수 있다. 한 명의 사용자는 자신에 해당된 하나의 카트를 편집할 수 있고, 여러 개의 상품을 주문 및 취소할 수 있다.

12.3 Related Schema

A. User

<u>Serial_Num</u>	Age	Gender	Password	ID
-------------------	-----	--------	----------	----

Primary Key (PK): Serial_Num

Foreign Key (FK): 없음

Function Dependency (FD):

$\text{Serial_Num} \rightarrow \{ \text{ID}, \text{Password}, \text{Gender}, \text{Age} \};$

Description: Entity User에 관한 테이블이다. 모든 속성에 Null을 허용하지 않는다.

B. Evaluation_Pd

<u>Serial_Num</u>	Product_Name	Price
-------------------	--------------	-------

Primary Key (PK): Serial_Num

Foreign Key (FK): Serial_Num

Function Dependency (FD):

$\text{Serial_Num} \rightarrow \{ \text{Product_Name}, \text{Price} \};$

Description: Entity Evaluation_Pd에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK를 FK로 갖는다.

C. Evaluation

<u>Serial_Num</u>

Primary Key (PK): Serial_Num

Foreign Key (FK): Serial_Num

Function Dependency (FD):

Serial_Num $\rightarrow \{ \}$;

Description: Entity Evaluation에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK를 FK로 갖는다.

D. Taste Model

<u>Product_Num</u>	Serial_Num
--------------------	------------

Primary Key (PK): Product_Num

Foreign Key (FK): Product_Num

Function Dependency (FD):

Product_Num $\rightarrow \{ \text{Serial_Num} \}$;

Description: Entity Taste Model에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity Product의 PK를 FK로 갖는다.

E. Product

<u>Product_Num</u>	Product_Name	Price	Count	Delivery
--------------------	--------------	-------	-------	----------

Primary Key (PK): Product_Num

Foreign Key (FK): 없음

Function Dependency (FD):

Product_Num $\rightarrow \{ \text{Product_Name, Price, Count, Delivery} \}$;

Description: Entity Product에 관한 테이블이다. 모든 속성에 Null을 허용하지 않는다.

F. Cart

<u>Serial_Num</u>	Product_Num	Count
-------------------	-------------	-------

Primary Key (PK): Serial_Num, Product_Num

Foreign Key (FK): Serial_Num, Product_Num

Function Dependency (FD):

$\text{Serial_Num} \rightarrow \{ \text{Product_Num}, \text{Count} \};$

Description: Entity Cart에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK와 Entity Product의 PK를 FK로 갖는다.

G. Order

<u>Serial_Num</u>	Product_Num
-------------------	-------------

Primary Key (PK): Serial_Num

Foreign Key (FK): Serial_Num

Function Dependency (FD):

$\text{Serial_Num} \rightarrow \{ \text{Product_Num} \};$

Description: Entity Order에 관한 테이블이다. 모든 속성에 Null을 허용하지 않으며, Entity User의 PK를 FK로 갖는다. Product_Num은 Multi-valued Attribute로 여러 값을 가진다.

12.4 SQL DDL

A. User

```
CREATE TABLE User (  
    'Serial_Num'    int            NOT NULL    AUTO_INCREMENT,  
    'ID'            varchar(45)    NOT NULL,  
    'Password'      varchar(45)    NOT NULL,  
    'Gender'        varchar(45)    NOT NULL,  
    'Age'           INT            NOT NULL  
    PRIMARY KEY ('Serial_Num'),  
};
```

B. Evaluation_Pd

```
CREATE TABLE Evaluation_Pd (  
    'Serial_Num'    int            NOT NULL    AUTO_INCREMENT,  
    'Product_Name'  varchar(45)    NOT NULL,  
    'Price'         int            NOT NULL,  
    PRIMARY KEY ('Serial_Num'),  
    KEY 'user_idx' ('Serial_Num'),  
    CONSTRAINT 'user_idx' FOREIGN KEY('Serial_Num') REFERENCES 'User'('Serial_Num')  
    ON DELETE NO ACTION ON UPDATE NO ACTION,  
};
```

C. Evaluation

```
CREATE TABLE Evaluation (  
    'Serial_Num'    int            NOT NULL    AUTO_INCREMENT,  
    PRIMARY KEY ('Serial_Num'),  
    KEY 'user_idx' ('Serial_Num'),  
    CONSTRAINT 'user_idx' FOREIGN KEY('Serial_Num') REFERENCES 'User'('Serial_Num')  
    ON DELETE NO ACTION ON UPDATE NO ACTION,  
);
```

D. Taste Model

```
CREATE TABLE Taste Model (  
    'Product_Num'    int            NOT NULL    AUTO_INCREMENT,  
    'Serial_Num'    int            NOT NULL,  
    PRIMARY KEY ('Product_Num'),  
    KEY 'product_idx' ('Product_Num'),  
    CONSTRAINT 'product_idx' FOREIGN KEY('Product_Num') REFERENCES  
    'Product'('Product_Num') ON DELETE NO ACTION ON UPDATE NO ACTION,  
);
```

E. Product

```
CREATE TABLE Product (  
    'Product_Num' int NOT NULL AUTO_INCREMENT,  
    'Product_Name' varchar(45) NOT NULL,  
    Price int NOT NULL,  
    Count int NOT NULL,  
    Delivery varchar(45) NOT NULL,  
    PRIMARY KEY ('Product_Num'),  
);
```

F. Cart

```
CREATE TABLE Cart (  
    'Serial_Num' int NOT NULL AUTO_INCREMENT,  
    'Product_Num' int NOT NULL AUTO_INCREMENT,  
    Count int NOT NULL,  
    PRIMARY KEY ('Serial_Num', 'Product_Num'),  
    KEY 'user_idx' ('Serial_Num'),  
    KEY 'product_idx' ('Product_Num'),  
    CONSTRAINT 'user_idx' FOREIGN KEY('Serial_Num') REFERENCES 'User'('Serial_Num')  
    ON DELETE NO ACTION ON UPDATE NO ACTION,  
    CONSTRAINT 'product_idx' FOREIGN KEY('Product_Num') REFERENCES  
    'Product'('Product_Num') ON DELETE NO ACTION ON UPDATE NO ACTION,  
);
```


G. Order

```
CREATE TABLE Evaluation (  
    'Serial_Num'    int            NOT NULL    AUTO_INCREMENT,  
    'Product_Num'   int            NOT NULL,  
    PRIMARY KEY ('Serial_Num'),  
    KEY 'user_idx' ('Serial_Num'),  
    CONSTRAINT 'user_idx' FOREIGN KEY('Serial_Num') REFERENCES 'User'('Serial_Num')  
    ON DELETE NO ACTION ON UPDATE NO ACTION,  
);
```

13. Testing Plan

13.1 Objective

시스템이 의도한 방향으로 실행되고 시스템 내부의 결함을 찾기 위해 testing을 한다. 이를 위해 설치단계에 미리 계획한다. 따라서 Testing Plan에서는 Testing Policy와 여러 Test Case에 대해 기술한다.

13.2 Testing Policy

“Eat It” 시스템의 개발에서는 크게 3단계로 나누어 testing을 진행한다. 첫 번째 단계로 Development Testing, 그리고 Release Testing, User testing으로 나뉘지며, Developing Testing은 다시 Component Testing, Integrating Testing, System Testing, Acceptance Testing의 4단계로 나뉜다.

A. Development Testing

개발 과정에서 수행되는 testing으로, Component Testing은 각 Component-(unit) 단위로 개발이 진행되면, 각 요소들이 개발한 후에 제대로 작동하는지 확인하는 testing이다. Integrating Testing은 Component Testing 후 각 요소들을 하나씩 점진적으로 합치면서 하는 testing이다. System Testing은 모든 하위 시스템을 하나로 합친 후, 그 시스템이 잘 작동하는지 testing 하는 것이다. Acceptance Testing은 사용자의 정보를 이용하여 시스템에 대한 사용자의 요구사항을 testing 하는 것이다.

B. Release Testing

사용자에게 출시하기 전에 최종 시스템을 testing하는 것이다. 요구사항 명세서에 작성되었던 요구사항이 제대로 반영되었는지를 확인한다.

C. User Testing

사용자가 사용자의 환경에서 시스템을 testing 하는 것이다.

13.3 Test Case

A. User Management System

A.1 Sign up with using Naver/Kakao

- 1) User: 'Naver/Kakao' 를 이용하여 회원가입을 시도한다.
- 2) 시스템 동작: User DB에 있는 ID와 중복 여부를 확인한다.

2-1) (Sign-up 성공)

시스템 알림: "회원가입이 완료되었습니다."

시스템 동작: User DB에 새로 만든 User 정보를 저장한다.

2-2) (Sign-up 실패)

시스템 알림: "이미 사용중인 ID입니다."

시스템 동작: Sign-up 시퀀스를 다시 시작한다.

A.2 Login with using Naver/Kakao

- 1) User: 'Naver/Kakao' 를 이용하여 로그인을 시도한다.
- 2) 시스템 동작: 입력 받은 ID/PW가 User DB에 있는지 확인한다.

2-1) (Log-in 성공)

시스템 알림: "로그인이 완료되었습니다."

시스템 동작: User DB, Taste Model DB에서 일치하는 데이터를 불러온다.

2-2) (Log-in 실패)

시스템 알림: "존재하지 않는 ID이거나, PW가 맞지 않습니다."

시스템 동작: Log-in 시퀀스를 다시 시작한다.

B. Taste Analysis System

1) User: Taste Model DB에 수정할 data를 입력한다.

2) 시스템 동작: User가 입력한 data를 Taste Model DB에 업데이트한다.

2-1) (Update 성공 – First evaluation)

시스템 알림: "취향 모델 생성이 완료되었습니다."

시스템 동작: 초기 화면으로 돌아간다.

2-2) (Update 성공 – Post evaluation)

시스템 알림: "취향 모델 수정이 완료되었습니다."

시스템 동작: 초기 화면으로 돌아간다.

2-3) (Update 실패)

시스템 알림: "취향 모델 생성에 실패했습니다."

시스템 동작: First Evaluation 시퀀스를 다시 시작한다.

C. Menu Recommendation System

1) User:

User의 Taste Model DB에서 data를 불러오기 위해 요구된 정보를 입력한다.

2) 시스템 동작: Taste Model DB에서 data를 불러온다.

2-1) (Update 성공)

시스템 알림: "추천 목록을 생성합니다."

시스템 동작: Taste Model DB를 이용하여 Product DB에서 entity들을 가져와 Item Display System에 전달한다.

2-2) (Update 실패 – First Evaluation을 거치지 않은 경우)

시스템 알림: "자신의 취향 평가를 먼저 해주시기 바랍니다."

시스템 동작: B.1로 돌아가 다시 시작한다.

D. Searching System

D.1 Product Search

1) User: 제품(음식)명을 입력하여 search product를 요청한다.

2) 시스템 동작: 해당하는 제품이 Product DB에 있는지 확인하고 그 결과를 출력하고 Item Display System으로 전달한다.

2-1) (Products exist)

시스템 알림: "검색이 완료되었습니다."

시스템 동작: 검색 결과를 출력한다.

2-2) (No Products)

시스템 알림: "해당하는 제품이 존재하지 않습니다."

시스템 동작: Product search 시퀀스를 다시 시작한다.

D.2 Category Search

1) User: 제품의 카테고리(범주)를 입력하여 search product를 요청한다.

2) 시스템 동작: 해당하는 제품이 Product DB에 있는지 확인하고 그 결과를 출력하고 Item Display System으로 전달한다.

2-1) (Products exist)

시스템 알림: "검색이 완료되었습니다."

시스템 동작: 검색 결과를 출력한다.

2-2) (No Products)

시스템 알림: "세부 필터에 해당하는 제품이 존재하지 않습니다."

시스템 동작: Product search 시퀀스를 다시 시작한다.

E. Item Display System

E.1 Display with Recommendation and Search System

1) User:

Recommendation System – User의 Taste Model DB에서 data를 불러오기 위해 User의 정보를 입력한다.

Search System – 검색하고자 하는 Product의 정보를 입력한다.

2) 시스템 동작: 화면에 표시할 상품들의 목록을 받는다.

2-1) (List exist)

시스템 알림: "결과는 다음과 같습니다."

시스템 동작: 상품들의 목록을 UI에 출력한다.

2-2) (No List – Recommendation System)

시스템 알림: "사용자의 취향 평가를 먼저 진행해주시오."

시스템 동작: First Evaluation 시퀀스로 돌아가 다시 시작한다.

2-3) (No List – Search System)

시스템 알림: "검색 결과가 존재하지 않습니다."

시스템 동작: Product search 시퀀스로 돌아가 다시 시작한다.

E.2 Store in Cart DB

1) User:

특정 상품에 대해 “장바구니에 담기”를 눌러 Cart DB에 저장을 요청한다.

2) 시스템 동작: User가 담은 상품의 정보를 Cart DB에 저장한다.

2-1) (저장 성공)

시스템 알림: “장바구니 담기에 성공하였습니다.”

시스템 동작: 상품의 정보를 Cart DB에 저장한다.

2-2) (저장 실패)

시스템 알림: “장바구니 담기에 실패하였습니다.”

시스템 동작: Cart System 시퀀스를 다시 시작한다.

F. Ordering System

F.1 Display Cart

1) User: User의 Cart DB에 있는 data를 불러올 것을 요청한다.

2) 시스템 동작: Cart DB에서 상품들의 정보를 읽어온다.

2-1) (Read 성공)

시스템 알림: “장바구니의 정보를 가져옵니다.”

시스템 동작: 장바구니 정보를 UI에 출력한다.

2-2) (Read 실패)

시스템 알림: “장바구니의 정보를 가져오는 것에 실패했습니다.”

시스템 동작: Cart Display 시퀀스를 다시 시작한다.

F.2 Edit Cart

- 1) User: 장바구니의 data를 수정한다.
- 2) 시스템 동작: User가 수정한 data를 Cart DB에 업데이트한다.

2-1) (Update 성공)

시스템 알림: "수정이 완료되었습니다."

시스템 동작: Cart DB의 data를 업데이트한다.

2-2) (Update 실패)

시스템 알림: "업데이트에 실패했습니다."

시스템 동작: Cart Display 시퀀스를 다시 시작한다.

F.3 Order

- 1) User: 선택한 product의 주문을 결정한다.
- 2) 시스템 동작: Kakao Pay API와 연동되어 결제 기능을 제공한다.

2-1) (성공)

시스템 알림: "결제가 완료되었습니다."

시스템 동작: Order DB를 업데이트한다.

2-2) (Update 실패)

시스템 알림: "결제에 실패했습니다."

시스템 동작: Cart Display 시퀀스를 다시 시작한다. (Order 전으로)

G. My Page System

G.1 Order Management

1) User: UI를 통해 '주문 관리'를 선택한다.

2) 시스템 동작: 주문 정보를 읽어온다.

2-1) (Read 성공)

시스템 알림: "주문 정보를 가져옵니다."

시스템 동작: 읽어온 주문 정보를 UI에 출력한다.

User: 주문 취소를 원하는 경우 취소 요청을 한다.

시스템 동작: 취소가 가능한지를 탐색한다.

2-1-1) (가능한 경우 - 배송 전 상태)

시스템 알림: "주문 취소가 완료되었습니다."

시스템 동작: Order DB에서 data를 삭제한다.

2-1-2) (불가능한 경우)

시스템 알림: "주문 취소가 불가능합니다."

시스템 동작: Order Management 시퀀스를 다시 시작한다.

2-2) (Read 실패)

시스템 알림: "주문 정보 가져오는 데에 실패했습니다."

시스템 동작: Order Management 시퀀스를 다시 시작한다.

G.2 Order Evaluation

- 1) User: UI를 통해 주문 메뉴 평가를 선택하여 평가 data를 입력한다.
- 2) 시스템 동작: 입력 받은 data를 Taste Model DB에 업데이트한다.

2-1) (Update 성공)

시스템 알림: "평가가 완료되었습니다."

시스템 동작: Taste Analysis System으로 돌아간다.

2-2) (Update 실패)

시스템 알림: "오류가 발생했습니다. 다시 해주시기 바랍니다."

시스템 동작: Evaluation 시퀀스를 다시 시작한다.

14. Development Environment

14.1 Objectives

Development Environment에서는 Programming Language, IDE와 같이 시스템을 개발하기 위해 필요한 Front-end, Back-end의 개발 환경과 외부 API(Application Programming Interface)에 대하여 서술한다.

14.2 Front-End Environment

A. Android Studio



[Figure 20] Logo of Android Studio

'Eat it!' 시스템은 Android Application이므로 Android 플랫폼 개발에 보편적으로 사용되는 Android Studio IDE를 사용하여 Front-End 개발을 진행한다. Android Studio는 Google에서 제공하는 Android 플랫폼 Application 개발용 IDE로 Android Application을 개발하기 위한 다양한 Tool과 Android Virtual Machine이 제공된다.

B. Java



[Figure 21] Logo of Java

본 시스템의 Front-End 개발 언어로는 Android Application 개발에 주로 사용되는 Java를 사용한다. Kotlin도 Android Application 개발에 사용되는 추세이지만 Java가 팀원들에게 더욱 친숙하기 때문에 빠른 개발을 위해 Java를 개발 언어로 선택했다. Java는 널리 사용되는 Object-Oriented Programming Language로 Application을 모듈화 하여 개발하기에 용이하다.

14.3 Back-End Environment

A. Python and Pycharm



[Figure 22] Logo of Pycharm

본 시스템의 Back-End 개발에는 Python 언어를 사용하고, 개발 환경으로는 Pycharm IDE를 사용한다. Back-End에서는 Server와 Machin-Learning을 위한 Framework들을 적용해야 하기 때문에 외부 Framework를 적용하기에 편리한

Python을 개발 언어로 선택했다. 또한 Python 가상환경을 생성, 관리하기에 용이하고 프로젝트를 효과적으로 구성할 수 있는 Pycharm을 IDE로 선택했다.

B. Django



[Figure 23] Logo of Django

Back-End에서 데이터베이스를 관리하고 Mobile Application과 통신하기 위해 오픈 소스 웹 프레임워크인 Django를 사용한다. Python 환경에서 사용할 수 있기 때문에 Django를 서버 개발용 Framework로 선택했다.

14.4 APIs and Tools

A. TensorFlow



[Figure 24] Logo of TensorFlow

Back-End에서 Deep Learning을 구현하기 위한 라이브러리로 TensorFlow를 사용한다. TensorFlow는 Google에서 제공하는 수학 라이브러리로, Machine Learning과 Deep Neural Network 구현을 위해 사용된다. Tensor 형태로 데이터를 표현하여 Machine Learning Model을 학습할 수 있으며 Python에서 사용하기에 용이하다.

B. NAVER and Kakao Login API



[Figure 25] Logo of NAVER and Kakao

‘Eat it!’ 시스템에서는 빠른 회원가입과 로그인을 통한 사용자 경험 향상을 위해 NAVER와 Kakao의 간편 로그인 기능을 지원한다. 간편 로그인 기능 구현을 위해 NAVER, Kakao SDK의 Login API를 프로젝트에 적용한다.

C. Github



[Figure 26] Logo of Github

본 프로젝트에서의 원활한 협업과 버전 관리를 위해 Github를 사용한다. Github는 Distributed Version Management 서비스인 Git을 지원하는 Web Hosting 서비스로, 가장 널리 사용되는 Git Hosting 사이트이다. Github를 통해 본래 Command-Line 서비스인 Git을 Web과 GUI Application으로 편리하게 이용할 수 있다. 팀장이 Master Branch를 관리하고 개발해야 하는 Subsystem마다 Branch를 생성하여 팀원들이 나누어 개발한다.

15. Index

15.1 Objective

Index에서는 본 문서에서 사용된 시각 자료의 인덱스를 철자 순서대로 기술한다.

15.2 Figure Index

[Figure 1] Logo of UML	13
[Figure 2] Example of Class Diagram	14
[Figure 3] Example of Sequence Diagram.....	14
[Figure 4] Example of State Diagram	15
[Figure 5] Example of ER Diagram.....	16
[Figure 6] Logo of draw.io	17
[Figure 7] Logo of JSON.....	53
[Figure 8] User Entity	61
[Figure 9] Evaluation_Pd Entity	61
[Figure 10] Evaluation Entity.....	62
[Figure 11] Taste Model Entity.....	62
[Figure 12] Product Entity.....	62
[Figure 13] Cart Entity.....	63
[Figure 14] Order Entity	63
[Figure 15] Survey Relationship	64
[Figure 16] Update Relationship	64
[Figure 17] Recommend Relationship	65
[Figure 18] Select Relationship	65
[Figure 19] Edit Relationship.....	66

[Figure 20] Logo of Android Studio	83
[Figure 21] Logo of Java	84
[Figure 22] Logo of Pycharm.....	84
[Figure 23] Logo of Django	85
[Figure 24] Logo of TensorFlow	85
[Figure 25] Logo of NAVER and Kakao	86
[Figure 26] Logo of Github	86

15.3 Table Index

[Table 1] Version History	12
[Table 2] Login Request	54
[Table 3] Login Response	54
[Table 4] Signup Request	55
[Table 5] Signup Response	55
[Table 6] Taste Analysis Request	55
[Table 7] Taste Analysis Response	56
[Table 8] Recommend Menu View Request	56
[Table 9] Recommend Menu View Response	56
[Table 10] Menu Search Request.....	57
[Table 11] Menu Search Response.....	57
[Table 12] Product Post Request.....	58
[Table 13] Product Post Response.....	58
[Table 14] Ingredient of product View Request.....	59
[Table 15] Ingredient of product View Response	59

.

15.4 Diagram Index

[Diagram 1] System Organization Block Diagram	18
[Diagram 2] User Management System Block Diagram	20
[Diagram 3] Taste Analysis System Block Diagram	20
[Diagram 4] Menu Recommendation System Block Diagram	21
[Diagram 5] Searching System Block Diagram	22
[Diagram 6] Item Display System Block Diagram	22
[Diagram 7] Ordering System Block Diagram	23
[Diagram 8] My Page System Block Diagram	24
[Diagram 9] Class Diagram of User Management System	25
[Diagram 10] Sequence Diagram of Signup System	27
[Diagram 11] Sequence Diagram of Login System	27
[Diagram 12] State Diagram of User Management System	28
[Diagram 13] Class Diagram of Taste Analysis System	29
[Diagram 14] Sequence Diagram of Taste Analysis System	32
[Diagram 15] State Diagram of Taste Analysis System	33
[Diagram 16] Class Diagram of Menu Recommendation System	34
[Diagram 17] Sequence Diagram of Menu Recommendation System	36
[Diagram 18] State Diagram of Menu Recommendation System	37
[Diagram 19] Class Diagram of Searching System	38
[Diagram 20] Sequence Diagram of Searching System	40
[Diagram 21] State Diagram of Searching System	41
[Diagram 22] Class Diagram of Item Display System	42
[Diagram 23] Sequence Diagram of Item Display System	44
[Diagram 24] State Diagram of Item Display System	44

[Diagram 25] Class Diagram of Ordering System	45
[Diagram 26] Sequence Diagram of Ordering System.....	47
[Diagram 27] State Diagram of Ordering System	48
[Diagram 28] Class Diagram of My Page System.....	49
[Diagram 29] Sequence Diagram 1 of My Page System.....	51
[Diagram 30] Sequence Diagram 2 of My Page System.....	51
[Diagram 31] State Diagram of My Page System.....	52
[Diagram 32] Overall ER Diagram	60