

1. 목표

- 영화 추천 서비스 구현
- HTML/CSS, Javascript(Vue.js/Vanilla JS, etc), Django, Database 등을 활용한 실제 서비스 설계
- Git을 통한 소스코드 버전 관리 및 협업
- 서비스 배포

2. 개발환경

1) (필수) Python Web Framework

A. Django 2.1.x

B. Python 3.6.x

2) 서비스 배포 환경

A. 서버 : Ubuntu / Amazon Linux 등

B. Database : MySQL / SQLite 등

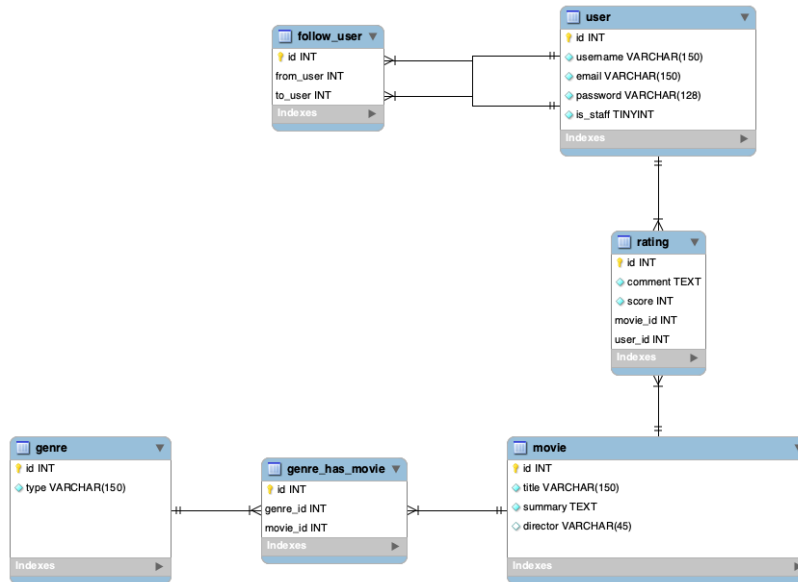
3. 서비스 개요

- A. 본 프로젝트는 영화를 주제로 한 서비스를 구현하는 것으로, 영화 데이터베이스를 필수적으로 가지고 있어야 한다. (최소 등록된 영화 레코드 50개 이상)
- B. 모바일 대응을 한 반응형 웹, Django REST API 서버 및 프론트엔드 프레임워크(Vue) 분리 등의 상세 구현 방식은 자유롭게 구성하되 프로젝트 README 상단에 구조도 혹은 설명을 명시한다.
- C. 모든 프로젝트의 과정은 git을 통한 소스코드버전관리를 진행하며, Gitlab을 통하여 팀원 간 프로젝트 소스코드를 관리한다.
- D. 다양한 기준으로 영화를 추천할 수 있는 기능은 반드시 구현하여야 하며, 부가적인 영화 커뮤니티에 필요한 기능등을 자유롭게 추가할 수 있다.
(예 - 채팅 서비스, 커뮤니티 서비스, 영화 관련 정보 아카이빙 서비스, 현재 박스오피스 정보 제공 등)
- E. 추천의 방식은 자유롭게 구성 가능하되 반드시 영화를 추천 받을 수 있어야 한다.
- F. 사용자를 위한 최소한의 HTML/CSS를 통한 웹 사이트 디자인을 하여야 한다.
- G. 완성된 서비스를 배포하고, 테스트 및 유지 보수를 진행한다.

4. 서비스 아키텍처

본 모델링은 최소한의 기준의 예시이며, 추가적인 필드 선언 및 모델링은 가능하다.
다만, 변경시 사전에 데이터베이스 모델링을 미리 팀원들과 협의하여 정한다.

A. ERD



B. 필수 기능

- 관리자 권한의 유저만 영화 등록 및 수정 / 삭제 권한을 가집니다.
(별도의 관리자 권한의 뷰가 구성되어야 합니다.)
- 관리자 권한의 유저는 유저 관리 (수정 / 삭제 권한)을 가집니다.
(별도의 관리자 권한의 뷰가 구성되어야 합니다.)
- 모든 로그인 된 유저는 영화에 대한 평점을 등록 / 수정 / 삭제 등을 할 수 있어야 합니다.
- 평점을 등록한 유저는 해당 정보를 기반으로 영화를 추천 받을 수 있어야 합니다.
- 데이터베이스에 기록되는 모든 정보는 유효성 검사를 진행해야 하며, 필요에 따라 해당하는 정보를 클라이언트 화면에 띄워줄 수 있어야 합니다.
- HTTP method와 상태 코드는 적절하게 반환되어야 하며, 필요에 따라 해당하는 에러 페이지도 구성을 할 수 있습니다.
- 이외 추가적인 기능은 자유롭게 구성 가능합니다.

5. 제출 방식

- A. 완성된 소스코드의 Gitlab 주소를 담임 교수에게 제출한다.
- B. 해당 저장소에는 반드시 README.md을 통해 아래의 내용이 기록되어야 한다.
 - i. 팀원 정보 및 업무 분담 내역
 - ii. 목표 서비스 구현 및 실제 구현 정도
 - iii. 데이터베이스 모델링
 - iv. 핵심 기능
 - v. 배포 서버 URL
 - vi. 기타(느낀점)