**CrossFilmz**

**Anthony Gruber, Avery Peiffer, Samir Sherlaker, Carter Smith**

As affordable, accessible streaming platforms have supplanted traditional television for the standard medium of entertainment, many users have experienced a paralysis in choosing the content they wish to consume. This is exacerbated by the platforms’ push towards releasing their own original content - Netflix alone released 371 original titles in 2019, more than was released by the entire television industry in 2005 ([source](https://variety.com/2019/tv/news/netflix-more-2019-originals-than-entire-tv-industry-in-2005-1203441709/)). Furthermore, for those who use multiple streaming platforms, it can be a great inconvenience to try and locate a title. Searching each platform individually can be tedious, and the only sources that give information about a title are long articles, bloated with ads, that spout paragraphs of unrelated information before getting to the point. In an attempt to provide a more personalized and convenient viewing experience to users, we want to create CrossFilmz, a movie rating and recommendation service with a focus on streaming platforms.

The functional and non-functional requirements for CrossFilmz are outlined in list form below.

**Functional Requirements:**

* User login page
* User account registration page
* Ability to add a movie rating
* User profile containing previous ratings
* Generated list of user suggestions
* Ability to filter based on streaming services
* Links to IMDb page or equivalent to get more information about the movie

**Non-functional Requirements:**

* Requests to APIs
* Security - API Keys secured in a protected file
* Security - OAuth or JWT for login
* Security - All DB requests are done via parameterized queries to prevent injection attacks
* Version Control - Github with branches for each team member
* Scaling - Design for horizontal and vertical scaling as the user base grows with Google Cloud
* Fast calculation of recommendations
* Fast filtering based on platforms

We would like the movie recommendation algorithm to be a content-based recommender system, as it will be shaped on a single user’s preferences; a collaborative recommender system, in which users are suggested content based on the ratings of other users, would not fit here. Movies will be tagged using keywords based on their genre, and recommendations will be generated based on what the user rates positively. As such, we believe that it would be sensible for the service’s backend to be written in Python, as the language provides several useful tools for accomplishing such a goal. The backend will use either the Django or Flask framework, and the frontend will be built using ReactJS. The service will be hosted on Heroku or Google Cloud. We also plan on making use of the Utelly API to find the availability of a title on various streaming platforms, which we will then use to filter a user’s recommendations. Additionally, there will need to be a database consisting of a user’s previous ratings and current preferences in order for the recommendation algorithm to provide suggestions.