

밥묵을래 ?



2조 최종 발표

부산대 맛집 어플

201924648 나인수 201924403 강수빈

201924539 이영인

발표순서

- ▶ 1. 앱 구조와 설명
- ▶ 2. API를 통한 맛집 위치기반 서비스 와 맛집 리뷰 블로그 검색 서비스 구현
- ▶ 3. 데이터 랭크 나누기
- ▶ 4. 부산 맛집 정보 서비스 API 를 통한 부산대 맛집 나열 및 시각화 구현하기
- ▶ 5. UI 디자인 및 앱 꾸미기
- ▶ 6. Fire Base 를 구축하고 로그인 서비스 구현하기
- ▶ 7. 맛집 게시판 구현하기

역할분담

- ▶ 이영인
 - ▶ 1. API를 통한 맛집 위치기반 서비스와 맛집 리뷰 블로그 검색 서비스 구현
 - ▶ 2. 데이터 랭크 나누기
- ▶ 강수빈
 - ▶ 1. 부산 맛집 정보 서비스 API를 통한 부산대 맛집 나열 및 시각화 구현하기
 - ▶ 2. UI 디자인 및 앱 꾸미기
- ▶ 나인수
 - ▶ 1. Fire Base를 구축하고 로그인 서비스 구현하기
 - ▶ 2. 맛집 게시판 구현하기

밥묵을래?



PNU Restaurant List



PNU

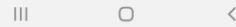
SignUp

ID

PW

Confirm PW

회원가입



PNU

LOGIN

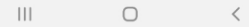
ID

PW

SIGN IN

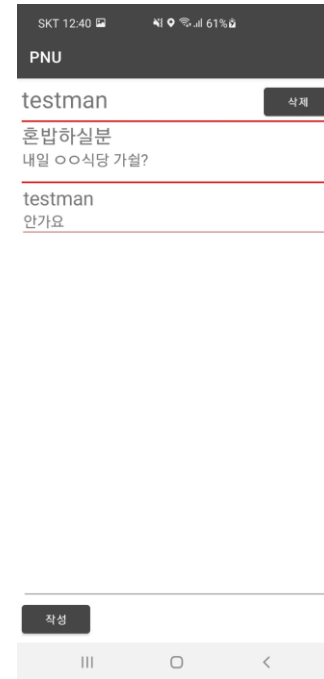
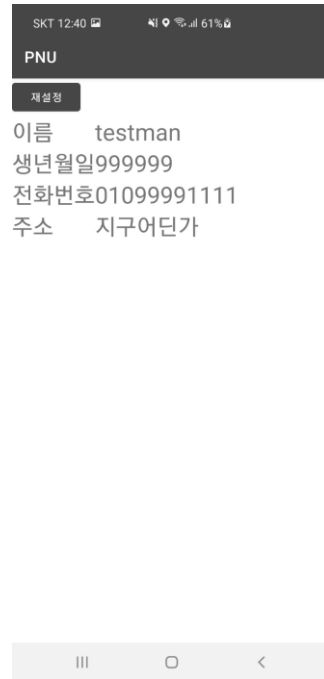
SIGN UP

RESET PASSWORD



앱 구조와 설명(1)

- ▶ 1. 로딩 화면
- ▶ 2. 회원가입 화면
- ▶ 3. 로그인 화면



앱 구조와 설명(3)

- ▶ 1. 맛집 게시판
- ▶ 2. 회원 정보
- ▶ 3. 게시글 및 댓글 화면
- ▶ 4. 게시글 작성 화면

API 연동을 통해서 위치기반 서비스, 식당 관련 리뷰 제공기능

- ▶ 네이버 지도 **API**를 사용하여 버튼 클릭 시 지도 화면에 띄우기, 네이버 지역 검색 **API**를 사용하여 식당의 좌표를 불러온 뒤 지도에 마커로 표시
- ▶ 네이버 블로그 검색 **API**를 사용하여 식당 이름을 클릭 시 관련 블로그 정보 불러오기(글 제목,내용,링크), **recycleView** 를 활용하여 식당 관련 블로그 정보 나열, 보고 싶은 글 클릭 시 해당 블로그 사이트로 연결

데이터 랭크 나누기

- ▶ 파이썬(selenium) 을 사용하여 카카오 맵을 크롤링(리뷰 수, 별점, CSV파일로 저장)
- ▶ 리뷰 수와 별점을 기준에 따라 다시 점수를 매겨 식당 다시 랭크를 매겨줌 -> CSV 파일 형태로 저장
- ▶ 리뷰(10개 이하 : 1점, 10~20 개 이하:2점 ... 100점 이상:10점)
- ▶ 별점(0.5점 이하 :1점, 0.5~1점: 2점 ... 4.5~5점 : 10점)

UI디자인

- ▶ 초기에 맛집 게시판 외 UI 디자인 틀 잡음.
- ▶ 후에 완성된 코드에서 UI 디자인 더 매끄럽게 수정.(앱 상단바 색상, 상단바 앱 이름 ..등등)
- ▶ 스레드를 이용해서 앱 실행 시 초기 로딩 화면 생성
- ▶ 여러 사람의 코드를 합치고 최적화 작업

맛집 API 를 이용한 나열 및 시각화

- ▶ 맛집 **API**를 사용하여 부산대 근처 맛집 정보들을 알아보려 했으나 공공 데이터 포털에 있는 맛집 **API**를 사용하여 부산지역 전체의 맛집을 담고 있어 생각한 앱의 방향성과 달라 맛집 **API** 사용을 포기하고, 직접 검색 시 나오는 부산대 앞 맛집 약 **85**개 정도를 수집.
- ▶ 이를 **list**로 보이게 하고, 클릭 시 해당 맛집 관련 검색결과가 뜨는 창으로 넘어가게 함

Firestore 구축 및 로그인 서비스 구현

- ▶ Firestore 를 앱과 연결하기 위해서 firestore에 앱 등록과 gradle 파일 수정.
- ▶ 로그인 서비스를 구현하기 위해서 firestore authentication 사용.
- ▶ 회원가입에 여러가지 제한을 둠(최소 비밀번호 자리 수, 아이디 이메일 형식...등)
- ▶ 이메일에 확인번호 전송을 통해 비밀번호 찾기 서비스 구현.

맛집 게시판 구현하기

- ▶ Firebase database 생성하고 앱 **gradle** 에 등록
- ▶ **Timestamp** 속성을 추가하여 해당 속성을 기준으로 내림차순 정렬로 게시글을 나열하기
- ▶ 나열된 게시글 중 하나를 클릭 시 전체 글을 볼 수 있으며 자신이 작성자일 경우 삭제 가능. 해당 페이지에 댓글을 쓸 수 있으며 댓글은 해당 문서의 하위 문서로서 계층화 되어있다.
- ▶ 회원 정보를 저장하는 문서를 추가하고 각 게시글과 댓글에는 작성자의 이름을 표시한다. 회원 정보를 원할 때 언제든지 수정할 수 있다.
- ▶ 맛집 게시판은 맛집 리스트 액티비티에서 전용 버튼 클릭을 통해 입장 가능하다.



Q&A

https://github.com/NAINSOO/final_2zo_project.git