Justin Hershberger

Terik Brunson

Matthew Morris

Caleb Yearsley

Corbin Cutler

Design Document

**Purpose statement**

Our idea is to combine a movie database, a rating system, and social media in order for movie lovers to view their friend’s as well as critic’s ratings of movies. They will be able to rate movies however they feel and share that with their friends. Users can find suggestions for new movies to watch or view their favorite actor/director’s filmography all in a social manner.

**Design Outline**

We will have a client-server model with a database to store information about the users, movies, and people involved with movies such as actors and directors. Our components are: a user, server, and a database. The user component will be able to create a session where they can access details about whichever movie they would like and be able to rate the movie and share the information with their friends. We will store the information about the user’s profile, friends, favorite movies, etc. in order to be accessible at a later session. When accessing a movie’s page, calls to iMdb’s api will be made and information will be sent to the user. Our database component will contain specific information about the user, movies, actors, directors, as well as a friends list. The server component will make calls to either an api or to our database to retrieve and send information to the user.

Here is a very basic diagram of the components interacting:

User ---- > < ---- Server ---- > < ---- API or Database

**Design Issues**

Movie not accessible- This issue is when a user tries to search for a movie that isn’t in the database. A call to the database for a movie that doesn’t exist in the database could cause problems. Possible solutions include: directing the user to a page that informs the user the movie isn’t in the database and to search for a different movie or directing the user to a suggested movies page informing them the movie they desire isn’t available.

Storage for our database- This issue is if we run out of space on our database to store each user’s, movie’s, actor’s, and director’s information. If we can’t store the information then our site doesn’t work. Possible solutions are to set up a larger database with servers to handle the load or to store less information.

Not enough information for a top 10 list- This issue is when we try to generate a personalized top 10 movie list base on a User’s friends list but if there aren’t enough users in our database with top 10 movie lists then we can’t generate a personalized one for the current user. Possible solutions to this issue is to provide them with iMdb’s top 10 list or to redirect them and say there isn’t enough information to generate a top 10 list but offer movie suggestions to rate.

Friend removal- This issue is when a user wants to remove a friend from their friends list. A solution to this is to have an easily accessible removal button that is directly connected to a friend token in the user’s information on the database.

Change of information- This issue is when a user wants to update or change information regarding their personal profile. The solution to this is to have an accessible update information button that allows the user to store the new information they desire and for us to update the token regarding the field they wanted to update in the database.

Accurate searching- This issue arises from movie titles and people associated with movies having very similar tags for searching. We want to deliver the most accurate results as possible. Possible solutions for this is to have very specific tags for each movie and person and hope the user searches the exact tag but this isn’t always the case so the best solution is to bring up a few results that best match what the user searched.

**Design Details**