

FINAL PRESENTATION

PROJECT **시대생 맛집 지도**

2024.12.10

UOSense



목차

소개

- 팀원 소개 및 프로젝트 소개
- Key Features

작업과정

- Sprint 회의
- Product backlog
- Burndown chart

설명

- Highlight Design
- Final Architecture

Live Demo

- 기능 실행

테스트 결과

- Test result 설명

Post moterm

- 프로젝트 과정에서 느낀 점
- 개선할 부분

Member Introduction¹



- 조장
- Front-End
- 개발담당(FE)

-Scrum
master
-Front-End
-소통담당

-BE
-개발담당(BE)

-Front-End
-문서담당

-Back-End
-디자인담당

Project Introduction

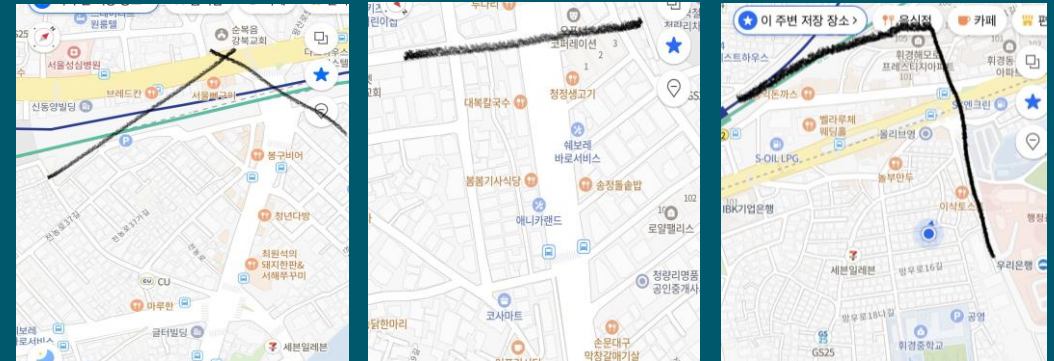
목적

-식당 선택 시 드는 시간과 고민 절감

- 시립대 학생들의 리뷰로만 구성해 신뢰성 제고
- 양질의 데이터를 바탕으로 맛집 선택의 질 향상
- 캠퍼스 주변에서 보다 만족스러운 외식 경험 제공

영역

-정문/쪽문/후문/남문 구역별 탐색



Key Features

식당 검색

식당 이름이나 음식 카테고리, 구역을 기준으로 식당을 검색

식당 정보 열람

식당의 주소, 전화번호, 영업 시간, 대표 메뉴 등의 정보 열람

리뷰 기능

식당에 대한 리뷰 작성. 다른 리뷰어의 리뷰에 대해 추천 및 신고 가능

즐거찾기

마음에 드는 식당 즐겨찾기 추가. 다른 사용자의 목록도 열람 가능

정보 수정 제안

특정 식당에 대한 정보 수정 혹은 신규 식당의 등록을 제안

Scrum[💬]

| Scrum 1 (11/14)

- Product Backlog 작성
- Story point 부여
- Sprint 구현 목록 작성
- Sprint 개발 기간 추정

| Scrum 2 (11/23)

- Coding standard 논의
- Code review process 논의
- 진행상황 점검
- Sprint 회고
→'Back-Front' 간 소통 강화
- 식당 분류 기준 설정
- Burndown chart 작성 시작

| Scrum 3 (11/30)

- 진행상황 점검
- API 명세서 확인(Swagger)
- Test case 관련 논의
- Front 코드 간략 소개
- 앱 아이콘 이미지 결정
- 검색 결과 반환 양식 결정

| Scrum 4 (12/2)

- 진행상황 점검
→Front, Back 구현 정도 파악
- Summary slide 구성
- 식당 데이터 입력
- UI 디자인 설명
- 테스트 방법에 대한 합의

| Scrum 5 (12/7)

- 진행상황 점검
- 'Front-Back' 통합
- API Path 형식 통일
- 서버 배포
- 기타 남은 작업 논의

Product Backlog

Aa 항목	≡ 사용자 스토리	≡ 필요작업	⚙ 진행상황	📌 중요도	🐘 SP
검색어 기반 식당 검색	사용자는 검색어를 입력해 등록된 식당을 조회한다.	1. 식당 데이터베이스 구축 2. 검색 기능 구현 3. 가격대, 평점 등의 정보 데이터베이스와 연결 4. 검색 UI 설계	● 완료	3	3
식당 데이터베이스 관리	관리자는 관리자 계정으로 로그인하여 식당 DB를 수정, 삭제하거나 새로운 식당 정보를 추가할 수 있다.	1. 관리자 계정 DB 접근 권한 부여	● 완료	2	2
식당 즐겨찾기 추가	사용자는 식당 세부 정보 화면에서 해당 식당을 즐겨찾기에 추가할 수 있다.	1. 즐겨찾기 기능 구현 2. 즐겨찾기 추가 시 버튼 이미지 변경 기능 구현 3. 즐겨찾기 해제 기능 구현 4. 즐겨찾기 목록 DB 구축...?	● 완료	8	2
다른 사람 즐겨찾기 목록 열람	사용자는 다른 사용자의 즐겨찾기 목록을 열람할 수 있다.	1. 다른 사용자의 즐겨찾기 목록 열람 기능 구현 2. 즐겨찾기 목록 공개/비공개 기능 구현 3. 열람한 즐겨찾기가 비공개 시 오류메시지 구현 4. 다른 사용자 즐겨찾기 목록 열람 시 UI 설계	● 완료	10	2
정보 수정 제안하기	사용자는 기존 식당 정보의 수정을 원하거나 새로운 식당 추가를 원할 시 관리자에게 해당 정보에 대한 수정을 제안할 수 있다. 관리자는 제안 내용의 승인을 검토하고 처리한다.	1. 정보 수정 제안 기능 구현 2. 관리자의 승인/대기/거절 상태(관리자 승인 프로세스) 구현 3. 알림 메시지 기능 구현 4. 정보 수정 제안 UI 설계	● 완료	11	3
데이터베이스 설계		1. 데이터베이스 설계	● 완료	1	2
		UI 설계			
사용자 로그인/로그아웃	사용자는 생성한 계정 정보를 바탕으로 로그인한다.	1. 사용자 정보 DB 구축 2. 사용자 정보 없을 시 오류메시지 구현 3. 자동 로그인 기능 구현 4. JWT 로그인 방식 구현	● 완료	7	2

Sprint

Sprint 1

- 검색어 기반 식당 검색
- 식당 데이터베이스 관리
- 데이터베이스 설계

Sprint 2

- 식당 정보 열람
- 사용자 로그인/로그아웃
- 사용자 계정 생성 및 관리
- 지도 기반 식당 검색

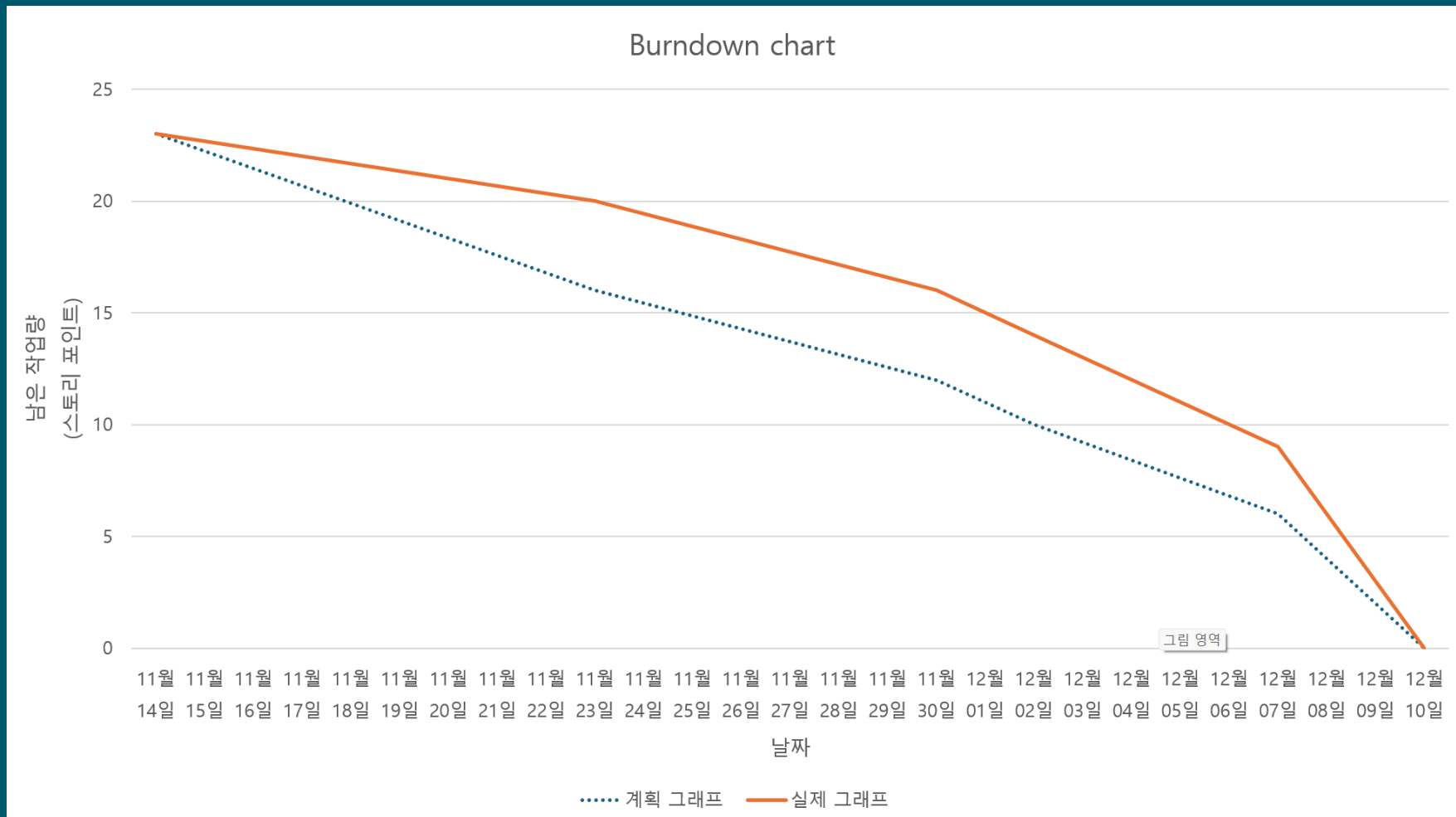
Sprint 3

- 리뷰 작성 및 관리
- 식당 즐겨찾기 추가

Sprint 4

- 타인 즐겨찾기 목록 열람
- 정보 수정 제안하기

Burndown chart²

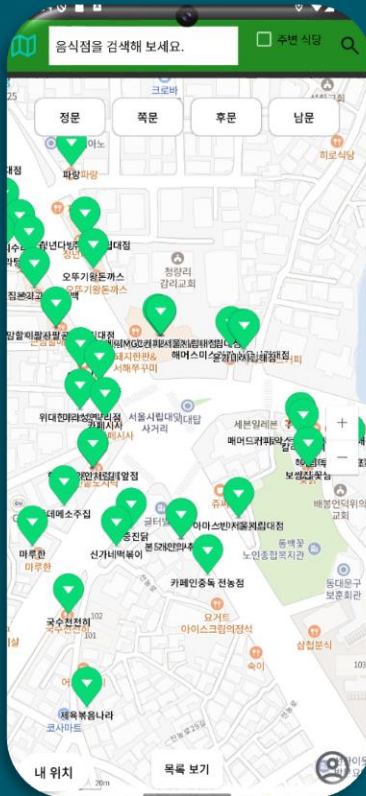
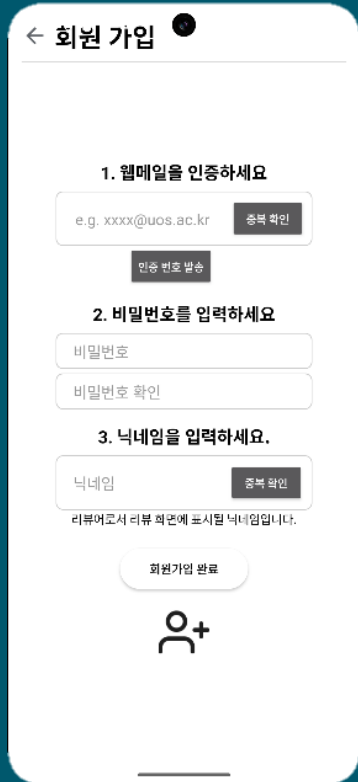


Highlight Design

01

회원가입 화면

웹메일 인증을 통한 회원가입 기능을 제공한다.



02

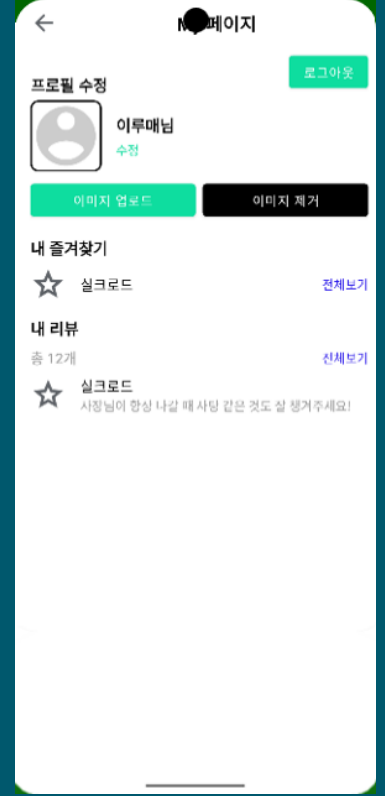
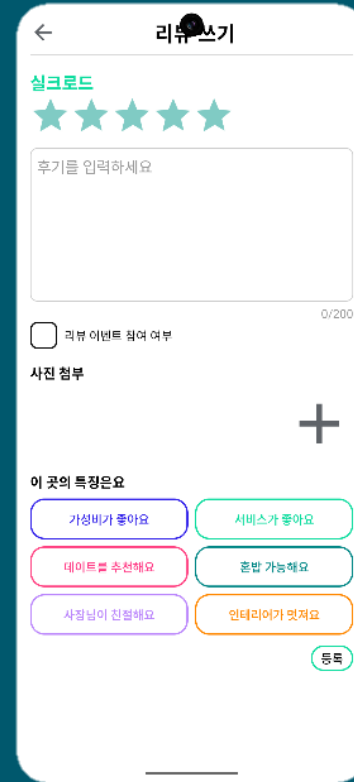
검색 결과 화면

지도의 마커와 리스트 형식으로 검색 결과가 제공된다.

03

리뷰 작성 화면

사용자는 별점과 텍스트 리뷰, 사진을 업로드할 수 있다.

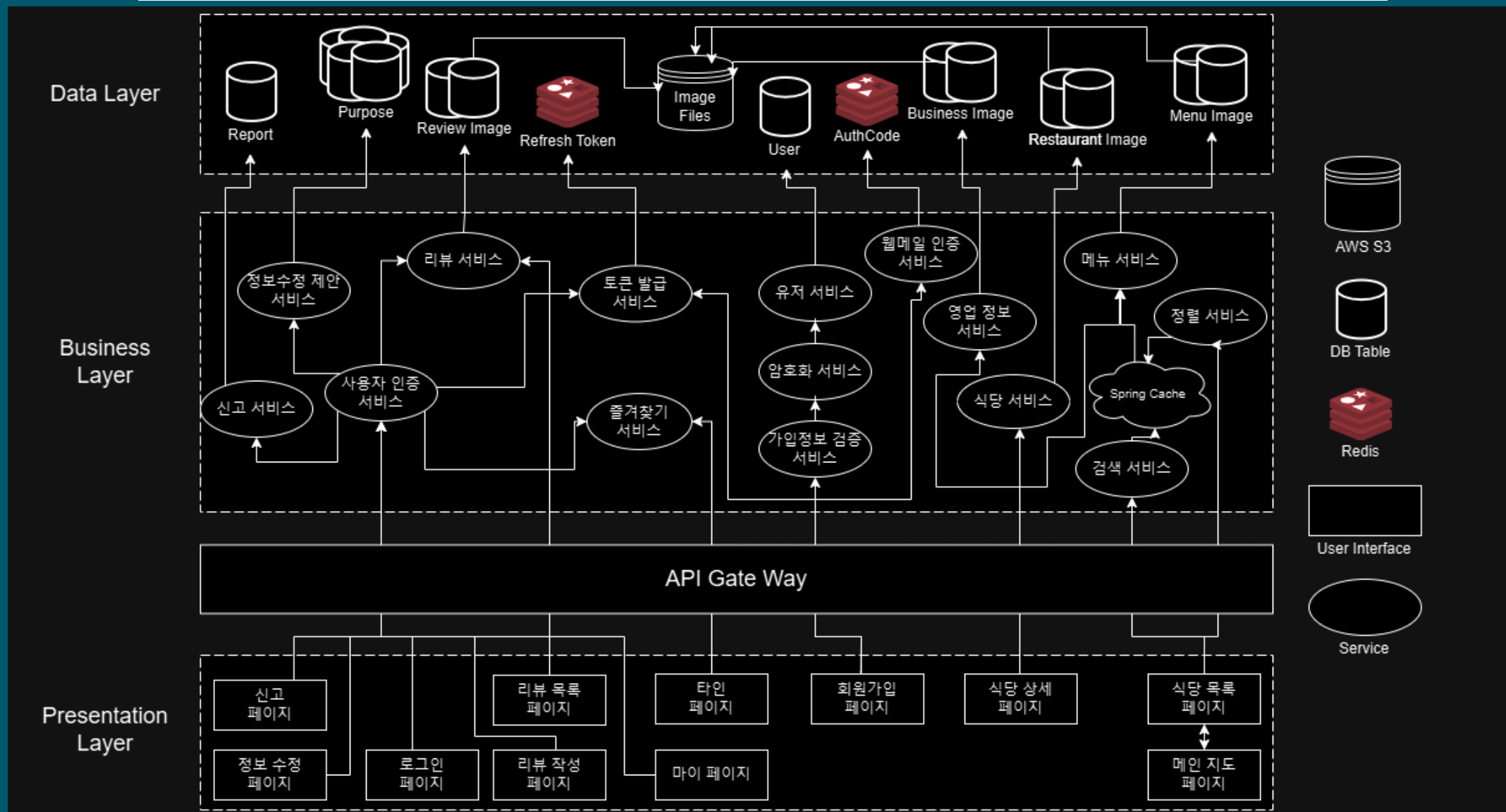


04

사용자 프로필 화면

사용자의 이름, 프로필 사진, 본인이 작성한 리뷰 등의 정보가 제공된다.

Final Architecture



Teck Stack



- Language: Kotlin, Java
- Framework: Spring boot
- Development Tool: Android Studio, IntelliJ, Git
- Collaboration Tool: Notion, GitHub
- Database: MySQL, Redis
- Storage: AWS S3
- Deploy: AWS EC2

Live Demo



Test result

← 회원 가입

1. 웹메일을 인증하세요

everglow02@uos.ac.kr 중복 확인

인증 번호 발송

S20256 제출

2. 비밀번호를 입력하세요

비밀번호

비밀번호 확인

3. 닉네임을 입력하세요.

닉네임 중복 확인

리뷰어로서 리뷰 화면에 표시될 닉네임입니다.

회원가입 완료

인증 번호가 확인되었습니다.

US

-웹메일 인증 완료

← 회원 가입

1. 웹메일을 인증하세요

everglow02@uos.ac.kr 중복 확인

인증 번호 발송

CQ0A5C 제출

2. 비밀번호를 입력하세요

비밀번호가 일치하지 않습니다.

3. 닉네임을 입력하세요.

Gomdadi 중복 확인

리뷰어로서 리뷰 화면에 표시될 닉네임입니다.

회원가입 완료

US

-비밀번호 확인 불일치

← 회원 가입

1. 웹메일을 인증하세요

everglow02@uos.ac.kr 중복 확인

인증 번호 발송

CQ0A5C 제출

2. 비밀번호를 입력하세요

비밀번호가 일치하지 않습니다.

3. 닉네임을 입력하세요.

DoDo 중복 확인

닉네임이 중복되었습니다.

리뷰어로서 리뷰 화면에 표시될 닉네임입니다.

회원가입 완료

US

-중복 닉네임 입력

←

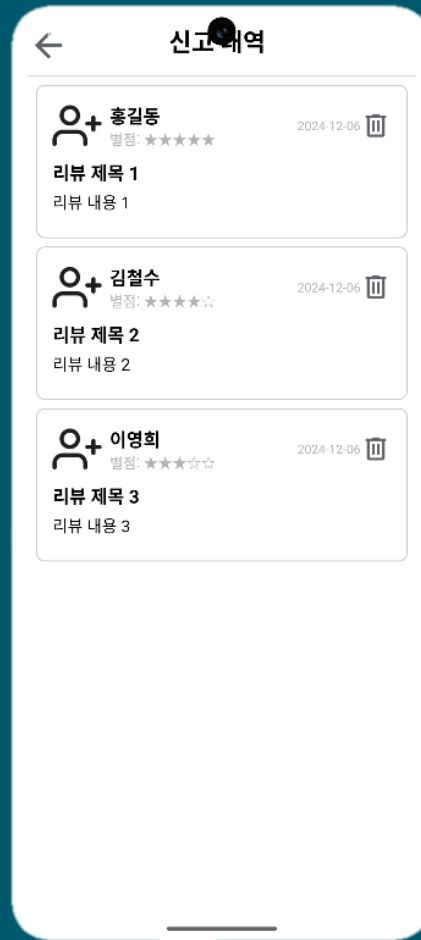
가입 완료!

시대생 맛집 지도
탐방하기

회원가입 성공!

-사용자 계정 생성 완료

Test result



-부적절한 리뷰 삭제



-정보수정제안

Postmortem¹



일정 준수

- 예상일정과 실제 일정 간의 차이
- 초기 기획에 시간을 과다 투입한 결과 개발이 상대적으로 촉박
- 작업량 과소평가



목표 달성

- 초기 기획보다 다소 축소
- 부적절한 리뷰 작성자에게 경고 메시지 전송
- 커뮤니티 기능
- 정보수정제안 기타 정보 입력



개선 방안

- 철저한 작업 시간 예측
- 기획과 개발의 합리적 시간 분배
- 작업 계획 세분화
- 여유 시간 포함



좋았던 점

- 상세한 초기 기획
- 상호간 즉각적인 소통
- 효율적인 업무 분담



THANK YOU
FOR WATCHING