



Design Specification Document

사용자 데이터 기반 카페 추천 시스템
카페 오아시스(Cafe Oasis)

| | | |
|---------|----------------------|----------|
| prof | 박영덕 교수님 | |
| Subject | 종합설계과제 | |
| Major | Computer Engineering | |
| Name | 우정현 | 21720847 |
| | 유아성 | 21720909 |
| | 김두현 | 21812131 |
| | 이다해 | 21912060 |

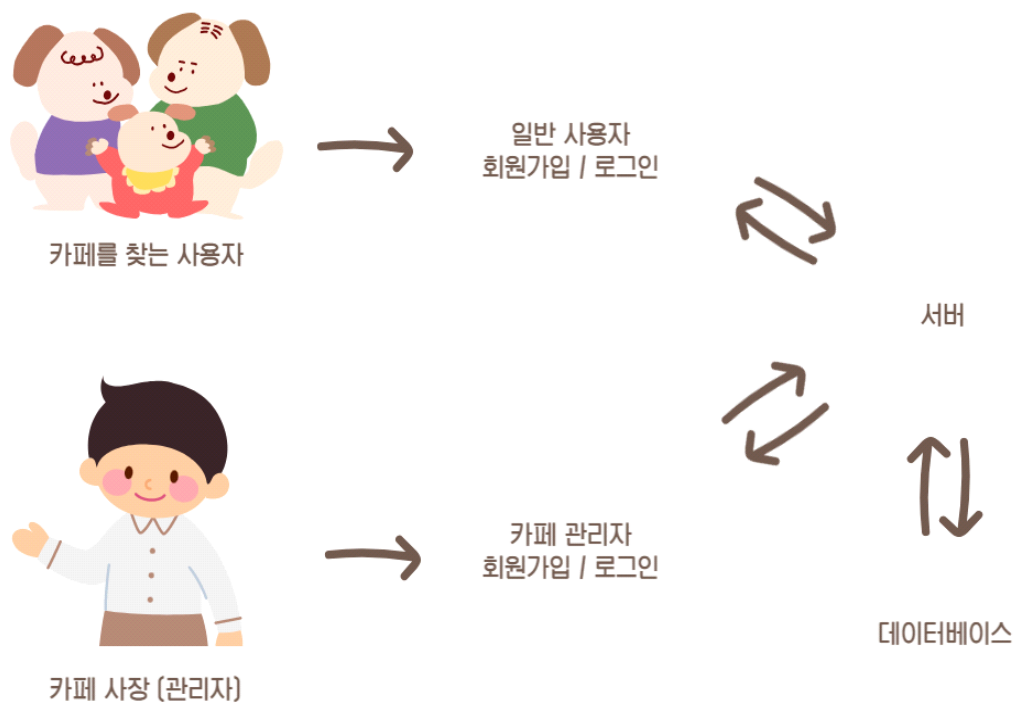
Table of Contents

| | |
|---------------------------------------|----|
| 1. 시스템 개요 | 2 |
| 1.1 프로젝트 개요 | 2 |
| 1.2 Block Diagram | 3 |
| 2. Database | 4 |
| 2.1 Table 및 ER 다이어그램 | 5 |
| 2.2 각 테이블의 자료구조 | 7 |
| 3. Security | 9 |
| 3.1 보안 점검 사항 | 9 |
| 3.2 안전한 데이터 전송 | 9 |
| 3.3 안전한 저장 | 9 |
| 3.4 사용자 인증 | 10 |
| 3.5 입력 유효성 검사 | 11 |
| 3.6 권한 부여 | 12 |
| 4. Server | 14 |
| 4.1 서버의 기능 | 14 |
| 4.2 서버 HTTP 메시지 구조 | 15 |
| 4.3 서버 요청 URL | 17 |
| 5. User Interface Specification | 19 |
| 5.1 스플래시 | 19 |
| 5.2 로그인 | 20 |
| 5.3 회원가입 | 21 |
| 5.4 홈 탭 | 23 |
| 5.5 맵 탭 | 24 |
| 5.6 프로필 탭 | 25 |

1. 시스템 개요

1.1 프로젝트 개요

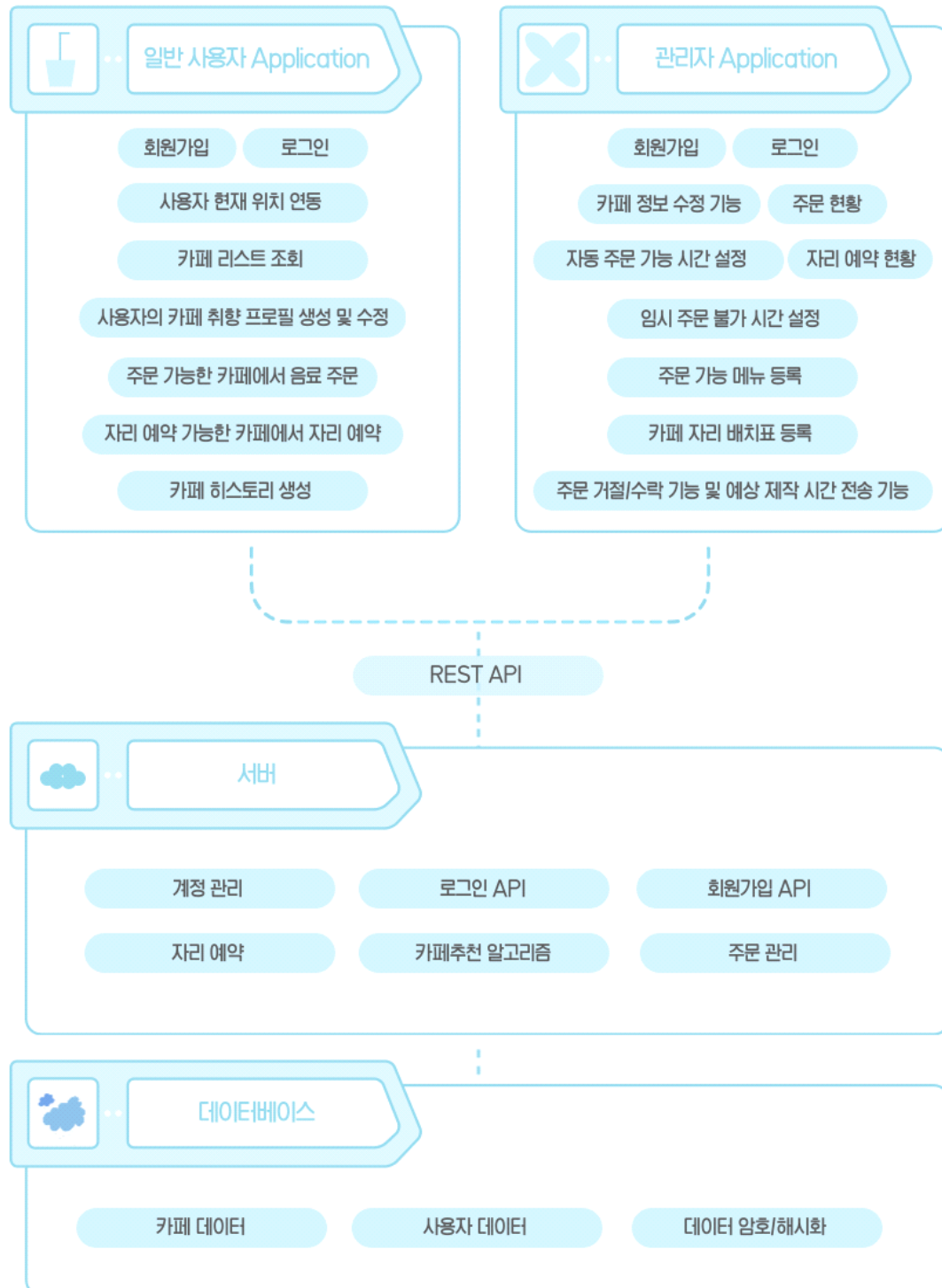
‘카페 오아시스’ 시스템은 사용자의 카페 취향 키워드를 활용한 콘텐츠 기반 카페 추천 시스템이다. 카페를 찾는 일반 사용자와 카페를 운영하고 있는 카페 사장(관리자)로 나뉘어지며, 각자 사용하는 애플리케이션이 다르다. 프로젝트의 개념도는 다음과 같다.



[그림 1] 프로젝트 전체 흐름도

1.2 Block Diagram

개발에 있어 전체 시스템 구조를 도식화한 Block diagram은 다음과 같다. 본 시스템은 크게 애플리케이션과 서버, 데이터베이스로 구분된다.



[그림 2] 전체 시스템 Block Diagram

2. Database

본 파트는 오아시스 시스템에서 요구되는 데이터베이스 설계 목록들이 기술된다. 본 DSD에서는 서버 측을 기준으로 설계된 데이터베이스를 작성한다. 현재 DSD 기준 오아시스 시스템에서는 카페 추천 기능을 중심으로 데이터베이스 설계를 목적으로 해 관계형 데이터베이스만 사용한다.

추후 카페 직원 기능이 업데이트됨에 따라 각 카페별 메뉴판 등록이나 카페의 좌석 예약, 카페 결제 시스템에 필요한 데이터는 관계형 DB로 설계하기보다 설계의 편의성, 성능 등을 고려하여 비 관계형 데이터베이스를 사용한다. 오아시스 시스템에서 카페 추천 기능을 중점으로 설계된 table 목록과 schema는 다음과 같다.

| Table 목록 | Schema |
|---------------|--|
| 사용자 | User(user_id, email, password, name, phone_no, registration_date, user_type) |
| 고객 | Customer (user_id*, nickname, age, sex) |
| 직원 | Staff(user_id*, cafe_id*, staff_position, staff_contact_info) |
| 카페 정보 | Cafe(cafe_id, cafe_name, cafe_rating, visitor_reviews, blog_reviews, cafe_phone_no, cafe_info, address, latitude, longitude) |
| 키워드 | Keyword(keyword_id, keyword_name) |
| 사용자 선호 카페 키워드 | UserKeyword(user_id*, keyword_id*) |
| 카페 키워드 | CafeKeyword (cafe_id*, keyword_id*, keyword_count) |
| 카페 평점 | CafeRating(rating_id, user_id*, cafe_id*, rating, rating_date) |
| 카페 방문 기록 | VisitHistory(visit_id, user_id*, cafe_id*, total_spend, visit_date) |

2.1 Table 및 ER 다이어그램

2.1.1 사용자 Table

사용자 테이블은 시스템의 모든 사용자에게 대한 기본 데이터를 보유한다. 여기에는 고유 식별자, 연락처 정보 및 등록 세부 정보가 포함되고 고객 또는 직원이 될 수 있는 다양한 사용자 유형을 구분한다.

2.1.2 고객 Table

고객 테이블은 고객인 사용자를 위해 사용자 테이블을 확장한다. 여기에는 별명, 나이, 성별과 같은 특정 고객 정보가 포함되고 각 고객은 user_id를 통해 사용자에게 연결된다.

2.1.3 직원 Table

직원 테이블은 다른 카페의 직원인 사용자를 위해 사용자 테이블을 확장한다. 특정 카페에서의 위치 및 연락처 정보에 대한 정보가 포함되고 각 스태프 구성원은 각각 user_id 및 cafe_id를 통해 사용자 및 카페에 연결된다.

2.1.4 카페 정보 Table

카페 정보 테이블은 개별 카페와 관련된 모든 정보를 담고 있다. 여기에는 카페 이름, 평가, 리뷰, 연락처 정보, 실제 주소, 지리적 좌표 및 관련 키워드와 같은 세부 정보가 포함된다.

2.1.5 키워드 Table

키워드 테이블은 카페와 연관될 수 있는 모든 잠재적 키워드를 저장한다. 각 키워드에는 고유 식별자가 존재한다.

2.1.6 사용자 선호 카페 키워드 Table

사용자 선호 카페 키워드 표는 사용자가 선호하는 카페 키워드로 연결된다. 이 연결은 사용자를 위한 개인화된 카페 추천에 도움이 된다.

2.1.7 카페 키워드 Table

카페 키워드 테이블은 카페를 관련 키워드에 연결하고 각 키워드의 수를 추적한다. 이것은 카페를 분류하고 추천 시스템을 개선하는 데 도움이 된다.

2.1.8 카페 평점 Table

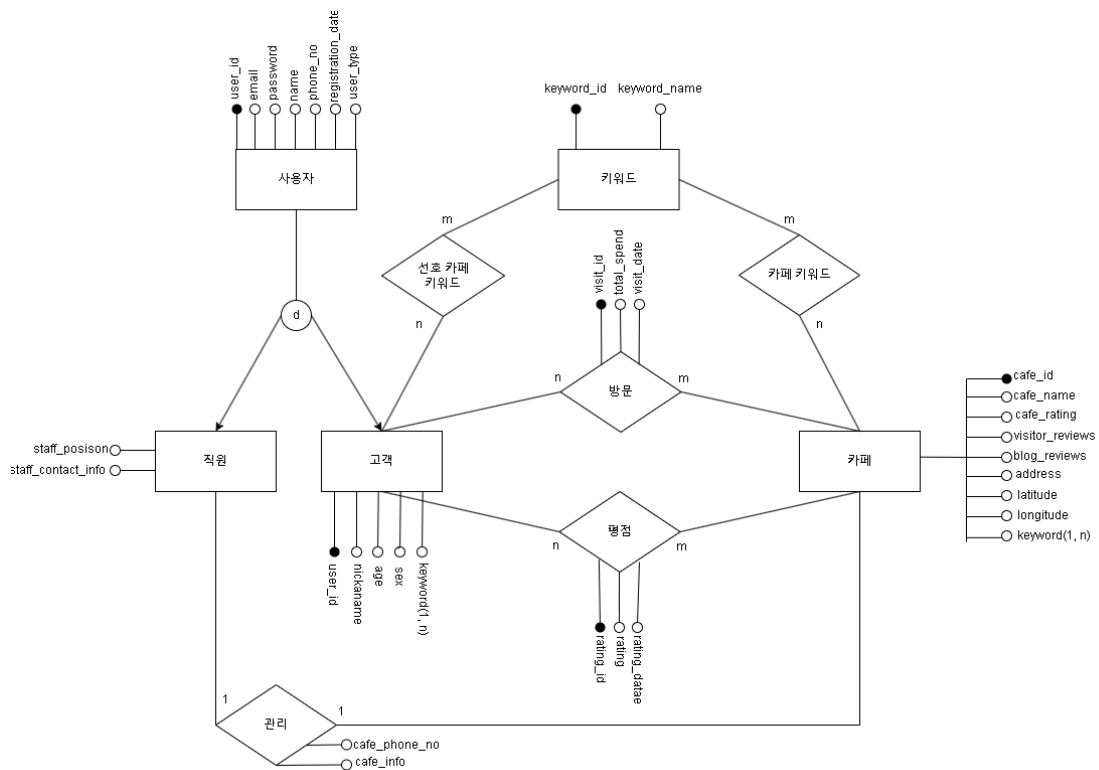
카페 평점 테이블은 각 카페에 대한 개별 사용자 평점을 저장한다. 이 데이터는 전체 카페 등급에 기여하고 사용자에게 관련 권장 사항을 제공하는 데 도움이 된다.

2.1.9 카페 방문 기록 Table

카페 방문 기록 테이블은 방문 중 총 지출과 함께 사용자의 모든 카페 방문을 기록한다. 이 데이터는 사용자 선호도 및 행동에 대한 귀중한 통찰력을 제공하며 추천 시스템을 개선하는 데 사용할 수 있다.

2.1.10 ER 다이어그램

오아시스 시스템의 서버 측 ER 다이어그램은 다음과 같다.



[그림 3] ER 다이어그램

2.2 각 테이블의 자료구조

설계된 table의 자료 형식은 아래의 표와 같다. MariaDB 기준으로 type을 정한다.

| Table Name | Attribute | Data Type |
|-------------|--------------------|--|
| User | user_id | INT |
| | email | VARCHAR(255) |
| | password | CHAR(60) |
| | name | VARCHAR(255) |
| | phone_no | VARCHAR(20) |
| | registration_date | TIMESTAMP |
| | user_type | ENUM('Customer', 'Staff') |
| Customer | user_id | INT |
| | nickname | CHAR(15) |
| | age | INT |
| | sex | ENUM('Male', 'Female') |
| Staff | user_id | INT |
| | cafe_id | INT |
| | staff_position | ENUM('General Employee', 'Manager', 'President') |
| | staff_contact_info | VARCHAR(255) |
| Cafe | cafe_id | INT |
| | cafe_name | VARCHAR(255) |
| | cafe_rating | DECIMAL(3,2) |
| | visitor_reviews | INT |
| | blog_reviews | INT |
| | cafe_phone_no | VARCHAR(20) |
| | cafe_info | TEXT |
| | address | TEXT |
| | latitude | DOUBLE(9,6) |
| | longitude | DOUBLE(9,6) |
| Keyword | keyword_id | INT |
| | keyword_name | VARCHAR(255) |
| UserKeyword | user_id | INT |
| | keyword_id | INT |
| CafeKeyword | cafe_id | INT |
| | keyword_id | INT |

| | | |
|--------------|---------------|---------------|
| | keyword_count | INT |
| CafeRating | rating_id | INT |
| | user_id | INT |
| | cafe_id | INT |
| | rating | DECIMAL(3,2) |
| | rating_date | TIMESTAMP |
| VisitHistory | visit_id | INT |
| | user_id | INT |
| | cafe_id | INT |
| | total_spend | DECIMAL(10,0) |
| | visit_date | TIMESTAMP |

3. Security

3.1 보안 점검 사항

오아시스 프로젝트를 개발함에 있어 보안 측면에서 안정적인 시스템이 구축돼야 오아시스를 사용하는 사용자가 믿고 서비스를 이용하고 개인정보 유출과 같은 이슈를 사전에 방지할 수 있기에 개발자가 설계해야 할 보안 점검 사항들이 존재한다. 서버와 클라이언트 간에 데이터를 안전하게 전송하고 저장하며 침입자들의 의도적인 해킹과 보안 공격에 대해 대응하기 위해 아래와 같이 몇 가지 고려 사항이 있다.

하지만 현재 개발하는 오아시스 프로젝트는 현재 당장으로는 시장 출시나 상업화를 위한 목적으로 설계된 제품이 아니기 때문에 필수적인 보안 사항만 고려한다.

- 안전한 데이터 전송
- 안전한 저장
- 사용자 인증
- 입력 유효성 검사
- 권한 부여

3.2 안전한 데이터 전송

‘안전한 데이터 전송’은 Android 애플리케이션과 Django 서버 간에 통신할 때 민감한 정보를 보호하고 사용자의 개인정보를 유지, 보호하는 데 중요하다. 이를 구현하기 위해 HTTPS(Hypertext Transfer Protocol Secure, 암호화 프로토콜)를 사용해 침입자가 데이터를 도청, 변조 및 중간 공격을 하지 못하도록 방지한다.

3.2.1 Https 사용 절차

- SSL/TLS 인증서 발급을 위해서 서버의 도메인 주소가 필요
- 본 프로젝트의 클라우드 서버로 사용되는 AWS EC2 서버의 IP 주소를 DNS(Domain Name System)로 변환하기 위해 가비아 사이트(<https://domain.gabia.com/>)를 통해 사용할 도메인 주소를 구매 -> www.cafeoasis.site
- 무료로 인증서를 발급하는 Let's Encrypt 인증 기관(CA)을 사용해 SSL 인증서 발급
- 안드로이드 앱에서 서버의 URL 주소를 https 프로토콜을 사용하도록 수정

3.3 안전한 저장

‘안전한 저장’은 서버와 클라이언트 각각에서 사용자의 개인 정보와 같은 민감한 데이터의 보안을 유지하며 안전하게 저장하는 것을 의미한다. 주로 데이터베이스에 데이터를 저장할 때 원래 데이터 값 그대로 저장하지 않고 암호화 과정을 거치거나 해싱 과정을 거친 뒤 데이터베이스에 저장한다.

3.3.1 데이터 암호화

AES-256 대칭 키 암호화 알고리즘을 사용해 중요하거나 민감한 데이터를 데이터베이스 또는 파

일 시스템에 저장할 때 데이터를 암호화한다. 본 오아시스 시스템에서는 회원가입 시 입력한 사용자의 연락처, 이메일 주소를 암호화 처리 후 저장한다.

3.3.2 패스워드 해싱

다른 데이터와 달리 패스워드를 해싱을 하는 이유는 서버에서 사용자의 기존 패스워드로 해독하거나 복구할 수 없게 하기 위함인데 서버에서도 해시된 암호의 해독을 방지하는 주된 이유는 다음과 같다.

- 사용자의 개인 정보 보호
- 사용자는 종종 여러 웹사이트 및 서비스에서 동일한 비밀번호를 재사용 하는 경우가 많아 공격자가 해시 알고리즘을 해독해 원래 암호를 발견할 수 있을 때 하나의 서비스 사이트에서 뿐만 아닌 추가적인 보안 문제 발생 가능
- 관리자의 악용 방지
- 서버에 접근할 수 있는 데이터베이스 관리자 또는 직원이 사용자의 암호를 해독할 수 있는 경우 자신의 권한을 오용하고 사용자 계정에 접근 가능
- 데이터 유출의 영향 완화

Django 프레임워크 내부에서는 이미 PBKDF2 해싱알고리즘을 사용해 사용자의 비밀번호를 안전하게 저장할 수 있다.

3.3.3 키 관리

‘키’라는 용어는 다양한 보안 프로토콜에서 사용되는 암호화 키를 의미한다. 안전한 방식으로 암호화 키를 관리하고 처리하는 모든 일련의 프로세스를 뜻하며 오아시스 프로젝트에서는 키 생성, 저장까지 과정을 담는다. 오아시스 시스템에서 요구되는 암호화 키는 AES 대칭 키와 API 키, JWT에 사용될 비대칭 키를 포함한다.

3.4 사용자 인증

Oasis 프로젝트의 사용자 인증은 사용자의 신원을 확인하고 시스템의 보안을 강화하며 안전하고 개인화된 사용자 경험을 보장하도록 설계된 중요한 절차이다. 사용자 인증에는 로그인 프로세스 중에 사용자가 제공한 신원 증명을 확인하는 과정이 포함된다. 또한 사용자가 app을 실행할 때마다 로그인 과정을 거쳐야 하는 번거로움을 없애고, 비밀번호와 이메일을 자주 전송하는 데 따른 잠재적인 보안 문제를 제거하기 위해 사용자 인증 프로세스에 JWT(JSON Web Tokens) 사용을 통합했다. 사용자가 app을 시작할 때 JWT가 클라이언트 측에서 자동으로 전송되고 서버는 이 JWT를 확인하여 사용자의 신원과 데이터에 대한 위조, 변조를 확인한다.

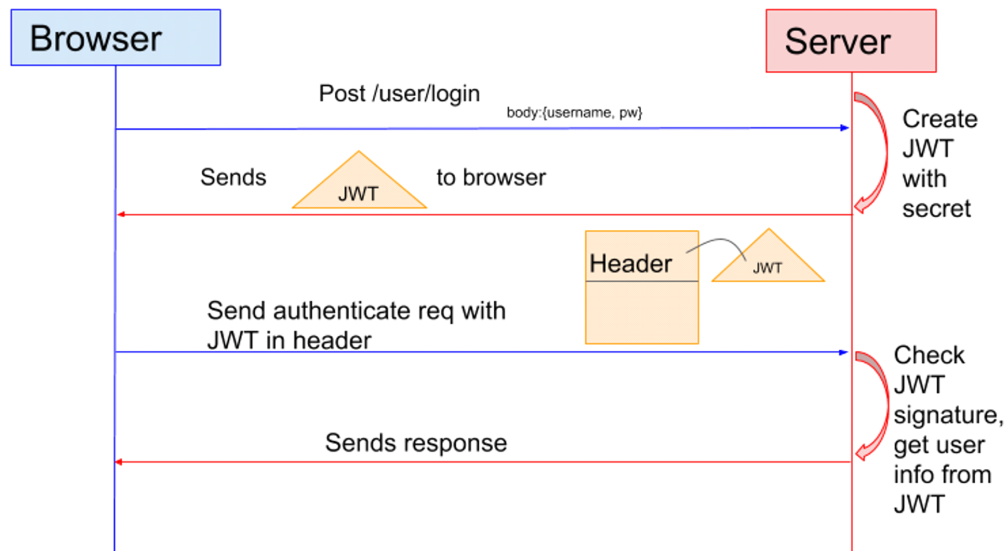
3.4.1 JWT

JWT(Json Web Token)은 웹에서 사용되는 json 형식의 토큰에 대한 표준 인증 방식이다. 주로 사용자의 인증(authentication)이나 인가(authorization) 정보를 서버와 클라이언트 간에 안전하게 통신하기 위해서 사용된다. 클라이언트가 서버로부터 인증을 받기 위한 필요한 정보들을 token

에 담아 암호화시켜 사용한다.

3.4.2 JWT 인증 과정

JWT를 이용한 클라이언트-서버 구조에서 사용자 인증, 인가 과정을 아래와 그림, 설명과 같다.



[그림 4] JWT 인증 과정

- ① 클라이언트에서 사용자의 인증 정보를 서버에 전달
- ② 서버는 인증 정보로 인증을 처리하고 JWT를 생성하여 클라이언트에 전달
- ③ 클라이언트는 JWT를 브라우저에 저장
- ④ 클라이언트는 이후 이루어지는 요청에 JWT를 이용
- ⑤ 서버는 JWT를 검증하여 인증을 처리
- ⑥ JWT가 만료되면 토큰을 재발급

3.5 입력 유효성 검사

‘입력 유효성 검사’는 사용자가 제공한 데이터가 예상 형식 및 값과 일치하는지 검증하는 과정이다. 이는 Injection Attack, XSS 등과 같은 다양한 유형의 보안 취약점을 방지한다. 실제 상업화된 제품 등급 소프트웨어에서는 항상 서버 측과 클라이언트 측 둘 다 유효성 검사가 필요하지만 오아시스 프로젝트에서는 안드로이드 측에서 유효성 검사 알고리즘을 우선으로 구현하며 추후 시스템이 업데이트를 거침에 따라 서버 측에서 또한 유효성 검사 과정을 구현한다.

유효성 검사는 사용자가 입력한 데이터에 대해 검사 과정을 거치므로 오아시스 app에서 사용자가 입력 or 조작한 모든 데이터에 대해 유효성 검사가 필요하다. 다음 표는 안드로이드 app에서 유효성 검사가 필요한 항목들과 검사 방식을 요약한 표이다.

| | 검사 항목 | 검사 방식 |
|--------|-----------|---|
| 로그인 | 이메일 | Android의 Patterns.EMAIL_ADDRESS.matcher(input).matches() 를 사용하여 입력한 이메일 주소가 유효한지 확인 |
| | 패스워드 | 오아시스 애플리케이션의 비밀번호 정책(3-? 비밀번호 정책 참조) 및 기타 규칙으로 사용자가 입력한 패스워드가 유효한지 확인 |
| 회원가입 | 이메일 | 로그인 과정과 동일 |
| | 패스워드 | 로그인 과정과 동일 |
| | 이름 | 사용자가 입력한 이름이 비어 있는 문자열 “ ” 인지 확인 |
| | 닉네임 | 사용자가 입력한 닉네임이 비어 있는 문자열 “ ” 인지 확인 특수 문자를 포함하지 않고 길이가 15자 이하인 문자열인지 확인 |
| | 연락처 | Android의Patterns.PHONE.matcher(input).matches()를 사용해 사용자가 입력한 연락처가 비어 있지 않은지 확인하고 유효한 숫자 형식인지 확인 |
| | 연령 | 사용자가 입력한 연령 값이 유효한 정수이고 알맞은 범위에 속하는지 확인 |
| | 성별 | 사용자가 선택한 성별 옵션 값이 “남자” 이거나 “여자” 값 중 하나인지 확인 |
| 카페 프로필 | 선호 카페 키워드 | <ul style="list-style-type: none"> - 사용자가 선택한 카페 키워드가 사전에 시스템에서 허용된 옵션 중 하나인지 확인 - 선택한 총 키워드 수가 5개를 초과하지 않는지 확인 |

3.6 권한 부여

'권한 부여'는 사용자가 인증된 후 자신이 사용할 수 있는 서비스의 범위를 결정하는 과정을 의미한다. 이는 사용자가 수행하려는 서비스가 해당 사용자에게 허용되는 서비스인지 확인하는 것으로, 잘못된 사용자가 시스템에서 수행하지 못하도록 하는 데 중요하다.

오아시스 프로젝트에서는 사용자(카페 손님)와 카페 관리자 사이에 권한 차이를 두어야 한다. 아래는 간단히 사용자와 카페 관리자 사이에 권한 차이를 분류한 이다.

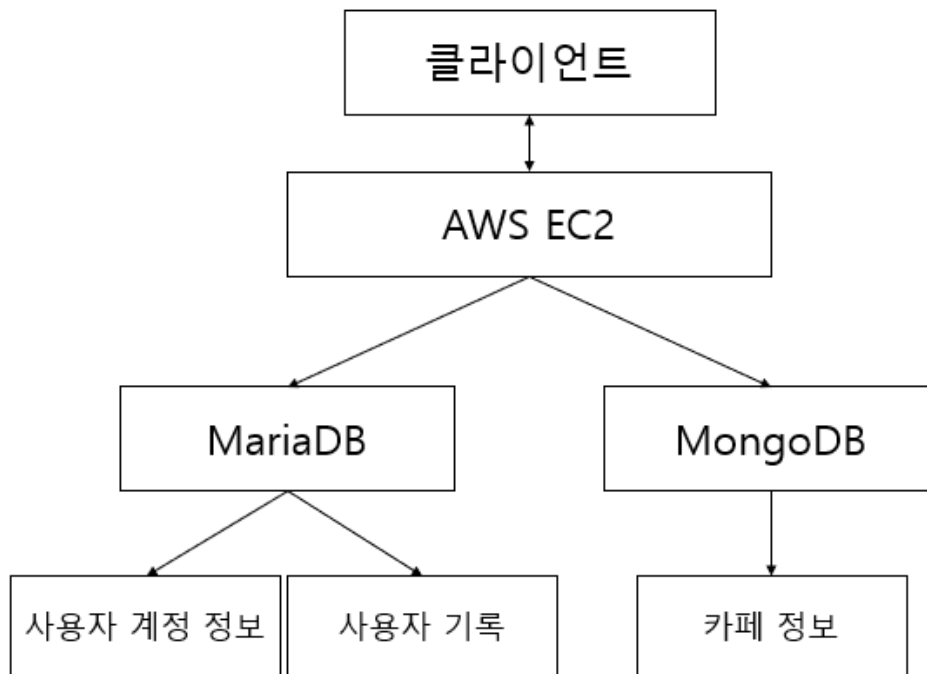
| | 일반 사용자 권한 | 카페 관리자 권한 |
|---------------|-------------------------------|---------------------------------|
| 1. 카페 정보 액세스 | 자세한 카페 정보를 검색하고 조회 가능 | 카페 정보를 추가 및 수정 가능 |
| 2. 카페 예약 및 결제 | 마음에 드는 카페 자리를 예약하거나 메뉴를 결제 가능 | 카페에서 이루어진 모든 예약을 확인하고 수용 인원에 따라 |

| | | |
|-------------|-------------------------------------|--|
| | | 확인하거나 거부 가능 |
| 3. 예약 현황 | 고객이 예약한 예약 상태와 업데이트 또는 변경 사항을 조회 가능 | 카페 전체의 예약 상태를 업데이트하고 변경 사항을 사용자에게 전달 가능 |
| 4. 프로필 관리 | 카페 프로필 관리에 접근 불가 | 사진, 설명, 리뷰 응답 등 카페 프로필을 관리 가능 |
| 5. 분석 및 보고서 | 카페 성과 분석 및 보고서에 액세스 불가 | Oasis 시스템 내에서 카페의 성과와 관련된 분석 및 보고서에 액세스 가능 |

4. Server

4.1 서버의 기능

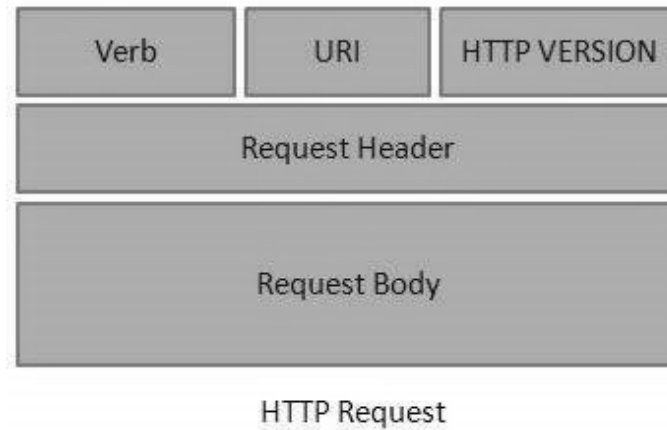
서버는 아마존 AWS EC2 linux에서 Django를 이용하여 웹 서버를 구축하여 사용한다. Django REST Framework를 사용하여 URL 별로 메소드를 처리하며 JSON 형태의 데이터를 클라이언트에서 수신받고 서버는 클라이언트에 요청한 값에 대해 처리한 값을 클라이언트에 송신한다. 그림은 개략적인 서버의 구조를 나타낸다.



[그림 5] Server

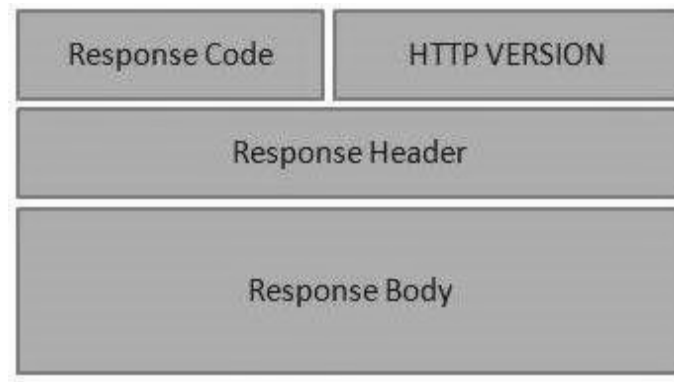
4.2 서버 HTTP 메시지 구조

클라이언트와 서버는 HTTP 프로토콜을 이용하여 통신한다. 아래 그림은 HTTP 요청/응답 메시지의 구조 및 설명이다.



[그림 6] HTTP 요청 메시지 구조

| 필드 | 내용 |
|----------------|--|
| Verb | HTTP 메소드를 정의(ex. GET, POST, DELETE, PUT, etc.) |
| URI | 통합 자원 식별자(Uniform Resource Identifier)로 웹 기술에서 사용하는 논리적 또는 물리적 리소스를 식별하는 고유한 문자열 시퀀스 |
| HTTP Version | HTTP 버전을 기술(ex. HTTP v1.1) |
| Request Header | 키-값 쌍의 메타데이터를 담은 HTTP 요청 메시지 |
| Request Body | 메시지 내용 |



HTTP Response

[그림 7] HTTP 응답 메시지 구조

| 필드 | 내용 |
|-----------------|---------------------------------------|
| Response Code | 받은 요청에 대한 서버 상태를 나타냄 (ex. 404 or 200) |
| HTTP Version | HTTP 버전을 기술 (ex. HTTP v1.1) |
| Response Header | 키-값 쌍의 메타데이터를 담은 HTTP 요청 메시지 |
| Response Body | 메시지 내용 |

4.3 서버 요청 URL

서버의 IP는 유동적이기에 <server ip>로 기술한다.

| URL | 기능 |
|------------------------------------|--|
| <server ip> /users/login | ID와 PW를 입력받아 서버에 로그인 요청을 보낸다. 서버는 user_accounts 데이터베이스에서 확인하고 일치하는 ID와 PW이 존재할 때만 상태 200을 송신한다. |
| <server ip> /users/signup | ID와 PW, e-mail 등을 입력받아 서버에 가입 요청을 보낸다. 서버는 user_accounts 데이터베이스에서 중복되는 ID를 확인하고 가입이 가능할 때만 상태 200을 송신한다. |
| <server ip> /users/findid | e-mail을 입력받아 서버에 아이디 찾기 요청을 보낸다. 서버는 user_accounts 데이터베이스에서 일치하는 e-mail이 존재할 때만 상태 200과 사용자 ID를 송신한다. |
| <server ip> /users/findpw | ID와 e-mail을 입력받아 서버에 비밀번호 찾기 요청을 보낸다. 서버는 user_accounts 데이터베이스에서 일치하는 e-mail이 존재할 때만 상태 200을 송신한다. 이후 비밀번호를 signup을 이용해 비밀번호를 재설정한다. |
| <server ip> /users/profile | 사용자의 개인정보 혹은 선호 키워드를 수정할 때 사용한다. 서버는 ID를 확인하여 User 칼럼을 찾고 응답의 값을 확인하여 중복되지 않고 유효한 값일 때에만 200을 송신한다. |
| <server ip> /cafe/search | 카페에 대한 검색을 수행한다. 서버는 카페 정보 데이터베이스에서 Cafe_ID 혹은 카페명으로 검색을 수행하고 200과 카페 정보를 응답으로 송신한다. |
| <server ip> /cafe/getinfo | 카페에 대해 정보를 얻는다. 서버는 카페 정보 데이터베이스에서 Cafe_ID 그리고 카페명으로 검색을 수행하고 200과 카페 정보를 응답으로 송신한다. |
| <server ip> /cafe /recommend | 카페에 대한 추천을 수행한다. 서버는 카페 정보 데이터베이스에서 사용자 선호 키워드, 사용자 위치와 카페 간의 거리 차이, 추천할 카페 개수 등의 변수로 AI 기반 추천을 수행하고 200과 카페 정보를 응답으로 송신한다. |

| URL | 기능 |
|-------------------------------------|--|
| <server ip> /cafe/edit | 소유주가 관리 앱 혹은 웹에서 카페 정보(프로필)를 수정할 때 사용한다. 서버는 클라이언트에서 요청을 수신받고 클라이언트의 권한을 확인하고 유효하다면 서버의 카페 정보 데이터베이스에서 수정을 반영하고 유효하지 않다면 오류 메시지를 반환한다. |
| <server ip> /cafe /edit_order | 소유주가 관리 앱 혹은 웹에서 주문 정보를 수정할 때 사용한다. 서버는 클라이언트에서 요청을 수신받고 클라이언트의 권한을 확인하고 유효하다면 카페 메뉴 데이터베이스에서 수정을 반영한다. |
| <server ip> /receptit/verify | 사용자가 영수증 리뷰를 등록할 때 영수증이 유효한지 판별하기 위해 사용한다. 서버는 학습된 AI 모델을 기반으로 영수증이 유효한지 판별하고 유효하다면 클라이언트에게 200을 송신한다. |
| <server ip> /receptit/upload | 사용자가 영수증 리뷰를 등록할 때 사용한다. 서버는 영수증과 리뷰를 리뷰 데이터베이스에 저장하고 200을 송신한다. |
| <server ip> /order/send | 사용자가 카페에 주문할 때 사용한다. 서버는 주문을 주문 데이터베이스에 저장하고 주문받은 카페에 알린다. |
| <server ip> /order/status | 소유주가 관리 앱 혹은 웹에서 주문 내역을 확인할 때 사용한다. 서버는 주문 데이터베이스에서 유효한 주문을 클라이언트에게 내역을 송신한다. |

5. User Interface Specification

5.1 스플래시



오아시스라는 이름은
여행자들이 뜨거운 사막에서
신선한 물과 그늘을 찾듯이,
카페를 찾는 이들에게 좋은 음료와
새로운 경험을 선사한다는 의미를 담고 있다.

[그림 8] 스플래시 화면

애플리케이션을 실행하면, 스플래시 화면이 나온다. 로그인이 되어 있지 않은 상태라면 로그인 화면으로, 로그인이 된 상태라면 홈 화면으로 넘어간다.

5.2 로그인



이메일

예) abc@gmail.com

비밀번호

password

☐

이메일 기억

☐

로그인 유지

로그인

회원가입



카카오 로그인

N Log In

[그림 9] 로그인 화면

이메일과 비밀번호를 입력하고 로그인 버튼을 누르면 홈 화면으로 넘어간다. 회원가입을 누르면 회원가입 화면으로 넘어간다.

5.3 회원가입



회원가입

이메일*

예)abc@gmail.com

이메일 확인*

예)abc@gmail.com

- ☐ 아래 약관에 모두 동의합니다.
- ☐ 이용약관 필수 동의 [자세히 보기](#)
- ☐ 개인정보 처리방침 필수 동의 [자세히 보기](#)
- ☐ 위치정보 이용약관 필수 동의 [자세히 보기](#)

다음

[그림 10] 회원가입 화면(1)

데이터베이스에 사용자 정보를 등록하기 위해 사용자는 이메일을 입력한다. 약관에 모두 동의했고, 데이터베이스에 중복된 이메일이 없다면 다음 버튼을 눌렀을 때 비밀번호 설정 페이지로 넘어간다.

←

회원가입

비밀번호*

영문, 숫자 조합 8~16자

비밀번호 확인*

영문, 숫자 조합 8~16자

←

회원가입

이름*

닉네임

나이

숫자

전화번호

010-1234-5678

다음

완료

[그림 11] 회원가입 화면(2)

비밀번호는 한 번 입력하도록 한 후, 확인을 위하여 한 번 더 입력받는다. 두 입력된 값이 동일해야 성공적으로 회원가입이 가능하다. 그리고 이름, 닉네임, 나이, 전화번호를 추가로 등록한다.

5.4 홈 탭



[그림 12] 홈 탭 화면

홈 화면에서는 카페 순위, 추천 카페, 북마크 카페 등을 확인할 수 있다.

5.5 맵 탭

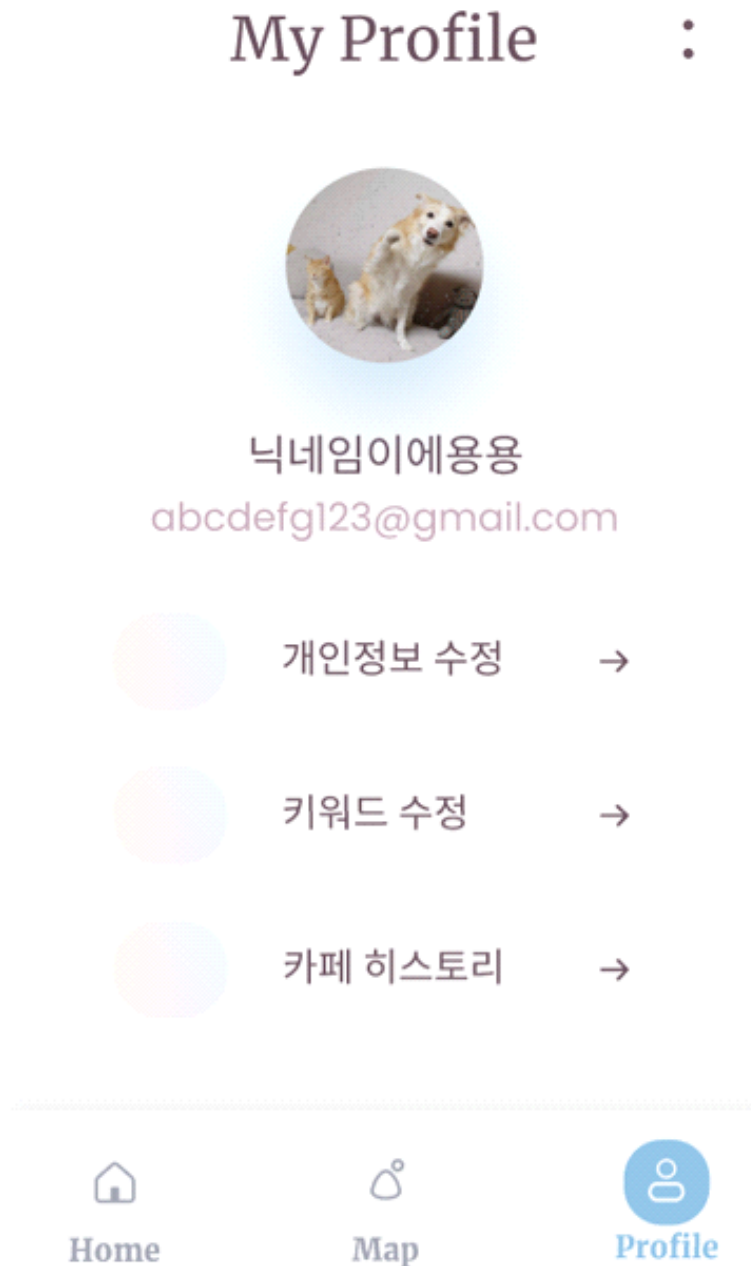
Oasis Map :



[그림 13] 맵 화면

맵 화면에서는 지도에서 카페를 찾아볼 수 있다. 내 위치 근처의 카페를 찾아볼 수도 있고, 마커 이벤트를 통해 카페의 정보도 찾아볼 수 있다.

5.6 프로필 탭



[그림 14] 프로필 탭 화면 (1)

프로필 탭의 초기화면이다. 내 닉네임, 이메일을 확인할 수 있으며, 개인정보 수정, 키워드 수정, 카페 히스토리 페이지로 이동할 수 있다.



개인정보 수정

이름

닉네임

나이

전화번호

#관심 키워드 수정

#따뜻함

키워드는 나중에 다시 수정할 수 있어요!

완료

완료

[그림 15] 프로필 탭 화면 (2)

좌측은 프로필 탭의 개인정보 수정 화면이다. 이름, 닉네임, 나이, 전화번호를 수정할 수 있다. 우측은 키워드 수정 화면이다. 관심 키워드를 수정할 수 있다.



Cafe History :



영수증 인식



히스토리 수정



Home



Map



Profile

[그림 16] 프로필 탭 화면 (3)

프로필 탭의 카페 히스토리 화면이다. 영수증 인식과 히스토리 수정 페이지로 이동할 수 있다.



영수증 인식

사진 촬영하기

사진 불러오기



내 히스토리

완료

완료

[그림 17] 프로필 탭 화면 (4)

좌측은 영수증 인식 화면이고, 영수증 사진 촬영 또는 영수증 사진 불러오기를 통하여 카페 히스토리에 사용자가 이용한 카페를 등록할 수 있다. 우측은 사용자의 히스토리 화면이다.