

MeetEat 포트폴리오

목차

1. 프로젝트 개요
2. 시장 분석
3. 개발 과정
4. 기술 스택
5. 구현 및 기능
6. 트러블 슈팅
7. 참고사항

1. 프로젝트 개요



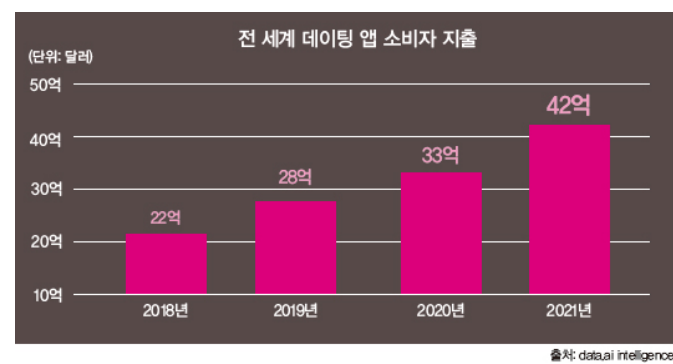
- 목적

만남을 희망하는 'User'와 홍보가 필요한 '동네 맛집'을 '음식'이라는 매개체로 이어주는 어플리케이션

2. 시장 분석

data.ai의 '모바일 현황 2022'에 따르면, 전 세계 데이팅 앱 소비자 지출이 2018년 이후 95% 증가해 42억 달러(한화 약 5조 1597억 원)를 넘었으며, '틴더'는 소비자 지출에서 틱톡과 유튜브에 이어 3위를 차지했다.

출처 : 서울시립대신문(<http://press.uos.ac.kr>)



관련 업계에선 국내 데이팅 앱 시장 규모를 2000억원으로 추산하고 있으며, 2~3년 이내 5000억원으로 2배 이상 커질 것으로 전망한다.

출처:세계비즈



결론

미잇은 급성장하는 데이팅 앱과 식당 예약 시장의 결합을 통해, 사용자들에게 새로운 소셜 다이닝 경험을 제공할 수 있습니다. 합석과 맛집 예약 기능을 결합한 이 서비스는 변화하는 트렌드에 맞춰 큰 시장 잠재력을 가지고 있습니다.

3. 개발 과정

개발 방법론

- **Agile** 방법론을 일부를 응용하여, 핵심 가치를 중심으로 **변화에 유연하게 대응**하고, 지속적으로 **개발 과정에서 개선**을 이루었습니다. 고정된 모델에 의존하지 않고, 자유롭게 변화를 반영하며 프로젝트를 진행했습니다.

협업 도구

- **Lean Canvas** (기획 및 전략 수립), **Git & GitHub** (버전 관리 및 프로젝트 관리), **Figma** (디자인 및 워크플로우 관리) 등을 활용하여 팀원 간의 효율적인 협업을 이끌어냈습니다.

주요 단계 및 업무

- 기획 단계:

Lean Canvas를 통해 프로젝트의 핵심 문제와 해결 방안을 정의하고, 전략을 수립했습니다.

- 설계 단계:

Spring Boot, JPA, MySQL을 기반으로 기술 스택을 설계하고, 전체 시스템의 아키텍처를 정의했습니다.

- 개발 단계:

회원가입, 로그인, 합석, 식당 예약 기능을 포함한 주요 기능을 구현했습니다.

- 테스트 단계:

Mock 객체를 이용해 단위 테스트를 진행하고, **Figma**를 통해 UI/UX를 개선했습니다.

4. 기술 스택

1. 프로그래밍 언어

- Java
- JavaScript

2. 백엔드 프레임워크

- Spring Boot
- Spring Security
- JPA

3. 프론트엔드 기술

- Thymeleaf

4. 데이터베이스

- MySQL

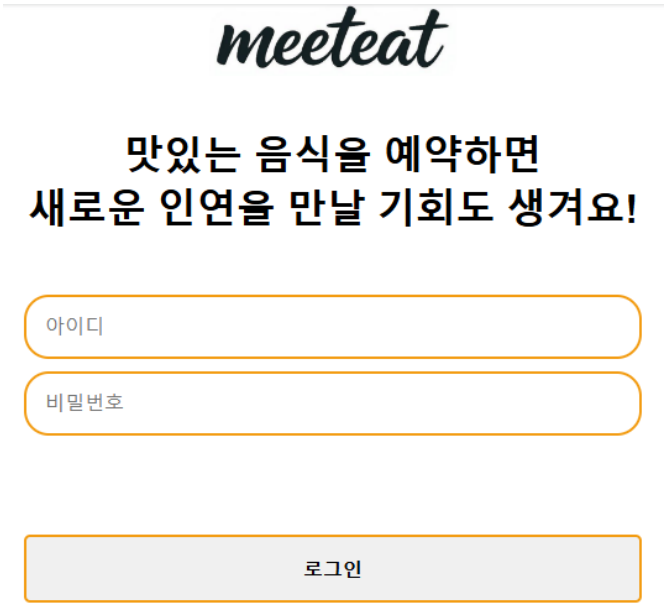
5. 빌드 도구

- Gradle

6. 버전 관리

- GitHub

5. 구현 및 기능



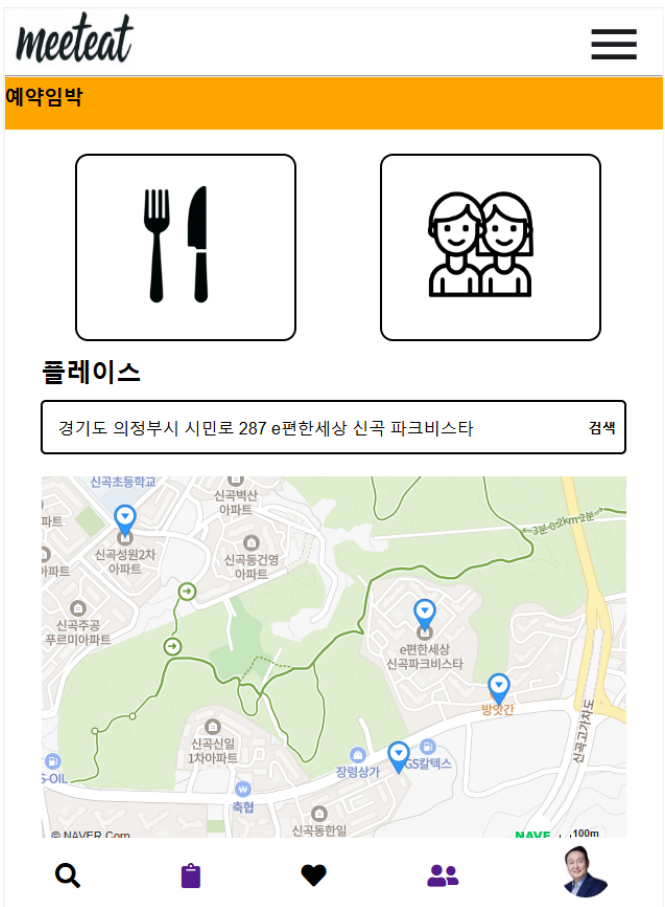
로그인

- Spring Security의 UserDetailsService 수정과 SecurityContext Container를 활용하여 인증, 인가 확인
- 회원 가입 시 네이버 이메일 전송 API를 통해 인증 코드를 전송하여 보안을 강화



이메일 인증

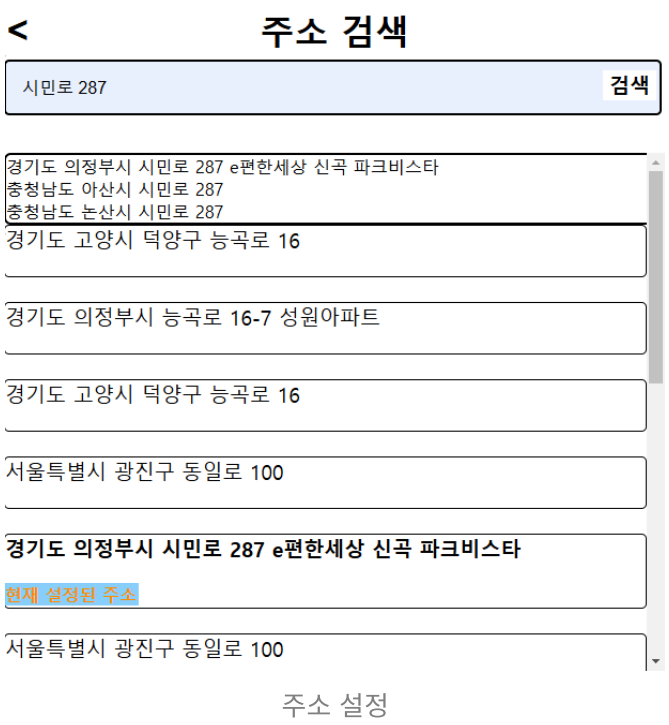
- 외부 SMTP 서버로 인증 코드 전송



메인 페이지

메인 페이지

- 네이버 지도 API 현재 위치 기반 주변 가게 마크 활성화
- 위치 변경 시 비동기 처리로 위치 재설정
- 예약 테이블 조회 이후 예약 날짜 임박한 데이터 알림



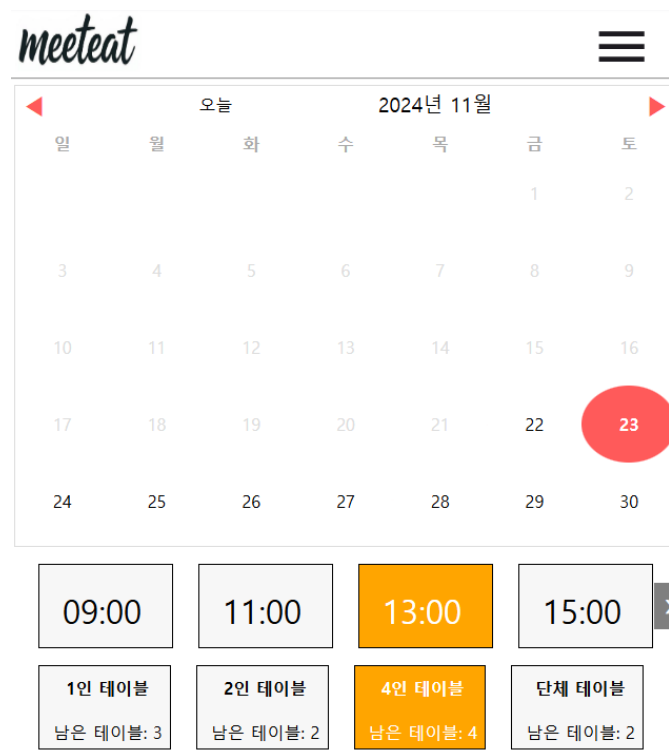
주소 설정 및 검색

- 네이버 주소 검색 API 주소 검색 시 ajax 비동기 통신으로 목록 생성
- 등록된 주소 클릭 시 활성화



식당 리스트

- 현재 시간 기준 영업 식당 활성화
- ajax 비동기 통신으로 카테고리 정렬



예약 캘린더 기능

- PostConstruct로 어플리케이션 실행 시 3주 날짜 데이터 업데이트
- Scheduled로 24:00 기준 데이터베이스 자동 업데이트

6. 트러블 슈팅

문제 배경과 해결

문제점

주소 변경 이후 메인 페이지 렌더링 시, 지도 마크가 제대로 표기되지 않는 문제가 발생했습니다.

해결

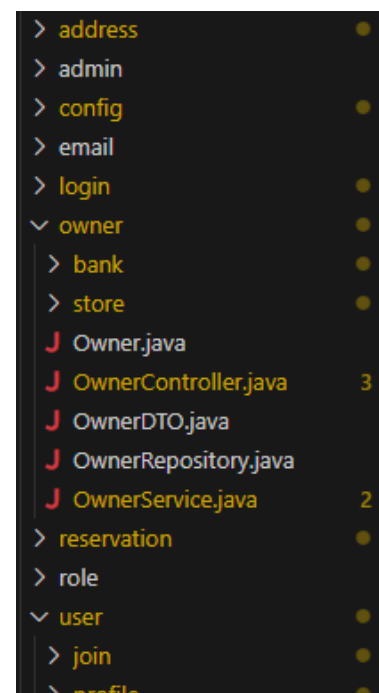
메인 페이지 렌더링 과정에서 JavaScript의 지도 마킹 이벤트를 DOMContentLoaded로 처리하여 해결하였습니다.

문제점

애자일 방법론에 따른 빈번한 수정 과정에서 서비스의 역할과 책임이 명확히 정의되지 않아 코드의 결합도가 높아지고 응집도가 낮아져, 모듈화가 제대로 이루어지지 않는 문제가 나타났습니다.

해결

이 문제를 DDD(Domain-Driven Design)를 적용하여 도메인 모델을 중심으로 서비스의 역할과 책임을 명확히 정의하고, 각 도메인에 맞는 모듈화를 구현함으로써 결합도를 낮추고 응집도를 높였습니다.



7. 참고사항

GitHub:

<https://github.com/YunSinHo/MeetEat>

프로젝트의 소스 코드와 관련 문서가 포함된 GitHub 리포지토리.

Figma 디자인:

<https://www.figma.com/design/wWvdGkg49bbfCamXIOLMg/MeetEat?node-id=8-364&node-type=frame>