

# 모바일 프로그래밍 팀 프로젝트



과 목	모바일 프로그래밍
담당 교수	최정열 교수
제출 일자	2024.06.10
팀장	20200050 박준범
팀원	20200837 박종민, 20190830 박재혁
팀명	코드웨이브

# 목 차

1. 개요.....	3
1.1 선정 배경 및 필요성.....	3
1.2 기존 앱과의 차별점 .....	4
2. 앱의 주요 기능.....	5
2.1 휴게소 리뷰.....	5
2.2 음식 메뉴 추천 / 비추천 .....	6
2.3 휴게소 정보 제공.....	7
2.4 로그인 / 회원가입 .....	8
2.5 휴게소 지도를 활용한 검색 및 필터링 .....	9
2.6 즐겨찾기 기능 .....	10
2.7 DB .....	11
3. 화면 설계서.....	14
3.1 종합 관계도.....	14
3.2 로그인 화면.....	16
3.3 회원가입 화면.....	17
3.4 메인 화면 .....	18
3.5 휴게소 정보 화면 .....	19
3.6 리뷰 정보 화면 .....	20
3.7 리뷰 작성 화면.....	21
3.8 음식 메뉴 화면.....	22
3.9 즐겨찾기 화면 .....	23
4. 시스템 설계서.....	24
4.1 클래스 및 메소드.....	24
4.2 DB 설계.....	26
4.3 외부 API 연동.....	28
5. 설계 제한 요소.....	29
6. 향후 계획.....	30
7. 업무 분장 및 팀원 공헌도.....	31
8. 참고문헌.....	32

## 표 목 차

〈표 3-1〉 로그인 화면 위젯 및 기능.....	15
〈표 3-2〉 회원가입 화면 위젯 및 기능.....	16
〈표 3-3〉 메인 화면 위젯 및 기능.....	17
〈표 3-4〉 휴게소 정보화면 위젯 및 기능.....	18
〈표 3-5〉 리뷰 정보 화면 위젯 및 기능.....	19
〈표 3-6〉 리뷰 작성 화면 위젯 및 기능.....	20
〈표 3-7〉 음식 메뉴 화면 위젯 및 기능.....	21
〈표 3-8〉 즐겨찾기 화면 위젯 및 기능.....	22

## 그 림 목 차

〈그림 1-1〉 리뷰로드.....	3
〈그림 2-1〉 리뷰 화면.....	5
〈그림 2-2〉 리뷰 작성 화면.....	5
〈그림 2-3〉 음식 메뉴.....	6
〈그림 2-4〉 휴게소 정보.....	7
〈그림 2-5〉 로그인 데이터 흐름.....	8
〈그림 2-6〉 지도와 필터링.....	9
〈그림 2-7〉 restarea.php.....	11
〈그림 2-8〉 메인화면 데이터 흐름.....	13
〈그림 3-1〉 전체 화면 관계도.....	14
〈그림 3-2〉 로그인 화면.....	15
〈그림 3-3〉 회원가입 화면.....	16
〈그림 3-4〉 메인 화면.....	17
〈그림 3-5〉 휴게소 정보 화면.....	18
〈그림 3-6〉 리뷰 정보 화면.....	19
〈그림 3-7〉 리뷰 작성 화면.....	20
〈그림 3-8〉 음식 메뉴 화면.....	21
〈그림 3-9〉 즐겨찾기 화면.....	22
〈그림 4-1〉 DB 테이블.....	27

## 1. 개요



〈그림 1-1〉 리뷰로드

### 앱 제목 : 리뷰 로드

#### 1.1 선정 배경 및 필요성

여행 중 휴게소는 매우 중요한 역할을 한다. 휴게소는 여행자들이 잠시 쉬며 음식을 먹고, 화장실을 이용하며, 주유를 하는 등 다양한 서비스를 제공한다. 그러나 모든 휴게소가 동일한 품질을 제공하지 않으며, 휴게소의 시설이나 청결 상태는 천차만별이다. 따라서 여행자들은 휴게소 선택에 있어 만족감을 느낄 수도 있고, 실망할 수도 있다.

이러한 문제점을 해결하기 위해 “리뷰로드”라는 앱을 개발하기로 하였다. 휴게소 리뷰 및 추천 기능을 갖춘 앱을 개발하기로 했다. 이 앱은 사용자들이 휴게소에 대한 리뷰를 작성하고, 다른 사용자들이 이를 확인하여 최적의 휴게소를 선택할 수 있도록 도와준다. 이를 통해 여행자들이 보다 편리하고 만족스러운 휴게소 이용 경험을 가질 수 있도록 하는 것이 목표이다.

## 1.2 기존 앱과의 차이점

### 기존 앱: 고속도로 휴게시설

기존 앱인 ‘고속도로 휴게시설’은 한국도로공사에서 제공하는 공공 데이터를 활용하여 전국 고속도로의 휴게시설 정보를 제공한다. 주요 기능으로는 고속도로 노선별 휴게소, 주유소, 졸음쉼터 검색, 휴게소별 대표 메뉴, 편의시설, 주유소 가격 정보 제공 등이 있다.

개발하고자 하는 리뷰로드와는 다음과 같은 차별성이 존재한다.

#### ① 리뷰 기능 추가

- 기존 앱에 없는 휴게소 리뷰 작성 기능을 추가하여 사용자들이 휴게소의 음식, 서비스 품질, 시설 상태 등에 대한 평가를 남길 수 있다.
- 다른 사용자들의 리뷰를 통해 보다 신뢰할 수 있는 정보를 얻을 수 있다.

#### ② 음식 메뉴 추천 / 비추천 기능

- 휴게소 음식 메뉴에 대한 추천 및 비추천 기능을 추가하여 사용자들이 보다 맛있는 음식을 선택할 수 있도록 선택을 도움

#### ③ 사용자 경험 개선

- 직관적이고 사용하기 쉬운 인터페이스를 제공하여 사용자가 쉽게 휴게소 정보를 검색하고 필터링 할 수 있다.
- 지도 기능을 통해 사용자들이 주변 휴게소를 쉽게 찾을 수 있도록 도움

#### ④ 회원 기능 : 로그인

- 로그인 및 회원가입 기능을 추가하여 개인 맞춤형 서비스를 제공함
- 사용자들이 자신의 리뷰와 추천 내역을 관리할 수 있음

#### ⑤ 실시간 정보 제공

- 휴게소 편의시설 정보 및 휴게소별 최신 리뷰를 실시간으로 제공하여 항상 최신 정보를 유지함

리뷰로드 앱은 사용자들이 보다 편리하고 만족스러운 여행을 할 수 있도록 돕는 것을 목표로 하며, 기존의 단순 정보 제공을 넘어서 사용자간의 소통과 신뢰를 기반으로 한 플랫폼을 제공함.

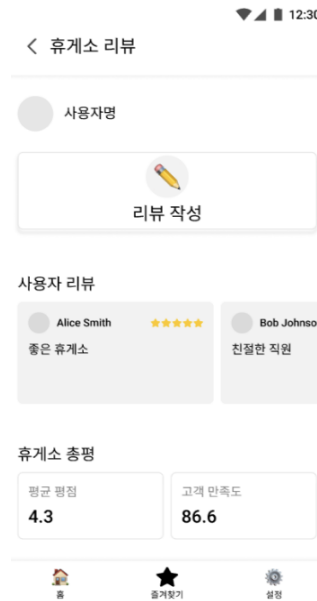
## 2. 앱의 주요 기능

### 2.1 휴게소 리뷰

- 사용자는 방문한 휴게소에 대한 리뷰를 작성하고 다른 사용자의 리뷰를 확인 가능.
- 리뷰를 통해 휴게소의 장단점을 파악하고, 더 나은 선택지 제공
- 리뷰 작성 시 별점, 텍스트 리뷰, 사진 첨부 기능을 제공



〈그림 2-1〉 리뷰 화면



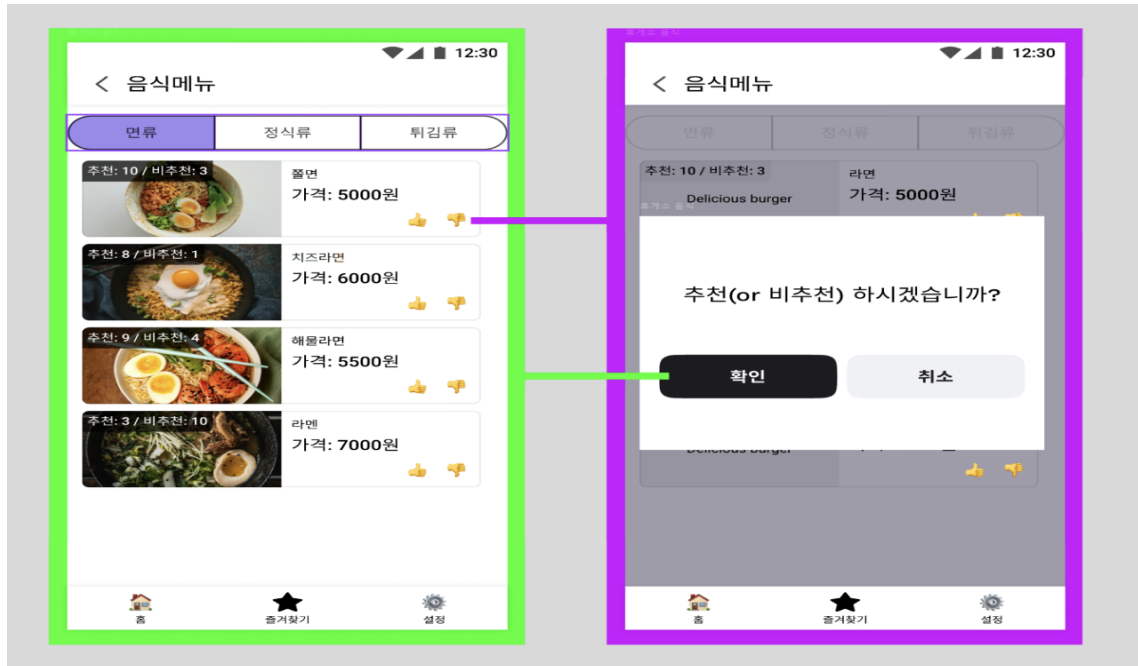
〈그림2-2〉 리뷰 작성 화면

그림<2-1>을 보면 휴게소에 대한 평점과 사용자들이 작성한 리뷰를 보여준다 리뷰는 DB에서 데이터를 가져와 출력한다.

그림<2-2>는 사용자가 별점 선택과 리뷰를 작성할 수 있게 하였고 사진이 첨부를 가능하게 하여 다양한 리뷰가 가능하게 한다. 완료 버튼을 누르면 DB에 작성한 리뷰가 저장된다.

## 2.2 음식 메뉴 추천 / 비추천

- 각 휴게소의 음식 메뉴에 대해 추천하거나 비추천할 수 있는 기능을 제공
- 사용자들이 다른 사람들이 추천하는 메뉴를 확인하고 선택의 질을 높여 줌.



<그림 2-3> 음식 메뉴

<그림 2-3>을 보면 음식 사진 왼쪽 상단에 추천 수와 비추천 수가 나와있어 사용자들에게 제공하고, 메뉴 오른쪽 아래에 손 모양을 통해 추천/비추천을 선택할 수 있다.

## 2.3 휴게소 정보 제공

- 각 휴게소의 위치, 편의시설, 운영시간 등의 상세 정보를 제공
- 



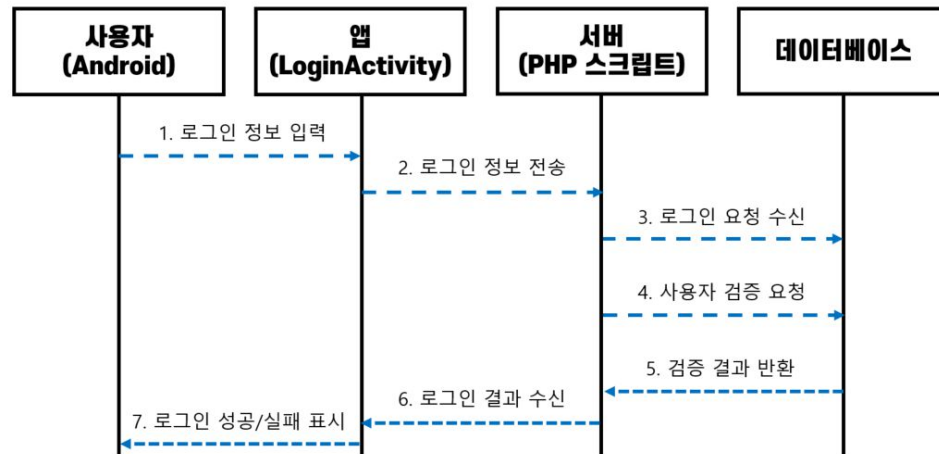
<그림 2-4> 휴게소 정보

그림<2-4>처럼 선택한 휴게소에 유가 정보, 대표 음식 메뉴, 휴게소 평가 통계, 편의시설에 대한 내용이 제공한다. 해당 내용들은 모두 DB에서 데이터를 가져와 제공한다.



## 2.4 로그인/회원가입

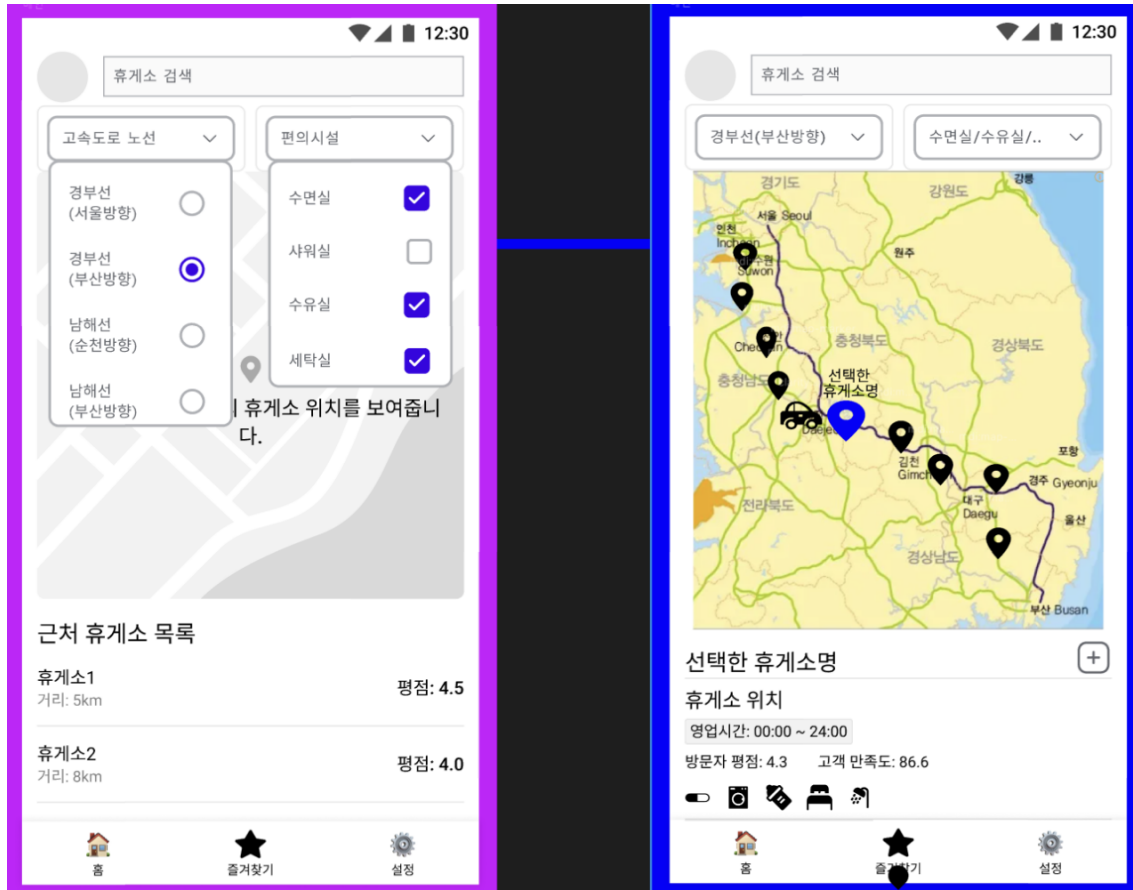
- 로그인 기능을 추가하여 사용자의 데이터를 저장
- 리뷰 및 추천을 로그인한 사용자만 가능하게 하여 정보에 대한 신뢰성 보장
- PHP를 활용하여 DB와 연결하여 로그인과 회원가입 기능 실행



〈그림 2-5〉 로그인 데이터 흐름

## 2.5 휴게소 지도를 활용한 검색 및 필터링

- 사용자가 원하는 조건에 맞춰 휴게소를 검색할 수 있는 기능을 제공
- 필터링 기능을 통한 원하는 휴게소를 찾는 시간을 줄여 줌.
- 지도 화면을 통해 주변 휴게소의 위치나 원하는 휴게소의 위치를 쉽게 찾을 수 있음.



〈그림 2-6〉 지도와 필터링

〈2-5〉에서 보이는 것과 같이 고속도로 노선과 편의시설을 선택하여 필터링을 하고 지도를 통해 그에 해당하는 결과값만을 보여주는 기능

## 2.6 즐겨찾기 기능

- 관심 있는 휴게소를 즐겨찾기에 추가하여 쉽게 접근할 수 있도록 함.
- 즐겨찾기 화면으로 이동하면 즐겨찾기 한 휴게소를 확인할 수 있고 즐겨찾기 해제도 가능하게 하여 편의성을 올림.

## 2.7 DB

- PHP를 활용하여 데이터 베이스 활용
- 데이터 베이스 구성은 MySQL을 사용하여 한다.  
메인 화면에서의 데이터 베이스를 활용.

```
<?php
header('Content-Type: application/json');

// 데이터베이스 연결 설정
$host = 'localhost';
$db = 'restarea31';
$user = 'restarea31'; // 데이터베이스 사용자 이름
$pass = 'tjdrufcozjarhd!'; // 데이터베이스 비밀번호

$dsn = "mysql:host=$host;dbname=$db;charset=utf8mb4";
$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES   => false,
];

try {
    $pdo = new PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    echo json_encode(['error' => 'Database connection failed: ' . $e->getMessage()]);
    exit;
}

$route = isset($_GET['route']) ? $_GET['route'] : '';
$facility = isset($_GET['facility']) ? $_GET['facility'] : '';
$search = isset($_GET['search']) ? $_GET['search'] : '';

$data = [];

try {
    $query = "SELECT * FROM restarea_combined WHERE 1=1";
    $params = [];

    if (!empty($route)) {
        $query .= " AND raRoute LIKE ?";
        $params[] = '%' . $route . '%';
    }

    if (!empty($facility)) {
        $query .= " AND raFacilities LIKE ?";
        $params[] = '%' . $facility . '%';
    }

    if (!empty($search)) {
        $query .= " AND raName LIKE ?";
        $params[] = '%' . $search . '%';
    }

    $stmt = $pdo->prepare($query);
    $stmt->execute($params);
    $data = $stmt->fetchAll();
} catch (\PDOException $e) {
    echo json_encode(['error' => 'Query failed: ' . $e->getMessage()]);
    exit;
}

echo json_encode($data);
?>
```

〈그림2-7〉 restarea.php

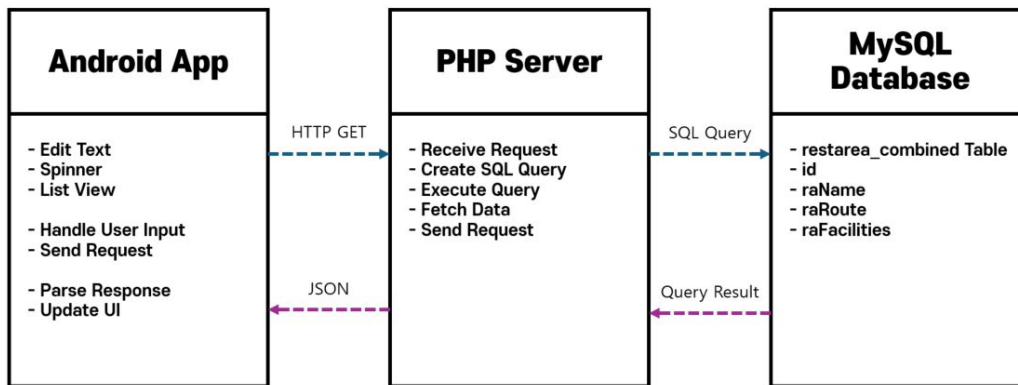
- ① 데이터베이스 연결 정보 설정
  - 데이터베이스 접속하기 위한 서버주소, 사용자이름, 비밀번호, 데이터베이스 이름을 설정
- ② 데이터베이스 연결 생성 및 확인
  - 설정한 정보를 활용하여 데이터베이스에 연결하고, 연결이 성공적으로 이루어졌는지 확인
- ③ GET 요청으로부터 데이터 받기
  - 사용자가 선택한 경로, 편의시설, 검색어를 받음
- ④ SQL 쿼리 작성
  - 입력 받은 정보를 기반으로 휴게소 정보를 조회하는 SQL 쿼리를 작성
- ⑤ 쿼리 실행 및 결과 확인
  - SQL 쿼리를 실행하고 결과를 확인하여 JSON 형태로 반환
  - 결과 반환
  - 검색된 휴게소 정보를 JSON 형태로 앱에 반환
- ① 연결 종료
  - 데이터베이스 연결을 종료함

‘MainActivity’는 사용자가 앱에서 경로, 편의시설, 검색어를 사용하여 휴게소 정보를 검색하고, 지도를 통해 위치를 확인할 수 있는 기능을 제공함. 또한 검색된 휴게소의 상세 정보를 확인할 수 있다

- 주요 컴포넌트
  - ① 사용자 인터페이스 (UI)
 

activity\_register.xml 파일에 정의된 로그인 화면 구성요소

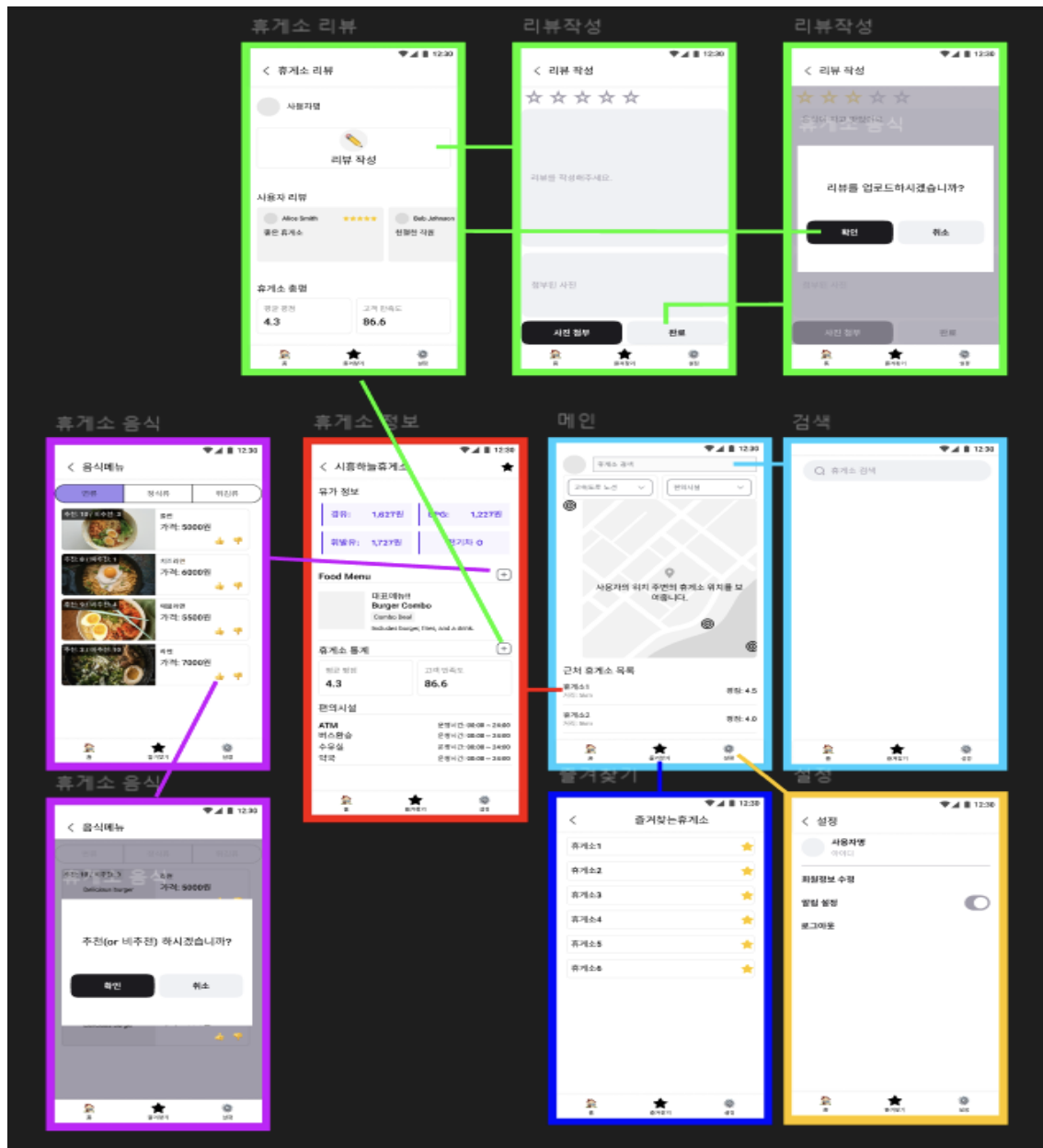
    - 경로 선택 스피너 (Spinner)
    - 편의시설 선택 스피너 (Spinner)
    - 검색어 입력 텍스트박스 (EditText)
    - 검색 결과 리스트뷰 (ListView)
    - 지도 (GoogleMap)
  - ② 서버와의 통신
    - 휴게소 정보를 검색하기 위해 ‘Volley’ 라이브러리를 사용하여 서버에 요청을 보냄
    - 서버 URL : ‘https://restarea31.mycafe24.com/db/restareaList.php’
  - ③ 데이터베이스
    - 휴게소 정보가 저장된 데이터베이스
    - 경로, 편의시설, 검색어에 따라 휴게소 정보를 조회함



〈그림 2-8〉 메인화면 데이터 흐름

- ① 사용자가 앱에서 검색어를 입력하거나 스피너에서 옵션을 선택함
- ② 앱은 사용자의 입력에 따라 HTTP GET요청을 PHP서버로 전송함
- ③ PHP서버는 요청을 받아 데이터베이스 쿼리를 생성하고, MySQL 데이터베이스에서 데이터를 검색함
- ④ MySQL 데이터베이스는 요청된 데이터를 반환하고,  
PHP서버는 데이터를 JSON 형식으로 변환
- ⑤ 앱은 PHP서버로부터 JSON 응답을 받아 사용자 인터페이스에 데이터를 표시

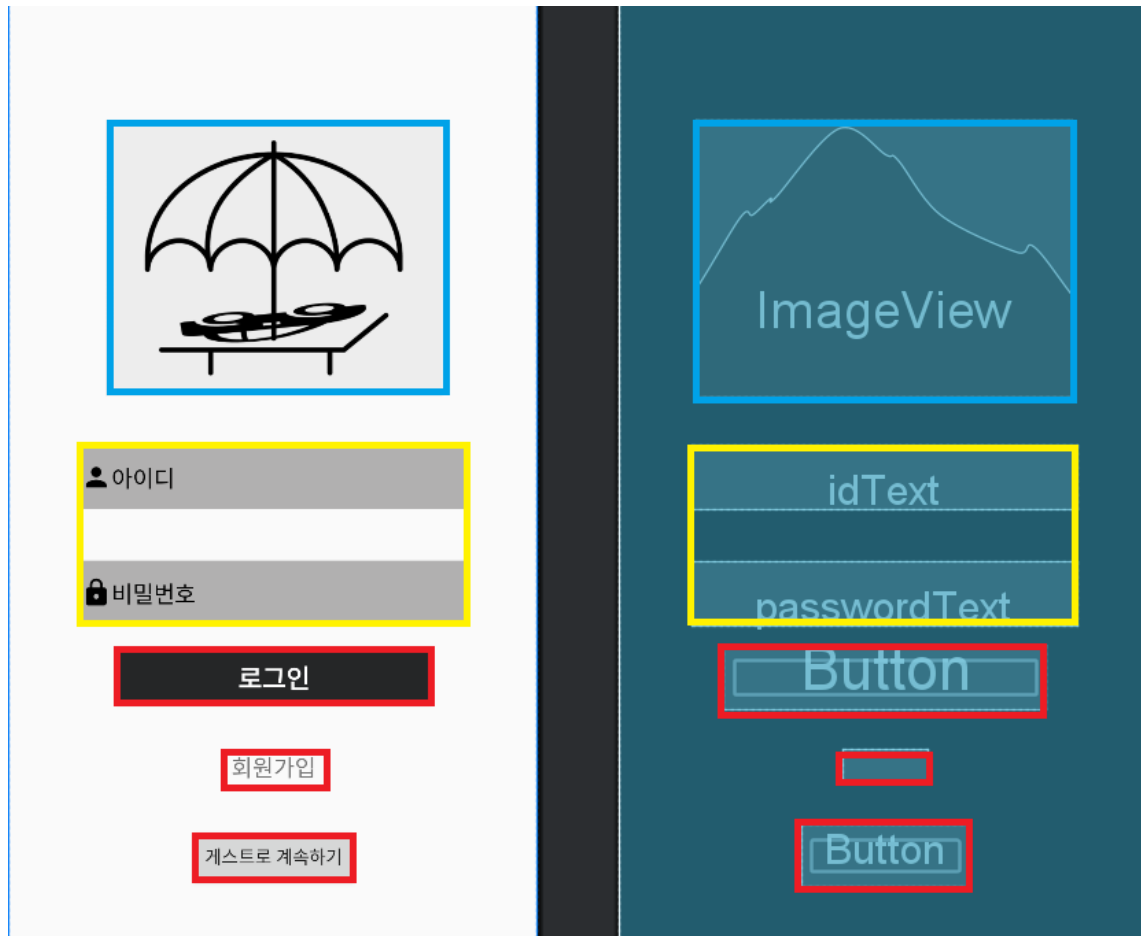
### 3.1 종합 관계도



〈그림 3-1〉 전체 화면 관계도

- 하단 바 → 메인 화면, 즐겨찾기, 설정 화면으로 이동
- 메인 화면 → 휴게소 정보 화면, 검색 창 화면
- 휴게소 정보 화면 → 휴게소 음식 메뉴 화면, 휴게소 리뷰 화면
- 휴게소 리뷰 화면 → 휴게소 리뷰 작성 화면

### 3.2 로그인 화면



〈그림 3-2〉 로그인 화면

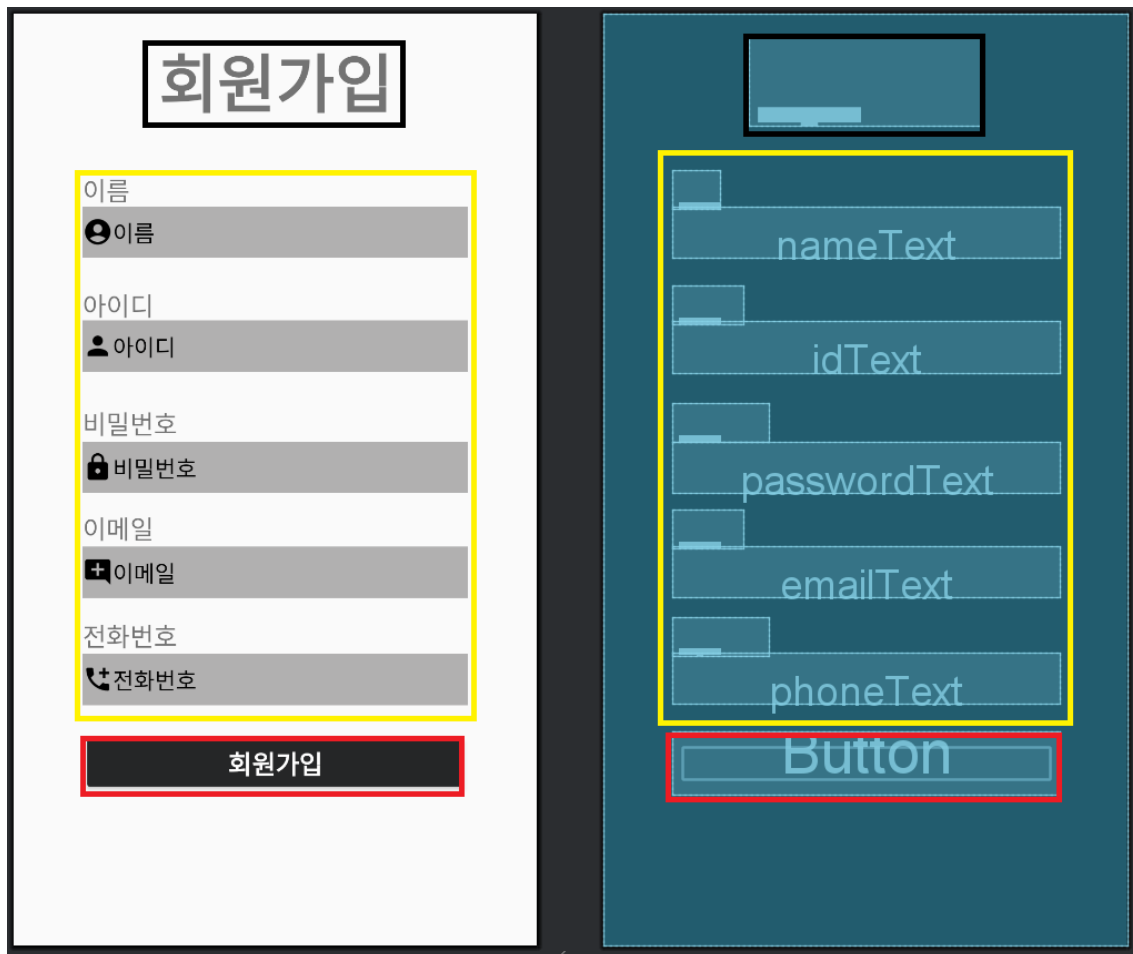
로그인 화면으로 로그인과 게스트모드 버튼을 통해 ‘로드 리뷰’ 메인 화면으로 이동할 수 있고 회원가입 버튼을 통해 회원가입 화면으로 이동이 가능하다.

〈표 3-1〉 로그인 화면 위젯 및 기능

위젯	구분	기능
ImageView	메인 이미지	‘로드 리뷰’를 소개하는 이미지를 나타낸다.
idText	아이디	아이디 입력
passwordText	비밀번호	비밀번호 입력
Button	로그인	클릭 시 입력된 정보를 확인 후 값이 맞는다면 메인 화면으로 이동한다.
	회원가입	클릭 시 회원가입 화면으로 이동한다.
	게스트로 계속하기	클릭시 메인 화면으로 이동한다.



### 3.3 회원가입 화면



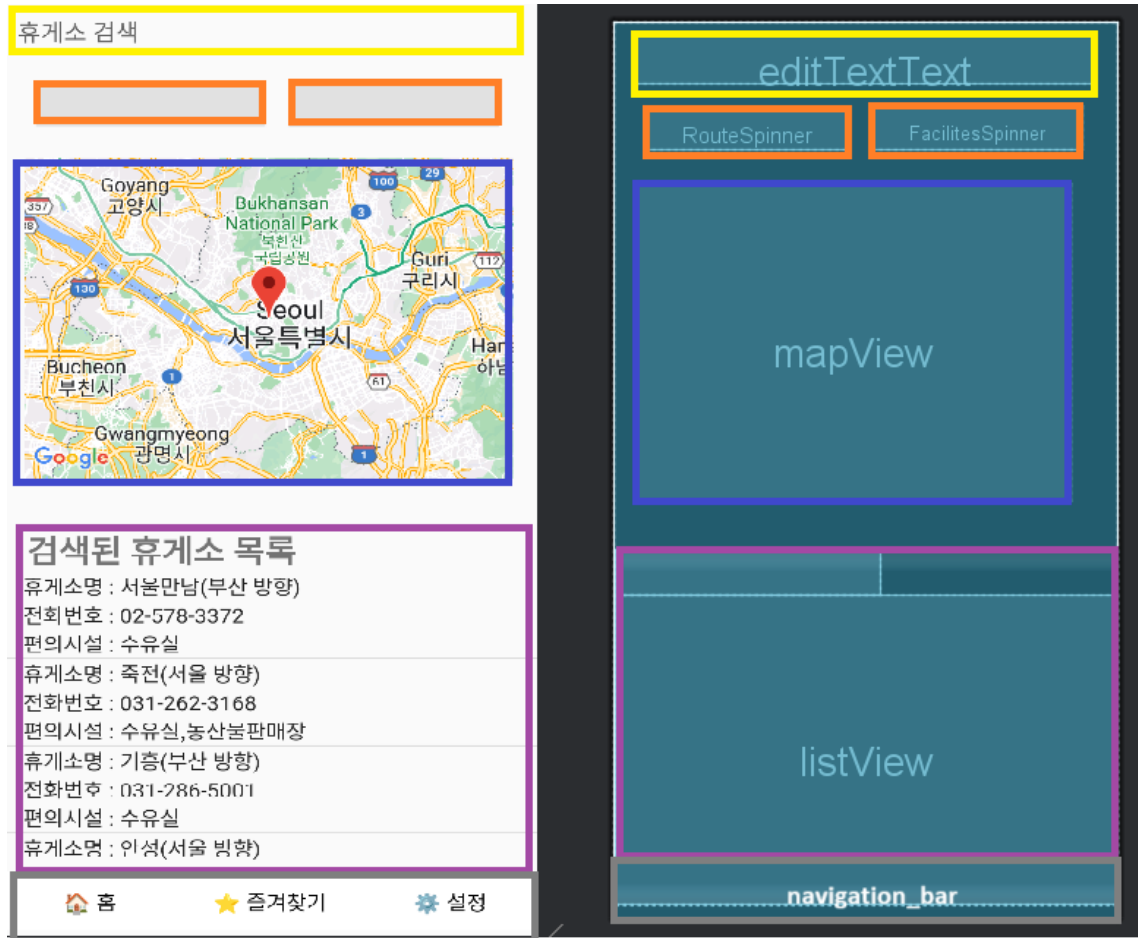
〈그림 3-3〉 회원가입 화면

로그인 화면에서 회원가입 버튼을 눌러 이동하면 출력되는 화면이다. 회원 가입 버튼을 누르면 정상적인 값들이 입력되었는지 확인 후 맞다면 데이터베이스에 값들을 추가하고 자동으로 로그인까지 진행 시키고 메인 화면으로 이동한다.

〈표 3-2〉 회원가입 화면 위젯 및 기능

위젯	구분	기능
TextView	회원가입	회원가입 텍스트를 나타낸다.
nameText	이름	이름 입력
idText	아이디	아이디 입력
passwordText	비밀번호	비밀번호 입력
emailText	이메일	이메일 입력
phoneText	전화번호	전화번호 입력
Button	회원가입	클릭시 회원가입이 완료 후 메인화면으로 이동

### 3.4 메인 화면



〈그림 3-4〉 메인 화면

리뷰 로드의 메인 화면이다. 검색을 통해서 원하는 휴게소를 찾을 수 있으며 지도로 휴게소의 위치를 표시하여 직관적인 정보를 제공한다.

〈표 3-3〉 메인 화면 위젯 및 기능

위젯	구분	기능
editText	휴게소 검색	휴게소 명으로 검색을 수행한다.
Spinner	고속도로 선택	고속도로를 선택하여 필터링을 한다.
	편의시설 선택	편의시설을 선택해서 필터링을 한다.
Fragment	Map	휴게소의 위치를 지도로 보여준다.
listView	휴게소 목록	초기에는 가까운 순으로 휴게소를 출력하고 검색 수행 시 검색된 휴게소를 출력한다.
Navigation_bar	하단 바	3개의 버튼으로 구성 되어있고 각 버튼을 누르면 해당하는 화면으로 이동한다.

### 3.5 휴게소 정보 화면



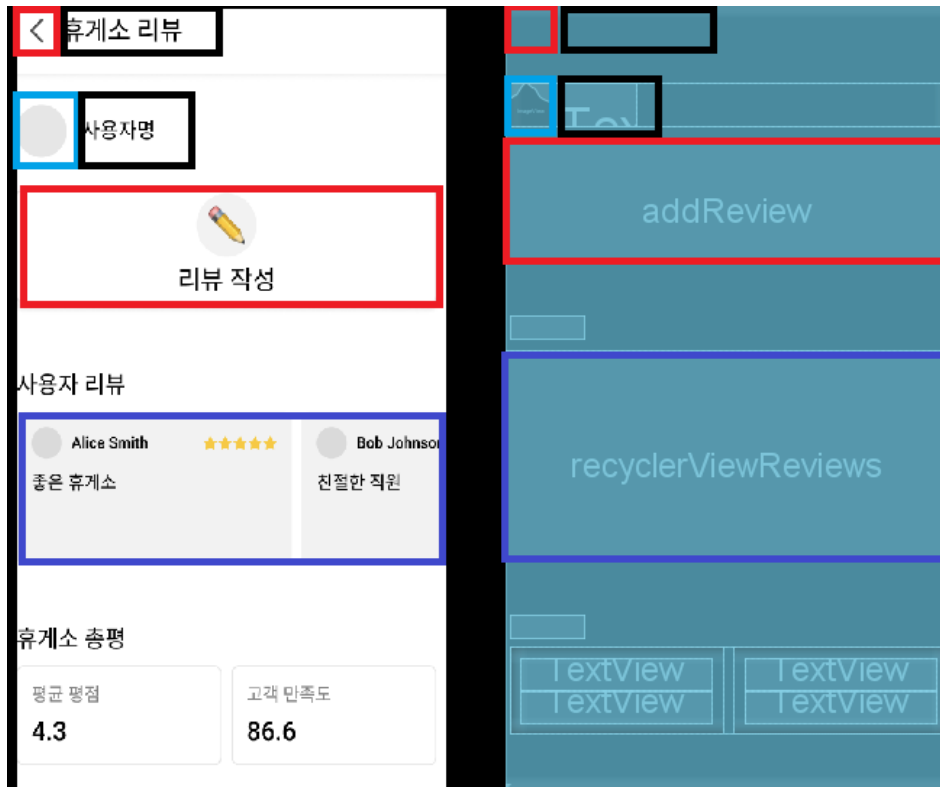
〈그림 3-5〉 휴게소 정보 화면

메인 화면에서 휴게소를 선택하면 출력되는 화면이다. 사용자는 이 화면에서 휴게소에 대한 전반적인 정보를 확인할 수 있고 리뷰, 음식 메뉴 화면으로 이동할 수 있다.

〈표 3-4〉 휴게소 정보화면 위젯 및 기능

위젯	구분	기능
TextView	휴게소 명	메인 화면에서 선택한 휴게소 명을 출력한다.
	유가 정보	선택한 휴게소에 유가정보에 대해 출력한다.
	휴게소 통계	선택한 휴게소의 평점, 고객 만족도를 출력한다.
listview	대표 메뉴	휴게소에서 가장 추천도가 높은 음식 메뉴를 출력한다.
	편의 시설	선택한 휴게소에 편의시설 데이터를 받아와 출력한다..
Button	<	이전 화면인 메인 화면으로 이동한다.
	음식 메뉴 +	음식 메뉴 화면으로 이동한다.
	휴게소 통계 +	리뷰 화면으로 이동한다.

### 3.6 리뷰 정보 화면



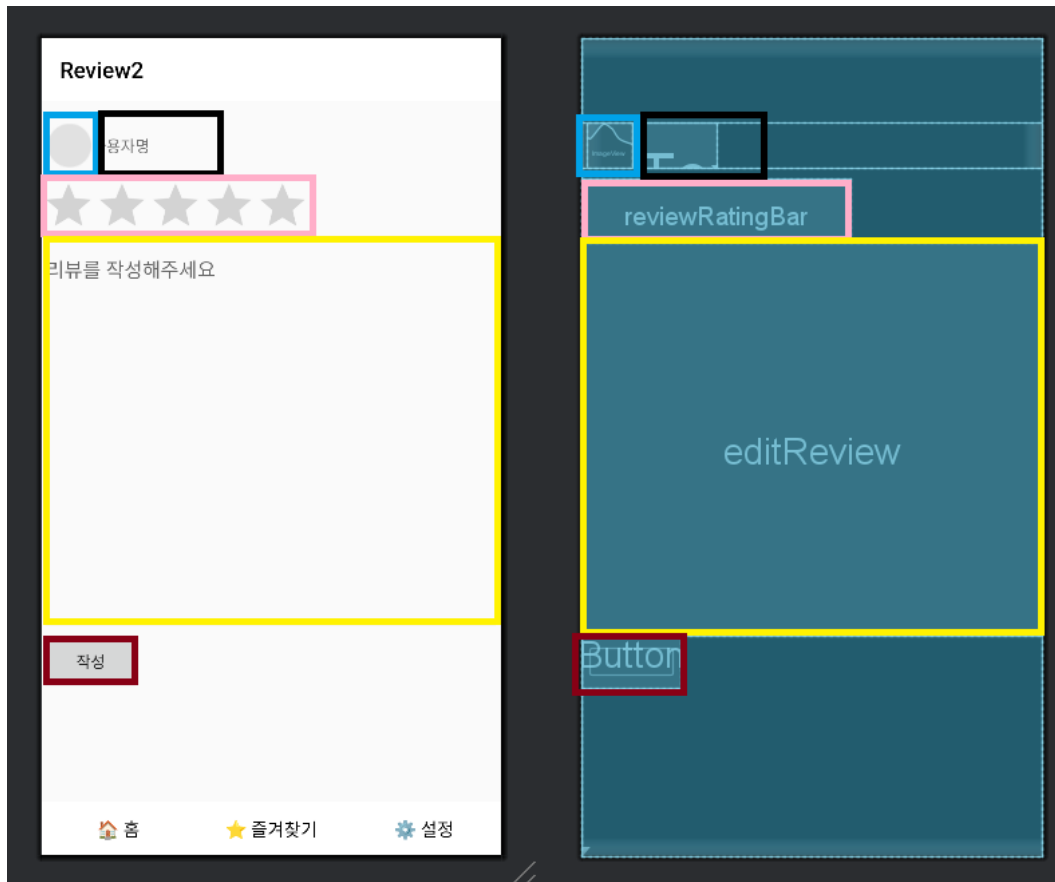
〈그림 3-6〉 리뷰 정보 화면

휴게소 정보화면에서 휴게소 통계 + 버튼을 누르면 출력되는 화면이다. 이 곳에서는 리뷰 작성과 사용자들이 작성한 리뷰를 볼 수 있는 화면이다.

〈표 3-5〉 리뷰 정보 화면 위젯 및 기능

위젯	구분	기능
imageView	프로필 사진	사용자의 프로필 사진을 보여준다
TextView	휴게소 리뷰	현재 어떤 화면에 있는지 출력하여 알려준다.
	사용자명	리뷰 작성 시 등록될 사용자 이름을 알려준다.
RecyclerView	사용자 리뷰	사용자들이 지금까지 작성한 리뷰를 출력해준다. horizontal형식으로 출력하여 양옆으로 넘기면서 리뷰를 볼 수 있다.
Button	<	이전 화면인 휴게소 정보 화면으로 이동한다.
	리뷰 작성	리뷰 작성 화면으로 이동한다.

### 3.7 리뷰 작성 화면



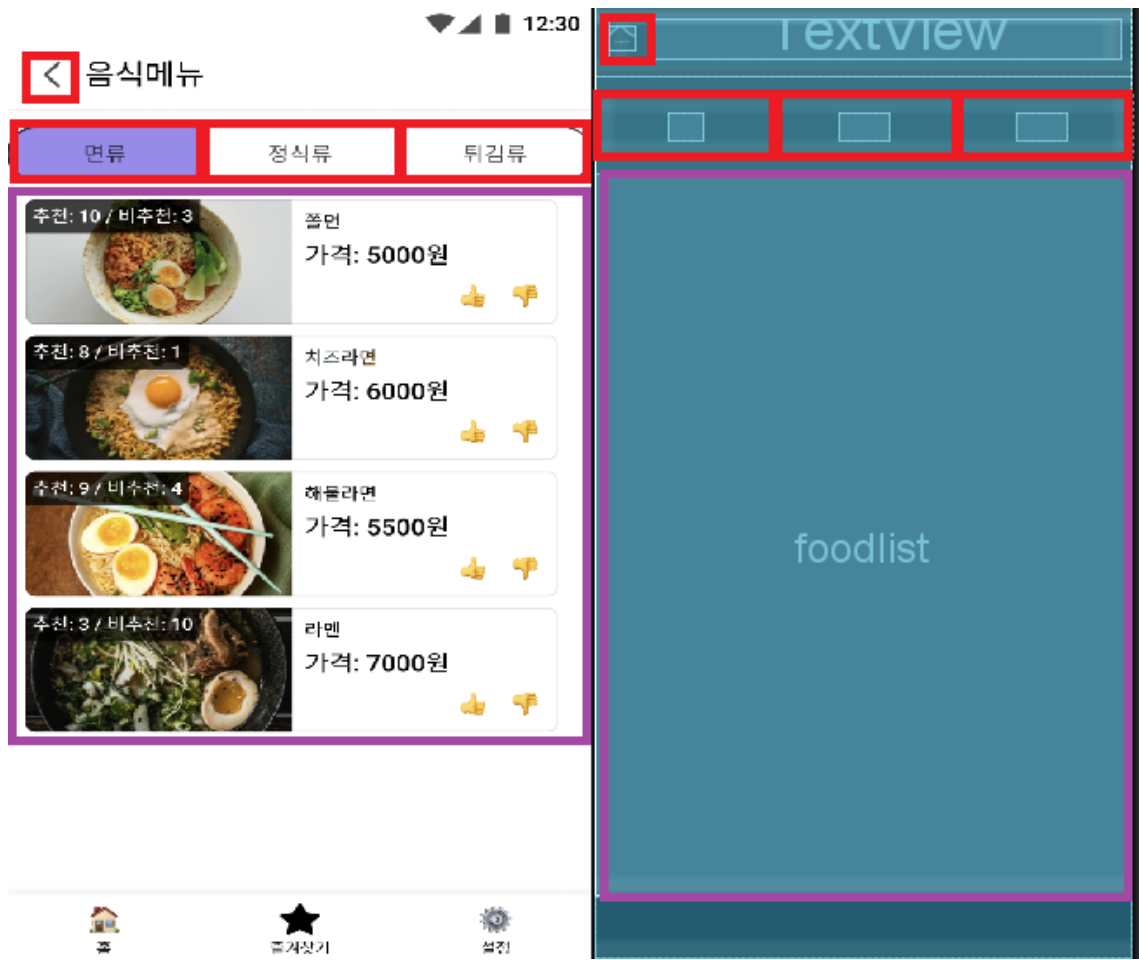
〈그림 3-7〉 리뷰 작성 화면

리뷰 정보화면에서 리뷰 작성 버튼을 누르면 출력되는 화면으로 해당 휴게소에 대해 리뷰를 추가 할 수 있는 화면이다.

〈표 3-6〉 리뷰 작성 화면 위젯 및 기능

위젯	구분	기능
imageView	프로필 사진	사용자의 프로필 사진을 보여준다.
textView	사용자명	리뷰 작성 시 등록될 사용자 이름을 알려준다.
RatiingBar	별점	원하는 별점을 선택할 수 있다.
editText	리뷰 작성	리뷰를 작성할 수 있다.
Button	작성	버튼을 누르면 선택한 별점과 작성한 리뷰를 데이터 베이스에 저장한다.

### 3.8 음식 메뉴 화면



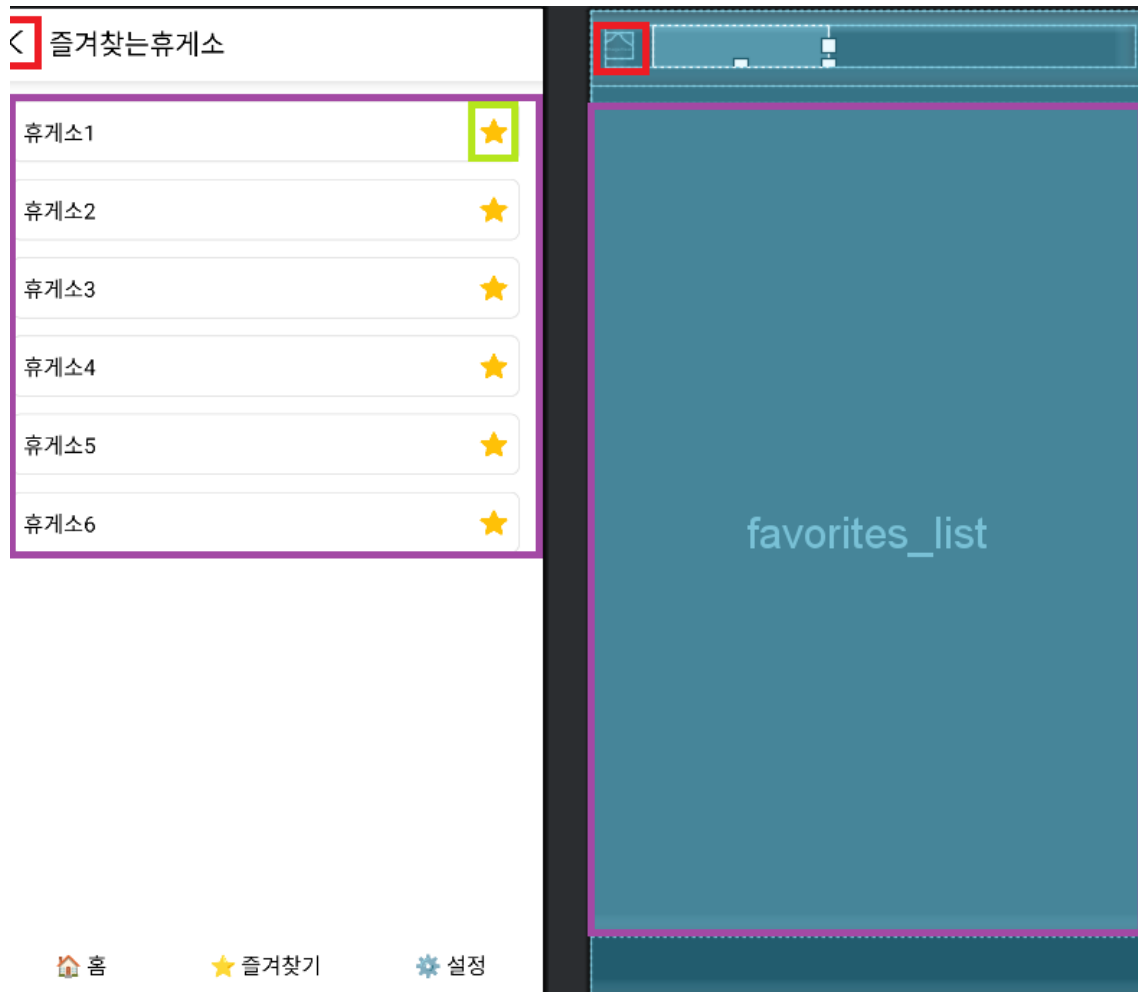
〈그림 3-8〉 음식메뉴 화면

휴게소 정보에서 음식 메뉴 + 버튼을 누르면 출력되는 화면이다. 상단에 카테고리를 누르면 해당 음식의 정보들이 출력되고 해당 음식을 추천/비추천이 가능하다.

〈표 3-7〉 음식메뉴 화면 위젯 및 기능

위젯	구분	기능
ListView	음식 메뉴	음식 데이터를 가져와 음식 메뉴를 출력한다. 리스트 요소로는 ImageView, TextView, Button이 있다.
Button	<	이전 화면인 휴게소 정보 화면으로 이동한다.
	카테고리	원하는 카테고리를 선택하면 출력되는 음식 메뉴 리스트를 변경한다.

### 3.9 즐겨찾기 화면



〈그림 3-9〉 즐겨찾기 화면

하단 바에 즐겨찾기 버튼을 클릭하면 출력되는 화면이다.

즐거찾기 한 휴게소를 확인이 가능하고 즐겨찾기 취소도 가능하다.

〈표 3-8〉 즐겨찾기 화면 위젯 및 기능

위젯	구분	기능
ListView	즐거찾는 휴게소	즐거찾기 한 휴게소들을 출력하여 보여준다
Button	<	하단 바에 즐겨 찾기를 누르기 이전 화면으로 이동한다.
checkbox	별 모양	클릭 시 즐겨찾기가 선택되거나 해제 된다.

## 4. 시스템 설계서

### 4.1 클래스 및 메소드

**MainActivity:** 앱의 메인 화면을 관리하는 클래스.

주요 메서드:

- onCreate(): 메인 화면을 초기화하고 UI를 설정.
- initView(): 메인 화면의 UI 요소들을 초기화.
- setupListeners(): 버튼 클릭 등 사용자 상호작용을 처리하는 리스너를 설정.
  - ➔ spinner, listView 등을 클릭하는데 사용된다.
- onMapReady(): Google Maps API 초기화 및 휴게소 위치 마커 추가.
- loadListView(): 선택된 조건에 따라 휴게소 리스트 데이터를 로드하고 화면에 표시.
  - ➔ 선택된 조건이란 초기 설정과 사용자가 휴게소 명을 검색하거나, spinner를 통해 필터링을 수행한 결과 값을 말함. (검색 기능 수행)

**LoginActivity:** 로그인 기능을 관리하는 클래스.

주요 메서드:

- onCreate(): 로그인 화면을 초기화하고 UI를 설정합니다.
- loginUser(): 사용자가 입력한 아이디와 비밀번호로 로그인 요청을 보내는 역할을 함. 로그인 요청을 보낼 URL을 설정하고 서버에서 응답 '로그인 성공'이면 성공 메시지를 띄우고 'MainActivity'로 이동
- navigateToGuestMode(): 게스트 모드로 메인 화면으로 이동.
- navigateToRegister(): 회원가입 버튼 클릭 시 회원가입 화면으로 이동

**RegisterActivity:** 회원가입 기능을 관리하는 클래스.

주요 메서드:

- onCreate(): 회원가입 화면을 초기화하고 UI를 설정.
- attemptSignUp(): 입력된 회원가입 정보를 검증하고 회원가입 처리를 시도.
- navigateToMain(): 회원가입이 성공하면 메인 화면으로 이동.

**ReviewMainActivity:** 리뷰 메인 화면을 관리하는 클래스입니다.

주요 메서드:

- onCreate(): 리뷰 메인 화면을 초기화하고 UI를 설정.
- calculateAverageRating(): 리뷰 목록의 평균 별점을 계산.
- onOptionsItemSelected(): 툴바의 뒤로 가기 버튼 클릭 시 이전 화면으로 이동.
- loadListReview: 해당 휴게소에 대한 리뷰 리스트 데이터를 로드하고 화면에 표시



**AddReviewActivity:** 리뷰 작성 화면을 관리하는 클래스.

- onCreate(): 리뷰 작성 화면을 초기화하고 UI를 설정.
- reviewCheck(): 사용자가 이미 리뷰를 작성했는지 확인.
- onActivityResult(): 리뷰 작성 후 결과를 처리하여 리뷰 데이터에 추가하고 화면 종료.

**FavoriteActivity:** 즐겨찾기 기능을 관리하는 클래스.

주요 메서드:

- onCreate(): 즐겨찾기 화면을 초기화하고 UI를 설정.
- loadFavorites(): 즐겨찾기한 휴게소 목록을 로드하고 화면에 표시.
- removeFavorite(): 선택된 휴게소를 즐겨찾기 목록에서 제거.
- addFavotite(): 선택된 휴게소를 즐겨찾기 목록에 추가

**FoodActivity:** 휴게소 음식 메뉴 정보를 관리하는 클래스입니다.

주요 메서드:

- onCreate(): 음식 메뉴 화면을 초기화하고 UI를 설정.
- 
- loadFoodMenu(): 휴게소의 음식 메뉴 리스트 데이터를 로드하고 화면에 표시.
- submitFoodRecommendation(): 사용자 추천/비추천을 서버에 제출.

**InfoActivity:** 휴게소 정보를 표시하는 클래스.

주요 메서드:

- onCreate(): 휴게소 정보 화면을 초기화하고 UI를 설정합니다.
- getRestAreaInfoFromWebService(): 웹 서비스로부터 휴게소 정보를 가져옵니다.
- displayRestAreaInfo(): 가져온 휴게소 정보를 화면에 표시합니다.
- handleVolleyError(): 네트워크 요청 실패 시 에러 처리합니다.

내부 클래스:

- RestAreaInfo: 휴게소 정보를 저장하는 모델 클래스.

## 4.2 DB 설계

**User Table:** 사용자 정보를 저장하는 테이블.

테이블:

- user\_id: 사용자 ID (Primary Key)
- email: 사용자 이메일
- password: 사용자 비밀번호 (암호화 저장)
- name: 사용자 이름
- profile\_image: 프로필 이미지 URL

관계:

- User 테이블은 Review 테이블 및 Favorite 테이블과 일대다 관계를 가짐.
  - 하나의 사용자는 여러 개의 리뷰를 작성할 수 있다.
  - 하나의 사용자는 여러 개의 즐겨찾기를 가질 수 있다.

**Review Table:** 사용자 리뷰 정보를 저장하는 테이블.

테이블:

- review\_id: 리뷰 ID (Primary Key)
- user\_id: 사용자 ID (Foreign Key)
- restarea\_id: 휴게소 ID (Foreign Key)
- rating: 리뷰 평점
- review\_text: 리뷰 내용
- review\_image: 리뷰 이미지 URL
- timestamp: 리뷰 작성 시간

관계:

- Review 테이블은 User 테이블과 다대일(N:1) 관계를 가짐.
  - 여러 개의 리뷰는 하나의 사용자에게 의해 작성될 수 있다.
- Review 테이블은 RestArea 테이블과 다대일(N:1) 관계를 가짐.
  - 여러 개의 리뷰는 하나의 휴게소에 대한 리뷰일 수 있다.

## RestArea Table: 휴게소 정보를 저장하는 테이블.

테이블:

- restarea\_id: 휴게소 ID (Primary Key)
- name: 휴게소 이름
- location: 위치 정보 (위도, 경도)
- facilities: 편의시설 정보
- brandstores: 브랜드 가게 정보
- diesel: 경유 가격 정보
- gasoline: 휘발유 가격 정보
- LPG: LPG가격 정보
- electronic: 전기차 충전소 유무 정보

관계:

- RestArea 테이블은 Review 테이블, FoodMenu 테이블, Favorite 테이블과 일대다(1:N) 관계를 가짐.
  - 하나의 휴게소는 여러 개의 리뷰를 받을 수 있다.
  - 하나의 휴게소는 여러 개의 음식 메뉴를 가질 수 있다.
  - 하나의 휴게소는 여러 사용자의 즐겨찾기에 포함될 수 있다.

## Favorite Table: 즐겨찾기 정보를 저장하는 테이블.

테이블:

- favorite\_id: 즐겨찾기 ID (Primary Key)
- user\_id: 사용자 ID (Foreign Key)
- restarea\_id: 휴게소 ID (Foreign Key)

관계:

- Favorite 테이블은 User 테이블과 다대일(N:1) 관계를 가짐.
  - 여러 개의 즐겨찾기는 하나의 사용자에게 의해 만들어질 수 있다.
- Favorite 테이블은 RestArea 테이블과 다대일(N:1) 관계를 가짐.
  - 여러 개의 즐겨찾기는 하나의 휴게소를 포함할 수 있다.

**FoodMenu Table:** 휴게소 음식 메뉴 정보를 저장하는 테이블입니다.

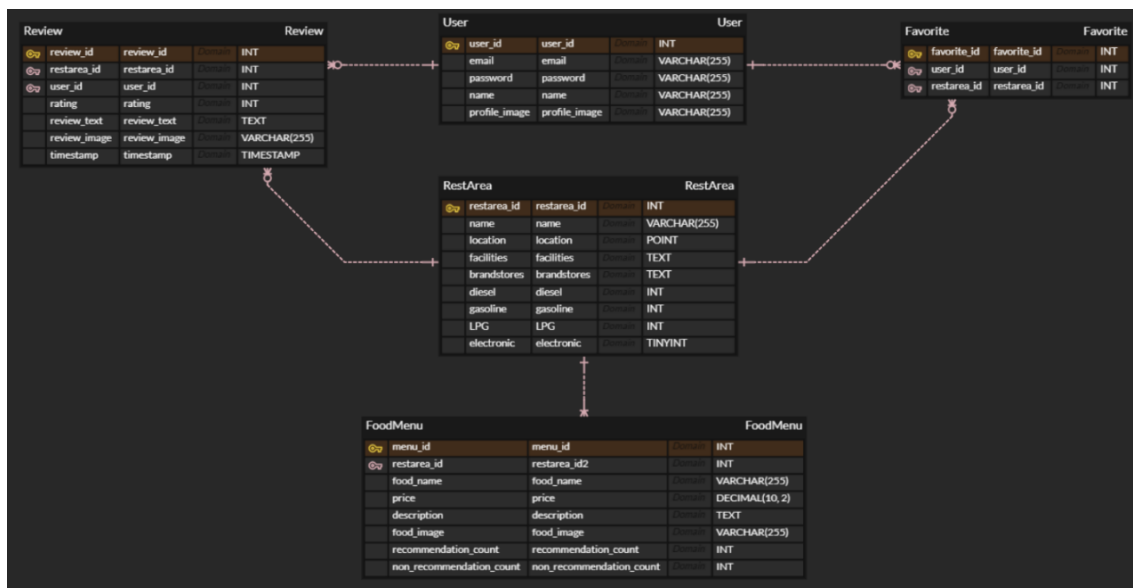
테이블:

- menu\_id: 메뉴 ID (Primary Key)
- restarea\_id: 휴게소 ID (Foreign Key)
- food\_name: 음식 이름
- price: 가격
- description: 음식 설명
- food\_image: 음식 이미지 URL
- recommendation\_count: 추천 수
- non\_recommendation\_count: 비추천 수

관계:

- FoodMenu 테이블은 RestArea 테이블과 다대일(N:1) 관계를 가짐.
  - 여러 개의 음식 메뉴는 하나의 휴게소에 속할 수 있다.

## DB 테이블 구성도



〈그림4-1〉 DB 테이블

### 4.3 외부 API 연동

**Google maps API 사용:** google maps api를 사용하여 휴게소의 위치 정보를 시각화  
주요 기능

- 지도 화면에 휴게소 위치 표시
- 사용자 위치 기반 주변 휴게소 검색
- 사용자가 필터링 한 조건에 따라 충족하는 조건의 휴게소 위치만 표시

## 5. 설계 제한 요소

### 경제성:

- 제한된 기간과 인력 내에서 프로젝트를 효율적으로 진행하기 위해 시간이 크게 소요될 것 같은 작업들을 선정하여 분량을 나누어 팀원들과 프로젝트를 진행하였습니다.
- 무료로 사용할 수 있는 오픈소스 라이브러리와 도구들을 적극 활용하여 비용을 최소화했습니다.

### 편리성:

- 주요 기능에 대한 접근성을 높이기 위해 직관적인 네비게이션과 필터링 기능을 도입했습니다.
- 사용자가 최대한 직관적이고 한눈에 정보를 제공받을 수 있도록 하기 위해 한 화면에 중요한 정보를 보다 많이 담았습니다.

### 윤리성:

- 모든 소스 코드는 직접 작성하거나 적절한 라이선스 하에 사용된 오픈소스 라이브러리를 활용하였으며, 타인의 코드를 무단 도용하지 않았습니다.
- 데이터 암호화를 도입하였습니다.

### 안정성:

- 주요 기능의 오류를 최소화하기 위해 팀원들과 지속적인 피드백을 진행했습니다.
- Volley라이브러리를 사용하여 데이터 처리를 할 때 에러가 발생할 때를 대비하여 에러 처리를 해 두었습니다.

### 유지 관리 용이성:

- 코드의 가독성과 유지보수성을 높이기 위해 기능별로 모듈화된 설계를 채택했습니다

## 6. 현재 개발 수준과 향후 계획

### 본 프로젝트의 현재 개발 수준

#### 기능 구현

- 화면간 액티비티 이동은 대부분의 화면에서 모두 잘 이루어지고 있습니다.
- 메인 화면에서 검색, 필터링, 지도를 통한 휴게소 위치 표시 기능이 구현되어야 현재 지도를 통한 휴게소 위치 표시 기능은 구현을 하지 못하고 검색과, 필터링 기능은 DB와 연동하여 구현을 하였습니다.
- 로그인/회원가입 기능도 DB와 연동하여 구현을 완료하였습니다.
- 리뷰 조회 및 리뷰 작성 화면에서 리뷰를 작성하면 DB에 저장하여 리뷰 조회 화면에 리사이클뷰를 통하여 사용자에게 정보 전달을 해주어야 하는데 현재 DB를 연동까지는 못하였습니다.

#### UI설계

- 디자인 개선은 많이 진행하지 못하였습니다.

### 향후 계획

본 프로젝트는 최종 완성된 앱의 70% 수준까지 개발되었습니다. 향후 계획으로는 개발하지 못한 추가 기능 구현과 기존 기능의 안정화 작업이 포함 됩니다. 예를 들어 구현하지 못한 지도를 통한 휴게소 위치 표시 기능과 리뷰 작성 시 DB연동을 구현 할 예정이고, 소셜미디어 통합, 고급 검색 기능, 주유소 최저가 검색 기능을 추가할 예정입니다.

베타테스트를 통해 수집된 사용자 피드백을 반영하여 UI/UX 개선 작업을 진행하겠습니다. 이를 통해 사용성 문제를 해결하고 더욱 직관적인 인터페이스를 제공하고자 노력할 것입니다. 또한 사용자 피드백을 지속적으로 반영하여 앱의 품질을 향상 시킬 것입니다.

## 7. 업무 분장 및 팀원 공헌도

### 박준범(팀장)

- 프로젝트를 진행하기 위하여 회의 주도, 프로젝트 분량 나누기 및 방향성을 잡음
- 보고서 작성
- 휴게소 정보, 음식 리스트, 즐겨찾기 화면 구현
- 각자 구현한 프로젝트 파일을 병합하고 각 액티비티 간의 이동 설정과 기능 보완(지도 표시, 게스트모드 진입 설정)
- 

### 박종민

- 보고서 작성
- 로그인화면 , 회원가입 화면 , 메인화면 구성
- PHP를 이용하여 외부 DB와 프로젝트 연결

### 박재혁



## 7. 참고 문헌