

# Software Requirement Specification



성균관대 1등 맛집 추천 앱

# 먹구스꾸

2024.05.26

Introduction to Software Engineering [SWE3002\_42]  
**Team 1**

Leader	2020312049 송새론
Member	2019312751 김광원
Member	2022310931 김민서
Member	2017313665 김태훈
Member	2020314841 박정호
Member	2017315471 신경덕
Member	2018310541 엄승주

# Table of Contents

## 1. Introduction

1.1 Purpose .....	8
1.2 Product Scope .....	8
1.3 Definitions, Acronyms, and Abbreviations .....	9
1.4 References .....	10
1.5 Overview .....	11

## 2. Overall Description

2.1 Product Perspective .....	12
2.1.1 Market Status .....	12
2.2 Product Functions .....	13
2.2.1 Create a User Account and Login .....	13
2.2.1.1 Sign-up	
2.2.1.2 Login	
2.2.1.3 Reset Password	
2.2.1.4 Profile Setting	
2.2.2 GPT-Based Recommendation .....	13
2.2.3 Use Diverse Filters for Better Search .....	14
2.2.3.1 Category	
2.2.3.2 Rating	
2.2.3.3 Comment	
2.2.3.4 Price	
2.2.3.5 Liked	
2.2.3.6 Sorting	
2.2.4 Search By Keyword .....	14
2.2.5 View by Map and List .....	15
2.2.5.1 Map	
2.2.5.2 List	
2.2.6 Overview of Restaurant Information .....	15
2.2.7 Write Reviews .....	15
2.2.8 Kingo-Pass .....	16
2.2.9 My Profile .....	16
2.2.9.1 Liked Restaurants	
2.2.9.2 Reviews Written by the User	
2.2.9.3 Change User Information	
2.3 User Classes and Characteristics .....	16
2.3.1 User .....	16

2.3.2 System Manager .....	17
2.4 Operating Environment .....	17
2.4.1 Hardware .....	17
2.4.2 Software .....	17
2.5 Implementation Constraints .....	17
2.6 User Documentation .....	19
2.7 Assumptions and Dependencies .....	19

### 3. Specific Requirements

3.1 External Interface Requirements .....	20
3.1.1 User Interfaces .....	20
3.1.2 Hardware Interfaces .....	33
3.1.3 Software Interfaces .....	33
3.1.4 Communication Interfaces .....	33
3.1.4.1 Email	
3.1.4.2 Application	
3.1.4.3 Network Server Protocol	
3.1.4.4 Message Format	
3.1.4.5 Communications Security and Encryption	
3.1.4.6 Data Transfer Speed and Synchronization Mechanism	
3.1.4.7 Authentication and Token Management	
3.2 Functional Requirements .....	35
3.2.1 Use Case .....	35
3.2.2 Use Case Diagram .....	43
3.2.3 Data Dictionary .....	44
3.3 Non-functional Requirements .....	48
3.3.1 Product Requirements .....	48
3.3.3.1 Email	
3.3.3.2 Application	
3.3.3.3 Network Server Protocol	
3.3.3.4 Message Format	
3.3.2 Organization Requirements .....	49
3.3.2.1 Environmental Requirements	
3.3.3 External Requirements .....	49
3.3.3.1 Regulatory Requirements	
3.3.3.2 Ethical Requirements	
3.3.4 Organizing the Specific Requirements .....	49
3.4 Logical Database Requirements .....	50
3.5 Design Constraints .....	50
3.6 Standards Compliance .....	50

3.7 Standards Architecture .....	51
3.8 System Evolution .....	52
3.8.1 Limitation and Assumption .....	52
3.8.2 Anticipated Changes .....	53

## List of Figures

[Figure 1] User Interface - Loading Page .....	21
[Figure 2] User Interface - Main Page .....	21
[Figure 3] User Interface - Login Page .....	22
[Figure 4] User Interface - Sign-up Page .....	23
[Figure 5] User Interface - Authentication Page .....	23
[Figure 6] User Interface - Profile Setting Page .....	23
[Figure 7] User Interface - Password Change - Check Email Page .....	24
[Figure 8] User Interface - Authentication Page .....	25
[Figure 9] User Interface - Password Page .....	25
[Figure 10] User Interface - Home Page .....	26
[Figure 11] User Interface - Map Page .....	27
[Figure 12] User Interface - List Page .....	28
[Figure 13] User Interface - Search Page .....	29
[Figure 14] User Interface - Restaurant Page .....	30
[Figure 15] User Interface - My Profile Page .....	31
[Figure 16] User Interface - Edit My Profile Page .....	32
[Figure 17] Use Case Diagram .....	43
[Figure 18] ER Diagram .....	47
[Figure 19] Context Model .....	49
[Figure 20] Standard Architecture .....	51

## List of Tables

[Table 1] Definition .....	10
[Table 2] User Interface - Loading and Main .....	20
[Table 3] User Interface - Login .....	21
[Table 4] User Interface - Sign-up .....	22
[Table 5] User Interface - Password Change .....	23
[Table 6] User Interface - Home .....	25
[Table 7] User Interface - Map .....	26
[Table 8] User Interface - List .....	27
[Table 9] User Interface - Search .....	29
[Table 10] User Interface - Restaurant Information .....	30
[Table 11] User Interface - My Profile .....	31
[Table 12] User Interface - Edit My Profile .....	32
[Table 13] Use Case of Sign-up and Profile Setting .....	35
[Table 14] Use Case of Login .....	36
[Table 15] Use Case of Finding Password .....	37
[Table 16] Use Case of Profile Edit .....	37
[Table 17] Use Case of Writing Reviews .....	38
[Table 18] Use Case of Editing Reviews .....	38
[Table 19] Use Case of Deleting Reviews .....	39
[Table 20] Use Case of Opening List of Reviews for Each Restaurant .....	39
[Table 21] Use Case of Opening Reviews that the User has Written .....	40
[Table 22] Use Case of Opening a Specific Review .....	40
[Table 23] Use Case of Liking/Unliking Restaurants and Reviews .....	41
[Table 24] Use Case of Searching Map .....	41
[Table 25] Use Case of Opening Recent History of the User .....	42
[Table 26] User .....	44
[Table 27] Restaurant .....	44
[Table 28] Menus .....	45

---

[Table 29] Categories .....	45
[Table 30] Restaurant Categories .....	45
[Table 31] Restaurant Likes .....	46
[Table 32] Restaurant Reviews .....	46
[Table 33] Review Likes .....	46
[Table 34] Review Images .....	47
[Table 35] Operating Information .....	47

# 1. Introduction

## 1.1 Purpose

본 문서는 성균관대학교 자연과학캠퍼스 학생들을 위한 맛집 추천 애플리케이션의 소프트웨어 요구 명세를 설명하기 위해 성균관대학교 2024 학년도 1학기 소프트웨어공학개론 수업 1조의 구성원에 의해 작성되었다. 문서의 주된 독자는 이 애플리케이션을 개발한 1조의 구성원을 포함, 주된 사용자층인 본교의 학생, 교수 및 교내 종사자 및 관계자 등이다.

1조는 본교 학생들이 학교 근처 맛집을 찾을 때 기존 지도와 맛집 추천 애플리케이션의 리뷰보다는 익명 성균관대 커뮤니티에 작성된 맛집 리스트를 참고하는 현상을 인지하였고, 기존 애플리케이션 추천 시스템의 신뢰성 문제를 파악하였다. 이에 따라 본교 학생들만 이용할 수 있는 맛집 추천 애플리케이션 '먹구스꾸'를 고안하였고, 이 문서에서는 이를 개발하기 위한 요구사항을 분석한다.

본 문서는 애플리케이션의 유저, 하드웨어, 소프트웨어 인터페이스 및 기능적인 요구사항 전반에 대해 서술한다. 또한, 애플리케이션의 기능을 각각 로그인 및 회원가입, 키워드 검색, 리뷰 작성, 다양한 필터 적용, GPT 기반 추천 시스템으로 나누어 각 동작에 대해 보다 구체적으로 서술한다.

## 1.2 Product Scope

1조는 본교 학생들이 학교 근처 맛집을 찾는 데 있어 기존 애플리케이션을 사용하기보다 학우들의 의견을 더욱 신뢰하는 것을 파악하였고, 이에 따라 학우들이 서로의 의견을 종합적, 체계적으로 나눌 수 있는 애플리케이션을 제작하고자 한다. 애플리케이션의 핵심 기능은 다음과 같다.

첫째, 회원가입 시 성균관대학교 학생 인증을 통해 리뷰의 신뢰성 증대와 애플리케이션 유저들 간의 유대감 증진을 통한 커뮤니티 활성화를 목표로 한다.

둘째, 키워드 검색 기능을 통해 먹고 싶은 음식 이름, 음식 설명, 음식점 설명, 음식점 카테고리를 입력하면 쿼리 기반 추천 시스템이 도입된 음식점 리스트를 추천 받고, 보다 빠르고 정확하게 원하는 음식점을 찾아볼 수 있는 것을 목표로 한다.

셋째, 유저가 음식점에 별점, 코멘트 리뷰, 사진을 남기고, 이를 기반으로 다른 유저가 참고하여 음식점을 선택하는 것뿐만 아니라, 평점순, 코멘트 내용, 리뷰 갯수를 기반으로 필터를 적용하여 보다 효과적으로 추천받을 수 있도록 하는 것을 목표로 한다.

넷째, 앞서 언급한 것과 같이 성균관대 학생들의 리뷰 뿐 아니라 음식 카테고리, 킹고패스 할인 여부, 찜 여부, 네이버 평점과 리뷰 갯수 기준, 최대와 최소 가격 설정 등 다양한 기준의 필터들을 적용하여 맞춤형으로 음식점을 검색하고 추천받을 수 있도록 한다.

마지막으로, 유저의 검색 기록, 찜한 음식점 기록, 리뷰 기록을 토대로 gpt기반의 추천 시스템을 활용해 '먹꾸스꾸의 오늘의 팍' 음식점 다섯개를 훔 하면 상단에 띄워줌으로써 취향에 맞는 새로운 음식점을 소개받을 수 있도록 하는 것을 목표로 한다.

### 1.3 Definitions, Acronyms and Abbreviations

아래 표는 본 문서에서 사용된 약어와 용어에 대한 설명이다.

[Table 1] Definition

약어	설명
UI	User Interface
UX	User Experience
API Gateway	Service for creating, publishing, maintaining, monitoring, and securing APIs

AWS	Cloud computing platform that provides a wide range of services including computing power, storage, and databases
AWS EC2	AWS Elastic Compute Cloud
AWS Lambda	AWS Serverless Computing Service
AWS S3	AWS Simple Storage Service
AWS SES	AWS Simple Email Service
Batch	Automated, scheduled execution of tasks or processing of data
Chat GPT	AI model for natural language understanding and generation
CI/CD	Continuous Integration & Continuous Deployment
CRUD	‘Create Read Update Delete’ Interaction
DB	DataBase
Elasticsearch	Distributed search and analytics engine
Kibana	Data visualization and exploration tool for log and time-series data
Kotlin	Modern, statically-typed programming language
MySQL	Kind of Relational DataBase Management System
Naver Geo API	API for accessing geolocation data provided by Naver
Nginx	Web server and reverse proxy server
Nori	Elasticsearch plugin for Korean language analysis using a morphological analyzer
Redis	In-Memory DataBase
Spring & Spring Boot	Framework for building Java-based applications

## 1.4 References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library  
<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

- Team 1. “Software Requirement Specification”. SKKU,  
[https://github.com/skkuse/2021spring\\_41class\\_team7](https://github.com/skkuse/2021spring_41class_team7)

## 1.5 Overview

본 소프트웨어 요구 명세서는 전반적인 문서와 애플리케이션에 대한 개요를 설명하는 첫 번째 장을 포함하여 총 3 장으로 구성되어 있다. 두 번째 장은 애플리케이션 전반적인 동작에 관한 제품의 관점을 포함하여 기능에 대한 상세한 소개, 유저 클래스와 특성, 제약사항, 유저 요구사항 문서, 가정에 대해 기술한다. 세 번째 장은 구체적인 요구사항을 기술한다. 먼저 외부 인터페이스 요구사항으로서 유저, 하드웨어, 소프트웨어, 커뮤니케이션 인터페이스에 대해 분석한다. 다음으로, 기술적과 비기술적 요구사항을 설명한 후, 데이터베이스 요구사항, 디자인 요소에서의 제약사항, 그리고 시스템 특징 등을 서술한다.

## 2. Overall Description

### 2.1 Product Perspective

'먹꾸스꾸'는 맛집을 찾는 성균관대학교 학생들을 위한 애플리케이션이다. 이 애플리케이션은 성균관대학교 학생들이 맛집을 찾는 과정에서 겪는 신뢰도와 정확도 문제를 개선하고, 회원가입 시 성대생 인증 절차 도입, 키워드 검색 시 쿼리 기반 추천 시스템 사용, 리뷰 작성, 다양한 필터 적용, 그리고 GPT 기반 추천 시스템을 홈 화면 상단에 띄우며 보다 효과적이고 편리하게 맛집을 추천받을 수 있도록 하고자한다. 사용자는 해당 애플리케이션을 통해 맛집을 검색, 찜하고, 리뷰를 작성할 수 있으며, 다른 이용자의 리뷰를 볼 수 있다. 또한, 카테고리, 킹고패스 여부, 찜 여부, 네이버 평점 및 리뷰 갯수, 최대와 최소 가격 설정 등 다양한 필터 적용을 통해 맞춤화된 검색 결과를 받아볼 수 있다.

#### 2.1.1 Market Status

현재 성균관대학교 학생들이 맛집을 찾기 위해 가장 많이 사용하는 어플리케이션은 '네이버 지도'와 '구글 지도' 어플리케이션으로, 성대생만을 위한 전용 어플리케이션이 존재하지는 않는다. 하지만 다양한 이용층이 존재하는 어플리케이션이라 허위성 광고나 별점 터러 등과 같이 신뢰를 저하시키는 요소가 있고, 학우들 사이에서 소문난 작은 맛집은 오히려 반영되지 않는 명확한 한계가 존재한다. 따라서 현재 가장 신뢰를 얻는 정보는 성균관대학교 학생 전용 커뮤니티 '에브리타임'에 올라온 맛집 리스트로, 이를 통해 성균관대 학생들은 지도 어플리케이션의 평점보다 같은 학우가 직접 작성한 글을 선호하는 경향이 있음을 파악했다. 하지만 이 역시 저장된 글을 매번 찾아서 봐야하며, 음식점 이름을 지도 어플리케이션에 다시 검색해야하는 번거로움이 존재한다. 이러한 이유로 지도 어플리케이션과 성균관대학교 커뮤니티 어플리케이션을 접목한 성균관대 학생 전용 맛집 추천 어플리케이션 '먹꾸스꾸'를 개발하고자 한다.

## 2.2 Product Function

### 2.2.1 Create a User Account and Login

‘먹구스꾸’ 서비스를 이용하기 위한 첫 번째 단계이다. 첫 서비스 이용 시 회원가입 창을 통해 회원가입 절차를 진행해야 하며, 회원인 경우 로그인을 하여 서비스에 접속할 수 있다. 로그인 시 내부적으로 ‘Access Token’을 발급받아 장시간 로그인으로 인한 취약점을 방지한다.

#### 2.2.1.1 Sign-up

회원이 아닌 경우 회원가입 창을 통해 회원가입을 진행한다. 회원가입 시 성균관대학교 이메일로만 가입이 가능하며, 이름, 이메일 주소, 비밀번호를 기입하면 해당 이메일로 인증번호가 전송된다. 다시 앱으로 돌아와 인증번호를 기입하면 프로필을 설정할 수 있게 된다.

#### 2.2.1.2 Login

이미 회원일 경우, 이메일 주소와 비밀번호를 기입하면 접속된다.

#### 2.2.1.3 Reset Password

비밀번호 초기화를 위해 가입했던 이메일 주소를 기입하면 해당 이메일로 인증번호가 전송된다. 다시 앱으로 돌아와 인증번호를 기입하면 새로운 비밀번호를 설정할 수 있게 된다.

#### 2.2.1.4 Profile Setting

회원가입의 최종 단계로, 닉네임(이름과 별개)과 프로필 이미지를 설정할 수 있다. 이미지는 사용자의 갤러리에 접근하여 가져오며 지정하지 않을 경우 기본 이미지로 대체된다. 회원가입을 성공하면 별도로 로그인을 진행할 필요 없이 접속된다.

### 2.2.2 GPT-Based Recommendation

유저의 검색 기록, 짬한 음식점 기록, 리뷰 기록을 토대로 gpt 기반의 추천 시스템을 활용한다. 이를 통해 ‘먹구스꾸의 오늘의 핵’ 음식점 5개를 홈 화면 상단에 보여준다. 사용자의 취향에 맞는

새로운 음식점을 소개받을 수 있다.

### 2.2.3 Use Diverse Filters for Better Search

#### 2.2.3.1 Category

사용자가 원하는 음식점 카테고리 1개 이상을 선택해 검색할 수 있다.

#### 2.2.3.2 Rating

음식점의 평점이 일정 평점 이상인 음식점만 조회할 수 있다.

#### 2.2.3.3 Comment

음식점의 댓글 수가 일정 수 이상인 음식점만 조회할 수 있다.

#### 2.2.3.4 Price

음식점의 메뉴 가격을 기준으로 사용자가 지정한 가격 범위(~ 이상, ~ 이하)를 기준으로 조회할 수 있다. 이때 기준이 되는 메뉴는 모든 메뉴를 기준으로 한다.

#### 2.2.3.5 Liked

사용자가 찜한 음식점만 지도에서 조회할 수 있다.

#### 2.2.3.6 Sorting

기본 순으로 정렬된 음식점을 가까운 순, 평점 높은 순, 댓글 많은 순, 찜 많은 순에 따라 정렬할 수 있다.

### 2.2.4 Search By Keyword

키워드 검색 기능을 통해 먹고 싶은 음식 이름, 음식 설명, 음식점 설명, 음식점 카테고리 등을 입력하면 쿼리 기반 추천 시스템이 도입된 음식점 리스트가 출력된다. 또한 앞서 말한 필터

기능도 검색 창에 포함되어 있어, 사용자 맞춤형으로 검색할 수 있다. 필터 기능 아래에 사용자의 최근 검색어도 함께 보여 준다.

### 2.2.5 View by Map and List

#### 2.2.5.1. Map

사용자는 지도에서 현재 자신의 위치를 핀으로 제공받는다. 이를 위해 위치 정보 제공 허용 등의 알림 창이 표시되고 이를 동의할 경우에만 정확한 위치가 표시된다. 간략한 가게 정보(가게 이름 및 평점)만을 지도에 표시하여 사용자가 필터를 적용하여 볼 수 있도록 한다.

#### 2.2.5.2. List

기본적으로 아무 필터도 적용되지 않은 가게 정보를 아래로 스크롤 할 수 있는 리스트 형태로 보여 준다. 지도보다 좀 더 상세한 가게 정보(이미지, 이름, 평점, 리뷰 수, 찜 수, 메뉴, 리뷰 등)를 나타내며 사용자가 적용할 수 있는 필터도 좀더 다양하다.

### 2.2.6 Overview of Restaurant Information

가게의 정보를 상세하게 볼 수 있는 기능이다. 이름, 카테고리, 찜, 평점, 리뷰 작성, 위치, 운영시간, 전화번호 등의 정보가 화면에 구분되어 표시된다. 또한 가게 이미지를 조회할 수 있으며, 메뉴와 리뷰를 리스트 형식으로 나열되어 있어 아래로 스크롤하면 볼 수 있다.

### 2.2.7 Write Reviews

가게 상세 페이지에서 연필 아이콘을 클릭하면 사용자는 해당 가게에 대해 리뷰를 작성할 수 있다. 평점을 기입하지 않거나 내용을 작성하지 않으면 비워져있다는 알림이 뜬다. 평점을 기입하고, 해당 가게와 관련된 사진을 첨부하고, 내용을 작성하면 리뷰 등록이 된다.

## 2.2.8 Kingo-Pass

성균관대학교와 제휴 맺은 업체를 목록화하여 좌우로 스크롤할 수 있도록 보여 준다. 사용자는 해당 업체들의 할인 혜택 정보를 제공받을 수 있다. 가게 이미지나 홍보글을 클릭하면 해당 가게 상세 화면으로 이동하게 된다.

## 2.2.9 My Profile

사용자는 자신의 프로필에서 자신의 활동 내역인 자신이 찜한 가게와 사용자가 작성한 리뷰를 볼 수 있고 사용자의 정보를 수정할 수 있다.

### 2.2.9.1 Liked Restaurants

사용자가 찜한 가게를 좌우로 스크롤할 수 있도록 하여 모든 리스트를 보여준다. 가게의 정보는 가게 이미지와 이름만 보여지며, 해당 가게를 클릭할 경우 가게 상세 화면으로 이동하게 된다.

### 2.2.9.2 Reviews Written by the User

사용자가 작성한 리뷰를 화면 상에는 하나만 보여주되 옆으로 스크롤 할 수 있어, 작성한 모든 리뷰를 담고 있다. 해당 리뷰를 클릭할 경우 가게 상세 화면으로 이동하게 된다.

### 2.2.9.3 Change User Information

사용자의 개인정보를 수정하는 기능이다. 이메일 주소를 제외한 닉네임과 이름 항목은 수정이 가능하며 중복 확인을 통해 고유한 값을 갖도록 한다.

## 2.3 Characteristics

### 2.3.1 User

이 어플리케이션의 사용자는 성균관대학교 학생 및 교수진과 관계자이다. 사용자는 다음과 같은 조건들을 갖추고 있다고 가정한다. 첫째, 기존 지도 어플리케이션의 검색과 추천 시스템

이용에 불편함을 느끼고 있다고 가정한다. 둘째, 기본적인 한국어를 읽고 이해할 수 있다고 가정한다. 마지막으로, 자신의 기기에 어플리케이션을 설치하고 사용할 수 있는 기본지식을 갖고 있다고 가정한다.

### 2.3.2 System Manager

어플리케이션의 시스템 관리자는 시스템 전문가로 제한한다. 시스템 관리자는 다음과 같은 조건들을 갖추고 있다고 가정한다. 첫째, 이들은 ‘먹꾸스꾸’ 시스템에 대한 충분한 지식이 있어야하고, 전반적인 시스템 상황을 이해할 수 있어야 한다. 둘째, 시스템 오류나 그러한 종류의 문제를 탐지하고, 다룰 수 있는 능력이 있다고 가정한다. 셋째, 새로운 시스템 변경사항을 반영할 수 있는 능력이 있다고 가정한다. 마지막으로, 소프트웨어 윤리의식도 가지고 있어야한다.

## 2.4 Operating Environment

### 2.4.1 Hardware

해당 애플리케이션은 Android 및 iOS operating system 기반의 mobile phone을 위해 설계되었다. 구동 가능한 사용자 기기는 최소 512MB의 RAM 용량과 64bit의 CPU, 200MB의 DISK가 디바이스에 내장되어 있음을 전제한다.

### 2.4.2 Software

해당 애플리케이션은 최소 Android OS 5.0(API 레벨 23) 혹은 IOS 13.4 이상의 버전을 전제하고 있다. 또한 Android OS 10.0(API 레벨 29) 버전에서 최상의 환경을 제공한다. 따라서 이보다 낮은 사양의 경우 제공하는 모든 서비스를 기획한대로 원활하게 이용할 수 없다.

## 2.5 Implementation Constraints

해당 애플리케이션은 다음의 체크리스트를 고려하여 Implementation 될 것이다. 아래는 Implementation 방향에 대한 세부 사항이 기술되어 있다.

- ✓ 시스템은 사용자의 편의와 요구사항을 고려해야 하며, 해당 내용에 따라 수정 가능하다.
- ✓ 어떤 플랫폼에서 해당 앱으로 접속하더라도 항상 동일한 UI, 기능을 제공해야 한다.
- ✓ 시스템은 최소 512MB 이상의 RAM을 가진 모바일 기기에서 실행되어야 한다.
- ✓ 시스템 설치 및 구동을 위해 디바이스에 최소 200MB의 여유 저장공간이 있어야 한다.
- ✓ 사용자 디바이스와 서버는 HTTP 프로토콜로 통신한다.
- ✓ 소스 코드는 메모리 낭비를 방지하기 위해 적절히 최적화 될 것이다.
- ✓ 시스템 비용과 이를 유지하기 위한 비용 모두 발생한다.
- ✓ 기본적으로 한국어를 지원한다.
- ✓ 시스템을 사용하기 위해, 사진 등록을 위한 갤러리 접근, 지도 사용을 위한 위치 정보 제공에 대한 허용 권한을 요구할 수 있으며, 권한 허용이 되지 않을 시 서비스 이용에 차질이 발생할 수 있다.
- ✓ 검색 결과 반환과 음식점 리스트 로딩 속도는 1sec를, 좋아요 여부가 UI에 반영되는 속도는 1sec를, 자동완성 키워드 데이터를 보여주는 속도는 0.1sec를, 지도 표시 속도는 1sec를 넘기지 않아야 한다.
- ✓ 최소 Python 3.7 버전 이상, springBoot 2.7.5 버전 이상, Java 17 버전 이상, Kotlin 1.9 버전 이상, React Native 12.3.6 버전 이상, Elasticsearch 8.2.0 버전, Kibana 8.2.0 버전, Redis 6.0.2 버전, Mysql 8.0 버전으로 개발한다.
- ✓ 최소 Android 5.0 (API 레벨 29) 버전 및 iOS 13.4 버전으로 개발한다.
- ✓ Android 10.0 (API 레벨 29) 버전을 사용하여 시스템을 Emulate 한다.

## 2.6 User Documentation

사용자가 애플리케이션 및 서비스를 이용할 수 있도록 지원하기 위해서 몇 가지 문서가 필요하다. 첫째, 최소 하드웨어 및 소프트웨어 요구사항을 다룬 문서이다. 둘째, 사용자 매뉴얼이 필요하다. 사용자 매뉴얼의 내용은 기기에 익숙하지 않은 사용자를 돋기 위한 애플리케이션 설치 가이드, 시스템 시작 방법 및 다양한 기능 사용 방법에 대한 설명을 내포하고 있다. 회원가입 및 지도/리스트를 통한 맛집 검색 등의 특징적인 기능에 대해서는 서비스 이용 전과정을 설명하는 모바일 기기 스크린샷 및 매뉴얼 내용을 설명하는 영상도 함께 제공한다. 마지막으로, 시스템 개발자와 연락할 수 있는 정보를 제공한다.

## 2.7 Assumptions and Dependencies

해당 애플리케이션은 기본적으로 Android 5.0 이상, iOS 13.4 이상의 device를 바탕으로 설계 및 구현되었다. 따라서 모든 기능과 컨텐츠는 해당하는 두 운영체제 내에서 정상적으로 작동한다.

### 3. Specific Requirements

#### 3.1 External Interface Requirements

##### 3.1.1 User Interfaces

[Table 2] User Interface - Loading and Main

Name	로딩 화면 및 메인 화면
Purpose/Description	사용자들에게 애플리케이션이 로딩되고 있는 중임을 나타내는 UX경험 증대 및 메인화면 안내 및 자동로그인
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	-
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>유저가 애플리케이션을 실행시 [Figure 1]과 같은 로딩 화면 렌더링</li> <li>로딩 완료후, [Figure 2]의 메인 화면 렌더링</li> <li>기기내에 사용자의 로그인 정보가 있는 경우 자동 로그인</li> <li>정보가 없는 경우 ‘시작하기’ 클릭시 로그인 화면으로 이동</li> </ol>
Format, Configuration of window	-
Data type	로그인 토큰 (String)
Instruction type	버튼 텍스트에 따른 지시
Exit message	-



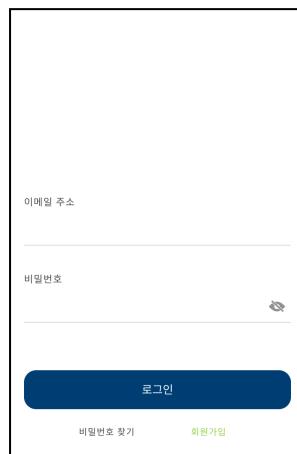
[Figure 1] User Interface - Loading Page



[Figure 2] User Interface - Main Page

[Table 3] User Interface- Login

Name	로그인 화면
Purpose/Description	애플리케이션을 사용하기 위해서는 기기에 로그인 정보가 없는 경우 로그인을 거쳐야 한다.
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	사용자의 계정 이메일과 계정 비밀번호의 입력 / 서버로부터의 확인
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>등록된 비밀번호와 이메일로 ‘로그인’ 하여 애플리케이션의 기능들을 사용할 수 있다.</li> <li>비밀번호를 잊어버린 경우, ‘비밀번호 찾기’를 눌러 비밀번호 찾기 화면으로 이동할 수 있다.</li> <li>처음으로 애플리케이션을 이용하는 경우 ‘회원가입’ 해야한다.</li> </ol>
Format, Configuration of window	-
Data type	이메일 주소 (String), 이메일 인증 번호 (Integer), 비밀번호(String)
Instruction type	-
Exit message	-



[Figure 3] User Interface - Login Page

[Table 4] User Interface - Sign-up

Name	회원가입 화면
Purpose/Description	애플리케이션을 처음 사용하는 유저는 회원가입을 거쳐야 한다.
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	이름, 성균관대 이메일, 사용할 비밀번호 / 유효한 이메일인지의 확인 이메일 인증 번호 / 인증번호 확인 닉네임, 프로필 사진 / 로그인 토큰
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>로그인에 사용할 이름, 이메일, 비밀번호를 확인하면 회원가입 버튼이 활성화 된다.</li> <li>사용자의 이메일로 인증 번호를 보낸다. (성균관대 skku 의 도메인만 인증이 가능하다)</li> <li>인증번호로 유효한 이메일인지 확인과정을 거친다.</li> <li>이메일의 인증이 완료 되면 개인의 프로필 사진과 닉네임을 입력 받아 회원가입을 진행한다.</li> <li>회원가입에 성공하면 로그인 토큰을 받아 앱의 메인 화면으로 넘어간다.</li> </ol>
Format, Configuration of window	-

Data type	이름(String), 이메일 주소 (String), 이메일 인증 번호 (Integer), 비밀번호(String), 프로필 이미지 (Image), 닉네임 (String)
Instruction type	-
Exit message	-

The sign-up page titled '회원가입' (Membership Registration) contains four input fields: '이름' (Name), '이메일 주소' (Email Address), '비밀번호' (Password), and '비밀번호 확인' (Password Confirmation). Below the fields is a large blue button labeled '회원가입' (Registration). At the bottom, a small note reads: '가입하시면 이용약관 및 개인정보 보호정책에 자동으로 동의하게 됩니다.' (By joining, you automatically agree to the Terms of Service and Privacy Policy).

The authentication page titled '인증번호 확인' (Verification Number Confirmation) features a single input field for '인증번호 입력' (Enter Verification Number). Below the input field is a dark grey button labeled '인증번호 확인' (Check Verification Number). A note at the bottom states: '가입하시면 이용약관 및 개인정보 보호정책에 자동으로 동의하게 됩니다.' (By joining, you automatically agree to the Terms of Service and Privacy Policy).

[Figure 4] User Interface - Sign-up Page [Figure 5] User Interface - Authentication Page

The profile setting page titled '프로필 설정' (Profile Setting) includes a circular camera icon for profile picture selection. Below it is a text input field for '닉네임' (Nickname). At the bottom is a blue button labeled '회원가입' (Registration).

[Figure 6] User Interface - Profile Setting Page

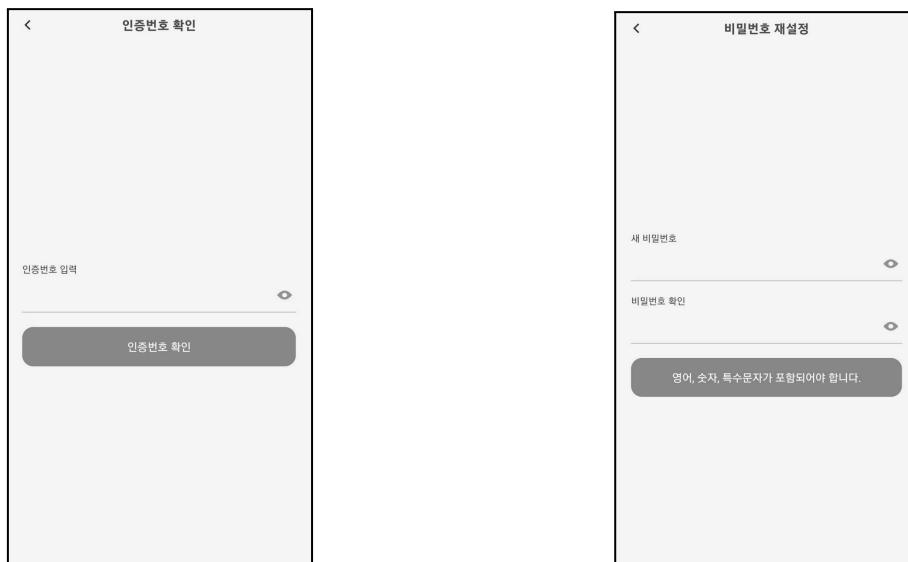
[Table 5] User Interface - Password Change

Name	비밀번호 변경 화면
------	------------

Purpose/Description	비밀번호를 분실한 유저는 이를 변경해야 한다.
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	성균관대 이메일 / 유효한 이메일인지의 확인 이메일 인증 번호 / 인증번호 확인 변경할 비밀번호 / 변경 성공여부
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>1. 비밀번호를 변경한 계정의 이메일로 인증 번호를 보낸다. (회원가입 되어있는 성균관대 skku 의 도메인만 인증이 가능하다)</li> <li>2. 인증번호로 유효한 이메일인지 확인과정을 거친다.</li> <li>3. 이메일의 인증이 완료 되면 변경할 비밀번호를 입력한다.</li> <li>4. 비밀번호 변경에 성공하면 로그인 화면으로 넘어간다.</li> </ol>
Format, Configuration of window	-
Data type	이메일 주소 (String), 이메일 인증 번호 (Integer), 새 비밀번호(String)
Instruction type	-
Exit message	-



[Figure 7] User Interface - Password Change - Check Email Page



[Figure 8] User Interface - Authentication Page    [Figure 9] User Interface - Password Page

[Table 6] User Interface - Home

Name	홈 화면
Purpose/Description	유저는 성균관대 근처의 추천 맛집, 할인 정보와 음식 카테고리를 한눈에 볼 수 있다
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	-
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>유저가 로그인 하면 서버에서 오늘의 픽, 킹고패스 정보를 가져와 보여준다.</li> <li>오늘의 픽 음식점을 '터치' 하여 자세한 정보를 볼 수 있다.</li> <li>각 카테고리를 '터치' 하여 카테고리 별 음식점의 리스트 페이지로 이동할 수 있다.</li> <li>킹고패스 적용 음식점을 '터치' 하여 자세한 정보를 볼 수 있다.</li> </ol>
Format, Configuration of window	-
Data type	카테고리(String), 음식점 정보 (Object)

Instruction type	-
Exit message	-

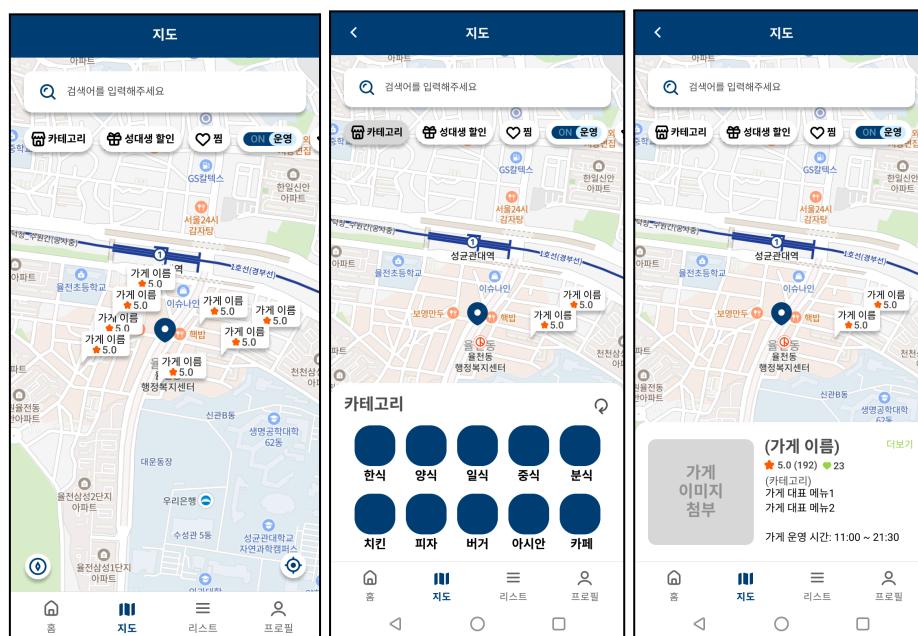


[Figure 10] User Interface - Home Page

[Table 7] User Interface - Map

Name	지도 화면
Purpose/Description	유저는 성균관대 근처의 음식점 정보를 지도에서 볼 수 있다
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	확인할 음식점의 조건 / 조건에 해당하는 음식점 정보
Format, Configuration	1. 유저는 지도와 함께 성균관대학교 주변의 음식점 정보를 확인

of screen & User Flow	<p>할 수 있다.</p> <ol style="list-style-type: none"> <li>2. 유저는 지도에서 자신의 위치도 확인할 수 있다.</li> <li>3. 유저는 확인하고 싶은 음식점의 조건을 설정하고 해당 음식점만 표시할 수 있다.</li> <li>4. 유저는 음식점의 자세한 정보를 확인하는 페이지로 이동할 수 있다.</li> <li>5. 유저는 원하는 음식점을 검색할 수 있다.</li> </ol>
Format, Configuration of window	-
Data type	카테고리(String), 음식점 정보 (Object)
Instruction type	-
Exit message	-

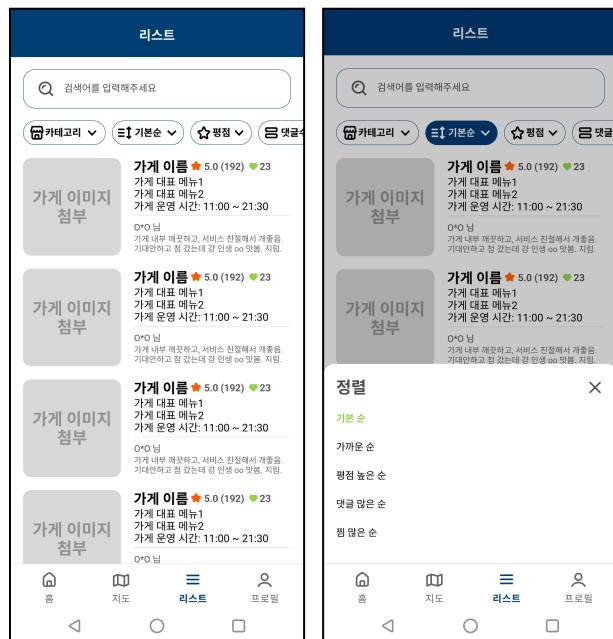


[Figure 11] User Interface - Map Page

[Table 8] User Interface - List

Name	리스트 화면
Purpose/Description	유저는 성균관대 근처의 음식점 정보를 목록으로 볼 수 있다
Input source/	Client / Server

Output destination	
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	확인할 의 조건 / 조건에 해당하는 식당 정보
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>1. 유저는 리스트로 성균관대학교 주변의 식당정보를 확인 할 수 있다.</li> <li>2. 유저는 확인하고 싶은 식당의 조건을 설정하고 해당 식당만 표시할 수 있다.</li> <li>3. 유저는 식당의 자세한 정보를 확인하는 페이지로 이동할 수 있다.</li> <li>4. 유저는 원하는 식당을 검색하는 검색페이지로 이동할 수 있다.</li> <li>5. 검색 결과를 가져와서 리스트로 볼 수 있다.</li> </ol>
Format, Configuration of window	-
Data type	카테고리(String), 음식점 정보 (Object)
Instruction type	-
Exit message	-



[Figure 12] User Interface - List Page

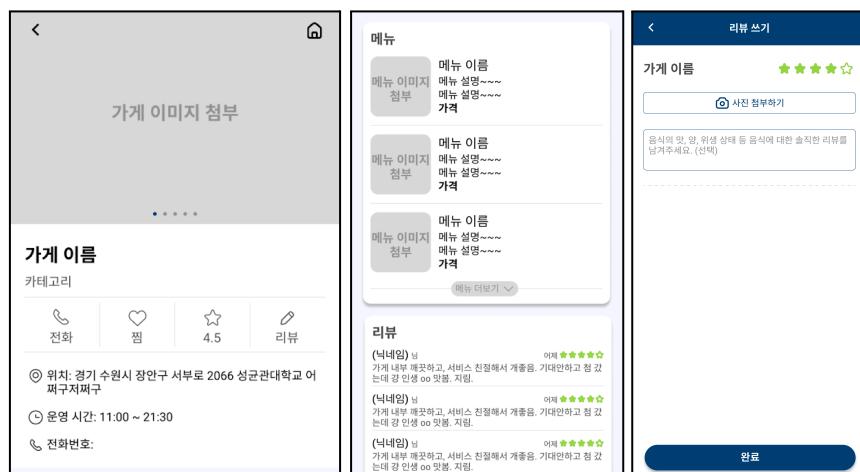
**[Table 9] User Interface - Search**

Name	검색 화면
Purpose/Description	유저는 성균관대 근처의 식당 정보를 검색으로 필터링 할 수 있다.
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	검색하고 싶은 키워드 / 검색 결과
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>1. 유저는 검색 창에 검색어를 입력하여 식당이름, 카테고리, 메뉴 이름, 메뉴 설명을 검색할 수 있다.</li> <li>2. 이 검색의 결과는 리스트나 지도 페이지로 넘겨진다.</li> <li>3. 이전에 검색했던 검색어는 최근 검색어로 저장되어 다시 사용할 수 있다.</li> </ol>
Format, Configuration of window	-
Data type	검색어 (String)
Instruction type	-
Exit message	-

**[Figure 13] User Interface - Search Page**

[Table 10] User Interface - Restaurant Information

Name	음식점 화면
Purpose/Description	유저는 성균관대 근처의 식당 정보를 자세히 볼 수 있다.
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	식당의 기본 정보 / 식당의 상세 정보 및 리뷰
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>유저는 식당의 상세 정보를 확인 할 수 있다.</li> <li>식당의 전화번호로 전화를 걸 수 있다.</li> <li>찜 하기 버튼을 통해 식당을 저장 할 수 있다.</li> <li>유저는 식당의 메뉴를 확인 할 수 있다.</li> <li>유저는 다른 사람의 리뷰를 볼 수 있고, 자신이 리뷰를 작성 할 수 있다.</li> </ol>
Format, Configuration of window	-
Data type	리뷰 사진 (Image), 리뷰 내용 (Text)
Instruction type	-
Exit message	-



[Figure 14] User Interface - Restaurant Page

[Table 11] User Interface - My Profile

Name	유저 정보화면
Purpose/Description	유저가 찜한 가게, 작성한 리뷰를 확인할 수 있다.
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>유저 본인이 찜한 가게들을 옆으로 스크롤하면서 모두 확인해볼 수 있다.</li> <li>유저 본인이 작성한 리뷰들을 옆으로 넘겨가며 모두 확인해볼 수 있다.</li> <li>사용자 이름을 '터치' 하여 내 정보 수정 화면으로 넘어갈 수 있다.</li> </ol>
Format, Configuration of window	-
Data type	카테고리(String), 음식점 정보 (Object)
Instruction type	-
Exit message	-



[Figure 15] User Interface - My Profile Page

**[Table 12] User Interface - Edit My Profile**

Name	유저 닉네임 수정 및 로그아웃, 회원탈퇴 화면
Purpose/Description	유저 본인의 닉네임을 수정하고 로그아웃, 회원탈퇴를 할 수 있다.
Input source/ Output destination	Client / Server
Unit	Screen
Time/Velocity	-
Relationship with other input/outputs	변경할 닉네임 / 중복여부
Format, Configuration of screen & User Flow	<ol style="list-style-type: none"> <li>원하는 닉네임을 입력 후 중복확인 버튼을 누르면 중복여부를 확인할 수 있다.</li> <li>수정하기 버튼을 누르면 입력한 닉네임으로 내 정보 수정이 이루어진다.</li> <li>로그아웃을 누르면 로그아웃이 되고 회원탈퇴를 누르면 먹구스꾸에서 탈퇴가 가능하다.</li> </ol>
Format, Configuration of window	-
Data type	닉네임(String)
Instruction type	-
Exit message	-

**[Figure 16] User Interface - Edit My Profile Page**

### 3.1.2 Hardware Interfaces

'먹구스꾸'는 5.0 이상 버전의 Android와 13.4 이상 버전의 iOS를 지원한다. 이 애플리케이션을 설치하기 위해서는 최소 200MB의 여유 저장공간이 있어야 한다. 또한, 앱의 기능을 사용하기 위해서는 사진 등록을 위한 갤러리 접근 허용 권한과, 지도 사용을 위한 위치 정보 허용 권한이 필요하다.

### 3.1.3 Software Interfaces

'먹구스꾸'는 성균관대학교 이메일을 통해서만 가입과 사용이 가능하다.

### 3.1.4 Communication Interfaces

#### 3.1.4.1 Email

- **회원가입 시 이메일 인증:** 회원가입 절차의 일환으로 사용자는 이메일 인증을 완료해야 합니다. 서버는 사용자가 입력한 이메일 주소로 인증 링크를 포함한 이메일을 전송합니다. 사용자가 해당 링크를 클릭하면 인증이 완료됩니다.
- **비밀번호 재설정 시 이메일 인증:** 비밀번호 재설정을 요청할 경우, 서버는 사용자의 이메일 주소로 비밀번호 재설정 링크를 전송합니다. 사용자는 해당 링크를 통해 새로운 비밀번호를 설정할 수 있습니다.

#### 3.1.4.2 Application

- 사용자는 애플리케이션을 통해 접근합니다.

#### 3.1.4.3 Network Server Protocol

- **HTTP/HTTPS:** 클라이언트와 서버 간의 모든 통신은 HTTP 또는 HTTPS를 통해 이루어집니다.

### 3.1.4.4 Message Format

- 모든 통신 메시지는 JSON 포맷을 사용하여 교환됩니다.

### 3.1.4.5 Communications Security and Encryption

- 모든 토큰은 암호화되어 전송됩니다.

### 3.1.4.6 Data Transfer Speed and Synchronization Mechanism

- 데이터 전송 속도는 네트워크 상태에 따라 다를 수 있으며, 실시간 동기화를 위한 별도의 메커니즘을 사용하지 않습니다. 클라이언트는 필요 시 서버에 데이터를 요청하여 최신 상태를 반영합니다.

### 3.1.4.7 Authentication and Token Management

1. 회원가입과 로그인 시:

- a. 서버에서 클라이언트로 Access Token과 Refresh Token을 전달합니다.

2. 토큰 저장 및 검증:

- a. 서버는 토큰 발급 즉시 Refresh Token을 Redis에 저장합니다.

3. Access Token이 만료된 경우:

- a. 클라이언트는 서버에 새로운 Access Token 발급을 요청합니다.

- b. 서버는 Redis에서 저장된 Refresh Token을 검증합니다.

- c. Refresh Token이 유효하면 새로운 Access Token을 발급하여 클라이언트에게 전달합니다.

- d. Refresh Token이 유효하지 않으면 클라이언트에게 401(UnAuthorized) 상태 코드와 함께 다시 로그인이 필요함을 알리는 정보를 전달합니다.

4. 클라이언트 요청:

- a. 클라이언트는 모든 요청 시 HTTP 헤더에 Access Token을 포함하여 서버에 보냅니다.

5. 서버의 토큰 검증 및 인가:
- 서버는 Access Token의 유효성을 확인합니다.
  - Access Token이 유효하지만 해당 API에 대한 권한이 없는 경우 401(UnAuthorized) 상태 코드로 클라이언트에 응답합니다.
  - Access Token이 만료된 경우, 클라이언트는 새로운 Access Token을 발급받기 위해 Refresh Token을 사용하여 서버에 요청합니다.
  - 서버는 Refresh Token의 유효성을 확인하고, 유효하다면 새로운 Access Token을 발급합니다.
  - Refresh Token이 유효하지 않으면 401(UnAuthorized) 상태 코드와 함께 다시 로그인이 필요함을 알리는 정보를 클라이언트에 전달합니다.

## 3.2 Functional Requirements

### 3.2.1 Use Case

[Table 13] Use Case of Sign up and Profile setting

Use case name	회원가입 및 프로필 설정
Actor	유저
Description	회원가입하지 않은 유저가 애플리케이션에 회원 정보를 등록한다.
Normal Course	<ol style="list-style-type: none"> <li>회원가입을 위해 아래 정보를 기입한다.             <ol style="list-style-type: none"> <li>이름</li> <li>이메일 주소</li> <li>이메일 인증 번호</li> <li>비밀번호</li> <li>비밀번호 확인</li> <li>프로필 이미지</li> <li>닉네임</li> </ol> </li> <li>이메일 인증번호가 확인이 되어야만 회원가입이 가능하다.</li> </ol>

	<p>3. 비밀번호의 유효성이 검증되어야만 회원가입이 가능하다.</p> <p>4. 1-(d), 1-(e) 과정에서 비밀번호가 일치되어야만 회원가입이 가능하다.</p> <p>5. 2, 3, 4 가 모두 검증되지 않으면 회원가입이 불가능하다.</p>
Pre-Condition	<p>1. 같은 이메일을 가진 사용자가 등록되지 않아야 한다.</p> <p>2. 필수 입력 정보들이 모두 기입되고, 해당 정보들의 유효성이 확인되어야만 한다.</p>
Post-Condition	<p>1. 유저의 비밀번호는 암호화되어 DB에 저장된다.</p> <p>2. 프로필 이미지의 경우 API Gateway를 통해 AWS Lambda로 S3에 정적 파일로 저장된다. DB에는 등록된 이미지의 URL이 저장된다.</p>
Assumptions	해당 사항 없음

[Table 14] Use Case of Login

Use case name	로그인
Actor	유저
Description	회원가입이 완료된 유저가 애플리케이션을 사용하기 위해 로그인한다.
Normal Course	<p>1. 유저가 회원가입 과정에서 등록한 정보들로 로그인한다. 필요한 정보들은 다음과 같다.</p> <ul style="list-style-type: none"> <li>a. 이메일 주소</li> <li>b. 비밀번호</li> </ul> <p>2-1. 이메일 주소와 이에 맞는 비밀번호가 유효한 경우 로그인에 성공한다.</p> <p>2-2. 정보들이 유효하지 않은 경우, 로그인에 실패한다.</p>
Pre-Condition	이미 회원가입이 완료된 유저여야 한다.
Post-Condition	로그인에 성공하여 애플리케이션의 기능들을 사용할 수 있다.
Assumptions	해당 사항 없음

**[Table 15] Use Case of Finding Password**

Use case name	비밀번호 찾기
Actor	유저
Description	등록된 이메일 정보로 비밀번호를 초기화하여 다시 로그인할 수 있게 한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 회원가입시 등록한 이메일을 기입한다.</li> <li>2. 비밀번호 초기화 버튼 요청을 보내면, 등록된 이메일인지 검증한다.             <ol style="list-style-type: none"> <li>a. 등록된 이메일일 경우                     <ol style="list-style-type: none"> <li>i. AWS SES를 통해 등록된 이메일로 인증번호를 보낸다.</li> <li>ii. 유저는 해당 인증번호를 통해, 유저가 직접 비밀번호를 변경한다.</li> </ol> </li> <li>b. 등록되지 않은 이메일일 경우                     <ol style="list-style-type: none"> <li>i. 유저에게 등록되지 않은 이메일임을 알려준다.</li> </ol> </li> </ol> </li> </ol>
Pre-Condition	없음
Post-Condition	유저가 원하는 비밀번호로 변경하여 다시 로그인할 수 있다.
Assumptions	유저가 비밀번호 정보를 잊어버린 상황을 가정한다.

**[Table 16] Use Case of Profile Edit**

Use case name	유저 수정
Actor	유저
Description	유저의 닉네임과 프로필 이미지를 수정한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 변경하고 싶은 닉네임을 기입한다.</li> <li>2. 변경하고 싶은 프로필 이미지를 등록한다.</li> <li>3. 닉네임의 중복확인 과정을 거친 후, 프로필 정보를 수정한다.</li> </ol>
Pre-Condition	닉네임이 중복되지 않아야 한다.
Post-Condition	1. 유저의 닉네임이 DB에서 변경된다.

	2. 프로필 이미지의 경우 API Gateway를 통해 AWS Lambda로 S3에 정적 파일로 저장된다. DB상에서 프로필 이미지 URL이 변경된다.
Assumptions	유저가 비밀번호 정보를 잊어버린 상황을 가정한다.

[Table 17] Use Case of Writing Reviews

Use case name	리뷰 작성
Actor	유저
Description	식당을 방문한 후 리뷰를 작성한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 유저는 다음과 같은 정보들을 등록한다.             <ol style="list-style-type: none"> <li>a. 리뷰 텍스트</li> <li>b. 첨부 이미지</li> <li>c. 별점</li> </ol> </li> <li>2. 리뷰가 비어있으면 안된다.</li> <li>3. 별점이 반드시 표시되어야한다.</li> <li>4. 2, 3의 유효성이 검증되지 않으면 리뷰 등록에 실패한다.</li> </ol>
Pre-Condition	리뷰 텍스트와, 별점은 반드시 필요하다.
Post-Condition	리뷰 이미지의 경우 API Gateway를 통해 AWS Lambda로 S3에 정적 파일로 저장된다. DB상에서 해당 이미지 URL이 포함된 정보를 저장한다.
Assumptions	해당 사항 없음

[Table 18] Use Case of Editing Reviews

Use case name	리뷰 수정
Actor	유저
Description	자신이 등록한 리뷰를 수정한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 유저는 다음과 같은 정보들을 수정한다.             <ol style="list-style-type: none"> <li>a. 리뷰 텍스트</li> <li>b. 첨부 이미지</li> <li>c. 별점</li> </ol> </li> </ol>

	2. 리뷰가 비어있으면 안된다. 3. 별점이 반드시 표시되어야한다. 4. 2, 3의 유효성이 검증되지 않으면 리뷰 수정에 실패한다.
Pre-Condition	1. 자신이 등록한 리뷰만 수정할 수 있다. 2. 리뷰 텍스트와 별점은 반드시 필요하다.
Post-Condition	리뷰 이미지의 경우 API Gateway를 통해 AWS Lambda로 S3에 정적 파일로 저장된다. DB상에서 수정된 이미지 URL, 리뷰 텍스트, 별점으로 기존의 리뷰를 수정한다.
Assumptions	해당 사항 없음

[Table 19] Use Case of Deleting Reviews

Use case name	리뷰 삭제
Actor	유저
Description	자신이 등록한 리뷰를 삭제한다.
Normal Course	리뷰 삭제 버튼을 눌러 삭제 API를 호출한다.
Pre-Condition	자신이 등록한 리뷰만 삭제할 수 있다.
Post-Condition	해당 리뷰 정보가 DB에서 삭제된다.
Assumptions	해당 사항 없음

[Table 20] Use Case of Opening List of Reviews for Each Restaurant

Use case name	음식점별 리뷰 리스트 조회
Actor	유저
Description	특정 음식점의 리뷰 리스트들을 조회한다.
Normal Course	식당 고유 ID를 통해 해당 식당에 등록된 리스트들을 Pagination으로 조회한다. 이 때, Pagination으로는 세가지 정보가 포함된다. <ol style="list-style-type: none"> <li>페이지 번호</li> <li>페이지 크기</li> </ol>

	c. 정렬 기준 해당 정보를 통해, 정렬 기준에 맞는 리뷰들의 수를 페이지 단위로 조회한다.
Pre-Condition	해당 사항 없음
Post-Condition	해당 사항 없음
Assumptions	해당 사항 없음

[Table 21] Use Case of Opening Reviews that the User has Written

Use case name	내가 작성한 리뷰 조회
Actor	유저
Description	유저가 작성한 리뷰 리스트를 조회한다.
Normal Course	로그인한 유저의 고유 ID를 통해 해당 식당에 등록된 리스트들을 Pagination으로 조회한다. 이 때, Pagination으로는 세가지 정보가 포함된다. a. 페이지 번호 b. 페이지 크기 c. 정렬 기준 해당 정보를 통해, 정렬 기준에 맞는 리뷰들의 수를 페이지 단위로 조회한다.
Pre-Condition	해당 사항 없음
Post-Condition	해당 사항 없음
Assumptions	해당 사항 없음

[Table 22] Use Case of Opening a Specific Review

Use case name	리뷰 단건 조회
Actor	유저
Description	리뷰 리스트에서 단건의 상세 리뷰 정보를 조회한다.
Normal Course	1. 고유 리뷰 ID를 통해, 다음과 같은 정보를 조회한다. a. 리뷰 ID

	<ul style="list-style-type: none"> <li>b. 리뷰 텍스트</li> <li>c. 리뷰 이미지 URL 리스트</li> <li>d. 유저가 해당 리뷰를 좋아요 했는지 여부</li> <li>e. 리뷰가 받은 좋아요 수</li> <li>f. 리뷰를 작성한 유저 ID</li> <li>g. 리뷰 작성한 유저 프로필 이미지</li> <li>h. 리뷰 작성한 유저 닉네임</li> <li>i. 리뷰 조회수</li> <li>j. 리뷰를 작성한 식당 고유 ID</li> <li>k. 별점</li> </ul> <p>2. 조회 후, 해당 리뷰의 조회수를 추가한다.</p>
Pre-Condition	해당 사항 없음
Post-Condition	조회 API 호출마다 리뷰의 조회수를 1씩 증가한다.
Assumptions	해당 사항 없음

[Table 23] Use Case of Liking/Unliking Restaurants and Reviews

Use case name	음식점과 리뷰 좋아요/좋아요 취소
Actor	유저
Description	음식점에 대해 좋아요 설정하거나 취소한다
Normal Course	유저는 아래의 두 가지에 대해 좋아요를 설정할 수 있다 a. 음식점 b. 리뷰
Pre-Condition	본인이 등록한 리뷰는 좋아요를 설정할 수 없다
Post-Condition	좋아요 여부 정보가 DB에 반영되고 타 유저의 페이지에 반영된다.
Assumptions	해당 사항 없음

[Table 24] Use Case of Searching Map

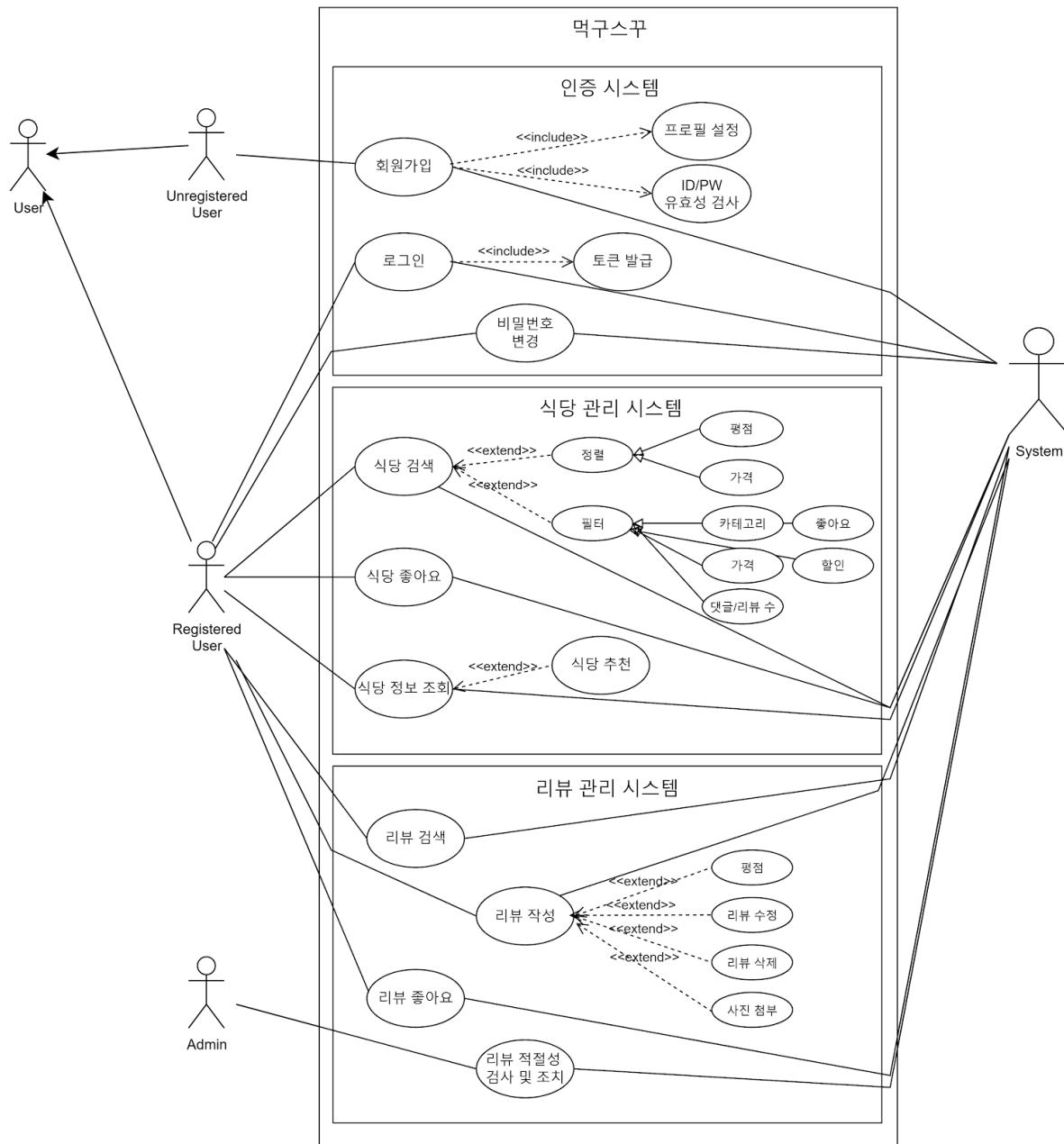
Use case name	지도 탐색
Actor	유저

Description	지도를 기반으로 음식점을 검색 및 조회한다
Normal Course	<ol style="list-style-type: none"> <li>1. 자신의 위치에 기반하여 지도가 표시된다.</li> <li>2. 음식점 카테고리 별로 지도에서 음식점을 검색 및 조회할 수 있다.</li> <li>3. 지도에 표시된 음식점을 클릭하여 정보를 조회할 수 있다.             <ol style="list-style-type: none"> <li>a. 클릭 후 음식점 정보 페이지로 이동한다.</li> </ol> </li> <li>4. 원하는 식당을 검색하여 지도에서 찾을 수 있다.</li> </ol>
Pre-Condition	사전에 위치 정보 제공 동의가 필요하다.
Post-Condition	해당 사항 없음
Assumptions	해당 사항 없음

[Table 25] Use Case of Opening Recent History of the User

Use case name	
Actor	유저
Description	유저가 검색한 히스토리를 조회한다.
Normal Course	<ol style="list-style-type: none"> <li>1. 음식점을 검색하면 유저가 검색한 검색어가 저장 된다.</li> <li>2. 검색 페이지 화면을 보면 최근검색어를 볼 수 있다.</li> </ol>
Pre-Condition	해당 사항 없음
Post-Condition	최근검색어는 5개까지만 제공한다.
Assumptions	해당 사항 없음

### 3.2.2 Use Case Diagram



[Figure 17] Use Case Diagram

### 3.2.3 Data Dictionary

아래는 MySQL 테이블 스키마 정보이다.

[Table 26] User

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	유저 고유 번호
email	varchar(255)	Unique	유저 이메일
nickname	varchar(255)	Unique	유저 닉네임
password	varchar(255)		비밀번호(암호화)
profileImageUrl	varchar(255)		프로필 이미지 Url
roles	varchar(255)		역할
withdrawal	boolean		

[Table 27] Restaurant

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	식당 고유 번호
name	varchar(64)	NOT NULL	식당 이름
original_categories	varchar(64)	NOT NULL	카테고리
review_count	int	NOT NULL	리뷰 개수
like_count	int	NOT NULL	좋아요 개수
address	varchar(256)		주소
contact_number	varchar(32)		전화번호
rating_avg	double		평균 평점
representative_image_url	text		대표 이미지 Url
view_count	int		조회수

discount_content	varchar(128)		할인 내용
------------------	--------------	--	-------

[Table 28] Menus

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	메뉴 고유 번호
restaurant_id	bigint	FK NOT NULL	restaurants 테이블 id
name	varchar(255)	NOT NULL	메뉴 이름
price	int	NOT NULL	가격
description	varchar(512)	NOT NULL	설명
isRepresentative	boolean	NOT NULL	대표 메뉴 여부
image_url	varchar(512)	NOT NULL	이미지 Url

[Table 29] Categories

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	카테고리 고유 번호
name	varchar(255)	NOT NULL	카테고리 이름

[Table 30] Restaurant Categories

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	식당 카테고리 고유 번호
category_id	bigint	FK NOT NULL	categories 테이블 id
restaurant_id	bigint	FK NOT NULL	restaurants 테이블 id

**[Table 31] Restaurant Likes**

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	식당 좋아요 정보 고유 번호
restaurant_id	bigint	FK NOT NULL	restaurants 테이블 id
user_id	bigint	FK NOT NULL	users 테이블의 id

**[Table 32] Restaurant Reviews**

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	식당 리뷰 정보 고유 번호
createdAt	datetime(6)		생성일
modifiedAt	datetime(6)		수정일
content	varchar(255)	NOT NULL	내용
like_count	bigint	NOT NULL	좋아요 개수
rating	double	NOT NULL	평점
restaurantId	bigint	FK NOT NULL	restaurants 테이블 id
view_count	bigint	NOT NULL	조회 수
user_id	bigint	FK NOT NULL	users 테이블 id

**[Table 33] Restaurant Likes**

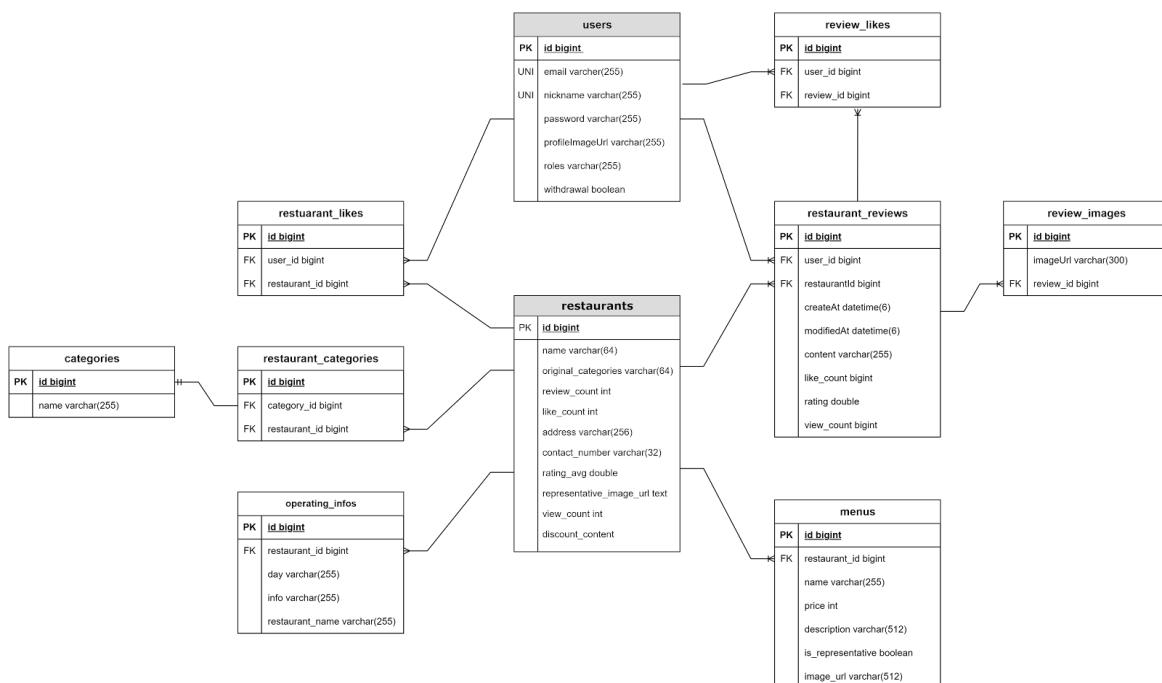
Name	Type	Constraint	Description
id	bigint	PK NOT NULL	좋아요한 리뷰 고유 번호
review_id	bigint	FK NOT NULL	reviews 테이블 id
user_id	bigint	FK NOT NULL	users 테이블 id

[Table 34] Review Images

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	리뷰 이미지 고유 번호
imageUrl	varchar(300)	NOT NULL	이미지 Url
review_id	bigint	FK NULL	reviews 테이블 id

[Table 35] Operating Information

Name	Type	Constraint	Description
id	bigint	PK NOT NULL	운영 시간 고유 번호
restaurant_id	bigint	FK NOT NULL	restaurants 테이블 id
day	varchar(255)	NOT NULL	요일
info	varchar(255)	NOT NULL	운영 정보
restaurant_name	varchar(255)	NOT NULL	식당 이름



[Figure 18] ER Diagram

## 3.3 Non-functional Requirements

### 3.3.1 Product Requirements

#### 3.3.1.1 Usability Requirements

- 언어는 한국어를 지원한다.
- 사용자가 페이지의 버튼과 텍스트, 아이콘만을 보고도 기능을 이해할 수 있어야 한다.
- 사용자가 쉽게 이전 페이지로 되돌아갈 수 있어야 한다.
- 사용자가 원하는 식당 정보를 쉽게 찾을 수 있어야 한다.

#### 3.3.1.2 Efficiency Requirements

- 검색 결과 반환과 식당 리스트 로딩 속도는 1sec를 넘지 않아야 한다.
- 좋아요 여부가 UI 반영되는 속도는 1sec를 넘지 않아야 한다.
- 자동완성 데이터를 보여주는 속도는 0.1sec를 넘지 않아야 한다.

#### 3.3.1.3 Dependability Requirements

- 유저가 가입한 계정 정보로 항상 로그인이 가능해야 한다.
- 사용자의 좋아요, 리뷰 정보를 손상 및 누락없이 다른 사용자가 조회할 수 있어야 한다.
- 관리자의 시스템 업데이트가 실 서비스의 중단 및 영향 없이 이루어져야 한다.

#### 3.3.1.4 Security Requirements

- 유저의 모든 개인정보는 암호화되어 DB에 저장되어야 한다.
- 사용자 계정에 대해 토큰을 발행하여 주기적으로 인증을 수행한다.
- 사용자가 업로드하는 리뷰, 평점 등의 모든 데이터는 익명성이 보장되어야 한다.
- 보안 취약점을 주기적으로 관리해야 한다.

### 3.3.2 Organization Requirements

#### 3.3.2.1 Environmental Requirements

- “먹구스꾸 V1.0” 모바일 Android 애플리케이션을 통해서만 사용할 수 있다.
- 위의 애플리케이션을 설치할 수 있을 만한 모바일 하드웨어 사양(64bit CPU, RAM 512MB, DISK 200MB)을 가지고 있어야 한다.
- 어떤 플랫폼에서 해당 앱으로 접속하더라도 항상 동일한 UI, 기능들을 제공해야 한다.

### 3.3.3 External Requirements

#### 3.3.3.1 Regulatory Requirements

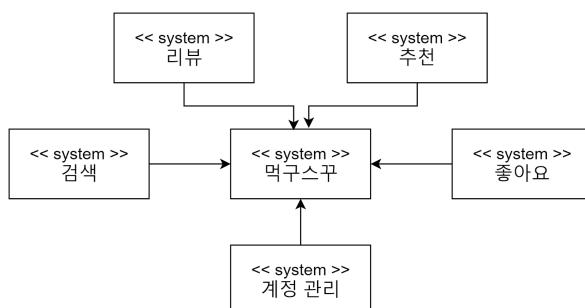
- 본 명세서에 명시된 오픈소스들의 라이선스 규정에 맞게 서비스를 개발해야 한다.
- 먹구스꾸 서비스에 필요한 카메라, 위치 정보 제공에 대한 동의를 사전에 얻어야 한다.

#### 3.3.3.2 Ethical Requirements

- 사용자가 생산한 모든 데이터를 사적 용도로 활용하지 않아야 한다.
- 사용자에게 거짓된 정보를 제공해선 안되고 주기적으로 검증해야 한다.

### 3.3.4 Organizing the Specific Requirements

해당 장은 통합 모델링 언어, UML(Unified Modeling Language)을 통해 시스템 모델을 설명한다. 여러 다이어그램을 통해 서비스의 구조와 요구사항들을 확인하는 것을 목표로 한다.



[Figure 19] Context Model

### 3.4 Logical Database Requirements

MySQL 데이터베이스를 통해 유저, 식당, 리뷰 등 서비스에 필요한 전반적인 정보를 관리한다. 시스템은 사용자의 ID, Password 뿐만 아니라 식당 정보, 리뷰 정보 등을 데이터베이스에 저장한다. 각 정보들은 3.2.3 장에서 명시한 대로 적절한 연관관계와 제약조건을 가지고 있으며 관리자는 해당 스키마 정보를 주기적으로 관리해야 한다.

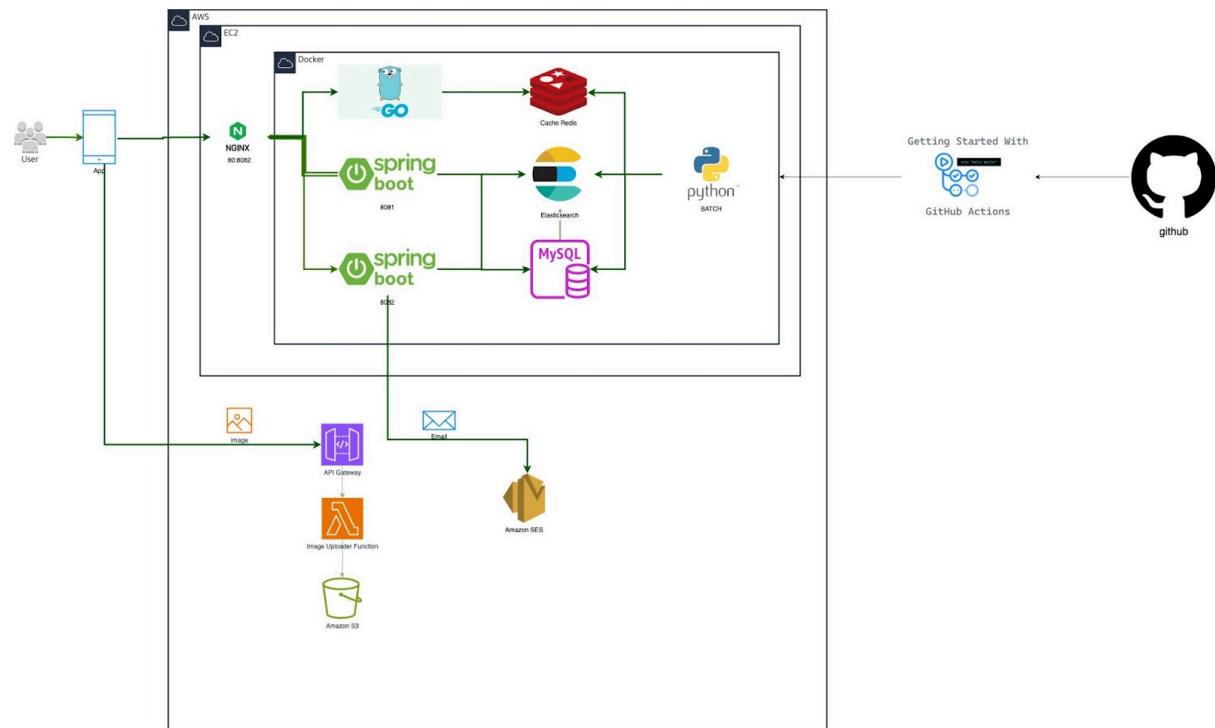
### 3.5 Design Constraints

본 시스템은 “먹구스꾸 V1.0”이 설치가능한 모든 플랫폼(안드로이드 등)에서 접속할 수 있어야 한다.

### 3.6 Standards Compliance

- React Native로 안드로이드, IOS를 지원하는 앱을 개발한다.
- Github Action을 통한 CI/CD를 구축한다.
- Nginx를 통해 서비스 지연 없는 무중단 배포를 구성하여 개발한다.

### 3.7 Standards Architecture



[Figure 20] Standard Architecture

1. 무중단 배포를 지원하기 위해 Nginx와 Spring Boot 서버 2대로 구성 한다.
2. 자동완성, 이미지 업로드를 제외한 호출은 전부 Spring Boot 서버로 호출한다.
3. 이미지 업로드 기능을 사용하면 API Gateway에서 Reverse Proxy를 통해 Lambda Function을 호출하고 S3에 이미지를 업로드한다.
4. 자동완성 기능을 사용하면 goLang API 서버를 호출하고, goLang 서버에서는 redis에 저장된 자동완성 데이터를 조회한다.
5. 이메일 전송 기능을 사용하면 Spring Boot API 서버를 호출하고, AWS SES 를 통해 이메일 전송을 한다.
6. 다양한 검색 조건과 품질 좋은 검색 성능을 위해 Elasticsearch를 사용한다.
7. 최근 검색어, Refresh Token, 자동완성 데이터는 빠른 조회가 필요한 반면에 휘발성으로 관리해도 괜찮은점을 고려하여 Redis를 사용한다.

8. Github Action으로 일 단위 batch로 python 언어를 통해
  - a. csv를 읽어서 naver geo API를 호출하여 음식점의 longitude, latitude 값을 csv에 저장한다.
  - b. csv를 읽어서 redis에 자동완성 데이터를 적재한다.
  - c. csv를 읽어서 elasticsearch에 음식점, 메뉴 등의 정보를 적재한다.
  - d. csv를 읽어서 mysql에 음식점, 메뉴 등의 정보를 적재한다.
  - e. mysql에 저장된 좋아요 갯수, 리뷰 갯수 등의 정보를 읽어서 elasticsearch로 적재한다.

## 3.8 System Evolution

이 섹션에서는 시스템에 기반이 되는 가정들을 설명하고 예상되는 변화들에 대해 설명한다.

이는 훗날 시스템 관리자가 시스템을 개선하는 데 도움이 되도록 한다.

### 3.8.1 Limitation and Assumption

본 어플리케이션의 추천 시스템이 유의미한 결과를 내기 위해서는 보다 많은 사용자가 이 플랫폼을 이용하여야 한다. 따라서 사용자가 적어 리뷰 수가 적다면 유의미하지 않은 추천 결과를 내게 된다. 이는 특히 초기 단계에서 해당 시스템이 겪을 수 있는 주요 문제점 중 하나이다. 이러한 상황을 완화하기 위해서는 초기 사용자 유입 전략과 데이터 수집 방안이 중요하다.

또한, 추천 시스템의 정확도는 사용자가 제공하는 데이터의 질에 크게 의존한다. 사용자가 리뷰나 평점을 성실히 작성하지 않거나, 편향된 데이터를 입력하는 경우, 추천 결과의 신뢰도가 낮아질 수 있다. 이 문제를 해결하기 위해서는 사용자에게 정확하고 솔직한 리뷰를 작성하도록 유도하는 인센티브 시스템을 도입할 필요가 있다.

마지막으로, 추천 시스템의 알고리즘 자체도 한계점을 가진다. 현재 사용되는 알고리즘이 특정 패턴이나 트렌드에 과도하게 의존할 경우, 다양성 있는 추천이 어렵고, 사용자에게 획일적인

콘텐츠만 제공될 가능성이 있다. 이를 방지하기 위해서는 알고리즘의 다양성을 확보하고,

주기적인 업데이트와 검증 과정을 통해 추천 결과의 질을 지속적으로 개선해 나가야 한다.

### 3.8.2 Anticipated Changes

본 어플리케이션의 시스템은 지속적인 개선과 발전을 통해 사용자 경험을 최적화하고자 한다.

이를 위해 예상되는 주요 변경 사항과 그에 따른 개선 방향은 다음과 같다:

#### 1. 데이터 수집 및 분석의 고도화

보다 다양하고 구체적인 식당 정보를 제공하고 사용자의 활동 데이터를 정교하게 수집할 예정이다. 이를 통해 사용자 행동 패턴을 더욱 정확하게 분석하고, 개인화된 추천의 정확도를 높일 수 있을 것이다. 예를 들어, 사용자의 클릭, 스크롤, 체류 시간 등의 데이터를 추가적으로 수집하여 분석함으로써 추천 시스템의 학습 데이터베이스를 강화할 계획이다.

#### 2. 실시간 데이터 처리 능력 향상

실시간 데이터 처리 인프라를 확충하고 효율화하여 시스템이 최신 데이터를 신속하게 반영하고 분석할 수 있도록 개선할 것이다. 사용자에게 더욱 시의성 있는 서비스를 제공하기 위해 데이터 스트리밍 기술을 도입하고, 서버 인프라를 확장하여 데이터 처리 속도를 높일 계획이다.

#### 3. 사용자 피드백 반영 시스템 도입

사용자 피드백을 실시간으로 반영하는 시스템을 구축할 예정이다. 신고나 피드백 기능을 통해 사용자가 명시적으로 부적절하거나 아쉬운 부분에 대해 관리자에게 의견을 남기고 이를 반영하는 시스템을 강화할 것이다. 또한 사용자가 제공하는 행동 피드백을 즉각적으로 분석하고, 이를 추천 알고리즘에 반영하여 추천 결과의 품질을 향상시킬 것이다.