

---

# Requirement Specification

---

## Eat it!

---



제출일	2019. 05. 05	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 .....	7
1.1	Objective .....	7
1.2	Readership .....	7
A.	User Requirements Readership.....	7
B.	System Requirements Readership.....	7
1.3	Document Structure.....	7
A.	Preface .....	7
B.	Introduction .....	7
C.	Glossary .....	8
D.	User Requirements Definitions .....	8
E.	System Architecture .....	8
F.	System Requirements Definitions .....	8
G.	System Models .....	8
H.	System Evolution .....	9
I.	Appendices .....	9
J.	Index .....	9
1.4	Version History.....	9
2.	Introduction .....	10
2.1	Objective .....	10
2.2	Needs .....	10
2.3	Eat it! .....	12
2.4	Expected Effects .....	14
A.	Improve User Experience.....	14
B.	Increase Seller's Sales.....	15

3. Glossary .....	16
3.1 Objective .....	16
3.2 Term Definition .....	16
4. User requirements definitions .....	19
4.1 Objective .....	19
4.1 Functional Requirements .....	19
A. Sign Up .....	19
B. Log In .....	19
C. First Menu Evaluation .....	19
D. Search .....	19
E. Main Menu .....	20
F. Shopping Cart .....	20
G. Order .....	20
H. My Page .....	20
I. Post Evaluation .....	21
4.2 Non-functional Requirements .....	21
A. Product Requirements .....	21
B. Organizational Requirements .....	22
C. External Requirements .....	23
5. System architecture .....	24
5.1 Objective .....	24
5.2 Overall Architecture .....	24
5.3 First Evaluation System .....	25
5.4 Menu Recommendation System .....	26
5.5 Ordering System .....	27
5.6 Post Evaluation System .....	28

6.	System Requirements Specifications .....	30
6.1	Objectives .....	30
6.2	Functional Requirements .....	30
A.	Login Function.....	30
B.	Explore First Evaluation Function .....	31
C.	Preference Analysis Function .....	31
D.	Main Menu Recommendation Function .....	32
E.	Menu Search Function .....	32
F.	Send to Cart Function .....	33
G.	Display Shopping Cart Function .....	33
H.	Edit and Submit Shopping Cart Function .....	34
I.	Order Process Function .....	34
J.	Display My Page Function .....	35
K.	Order Management Function.....	35
L.	Post Evaluation Function .....	36
6.3	Non-functional Requirements .....	36
A.	Product Requirements .....	36
B.	Organizational Requirements .....	37
C.	External Requirements.....	38
6.4	Scenario .....	38
A.	Signup/Login Scenario .....	38
B.	First Menu Evaluation Scenario .....	38
C.	Main Menu Recommendation Scenario .....	39
D.	Menu Search Scenario .....	39
E.	Add to Card and Order Scenario .....	39
F.	After Order Evaluation Scenario .....	40

7.	System Models .....	41
7.1	Objective .....	41
7.1	Context Models.....	41
A.	Context Diagram .....	41
B.	Process Diagram.....	42
7.2	Interaction Models.....	42
A.	Use-Case Diagram.....	42
B.	Tabular Description for each Use Case .....	43
C.	Sequence Diagram.....	46
7.3	Structural Models .....	48
A.	Class Diagram .....	48
7.1	Behavioral Models.....	49
A.	Activity Diagram.....	49
B.	State Diagram .....	50
8.	System Evolution .....	51
8.1	Objective .....	51
8.2	Limitation.....	51
8.3	Evolution .....	51
A.	Platform Extension.....	51
B.	Communication System .....	52
C.	Grocery Management .....	52
9.	Appendices.....	53
9.1	Objective .....	53
9.1	Database Requirements.....	53
A.	User Information .....	53
B.	Sample Information.....	53

C.	Product Information .....	54
D.	Shopping Chart Information.....	54
E.	Order Information .....	54
9.2	Hardware Requirements .....	54
10.	Index .....	55
10.2	Table Index .....	55
10.3	Figure Index .....	56
10.4	Diagram Index .....	56

# 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

### A. User Requirements Readership

User Requirements는 본 시스템을 사용할 고객을 예상 독자로 간주한다. 따라서 전문성이 없는 독자를 고려하여 자연어와 다이어그램 형태를 통해 본 시스템이 제공하는 서비스들과 운영적인 제약사항에 대해 기술한다.

### B. System Requirements Readership

System Requirements는 시스템을 개발하는 소프트웨어 개발자나 시스템의 설계자와 같은 개발진을 주요 독자로 간주한다. 시스템의 기능과 서비스, 운영상의 제약사항들을 세부적으로 기술하여 구조화된 문서 형태로 정리한다. 구현되어야 하는 기능들에 대한 부분은 고객과 계약자 사이의 계약에 포함될 수 있다.

## 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 서비스의 필요성에 대해 논한다. 어떠한 배경을 바탕으로 시스템이 제안되었는지 밝히고 Eat it! 서비스의 기능, 그리고 본 서비스가 다른 시스템들과 어떻게 상호작용하는지 설명한다. 마지막으로 본 서비스의 시장 도입 효과에 대하여 논한다.

## C. Glossary

Glossary에서는 본 문서에서 사용된 기술적 용어들을 정의한다. 배경지식이 없는 독자들이 문서를 이해하는 데에 어려움이 없도록 하는 것이 취지이다.

## D. User Requirements Definitions

User requirement definition에서는 사용자에게 어떤 서비스를 제공하는지에 대해서 설명한다. Functional requirement와 non-functional requirement로 분류해 서술하며, 이는 시스템을 구성하는 기능들을 모두 포함한다. 자연어와 다이어그램을 활용하여 사용자의 이해를 돕고, 제품이나 프로세스가 따라야하는 표준이 있다면 이를 명시한다.

## E. System Architecture

System architecture에서는 시스템의 구조에 대해 대략적으로 설명한다. 도식도 (diagram)를 통하여 시스템의 전체적인 구조, 관계를 설명하고 각 시스템의 기능, 구조를 추상적으로 설명한다.

## F. System Requirements Definitions

시스템 Requirements specification 단계는 Functional Requirements와 Non-functional Requirements의 요구사항을 구체적으로 명시하는 것이다. 구조는 한 기능별 이름, 설명, 입력, 출력, 처리, 조건 6 부분으로 나눌 수 있다.

## G. System Models

System Models에서는 본 시스템과 그 구성요소, 그리고 동작 환경 사이의



관계들을 시각적인 모델로 나타낸다. 이는 Context Diagram, Process Diagram, Use-Case Diagram 등 다양한 Diagram을 포함한다.

## H. System Evolution

System Evolution에서는 본 시스템이 기반으로 하는 기본적인 가정들을 정의하고 hardware evolution, user needs의 변화로 인한 발생 가능한 변경사항들을 기술한다. 또한 시스템을 이러한 변경사항에 맞춰 발전시킬 방향성에 대해 논한다. 이는 추후 시스템의 변경에 제약이 될 design decision을 피하기 위한 과정이다.

## I. Appendices

Appendices에서는 개발되는 어플리케이션과 관련된 자세하고 구체적인 정보를 제공한다. 먼저 시스템에서 사용되는 데이터의 논리적 조직을 정의하고 데이터 사이의 관계에 대해 기술한다. 또한 시스템을 사용하기 위해 필요한 hardware의 요구사항에 대해 설명한다.

## J. Index

Index에서는 본 문서에서 사용된 그림과 다이어그램, 테이블에 대한 인덱스를 철자 순서대로 기술한다.

## 1.4 Version History

Version	Modified Date	Explanation
0.1	2019. 04. 28	요구사항 명세서 10장에 대한 문서 목차 작성
1.0	2019. 05. 01	Preface 작성 및 완성
2.0	2019. 05. 04	명세서 전 항목 작성 완료
3.0	2019. 05. 04	각 항목들 통합 후 Glossary, Preface 작성
4.0	2019. 05. 05	Index 작성, 추가 사항 및 수정 사항 반영

[Table 1] Version History

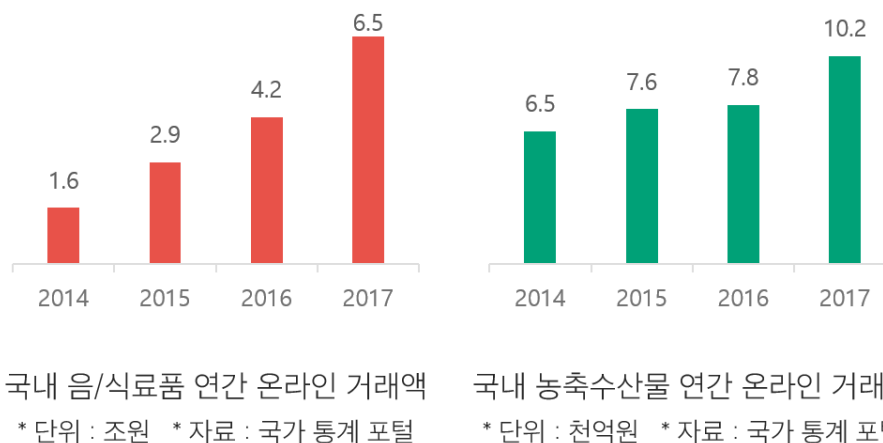
## 2. Introduction

### 2.1 Objective

Introduction에서는 본 서비스의 필요성에 대해 논한다. 어떠한 배경을 바탕으로 시스템이 제안되었는지 밝히고 Eat it! 서비스의 기능, 그리고 본 서비스가 다른 시스템들과 어떻게 상호작용하는지 설명한다. 마지막으로 본 서비스의 시장 도입 효과에 대하여 논한다.

### 2.2 Needs

최근 몇 년간 국내 온라인 식품 쇼핑 시장은 꾸준히 성장해왔다. 맞벌이 가정과 일인 가구가 대폭 증가하고 인터넷 쇼핑의 접근성이 나날이 좋아지고 있기 때문이다. 특히 20대, 30대의 젊은 세대들은 시간을 들여 나가서 장을 보는 것보다 스마트폰을 통해 터치 몇 번으로 식료품을 주문할 수 있는 식료품 쇼핑 어플리케이션이나 배달 어플리케이션을 선호하는 추세이다.

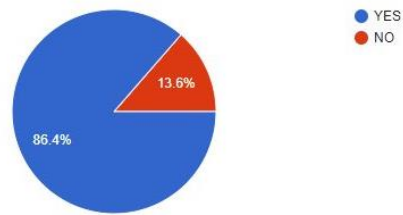


[Figure 1] 연간 국내 식료품 거래액

다음 통계청의 자료에 따르면 국내 음/식료품 연간 온라인 거래액은 2014년의 1.6조원에서 2017년 6.5조원까지 3년간 폭발적으로 성장했다. 또한 음식의 재료가 되는 농축수산물의 거래액도 2014년 6천5백억원에서 2017년 약 1조원으로 증가했다.

웹사이트나 어플리케이션 등 온라인으로 식품을 구매한 적이 있습니까?  
(网购过食品吗?)

(103 条回复)



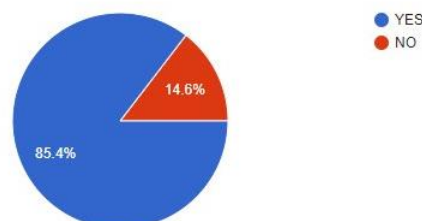
[Figure 2] 설문조사 결과 1

또한 우리는 젊은 층의 인터넷 식품 쇼핑 이용 실황에 대해 조사하기 위해 20대 103명을 대상으로 설문조사를 시행했다. 그 결과 103명 중 86.4%가 웹사이트 혹은 어플리케이션을 통해 식품을 구매한 적이 있다고 응답했다. 20대중 대다수가 온라인 인터넷 쇼핑을 경험해 본 것이다. 실제로 이마트, 홈플러스, 롯데마트 등의 대형 마트에서 홈페이지와 스마트폰 어플리케이션을 통한 온라인 주문 서비스를 활발히 운영 중이며 마켓컬리와 같은 온라인 전용 식품 쇼핑 서비스 또한 많은 사용자들이 이용하고 있는 상황이다.

그러나 시장의 성장과는 별개로 온라인 식품 쇼핑 서비스에 대한 만족도가 높기만 한 것은 아니다. 직접 마트에 가지 않고 컴퓨터나 스마트폰만을 사용하여 간편하게 식료품을 주문할 수 있다는 큰 장점이 있지만, 온라인몰 특성상 그 종류가 너무 방대하여 주문할 상품을 하나하나 고르는 데에서 오는 피로가 크다.

상품을 하나하나 찾아서 구매하는 것에 귀찮음을 느낀 적이 있습니까?  
(有觉得买的时候一个一个找着买麻烦的时候吗?)

(103 条回复)



[Figure 3] 설문조사 결과 2

설문조사의 다음 항목에 대한 사용자들의 응답들을 보면 온라인 식품 쇼핑에서 상품을 하나하나 찾아서 구매하는 것에 불편함을 느낀 사용자가 85.4%로, 대부분의 사용자들이 불편을 경험한 것으로 나타났다.

사용자가 온라인 식품 쇼핑 서비스를 이용하려 할 때의 사용자 경험은 주로 다음과 같다. 어플리케이션을 실행하면 주요 상품들을 표시하는 메인 화면을 띄운다. 이 때 사용자는 사려고 하는 특정 상품을 떠올린 뒤 카테고리 탭에서 그에 맞는 카테고리를 선택한다. 그 후 세부 카테고리를 선택하고, 인기순이나 가격순, 추천순으로 정렬한 뒤 마음에 드는 품목을 장바구니에 담는다. 이러한 과정을 거치지 않고 검색을 이용하여 원하는 상품을 바로 검색할 수는 있지만, 두 경우 모두 내가 무슨 품목을 살 것인지 파악한 상태여야 빠르게 쇼핑을 할 수 있다. 무엇을 구매할 지 확실히 정한 상태가 아니라면 자신의 취향과는 관계없이 노출되는 수많은 상품들을 하나하나 보며 골라야 하고 이 과정에서 피로감을 느끼게 된다. 이러한 불편함은 비단 온라인 서비스 뿐만 아니라 오프라인 매장에서 쇼핑을 할 때에도 겪을 수 있는 경험이다.

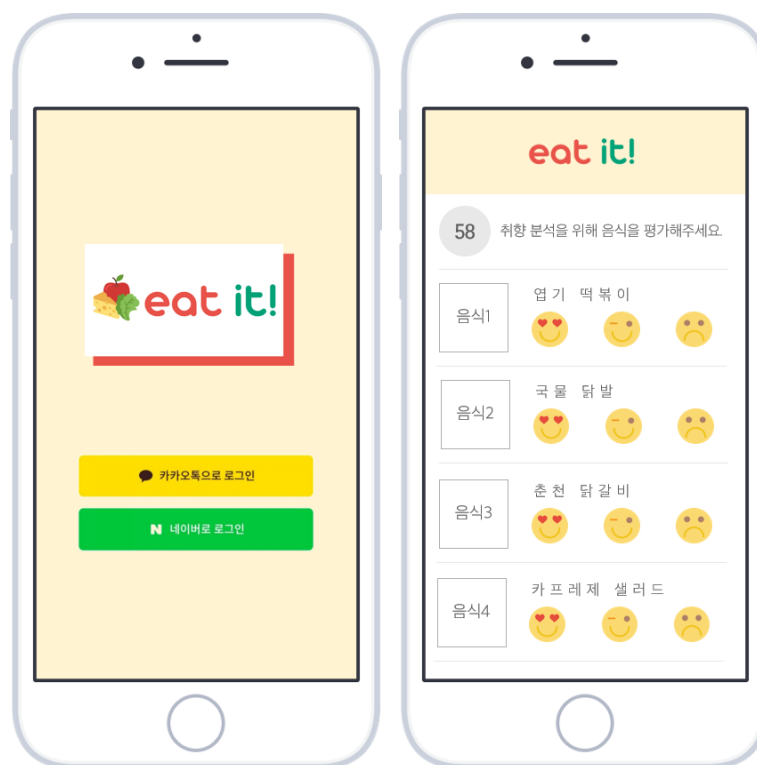
따라서 온라인 식품 쇼핑 시장의 성장에 맞춰 식품 쇼핑 서비스의 사용자 경험을 향상하기 위해 상품 선택의 피로를 줄이기 위한 해결책이 필요하다. 이러한 필요성에 따라 사용자의 음식 취향을 파악하여 사용자가 좋아할 만한 메뉴들을 추천함으로써 선택의 피로를 덜고 사용자 경험을 향상시키는 온라인 음식 쇼핑 시스템을 계획하게 되었다.

## 2.3 Eat it!

Eat it!은 사용자의 음식 취향을 Deep Learning 모델로 파악하여 사용자가 좋아할 만한 음식을 추천하는 온라인 음식 쇼핑 시스템이다. 많은 양의 상품들을 전부 노출해 선택에 피로감을 느끼게 했던 기존의 음식 쇼핑 서비스에서 탈피하고 사용자에게 적합하다고 예측되는 상품들만 제공하여 사용자 경험을 높이는 것을 목표로 한다. 특정 요리를 하기 위해 필요한 재료들을 검색하고 하나하나 담는 것이 아닌,

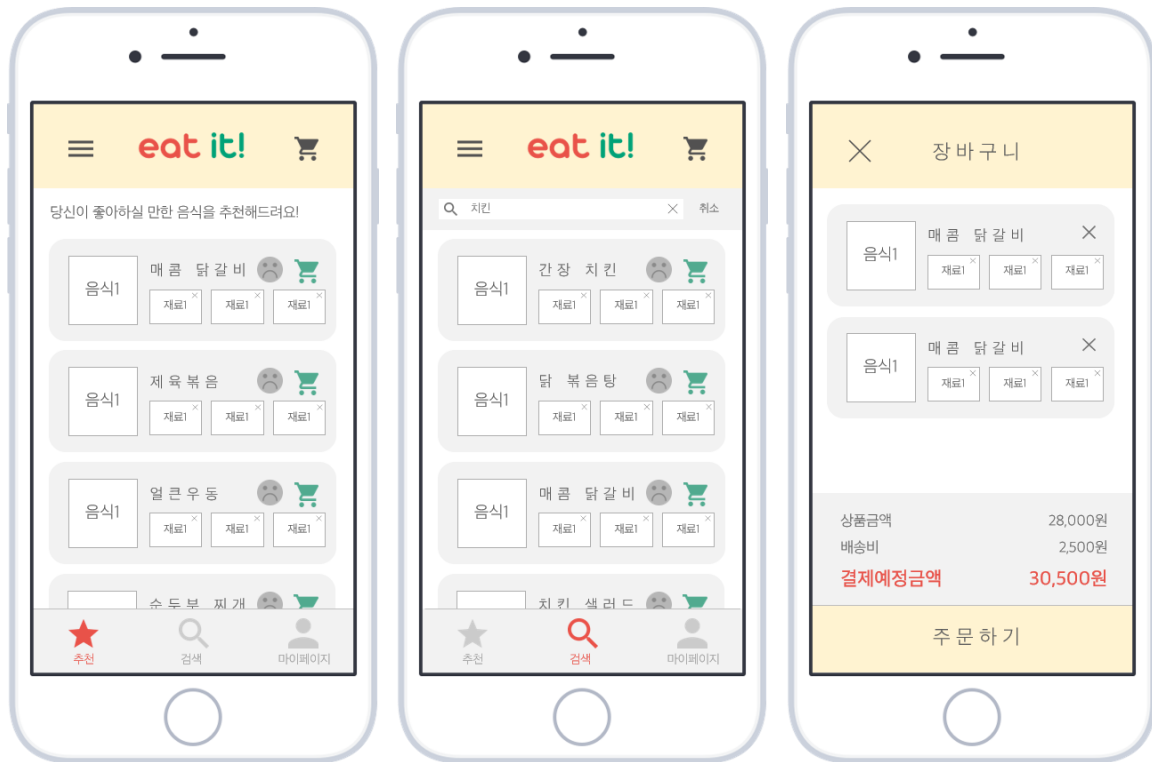
추천된 요리를 선택하면 필요한 재료들을 한 번에 담아 구매하는 방식으로 식품 쇼핑에 소모되는 시간이나 피로감을 최소화한다.

본 시스템은 크게 두 부분으로 구성된다. 가입한 유저가 정해진 숫자의 미리 제공되는 메뉴들을 본인 취향대로 평가하고 그 평가를 기반으로 개인에 맞는 취향 모델을 구축하는 시스템과 실질적인 쇼핑 서비스를 제공하는 시스템이다. 후자는 사용자에게 맞는 메뉴 추천, 구매, 주문 내역 관리, 구매한 상품 평가를 통한 취향모델 업데이트 기능을 제공한다.



[Figure 4] 가입과 취향 분석을 위한 메뉴 평가 mockup

처음 가입한 사용자는 특정 수 이상의 미리 제공된 메뉴들을 평가한 후 쇼핑 서비스를 이용할 수 있다. 사용자가 평가한 데이터로 맞춤형 취향 모델을 구축하기 때문이다. 보편적인 음식 메뉴 세트를 좋음, 보통, 싫음 세 가지 class로 나누어 평가하게 한 후 이 데이터를 machine learning model의 training data로 학습한다. 만족스러운 사용자 경험을 위해 추천의 정확도가 중요하기 때문에 기계학습 모델은 deep learning 모델을 사용하여 모델의 성능을 높인다.



[Figure 5] 쇼핑 메뉴 mockup

사용자의 평가 데이터로 취향 모델을 학습한 후에는 쇼핑 서비스를 제공할 수 있다. 가장 중요한 기능은 메뉴 단위로 사용자에게 적합한 음식을 추천하는 추천 기능이다. 학습된 모델을 바탕으로 상품들 중 사용자가 좋아할 것으로 예측되는 상품들을 우선적으로 노출한다. 사용자의 필요에 따라 조리된 정도를 선택할 수 있고 원하는 메뉴에 필요한 재료들을 한 번에 장바구니에 담는다. 사용자가 원하는 특정 상품을 검색 후 담을 수도 있다. 장바구니에 담은 후 금액과 담긴 상품들을 확인한 후 결제를 진행한다. 결제 후에는 마이페이지 탭에서 주문한 내역을 확인하고 구입했던 상품을 평가함으로써 사용자의 취향 모델을 업데이트하여 정확도를 높인다.

## 2.4 Expected Effects

### A. Improve User Experience

본 서비스의 가장 큰 기대 효과는 사용자의 쇼핑 경험 향상이다. 사용자는 말 그대로 터치 몇 번을 통해 자신이 좋아하는 음식을 구매할 수 있으며, 요리를 하기

위해 재료와 레시피를 검색해서 하나하나 구입할 필요가 없으므로 시간이 부족하거나 식품 쇼핑에 귀찮음을 느끼는 사용자에게 최적의 서비스로 다가갈 것이다.

## **B. Increase Seller's Sales**

사용자에게 적합한 상품만을 추천함으로써 판매자의 이익 또한 증가시킬 수 있다. 일차적으로 사용자는 자신의 취향에 기반하여 추천된 상품을 무작위로 노출된 상품보다 신뢰성을 가지고 구매에 고려하게 된다. 또한, 한꺼번에 많은 상품의 목록을 훑는 것이 아니라 선별된 상품들만 노출되기 때문에 하나의 상품을 더욱 자세히 관찰하게 된다. 이에 따라 사용자의 취향 기반으로 추천되는 상품들은 구매 확률이 높아질 것이기 때문에, 판매자의 매출 증대를 기대할 수 있다.

## 3. Glossary

### 3.1 Objective

Glossary에서는 본 문서에서 사용된 기술적 용어들을 정의한다. 배경지식이 없는 독자들이 문서를 이해하는 데에 어려움이 없도록 하는 것이 취지이다.

### 3.2 Term Definition

Term	Definition
하드웨어(hardware)	중앙 처리 장치, 모니터, 자판, 컴퓨터 기억 장치, 그래픽 카드, 사운드 카드, 메인보드와 같은 컴퓨터의 물리적 부품. 본 문서에서는 Eat it! 서비스의 대상이 되는 모바일 기기를 말한다.
기계학습(Machine Learning)	인공 지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야
딥 러닝(Deep Learning)	여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화 (abstractions, 다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업)를 시도하는 기계학습(machine learning) 알고리즘의 집합
사용자 경험(User Experience)	사용자가 어떤 시스템, 제품, 서비스를 직, 간접적으로 이용하면서 느끼고 생각하게 되는 총체적 경험
오픈 마켓(Open Market)	존의 온라인 쇼핑몰과 다르게 개인 판매자들이 인터넷에 직접 상품을 올려 매매하는 곳. 대표적인 오픈 마켓 웹사이트로는 G마켓, 옥션, 11번가 등이 있다.
안드로이드(Android)	구글에서 개발한 모바일 운영 체제
iOS	애플의 모바일 운영 체제
Web(World Wide Web)	인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 전 세계적인 정보 공간
챗봇(Chat Bot)	음성이나 문자를 통한 인간과의 대화를 통해서 특정한 작업을 수행하도록 제작된 컴퓨터 프로그램



<b>행렬(Matrix)</b>	수나 기호, 수식 등을 네모꼴로 배열한 것으로, 괄호로 묶어 표시
<b>Primary Key</b>	관계형 데이터베이스에서 조(레코드)의 식별자로 이용하기에 가장 적합한 것을 관계 (테이블)마다 단 한 설계자에 의해 선택, 정의된 후보 키
<b>Foreign Key</b>	관계형 데이터베이스에서 외래 키(외부 키, Foreign Key)는 한 테이블의 필드(attribute) 중 다른 테이블의 행(row)을 식별할 수 있는 키
<b>가상 머신(Virtual Machine)</b>	컴퓨팅 환경을 소프트웨어로 구현한 것, 즉 컴퓨터를 에뮬레이션하는 소프트웨어. 본 문서에서는 모바일 디바이스의 가상 머신을 말한다.
<b>서버(Server)</b>	클라이언트에게 네트워크를 통해 정보나 서비스를 제공하는 컴퓨터(server computer) 또는 프로그램(server program)
<b>클라이언트(Client)</b>	네트워크를 통하여 서버라는 다른 컴퓨터 시스템 상의 원격 서비스에 접속할 수 있는 응용 프로그램이나 서비스
<b>사용자(User)</b>	"eat it"의 서비스를 이용하는 사람
<b>회원가입(Sign Up)</b>	"eat it"의 회원이 되기 위해 특정 정보들을 입력하고 시스템에 등록하는 행위
<b>로그인(Log in)</b>	"eat it"에 등록된 정보를 통해 시스템에 접속하는 행위
<b>계정 연동</b>	다른 서비스에서 사용하는 ID를 "eat it"에서도 사용할 수 있도록 서비스를 연결시키는 것을 말한다.
<b>검색(Search)</b>	사용자가 입력한 단어에 적합한 정보를 찾아 보여주는 기능
<b>마이페이지(My Page)</b>	사용자가 음식 구입 후에 주문 내역을 관리하고 음식에 대한 평가를 남길 때 사용하는 페이지
<b>업데이트(Update)</b>	기존의 정보나 데이터를 수정, 변경하거나 최신의 정보나 데이터를 추가하는 일
<b>데이터베이스(Data Base)</b>	서비스를 제공할 때 필요한 정보들을 저장하고 있는 저장소
<b>API: Application Programming Interface</b>	응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스

<b>다이어그램(Diagram)</b>	정보를 조율, 묘사, 상징화하여 2차원 기하학 모델(two-dimensional geometric model)로 시각화하는 기술
<b>User Database</b>	사용자의 로그인 아이디, 비밀번호 등 개인정보와 음식에 대한 취향 정보 포함
<b>User Analysis Database</b>	Evaluation System을 통해 분석된 사용자의 취향 모델을 저장하는 저장소
<b>장바구니(Cart)</b>	장바구니와 같은 개념으로 사용자가 찜한 목록을 저장해두는 항목
<b>텐서플로우(Tensorflow)</b>	기계학습용 엔진. 검색, 음성 인식, 번역 등의 기능을 수행할 수 있는 오픈 소스로 공유 가능한 소프트웨어

[Table 2] Term Definition

## 4. User requirements definitions

### 4.1 Objective

User requirement definition에서는 사용자에게 어떤 서비스를 제공하는지에 대해서 설명한다. Functional requirement와 non-functional requirement로 분류해서술하며, 이는 시스템을 구성하는 각 기능을 모두 포함한다. 자연어와 다이어그램을 활용하여 사용자의 이해를 돕고, 제품이나 프로세스가 따라야하는 표준이 있다면 이를 명시한다.

### 4.1 Functional Requirements

#### A. Sign Up

사용자에게 회원가입에 필요한 정보를 요청하는 기능이다. 사용자는 본 시스템의 보다 정확한 서비스를 제공받기 위해서는 회원가입을 해야 한다. 회원가입은 카카오 또는 네이버 계정의 연동을 통해 간편하게 가능하며, 사용자는 자신의 나이와 성별만을 입력하면 된다.

#### B. Log In

사용자가 회원가입으로 생성된 아이디를 통해 시스템에 로그인하는 기능이다. 사용자는 시스템에 로그인하여, 시스템의 서비스를 제공받을 수 있다.

#### C. First Menu Evaluation

사용자의 음식 취향을 분석하기 위한 기능이다. 본 시스템에 가입한 후 처음으로 로그인 하는 사용자는 먼저 자신에 취향에 맞게 시스템에서 제안된 100가지 음식을 평가해야 한다. 시스템은 평가 결과를 토대로 사용자의 취향을 분석하여 "음식추천"을 위한 모델을 생성한다.

#### D. Search

사용자가 검색어를 통해 음식을 검색할 수 있는 기능이다. 사용자는 검색창에 음식의 이름을 입력하여 자신이 원하는 것을 손쉽게 찾을 수 있다. 이때 시스템은 사용자가 원하는 음식뿐만 아니라 입력된 단어를 토대로 비슷한 음식 또한 같이 보여준다.

## E. Main Menu

“First Menu Evaluation”에서 분석한 취향을 토대로 사용자에게 그들이 좋아할 만한 음식을 추천해주는 기능이다. 추천된 음식들이 나열되며, 그 음식을 만드는데 필요한 재료 또한 동시에 보여준다. 사용자는 이를 통해 자신이 원하는 음식을 선택할 수 있다.

## F. Shopping Cart

사용자가 “Main Menu”에서 마음에 든 음식을 선택할 경우 이 “장바구니(Shopping Cart)”에 담기게 된다. 사용자는 이 “장바구니”를 통해 자신이 어떤 음식을 선택했는지 한 눈에 확인할 수 있으며, 선택한 음식의 개수 및 가격 또한 확인이 가능하다. 만약 사용자가 선택한 음식을 취소하고 싶을 경우엔 ‘x’ 버튼을 눌러 쉽게 취소할 수 있다.

## G. Order

본 기능은 “Shopping Cart”의 연장선으로, 사용자는 자신의 장바구니의 담긴 음식을 “주문하기(Order)” 버튼을 클릭하여 주문할 수 있다. 이때 사용자는 자신이 주문한 음식이 배송될 주소를 입력해야 하고, 결제는 카카오페이의 mockup용 API를 연동하여 실행된다.

## H. My Page

로그인한 사용자만이 이용할 수 있는 기능이다. 사용자는 “마이페이지(My

page)”에서 자신이 주문한 내역을 확인 및 관리할 수 있으며, 구입한 음식에 대한 평가를 남길 수 있다.

## I. Post Evaluation

사용자가 주문한 음식을 평가하는 기능이다. 사용자는 음식을 구입한 후에 마이페이지에서 “좋다”, “보통”, “싫다” 3가지 선택지를 기준으로 음식에 대한 평가를 남길 수 있다. 시스템은 이런 평가들을 통해 취향 분석 모델을 업데이트한다.

## 4.2 Non-functional Requirements

Non-functional requirement는 크게 product requirement, organization requirements, external requirement로 나눌 수 있다.

### A. Product Requirements

#### A.1 Usability Requirements

본 시스템은 추천 서비스이기에 앞서 음식을 주문하는 서비스이다. 따라서 수많은 음식 중에서 사용자가 원하는 음식을 찾기 쉬워야 하며, 음식에 대한 정보(필요한 재료, 가격 등)가 직관적이어야 한다.

#### A.2 Performance Requirements

본 시스템의 핵심 서비스는 “음식 추천”이다. 따라서 추천의 정확도가 낮으면 사용자의 이용율이 떨어질 수 있기 때문에 최대한 사용자의 취향에 근접해야 한다. 또한, 수많은 음식의 방대한 데이터를 나타내는 데에 문제가 없어야 하며, 음식을 검색했을 때 performance 측면에서 문제가 없도록 해야 한다.

#### A.3 Dependability Requirement

사용자에게 추천해준 음식이 그들의 취향에 맞지 않을 경우 사용자의 신뢰를 잃을 수 있으므로 최대한 많은 데이터를 모아 분석하여 취향에 근접한 결과를 내야만 한다. 또한, 주문/구매 서비스에 대해서도 오류가 발생할 경우 사용자의

신뢰를 잃을 우려가 있기 때문에, 이러한 방면에서도 문제를 만들어서는 안 된다.

#### **A.4 Security Requirement**

본 시스템의 경우 카카오 또는 네이버 계정을 연동하여 로그인하기 때문에 그에 대한 개인정보가 유출되어서는 안 되는 건 물론이고, 결제 수단에 대한 보안이 철저히 관리되어야 한다. 또한, 사용자가 시스템을 안심하고 이용할 수 있도록 데이터베이스 관리에 유의해야 한다.

### **B. Organizational Requirements**

#### **B.1 Environmental Requirements**

사용자가 언제 어디서든 사용할 수 있도록 모바일 디바이스는 상시 네트워크에 연결되어 있어야 한다. 또한, 데이터베이스를 관리하고 디바이스와 연결할 서버는 상시 구동되어야 한다.

#### **B.2 Operational Requirements**

시스템이 추천한 음식을 사용자가 마음에 들어 하지 않을 경우, 시스템은 계속해서 사용자의 요구에 맞는 음식을 추천할 필요가 있다. 따라서 사용자가 마음에 들어 하지 않는 데이터를 추가적으로 분석하여 요구에 근접한 결과를 도출하거나, 또는 사용자에게 보다 세부적인 질문을 하여 결과에 다가갈 수 있는 기능이 필요하다.

#### **B.3 Development Requirements**

보다 정확한 추천을 하기 위해 machine learning 기술, 더 나아가 deep learning 기술의 도입이 필요하다. 그리고 본 시스템은 안드로이드 기기를 대상으로 한 어플리케이션이기 때문에, 개발 언어는 Java를 사용해야 한다.

## **C. External Requirements**

### **C.1 Regulatory Requirement**

본 시스템은 사용자에게 로그인을 요구하기에 앞서 먼저 그들의 개인정보를 수집함에 대한 동의를 구해야 하며, 취급방침에 대한 알리를 제공해야 한다. 또한, 정보를 외부에 제공하게 될 경우엔 사용자에게 동의를 먼저 구해야 한다.

### **C.2 Safety/security Requirement**

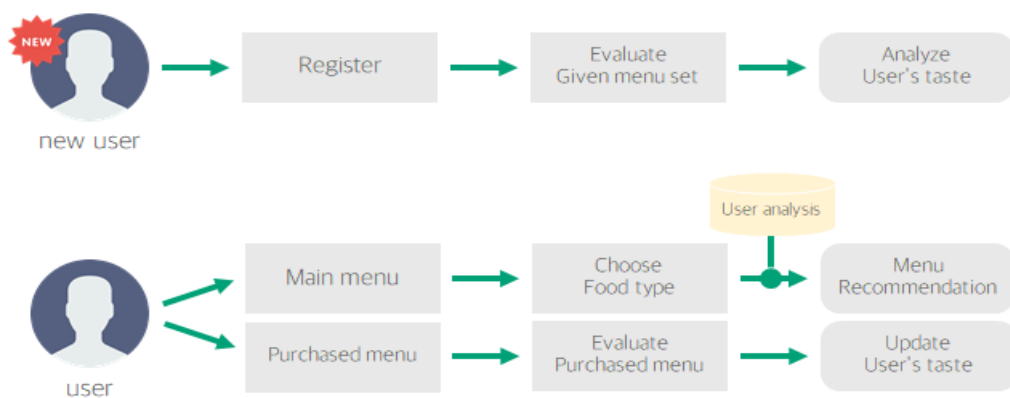
본 시스템은 사용자의 개인정보가 외부로 유출되지 않도록 해야 하며, 사용자의 결제에 오류가 발생해서는 안 된다.

## 5. System architecture

### 5.1 Objective

System architecture에서는 시스템의 구조에 대한 대략적인 설명을 한다. 도식도(diagram)를 통하여 시스템의 전체적인 구조, 관계를 설명하고 각 시스템의 기능, 구조를 추상적으로 설명한다.

### 5.2 Overall Architecture



[Diagram 1] Overall Architecture Flowchart

User가 "eat it"에 접속했을 때, 새로운 user인 경우에는 'Register'를 할 수 있도록 한다. 이 때, system은 'User database'에서 가입이력이 있는 지를 확인하고 가입이력이 없다면 'Register' 과정을 계속 진행하고 가입이력이 있다면 'log-in'을 바로 하도록 한다.

'Register'를 할 때, 주어진 100개의 메뉴에 대해서 먼저 평가하는 과정을 거친다. 음식에 대하여 매우 좋음, 보통, 매우 나쁨 이 세가지 단계로 평가를 할 수 있게 한다. User가 평가한 데이터를 바탕으로 취향을 분석 후 user별 모델을 생성하여 'User analysis database'에 저장한다.

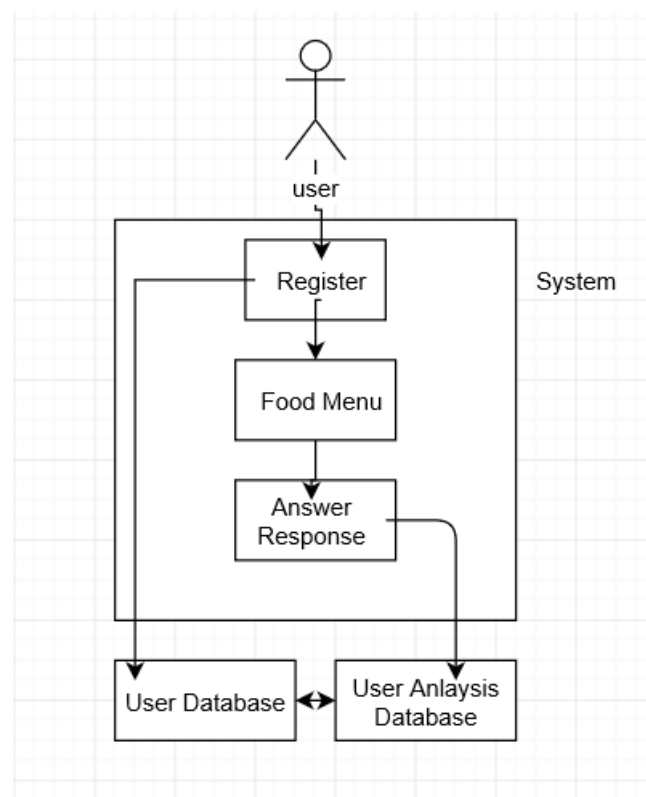
기존의 user 혹은 'Register'를 마친 user는 log-in을 하여 user가 구매하고 싶은



음식 메뉴를 검색한다. 혹은 'user analysis database'에 저장된 취향을 바탕으로 음식 메뉴와 그에 들어갈 수 있는 재료들을 추천한다. User가 검색 혹은 추천된 음식의 해당하는 재료들을 'Cart'에 담고 'order'과정을 진행한다.

'order'과정을 진행한 후에 구매한 메뉴에 대해서 평가를 하고 이 평가를 'Register'에서 생성한 'user analysis database'를 업데이트한다.

### 5.3 First Evaluation System



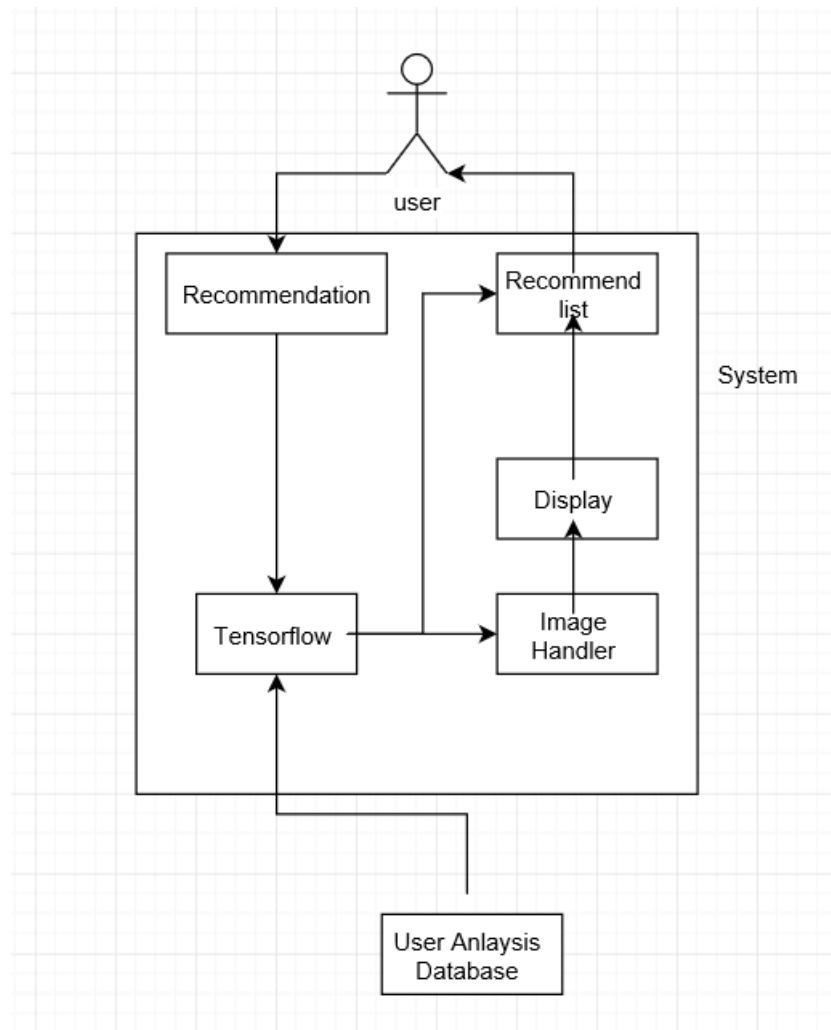
[Diagram 2] First Evaluation System

First Evaluation System은 User가 처음으로 'Eat it'에 접속했을 때 'Register'에서 user의 음식에 대한 취향을 추상적으로 분석할 수 있는 시스템이다.

User가 'Eat it'에 'Register'를 하고 접속할 때면, system은 user에게 음식 메뉴를 100개를 평가하도록 요청한다. 평가할 때 음식 메뉴별로 좋음, 보통, 나쁨 세 단계

중에서 선택하여 응답을 한다. 선호하는 음식 및 싫어하는 음식의 정보는 user의 'user database'에 연결된 'user analysis database'에 저장된다. 그 후에 user의 취향에 맞게 모델을 만들어 저장한다.

## 5.4 Menu Recommendation System



[Diagram 3] Menu Recommendation System

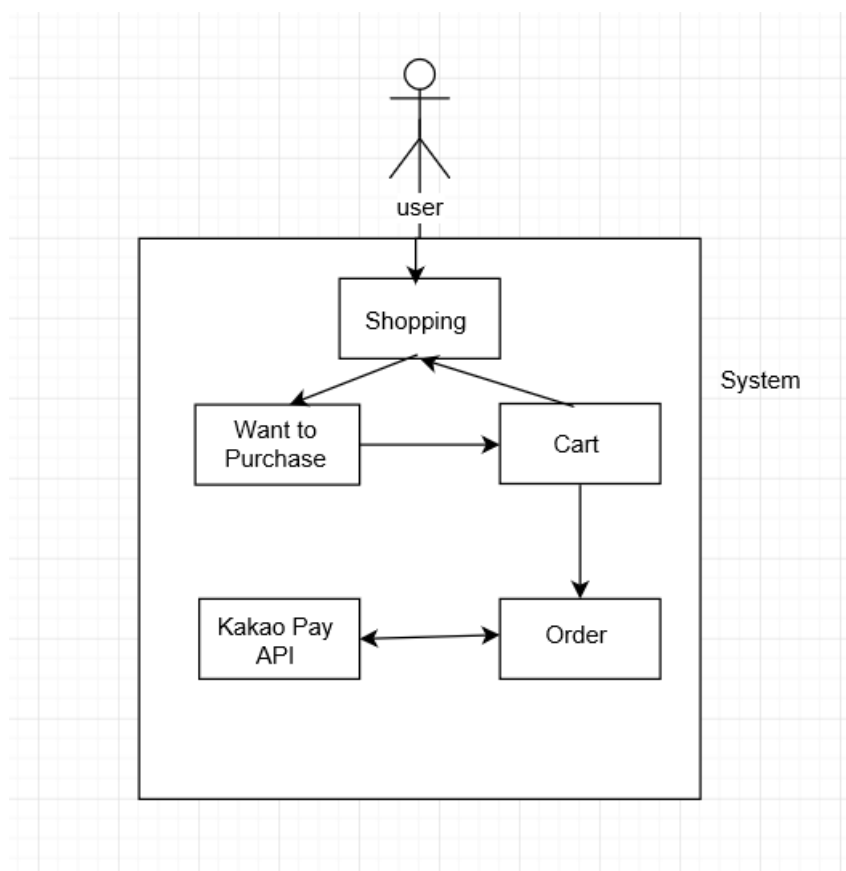
Menu Recommendation System 은 'Eat it'에서 음식 메뉴 추천을 요청할 때, user 의 취향 모델을 기반으로 음식들을 추천을 해주는 시스템이다.

이 시스템은 user 가 어플을 구동할 때, 'tensorflow'를 통해 user 의 취향 모델을 기반으로 식품목록 중 user 의 취향과 매칭률이 높은 상품을 화면에 띄워 user 에게

추천목록을 노출시킨다. 그리고 추천된 식품 항목별로 재료도 같이 보여주어 user 가 같이 볼 수 있게 'Image Handler'를 통하여 화면에 display 한다.

User 가 어플을 이용하면서 user analysis database 는 갱신이 되고 갱신된 user analysis database 를 통하여 새롭게 추천 목록을 만든다.

## 5.5 Ordering System



[Diagram 4] Ordering System

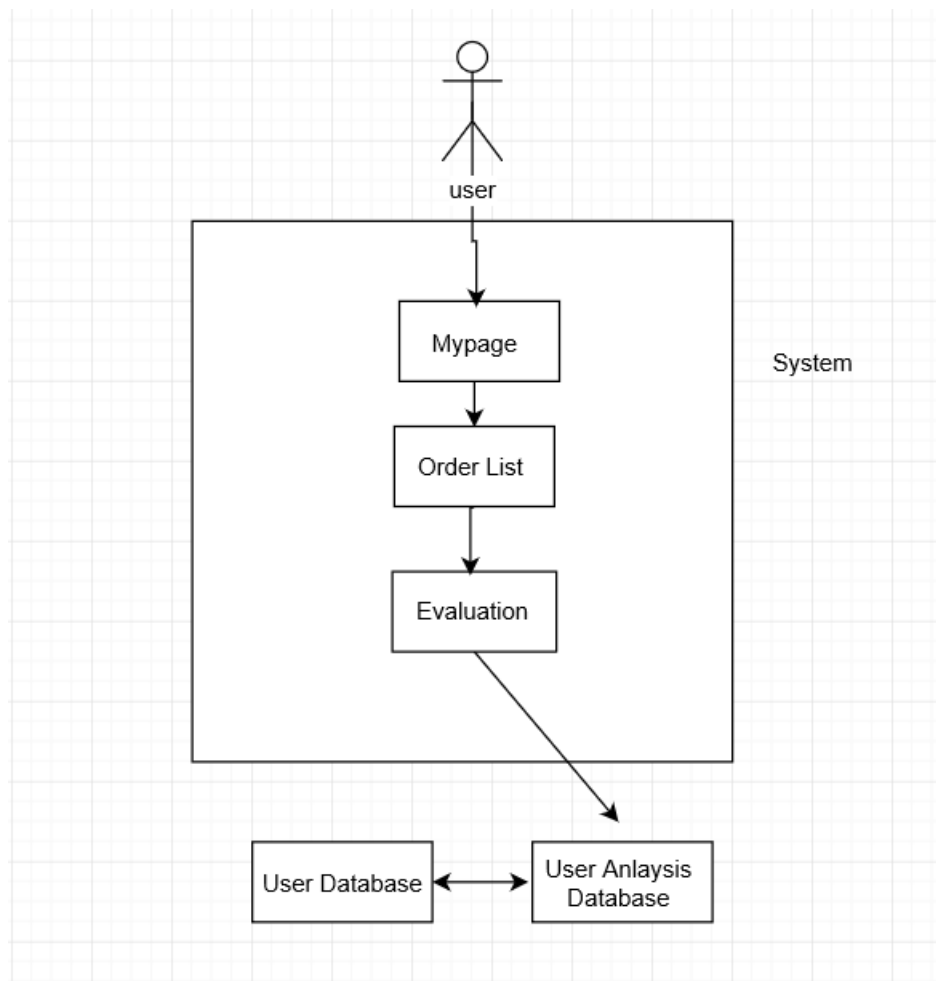
Ordering System은 user가 선택한 식품을 결제하고 구매할 수 있게 하는 시스템이다.

User가 추천 목록에 나오는 식품 또는 검색을 하여 나온 식품을 선택하여 식품 및 그의 재료의 상세 정보를 user에게 보여준다. 그 후에 user가 재료를 구매하고

싶은 경우 대형마트 카트 모양의 아이콘을 클릭하여 'Cart'에 담는다. 이 때 'Cart'에 담고 다른 식품들을 쇼핑할 수 있다.

쇼핑을 모두 마친 user는 'Cart'에 들어가 구매할 식품들을 선택하고 주문하기 버튼을 누른다. 결제할 때는 'Kakao pay API'를 연동한다.

## 5.6 Post Evaluation System



[Diagram 5] Post Evaluation System

Post Evaluation System은 user가 식품을 구매하고 사용한 후에 그 식품에 대한 평가를 요청하는 시스템이다.

First Evaluation System 처럼 좋음, 보통, 나쁨 세 단계로 구매한 식품에 대해 평가를 하도록 user에게 요청을 한다. User는 자신의 만족도의 맞게 평가를

한다.

여기서 다른 일반적인 시스템에서는 이 평가를 다른 user들에게 보여주어 'order' 과정에 도움을 주도록 하지만 'Eat it' 시스템에서는 이 시스템을 통한 평가를 기반으로 user analysis database에 다시 갱신되어 취향 모델을 갱신하는 것이 주 목적이다.

## 6. System Requirements Specifications

### 6.1 Objectives

시스템 Requirements specification 단계는 Functional Requirements와 Non-functional Requirements의 요구사항을 구체적으로 명시하는 것이다. 구조는 한 기능별 이름, 설명, 입력, 출력, 처리, 조건 6 부분으로 나눌 수 있다.

### 6.2 Functional Requirements

#### A. Login Function

이름	Login Function
기능	카카오톡/네이버로 로그인
설명	'Eat it'은 한국을 대상으로 한 서비스이기 때문에 카카오톡이나 네이버로 로그인 기능을 지원한다. 이 기능을 통해 사용자들이 더 빠르고 쉽게 'eat it'의 서비스를 이용할 수 있도록 한다.
입력	카카오톡이나 네이버의 API를 이용하여 사용자의 카카오톡이나 네이버 아이디를 얻는다. 처음에 로그인 시에 닉네임, 나이, 성별 등을 추가로 입력할 수 있다.
출력	로그인이 정상적으로 되면 마이페이지 화면으로 이동한다
처리	사용자의 카카오톡이나 네이버 내부 아이디라는 정보가 DB에 없으면 처음에 로그인 후에, 시스템은 사용자의 정보를 DB에서 저장한다.
조건	사용자가 자기의 정보를 제공하는 걸 동의해야 함.

[Table 3] Login Function

## B. Explore First Evaluation Function

이름	Explore First Evaluation Function
기능	처음에 로그인 후에 사용자의 취향평가
설명	사용자가 처음에 로그인 하면 음식에 대한 취향의 정보가 없으므로 사용자의 좋아하는 요리, 또는 식품을 통하여 기본적인 취향 분석 모델을 생성한다
입력	사용자는 좋아하는 요리와 음식
출력	구매할 식품과 메뉴 추천
처리	사용자가 입력한 정보를 받고 분석하여 선호할 가능성이 높은 요리와 식품이 나온다.
조건	사용자는 좋아하는 음식 입력

[Table 4] Explore First Evaluation function

## C. Preference Analysis Function

이름	Preference Analysis Function
기능	사용자의 취향을 machine learning으로 분석
설명	사용자가 'eat it' 서비스를 쓰면서 구매한 식품, 또는 검색한 메뉴, 좋아하는 음식 등을 정보를 수집하고 사용자의 취향 모델을 학습한다.
입력	사용자의 서비스 이용 과정에서 수집된 메뉴, 식품, 음식 정보
출력	사용자의 입맛, 좋아하는 요리 스타일
처리	사용자가 선호한 요리와 식품을 바탕으로 사용자의 입맛과 식사 습관을 분석한다.
조건	사용자는 앱을 한 번 쓰거나 계속 씀

[Table 5] Preference Analysis function

## D. Main Menu Recommendation Function

이름	Main Menu Recommendation Function
기능	사용자에게 메뉴 추천
설명	사용자의 입맛, 좋아하는 식품, 식사 습관 등 정보를 다 machine learning으로 분석하여 전달받은 후 이 정보들을 바탕으로 사용자에게 메뉴를 추천한다.
입력	Main Tab에 진입
출력	사용자가 좋아할 것이라 예측된 메뉴 노출
처리	사용자의 입맛, 좋아하는 식품, 식사 습관 등 정보를 받고 machine learning으로 사용자가 좋아할 수 있는 메뉴를 분석하고 출력한다.
조건	사용자의 취향 분석 학습모델 존재

[Table 6] Main Menu Recommendation Function

## E. Menu Search Function

이름	Menu Search Function
기능	메뉴를 검색하는 기능
설명	사용자가 메뉴의 이름, 들어가는 식품, 맛 등 요소로 메뉴를 검색할 수 있다.
입력	찾고자 하는 메뉴
출력	메뉴 정보를 담은 검색 결과 화면
처리	메뉴의 요소를 받고 DB에서 관한 메뉴를 찾아서 출력한다.
조건	메뉴에 관한 요소 입력해야 함.

[Table 7] Menu Search Function



## F. Send to Cart Function

이름	Send to Cart Function
기능	상품을 장바구니에서 담는다.
설명	사용자는 관심이 있는 상품을 상품 옆에 있는 '장바구니' 버튼으로 장바구니에서 넣을 수 있다.
입력	상품 옆에 있는 '장바구니' 버튼
출력	이 상품을 장바구니에서 담는다.
처리	관한 상품의 정보를 사용자의 장바구니에 보낸다. DB 업데이트.
조건	같은 상품이 장바구니에 없고 장바구니에 담을 수 있는 상품 제한을 초과하지 않는다.

[Table 8] Send to Cart Function

## G. Display Shopping Cart Function

이름	Display Shopping Cart Function
기능	장바구니 확인
설명	'장바구니' 버튼을 통하여 사용자는 자기의 장바구니에 무엇이 있는지 확인할 수 있다.
입력	'장바구니' 버튼
출력	장바구니 화면
처리	사용자가 '장바구니' 버튼을 누르면 DB에 사용자의 장바구니 정보를 받고 시스템으로 화면을 생성한다
조건	

[Table 9] Display Shopping Cart Function

## H. Edit and Submit Shopping Cart Function

이름	Edit and Submit Shopping Cart Function
기능	장바구니를 수정하고 장바구니에 있는 상품을 주문 요청
설명	장바구니에 상품의 수량 수정, 삭제 등 동작을 편집하고 장바구니에 상품을 주문하는 것이다.
입력	수량 수정, 삭제, 상품 선택
출력	다시 편집된 장바구니
처리	사용자의 동작에 따라 장바구니 DB 업데이트
조건	장바구니 빈 상태 아님.

[Table 10] Edit and Submit Shopping Cart Function

## I. Order Process Function

이름	Order Process Function
기능	상품을 주문하는 기능
설명	장바구니에 있는 상품을 주문하거나 상품을 직접 주문해도 된다. 주문할 요청을 하고 배송지와 받는 사람 등을 설정한 다음에 결제한다.
입력	'주문' 버튼, 배송지와 받는 사람의 정보, 결제수단 선택
출력	결제완료 화면, 또는 주문 내역
처리	'주문' 다음에 배송지와 받는 사람의 정보 입력 화면으로 이동, 다음에 결제 화면으로 이동, 결제 완료된 후에 DB 업데이트
조건	

[Table 11] Order Process Function

## J. Display My Page Function

이름	Display My Page Function
기능	마이페이지 화면을 확인하는 기능
설명	마이페이지에 화면에 사용자 자신의 여러 가지 정보를 볼 수 있고 수정할 수 있는 부분을 수정해도 된다. 예를 들면, 사용자의 닉네임 수정, 주문 내역 조회.
입력	'마이페이지' 버튼
출력	마이페이지 화면으로 이동
처리	사용자가 '마이페이지' 버튼을 누르면 시스템이 DB에 있는 사용자의 정보로 마이페이지 화면 생성
조건	

[Table 12] Display My Page Function

## K. Order Management Function

이름	Order Management Function
기능	주문내역 확인하는 기능, 수취확인 기능, 또는 반품이나 환불 기능
설명	사용자가 주문한 상품의 정보를 확인할 수 있고, 상품을 정상적으로 받은 후에 수취확인, 받은 상품은 정상상태가 아닌 경우, 반품과 환불 기능 지원한다.
입력	'주문내역' 버튼, '수취확인' 버튼, '환불신청'버튼
출력	'주문내역' 버튼을 누르면, 주문내역 화면으로 이동, '수취확인' 버튼을 누르면, 구매한 상품을 수취확인 된다. '환불신청'버튼을 누르면 환불 단계로 이동한다.
처리	사용자 DB에 관련된 정보를 보여주고 업데이트
조건	주문한 상품이 있어야 함.

[Table 13] Order Management Function

## L. Post Evaluation Function

이름	Post Evaluation Function
기능	상품을 평가하는 기능
설명	상품을 구매한 후에 상품에 대한 평가를 할 수 있다.
입력	상품평가의 글, 또는 사진
출력	그 상품의 평가 게시판에서 구매자가 평가한 내용을 업데이트
처리	평가한 내용을 서버로 보낸다.
조건	수취확인 된 상태에 평가 가능

[Table 14] Post Evaluation Function

## 6.3 Non-functional Requirements

Non-functional requirement은 크게 product requirement, organization requirements, external requirement로 나눌 수 있다.

### A. Product Requirements

#### A.1 Usability Requirements

본 시스템은 추천 서비스이기에 앞서 음식을 주문하는 서비스이다. 따라서 수많은 음식 중에서 사용자가 원하는 음식을 찾기 쉬워야 하며, 음식에 대한 정보(필요한 재료, 가격 등)가 직관적이어야 한다.

#### A.2 Performance Requirements

본 시스템의 핵심 서비스는 "음식 추천"이다. 따라서 추천의 정확도가 낮으면 사용자의 이용율이 떨어질 수 있기 때문에 최대한 사용자의 취향에 근접해야 한다. 또한, 수많은 음식의 방대한 데이터를 나타내는 데에 문제가 없어야 하며, 음식을 검색했을 때 performance 측면에서 문제가 없도록 해야 한다.

### A.3 Dependability Requirement

사용자에게 추천해준 음식이 그들의 취향에 맞지 않을 경우 사용자의 신뢰를 잃을 수 있으므로 최대한 많은 데이터를 모아 분석하여 취향에 근접한 결과를 내야만 한다. 또한, 주문/구매 서비스에 대해서도 오류가 발생할 경우 사용자의 신뢰를 잃을 우려가 있기 때문에, 이러한 방면에서도 문제를 만들어서는 안 된다.

### A.4 Security Requirement

본 시스템의 경우 카카오 또는 네이버 계정을 연동하여 로그인하기 때문에 그에 대한 개인정보가 유출되어서는 안 되는 건 물론이고, 결제 수단에 대한 보안이 철저히 관리되어야 한다. 또한, 사용자가 시스템을 안심하고 이용할 수 있도록 데이터베이스 관리에 유의해야 한다.

## B. Organizational Requirements

### B.1 Environmental Requirements

사용자가 언제 어디서든 사용할 수 있도록 모바일 디바이스는 상시 네트워크에 연결되어 있어야 한다. 또한, 데이터베이스를 관리하고 디바이스와 연결할 서버는 상시 구동되어야 한다.

### B.2 Operational Requirements

시스템이 추천한 음식을 사용자가 마음에 들어 하지 않을 경우, 시스템은 계속해서 사용자의 요구에 맞는 음식을 추천할 필요가 있다. 따라서 사용자가 마음에 들어 하지 않는 데이터를 추가적으로 분석하여 요구에 근접한 결과를 도출하거나, 또는 사용자에게 보다 세부적인 질문을 하여 결과에 다가갈 수 있는 기능이 필요하다.

### B.3 Development Requirements

보다 정확한 추천을 하기 위해 machine learning 기술, 더 나아가 deep

learning 기술의 도입이 필요하다. 그리고 본 시스템은 안드로이드 기기를 대상으로 한 어플리케이션이기 때문에, 개발 언어는 Java를 사용해야 한다.

## C. External Requirements

### C.1 Regulatory Requirement

본 시스템은 사용자에게 로그인을 요구하기에 앞서 먼저 그들의 개인정보를 수집함에 대한 동의를 구해야 하며, 취급방침에 대한 알리를 제공해야 한다. 또한, 정보를 외부에 제공하게 될 경우엔 사용자에게 동의를 먼저 구해야 한다.

### C.2 Safety/security Requirement

본 시스템은 사용자의 개인정보가 외부로 유출되지 않도록 해야 하며, 사용자의 결제에 오류가 발생해서는 안 된다.

## 6.4 Scenario

### A. Signup/Login Scenario

A.1 처음에 이 어플을 사용하는 사용자들이 카카오톡이나 네이버 계정으로 sign up하고 로그인 할 수 있다. 그 이유는 이 어플이 한국에서 생겨서 대부분 한국 사람들 다 가지는 계정으로 하면 더 쉽고 편하게 어플을 쓸 수 있다.

A.2 앞으로 다시 앱을 사용할 때에는 이미 sign up된 카카오톡이나 네이버 계정으로 하면 된다.

A.3 처음에 sign up 한 다음에 사용자의 닉네임 다시 설정할 수 있고 카카오톡이나 네이버의 회원정보 그대로 이어 써도 된다.

### B. First Menu Evaluation Scenario

B.1 처음에 로그인 한 다음에, 사용자의 기본적인 음식 취향과 식사습관 등 정보를 파악하기 위해 첫 번째 메뉴 평가를 해야 한다. 사용자가 자기의 일상 메뉴 정보를

입력하고, 또는 자기의 입맛과 식사 습관 등 정보를 입력한다.

B.2 이 정보들을 가지고 알고리즘을 통하여 사용자의 기본적인 음식 선호를 분석하고 결과를 출력한다.

### **C. Main Menu Recommendation Scenario**

C.1 사용자가 'eat it' 쓰면서 모든 입력되고 받을 수 있는 정보를 받는다.

C.2 가지는 정보를 통하여 사용자에게 맞춤형 메뉴를 추천한다. 예를 들면, 사용자가 구매한 식품, 좋아하는 요리, 검색한 요리법 등을 모든 정보를 분석하여 가능성이 높은 사용자는 좋아할 수 있는 메뉴를 사용자에게 추천하고 요리법도 같이 제공한다.

### **D. Menu Search Scenario**

D.1 'eat 'it'에 이미 등록된 메뉴를 검색할 수 있다. 메뉴의 이름, 들어가는 식품, 맛, 메뉴의 종류 (예: 중화요리) 등 메뉴와 관한 아무 요소를 입력하면 관련된 메뉴를 검색 결과로 나온다,

D.2 또는 사용자가 입맛이 변할 수도 있는 것 같은 원인들을 고려해서 검색하기 전에 사용자 자신의 취향 참고 여부 선택도 할 수 있다.

### **E. Add to Card and Order Scenario**

E.1 사용자가 상품을 보면서 장바구니에서 담을 수 있다. 구매하고 싶은 상품을 다 담고 주문하고 싶으면 장바구니로 이동하고 주문할 상품을 선택하고 결제한다.

E.2 결제 단계에 먼저 배송지와 받는 사람의 정보를 입력하고 결제 수단을 선택한다. 결제 수단을 신용카드, payco, 카카오페이 등 여러 가지 결제 수단을 지원한다.

E.3 결제 수단을 선택한 다음에 지불한다.

E.4 또는 장바구니 안 통하여 상품을 직접 주문해도 된다. 만약 상품을 직접

주문하면 바로 결제 단계로 이동한다.

#### **F. After Order Evaluation Scenario**

상품을 구매하고 수취확인 되면 상품에 대한 평가를 할 수 있다. 상품의 질, 배송 속도 등 모든 상품과 관련된 걸 다 평가할 수 있고, 평가한 내용을 상품 아래 상품평 내역에서 등록될 것이다.



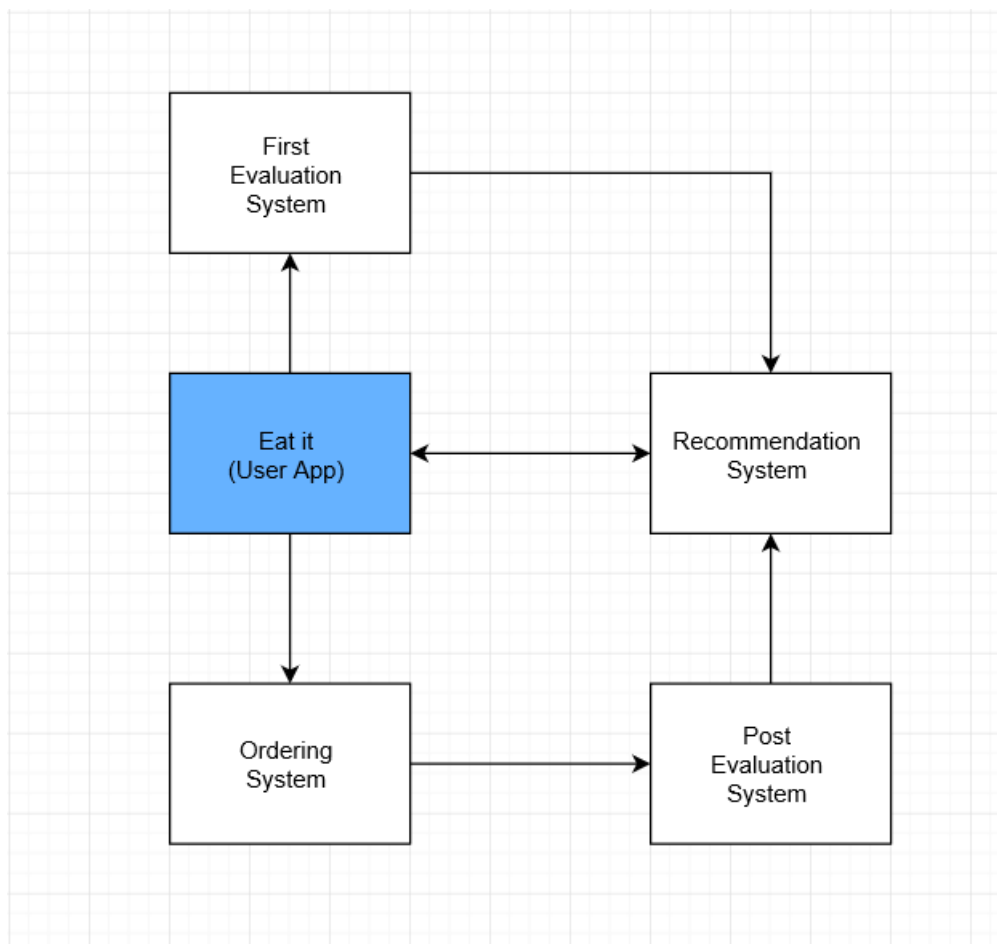
## 7. System Models

### 7.1 Objective

System Models에서는 본 시스템과 그 구성요소, 그리고 동작 환경 사이의 관계들을 시각적인 모델로 나타낸다. 이는 Context Diagram, Process Diagram, Use-Case Diagram등 다양한 Diagram을 포함한다.

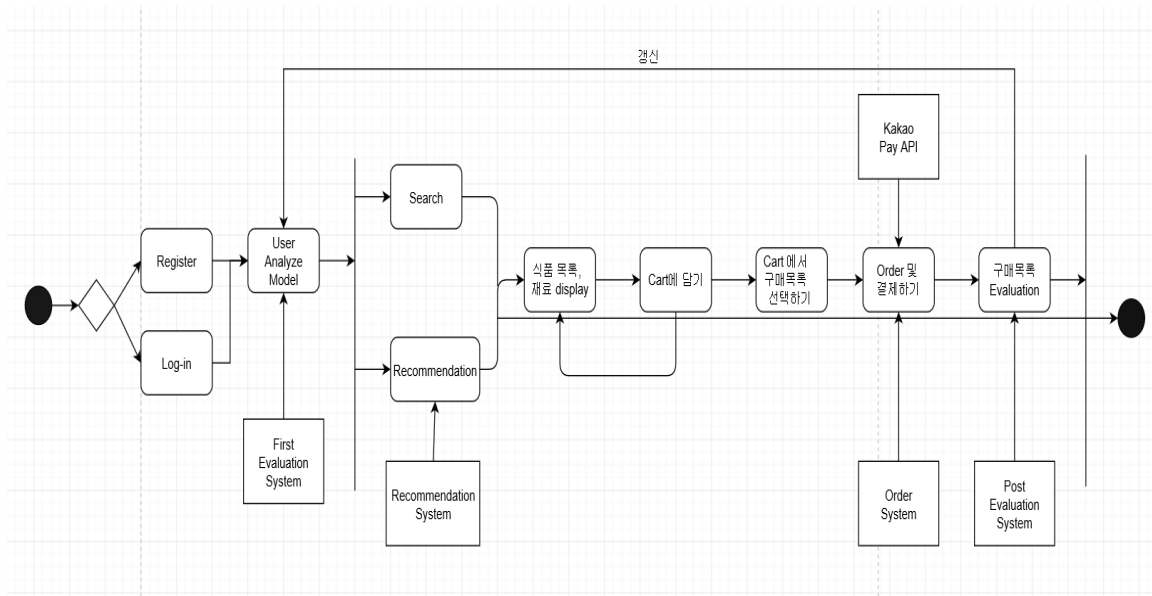
### 7.1 Context Models

#### A. Context Diagram



[Diagram 6] Context Diagram

## B. Process Diagram

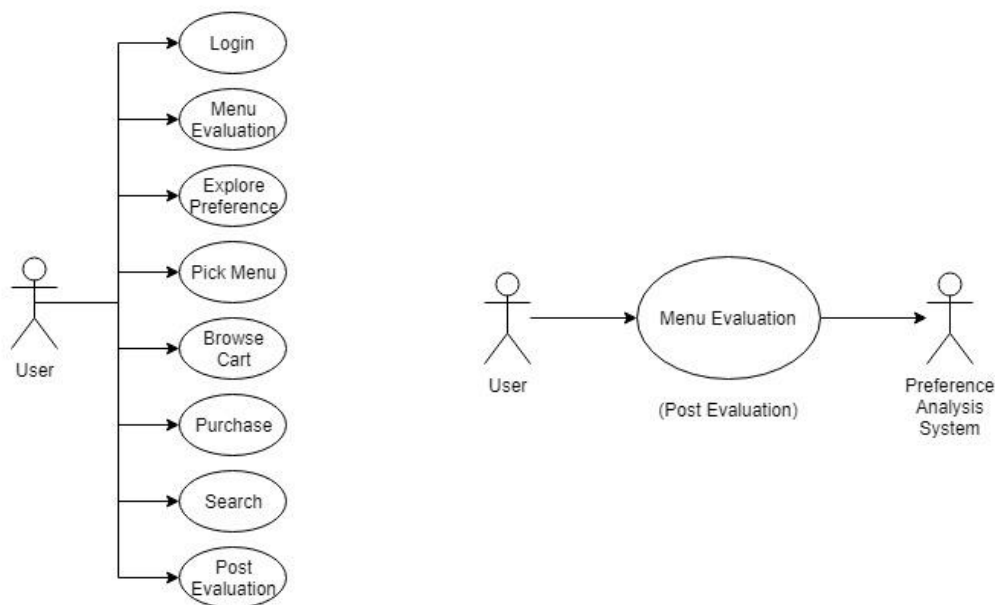


[Diagram 7] Process Diagram

## 7.2 Interaction Models

### A. Use-Case Diagram

USE-CASE Diagram



[Diagram 8] Use-case Diagram

## B. Tabular Description for each Use Case

### B.1. LOGIN

SUB SYSTEM	LOGIN
USE CASE	Login
ACTOR	Registered User, User DB
DESCRIPTION	사용자가 KAKAO 나 NAVER API 를 통해 로그인을 한다. 등록되어 있지 않은 사용자는 회원가입하기를 권유하고, 등록되어 있는 회원은 회원으로 로그인 한다.
STIMULUS	사용자가 아이디와 비밀번호를 입력하고 로그인 버튼을 누른다.
RESPONSE	입력한 정보가 UserDB에 존재하는 정보와 일치하는지 판단하고 해당 회원의 개인 DB를 열람할 권리를 준다.
COMMENTS	

[Table 15] LOGIN System

### B.2. MENU EVALUATION

SUB SYSTEM	Menu Evaluation
USE CASE	Menu Evaluation
ACTOR	Registered User, User DB, User analysis DB
DESCRIPTION	새로운 유저가 본인의 취향에 맞는 메뉴를 고른다.
STIMULUS	여러가지 종류의 메뉴가 주어지고, 주어진 메뉴를 사용자 입맛에 따라 고른다.
RESPONSE	User DB에 입력한 자료가 update된다.
COMMENTS	

[Table 16] MENU EVALUATION System

### B.3. EXPLORE PREFERENCE

SUB SYSTEM	Explore Preference
USE CASE	Explore Preference
ACTOR	User, User DB, User Analysis DB
DESCRIPTION	회원이 처음 Menu Evaluation때 골랐던 메뉴에 따른 결과가 보여진다
STIMULUS	Menu Evaluation때 골랐던 메뉴에 따라 비슷한 종류를 사용자에게 표시한다
RESPONSE	Menu Evaluation에 따라 User 가 고른 Menu 랑 유사한 결과를 보여줌
COMMENTS	

[Table 17] EXPLORE PREFERENCE System

### B.4. PICK MENU

SUB SYSTEM	Pick Menu
USE CASE	Pick Menu
ACTOR	User, User Analysis DB
DESCRIPTION	First Evaluation system때 user가 음식에 대한 취향을 토대로, 시스템은 user한테 여러가지 메뉴를 추천한다. 주어진 메뉴와 유저가 검색한 메뉴에서 user가 고른다.
STIMULUS	User가 주어진 메뉴들에서 고른것들은 사용자의 장바구니로 이동한다.
RESPONSE	User가 주어진 메뉴 중에서 장바구니에 담으면 한 곳에 User가 선택한 모든 메뉴 및 재료들이 담겨져 있다.
COMMENTS	

[Table 18] PICK MENU System

### B.5. BROWSE CART

SUB SYSTEM	Browse Cart
USE CASE	Browse Cart
ACTOR	Registered User
DESCRIPTION	사용자가 장바구니에 담아 놓은 메뉴들을 보여준다
STIMULUS	사용자가 추천된 메뉴와 검색한 메뉴에서 장바구니에 담은 메뉴들을 보여준다
RESPONSE	사용자가 장바구니에 담아놔던 아이템들이 보여진다

[Table 19] BROWSE CART System

### B.6. PURCHASE

SUB SYSTEM	Purchase
USE CASE	Purchase
ACTOR	Registered User, Kakao Pay API
DESCRIPTION	사용자가 장바구니에 담어져 있는 메뉴 및 재료들을 구매 할 수 있도록 해준다
STIMULUS	쇼핑을 마친 user는 Cart 에 들어가 구매할 식품들을 선택하고 주문하기 버튼을 누른다
RESPONSE	User에게 KakaoPay 화면을 보여주고 결제를 진행한다.

[Table 20] PURCHASE System

### B.7. SEARCH

SUB SYSTEM	Search
USE CASE	Search
ACTOR	Menu DB
DESCRIPTION	사용자가 원하는 식품을 Menu DB에서 검색한다
STIMULUS	user가 원하는 식품을 찾기 위해 Menu DB 에서 쿼리를 입력한다
RESPONSE	검색한 쿼리에 맞는 식품을 Menu DB에서 검색한 후 출력한다.

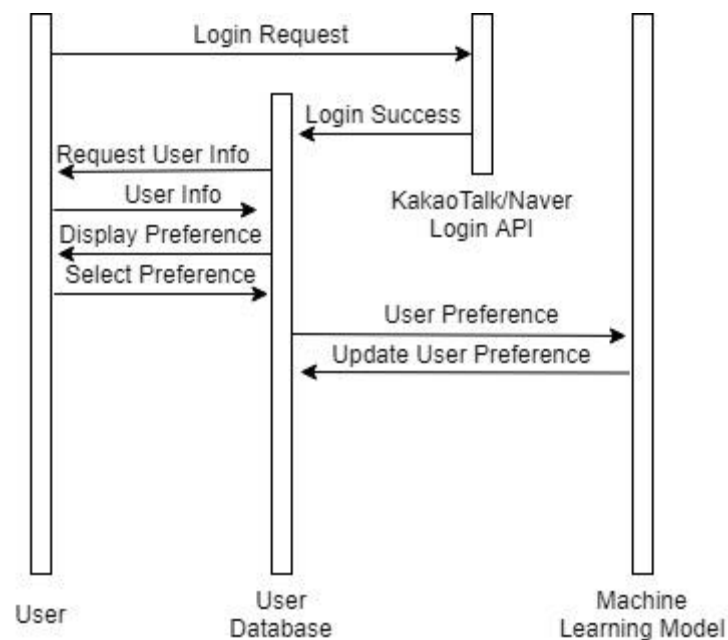
[Table 21] SEARCH System

## B.8. POST EVALUATION

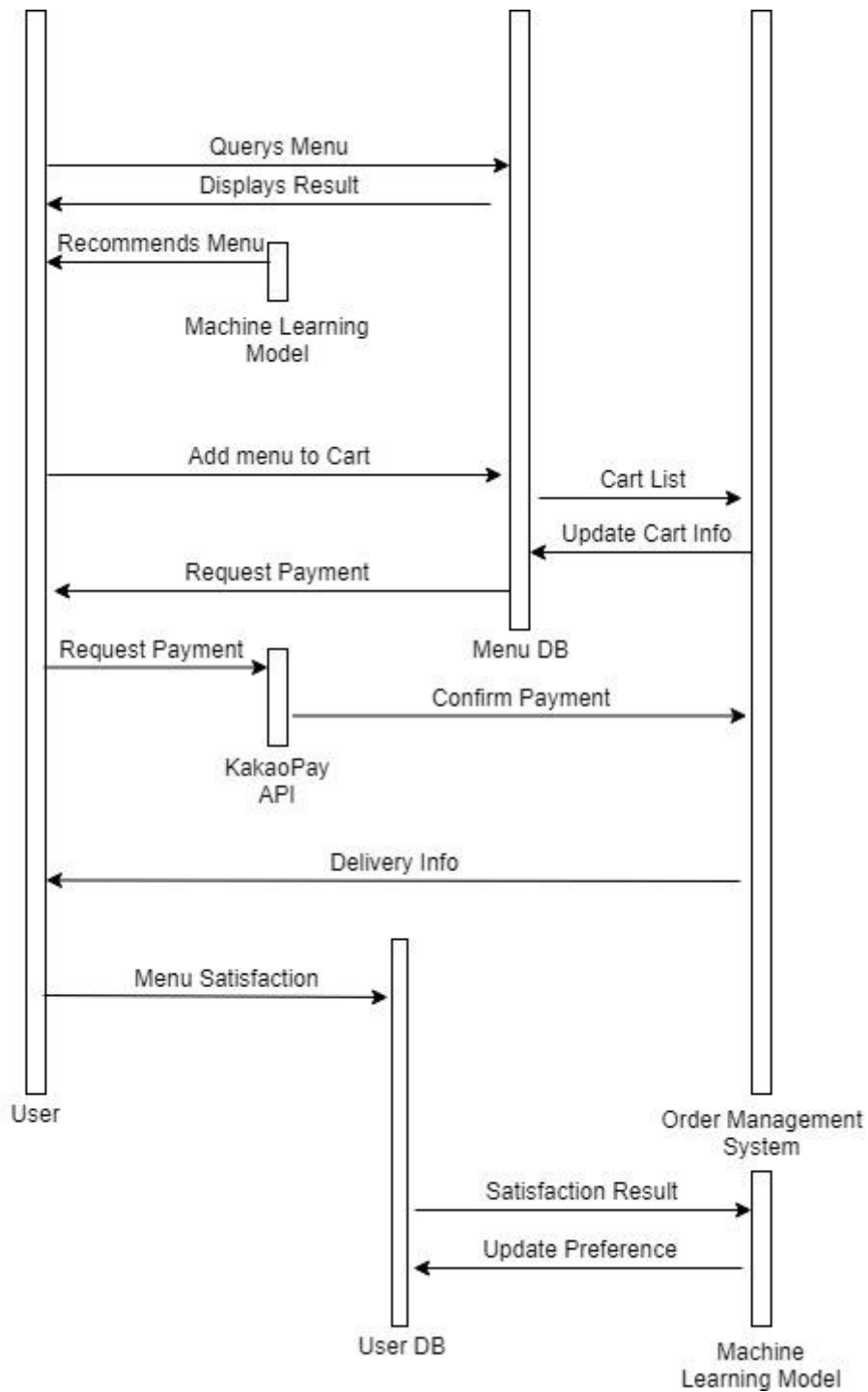
SUB SYSTEM	Post evaluation
USE CASE	Post evaluation
ACTOR	Registered User, User Analysis DB
DESCRIPTION	사용자가 구매한 식품들을 배송받고 그 배송을 토대로 만족도 조사를 한다. First Evaluation system 처럼 좋음 보통 나쁨 세 단계로 구매한 식품에 대해 평가를 하도록 User에게 요청을 한다
STIMULUS	User는 자신의 만족도의 맞게 평가를 한다. 좋음 보통 나쁨 세가지로 구매한 식품에 대해 평가를 하도록 user 에게 요청을 한다
RESPONSE	Post Evaluation system 을 통한 평가를 기반으로 user analysis DB에 다시 갱신되어 취향 모델을 갱신한다.
COMMENTS	

[Table 22] POST EVALUATION System

## C. Sequence Diagram



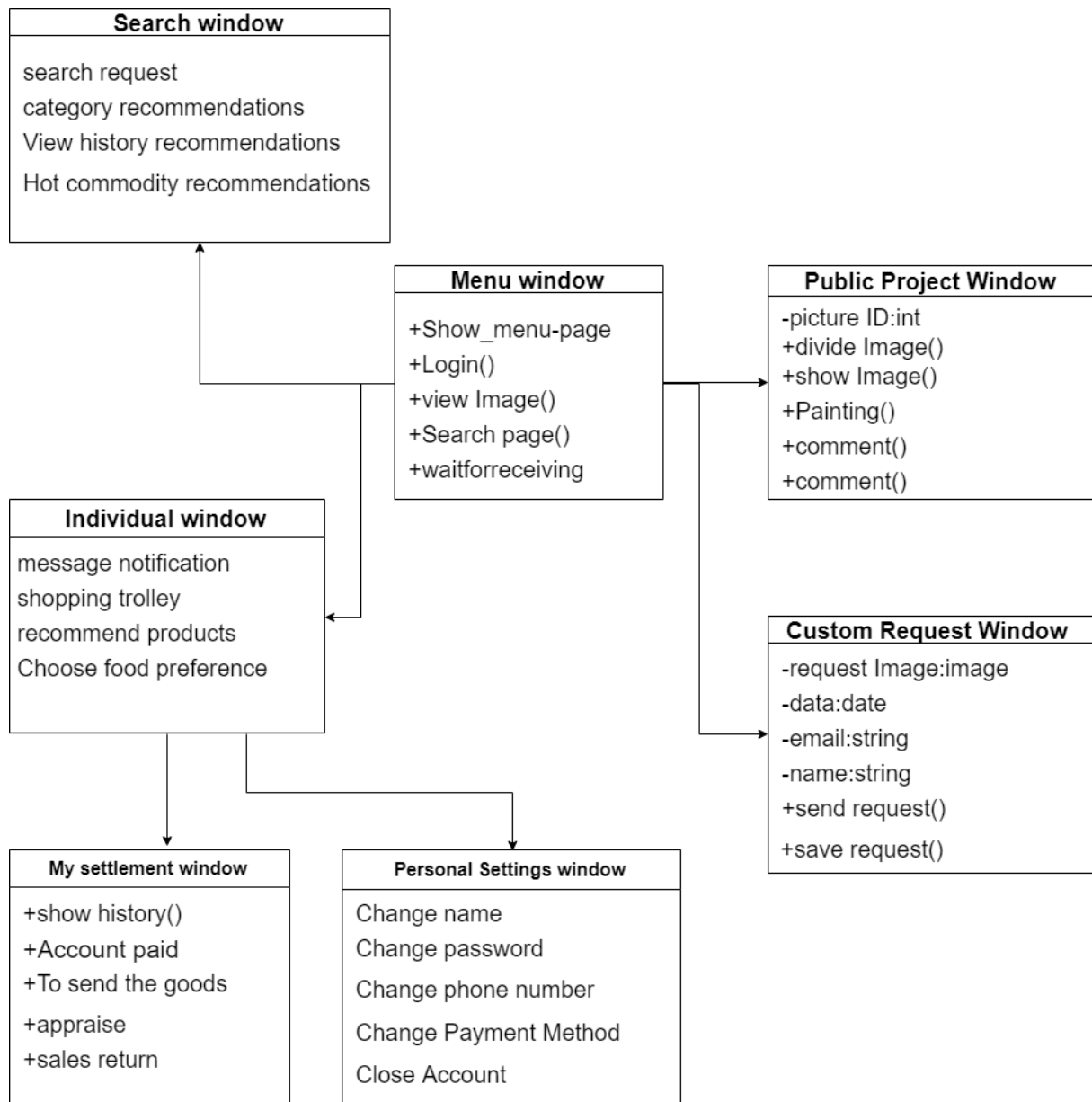
[Diagram 9] Registration & First Evaluation Sequence Diagram



[Diagram 10] Purchase Sequence Diagram

## 7.3 Structural Models

### A. Class Diagram

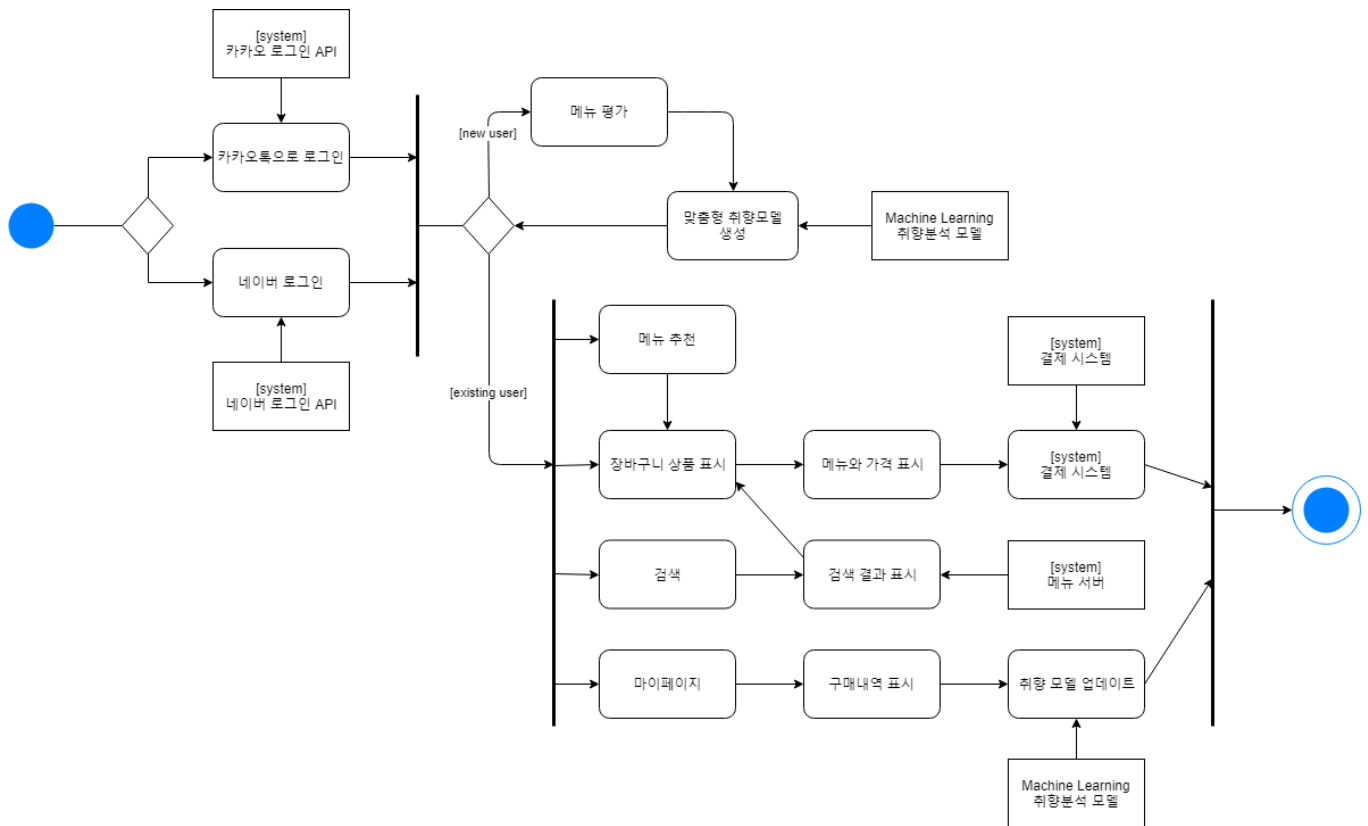


[Diagram 11] Class Diagram



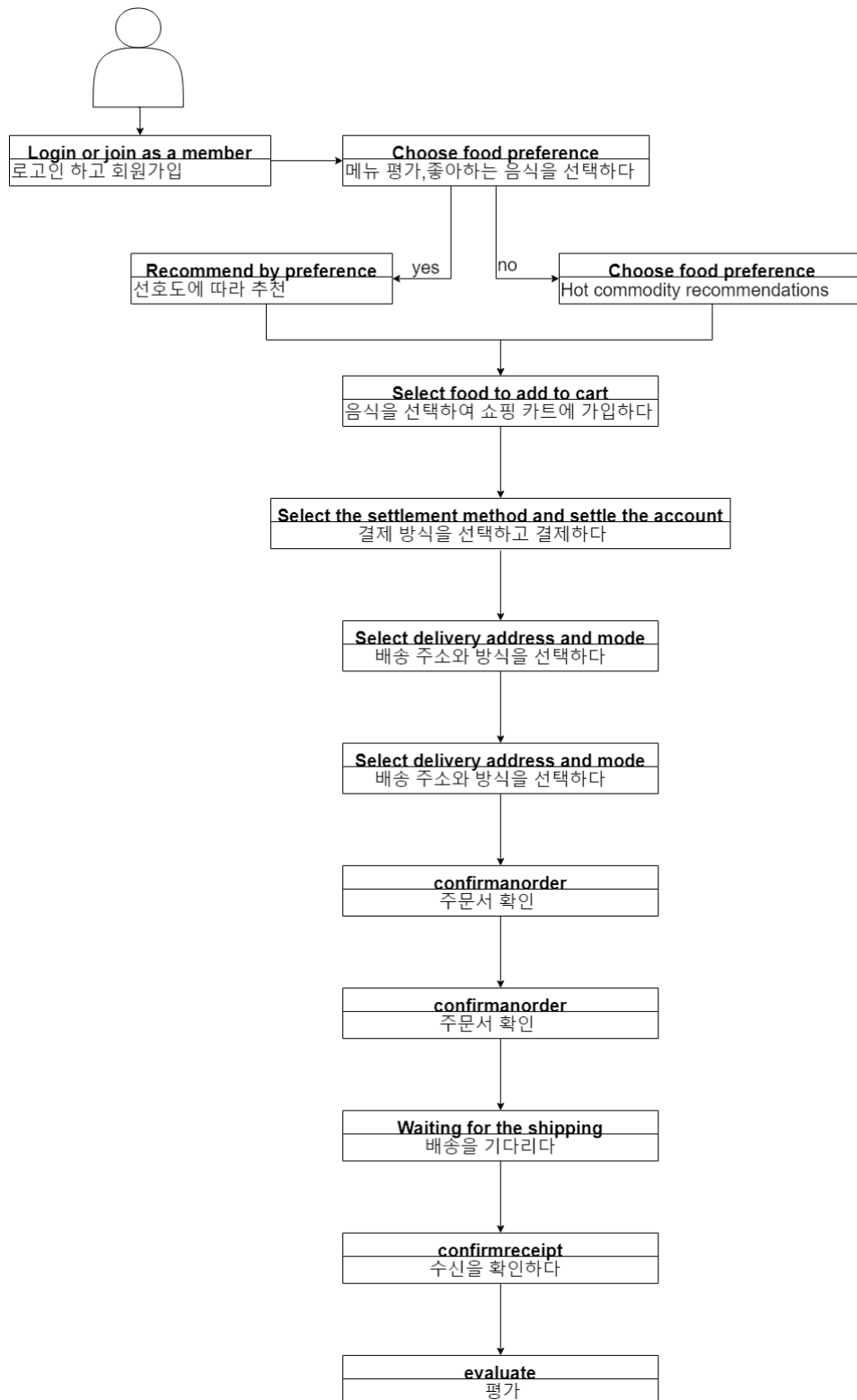
## 7.1 Behavioral Models

### A. Activity Diagram



[Diagram 12] Activity Diagram

## B. State Diagram



[Diagram 13] State Diagram

## 8. System Evolution

### 8.1 Objective

System Evolution에서는 본 시스템이 기반으로 하는 기본적인 가정들을 정의하고 hardware evolution, user needs의 변화로 인한 발생 가능한 변경사항들을 기술한다. 또한 시스템을 이러한 변경사항에 맞춰 발전시킬 방향성에 대해 논한다. 이는 추후 시스템의 변경에 제약이 될 design decision을 피하기 위한 과정이다.

### 8.2 Limitation

Eat it! 서비스는 지마켓과 같은 오픈 마켓 형태가 아니라 마켓컬리와 같이 모든 상품에 대해 마켓에서 등록과 배송을 담당하는 집중형 마켓 형태로 고안되었다. 이는 사용자가 원하는 상품들을 골라 구매하였을 때 모든 상품이 한 번에 배송되어 사용자 경험을 향상시키기 위함이다. 그러나 후자의 방식을 사용하기 때문에 사용자에게 제공할 수 있는 상품들의 수가 제한된다는 한계점이 발생할 수 있다.

또한, 대부분의 사용자들은 식품을 쇼핑할 때 사람들의 평가에 의존하는 경향을 보인다. 그런데 본 서비스는 개인 맞춤형 추천 솔루션으로 타인의 취향보다 개개인의 취향에 맞춘 쇼핑을 제공하기 때문에 타인의 의견이 배제된 느낌을 받을 수 있다. 따라서 타인과의 커뮤니케이션 측면을 보완할 필요성이 있다.

### 8.3 Evolution

#### A. Platform Extension

본 서비스의 가장 중요한 requirement중 하나는 접근성이다. 실생활에 밀접하게 자주 사용되는 서비스이기 때문에 사용자에게 가장 가까운 다양한 환경에서 접근할 수 있어야 한다. 이에 따라 보편적으로 가장 많이 사용되는 모바일 Android 플랫폼에서 개발하는데, 추후 iOS 플랫폼을 지원함으로써 서비스의 접근성과 대상 고객을

확장할 수 있다. 또한 온라인 식품 쇼핑을 할 때 web 형태를 선호하는 고객들 또한 많으므로 web, Android, iOS 플랫폼으로 확장한다면 접근성을 크게 높일 수 있을 것이다.

## **B. Communication System**

앞서 말한 커뮤니케이션의 부족을 보완하기 위해 상품의 후기 기능을 강화할 필요가 있다. 후기의 신뢰도를 높이기 위해 해당 상품의 구입 내역이 있는 고객들만 후기를 작성할 수 있도록 하고, 후기의 수를 늘리기 위해 보상을 제공한다. 기존의 단순한 상품 후기가 아니라 영화를 본 후 후기를 올리는 것처럼 SNS에 포스팅하는 것과 유사한 형태로 리뷰를 등록하도록 하고, 친구를 등록한 후 서로의 활동을 볼 수 있는 기능을 추가한다면 커뮤니케이션 요소를 대폭 강화할 수 있을 것이다.

또한 식품 쇼핑에서 문제가 발생하거나 궁금증이 생긴 경우 빠른 피드백이 매우 중요하다. 이를 해결하기 위해 챗봇 형태로 고객의 문의에 일차적으로 빠르게 응대할 수 있도록 하고 상담원과의 실시간 채팅 기능을 지원하여 빠른 대응으로 사용자 경험을 향상시킬 수 있다.

## **C. Grocery Management**

Eat it!에서는 기본적으로 메뉴를 선택할 경우 해당 메뉴에 필요한 재료들을 한번에 담는 기능을 제공하지만, 양파나 계란 등 냉장실에 대부분 보유하고 있는 재료들을 매번 클릭하여 제거하는 것에 사용자가 귀찮음을 느낄 수 있다. 이를 해결하기 위해 냉장고에 있는 식재료들을 등록하면 그 재료들을 자동으로 배제하는 기능을 추가할 수 있다. 또한 냉장고에 있는 재료들을 활용할 수 있는 메뉴를 추천하는 기능을 도입한다면 사용자의 남은 식재료 관리를 원활하게 할 수 있을 것이다.

## 9. Appendices

### 9.1 Objective

Appendices에서는 개발되는 어플리케이션과 관련된 자세하고 구체적인 정보를 제공한다. 먼저 시스템에서 사용되는 데이터의 논리적 조직을 정의하고 데이터 사이의 관계에 대해 기술한다. 또한 시스템을 사용하기 위해 필요한 hardware의 요구사항에 대해 설명한다.

### 9.1 Database Requirements

Eat it! 시스템은 relational data model을 사용하여 데이터를 Relation 형태로 데이터베이스에 저장한다. 본 시스템은 사용자 정보, 상품 정보, 장바구니, 그리고 구매 내역에 대한 Relation을 보유하여야 하고 사용자의 취향 분석 모델을 위해 별도의 데이터를 행렬 형태로 저장한다. Database Requirements에서는 이 relation 들의 schema를 간략하게 기술한다.

#### A. User Information

가입된 사용자의 정보를 저장한다. 사용자의 고유 번호인 uid를 primary key로 사용하며 사용자의 id, password, 이름, 닉네임, 주소, 전화번호를 모두 not null 형태로 보유해야 한다. 사용자의 취향 학습 모델을 위해서는 데이터를 Relation 외부의 matrix 형태로 저장하는데, 각 사용자의 모델을 식별하기 위해서도 본 table의 uid를 사용한다.

#### B. Sample Information

사용자의 첫 취향 분석 모델 학습을 위해 평가되어야 하는 샘플 메뉴들의 정보를 저장한다. 고유 번호인 sid를 primary key로 사용하며 메뉴의 이름과 Machine Learning 학습을 위해 필요한 training data label들의 값을 not null로 보유한다.

### C. Product Information

판매 가능한 상품들 각각의 정보를 저장한다. 고유 번호인 pid를 primary key로 사용하며 상품의 이름, 가격, Machine Learning 모델의 추천을 위한 label들의 값을 not null로 보유한다.

### D. Shopping Chart Information

사용자가 쇼핑 카트에 담은 상품들의 정보를 저장한다. User Information의 uid를 foreign key이자 primary key로 사용하고, 담은 상품들의 pid를 foreign key들로 저장한다.

### E. Order Information

사용자가 결제한 주문의 정보를 저장한다. 고유 번호인 oid를 primary key로 사용하며 User Information의 uid와 구매한 상품들의 pid를 foreign key들로 저장한다.

## 9.2 Hardware Requirements

Eat it! 시스템은 Android 8.0 Oreo 환경에서 개발된다. 따라서 본 서비스를 이용하기 위해서는 Android 운영체제의 스마트폰, 혹은 가상 머신이 필요하다. 또한 이 기기들은 Android 8.0 버전에서 개발된 어플리케이션을 지원하는 Android 버전이 설치되어 있어야 한다. 본 시스템에서 Machine Learning 모델의 학습은 서버 프로그램을 통해 이루어지므로 클라이언트 디바이스에서 Machine Learning을 위한 사양은 요구되지 않는다. 그러나 회원가입과 로그인을 하고 상품 정보를 받아오며 모델 학습에 필요한 label을 전송해야 하므로 기기는 네트워크와 연결되어 있어야 한다.

# 10. Index

## 10.1 Objective

Index에서는 독자의 편의를 위해 본 문서에서 사용된 그림과 다이어그램, 테이블에 대한 인덱스를 철자 순서대로 기술한다.

## 10.2 Table Index

[Table 1] Version History .....	9
[Table 2] Term Definition.....	18
[Table 3] Login Function .....	30
[Table 4] Explore First Evaluation function .....	31
[Table 5] Preference Analysis function .....	31
[Table 6] Main Menu Recommendation Function .....	32
[Table 7] Menu Search Function .....	32
[Table 8] Send to Cart Function .....	33
[Table 9] Display Shopping Cart Function.....	33
[Table 10] Edit and Submit Shopping Cart Function.....	34
[Table 11] Order Process Function.....	34
[Table 12] Display My Page Function .....	35
[Table 13] Order Management Function.....	35
[Table 14] Post Evaluation Function .....	36
[Table 15] LOGIN System .....	43
[Table 16] MENU EVALUATION System.....	43
[Table 17] EXPLORE PREFERENCE System .....	44
[Table 18] PICK MENU System.....	44
[Table 19] BROWSE CART System .....	45

[Table 20] PURCHASE System .....	45
[Table 21] SEARCH System .....	45
[Table 22] POST EVALUATION System.....	46

### 10.3 Figure Index

[Figure 1] 연간 국내 식료품 거래액.....	10
[Figure 2] 설문조사 결과 1 .....	11
[Figure 3] 설문조사 결과 2.....	11
[Figure 4] 가입과 취향 분석을 위한 메뉴 평가 mockup.....	13
[Figure 5] 쇼핑 메뉴 mockup.....	14

### 10.4 Diagram Index

[Diagram 1] Overall Architecture Flowchart.....	24
[Diagram 2] First Evaluation System.....	25
[Diagram 3] Menu Recommendation System.....	26
[Diagram 4] Ordering System.....	27
[Diagram 5] Post Evaluation System .....	28
[Diagram 6] Context Diagram .....	41
[Diagram 7] Process Diagram.....	42
[Diagram 8] Use-case Diagram.....	42
[Diagram 9] Registration & First Evaluation Sequence Diagram .....	46
[Diagram 10] Purchase Sequence Diagram.....	47
[Diagram 11] Class Diagram .....	48
[Diagram 12] Activity Diagram.....	49
[Diagram 13] State Diagram .....	50