Saldua, Samantha Louise

|  |  |
| --- | --- |
| Document design  Movie Team | Cespedes, Luis Joan |

**Introduction**

Movie Wise will be a responsive website that users can use to see information about movies, old, new, and upcoming. Information will include plot summaries, trailers, and critic reviews. While these will be the main functions, we are planning to implement a function where users can sign in, add movies to lists (ex: watchlist, favorites), giving movies ratings and reviewing them (will be separate from professional critic reviews). The website will have a browse function where users can browse by genre, recently released, alphabetical (if possible since there are A LOT of movies) and/or more. The website will also have a search function that will suggest movies as a user type.

**Product Features**

The following is the list of the main features of our application, some features are passive (which mean that the user does not have to perform any action), and some others features are active ( which means that the user have to perform some action).

**1.Website startup**

**1.1.** Request a list of movies to the API (OMDB API)

**1.2.** Load first list of movies into the website

**1.3.** Load movies posters

**1.4.** Set up the responsive design

**2.Change list of movies**

**2.1.** Pick movies by genre

**2.1.1.** Load new list of movies into the website

**2.2.** Set up the responsive design

**3. Select movie**

**3.1.** Request the specific movie information to the API (OMDB API)

**3.2.** Request the specific movie trailer to the API (YouTube API)

**3.2.1.** Set up a movie poster in cases of missing trailer

**3.3.** Set up the layout of the all the APIs responses

**4. Search Bar**

**4.1.** Search a specific movie title

**4.2.** Search a for a title with multiple matches

**4.3.** Search for undefined title

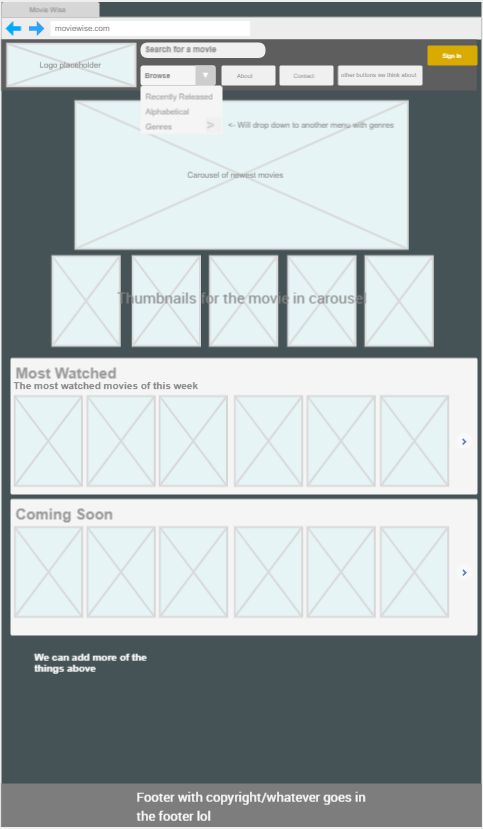
**5. Rating score**

**5.1.** Give a score to a specific movie

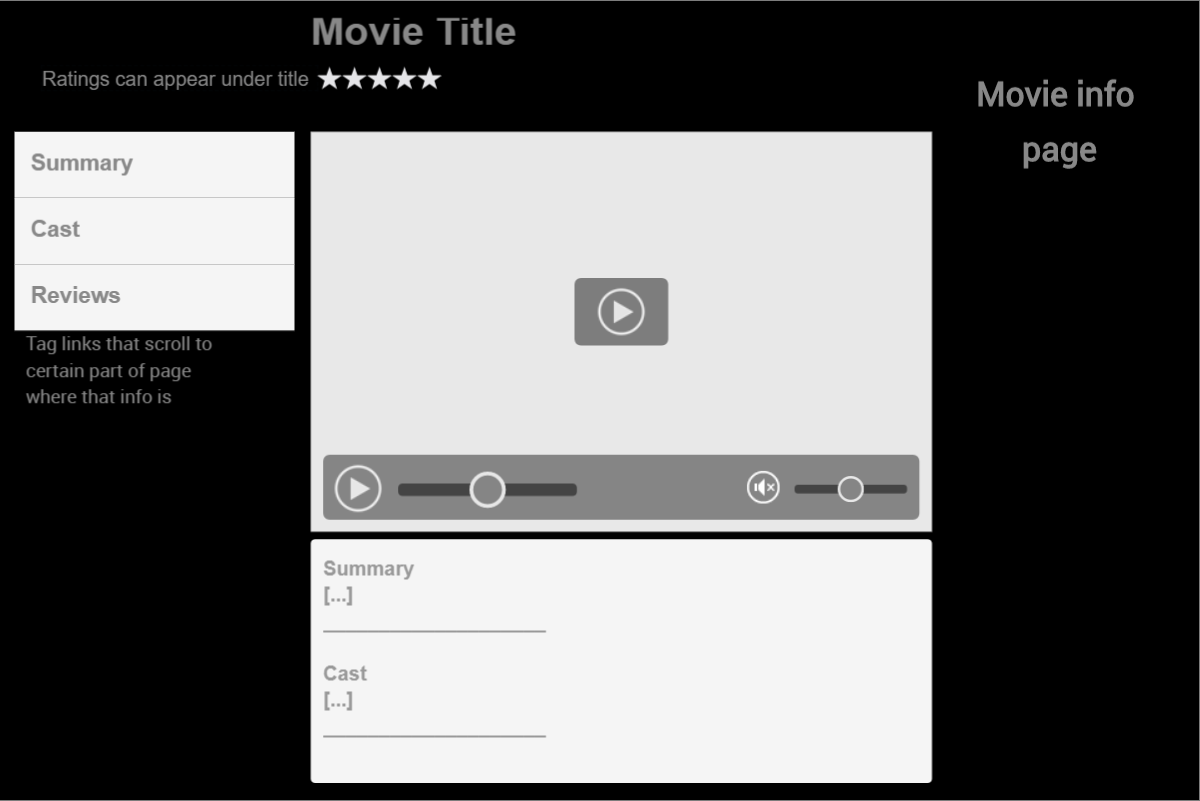
**5.1.1.** Load the average scored of the movie

**UI Sketch/ Design**

**Home Page:**



**Movie Information:**

****

**Searching for a Movie:**

****

**Requirements**

**Hardware Requirements**

The Movie Wise app will require the following hardware equipment in order to provide proper development.

**Server:** this will be the main computer that will host our web app, along with some other locals APIs (created by our team) which will be required for the proper function of the system.

**Mobile Devices:** our team will also need at least two mobile devices with different screen sizes in order to develop, test, and guarantee a proper responsive design. (Mainly a tablet, and a phone)

**Software Requirements**

The following software requirements are essential in order to the success of our application:

**Visual Studio Code:** this is a text editor that provides several tools that will help us with our coding writing.

**Web Browsers:** these are the tools that provide us a code compiler and a UI interpreter, our app will essentially run in the web browsers. as a team, our main focus will be provided support for the three main browsers (Google Chrome, Mozilla Firefox, Microsoft Edge)

**Development Technologies:**

**JavaScript:** this will be our main programing language because is basically the standard of web development, our team have also considered the idea of using React JS which is basically a JavaScript library.

**HTML5:** this is the structure language of the websites, as we all know this is also part of the standard web development.

**CSS:** this is the language in charge to provide style to our web app, the whole design of our system will be done on CSS.

**Java /Spring framework:** this will be our Backend language, we will probably have some features, that require the user information in order to create a comment section, this will turn our app in a semi-social media app of movies. since we are just a two-person team, we are not 100 percent sure of the implementation of this feature, but it is strongly likely to be implemented. We are looking forward to using the Java Spring Framework in order to complete this task so that way we can also take advantage of the Java spring Security embedded in the Spring framework

**MySQL:** This is will be our database management system, this goes along with our java backend, here we are planning to store all the required information.

**OMDB API:** this is the main key to our project, this is a third-party API that provides the general information about the movies, this API will return us JSON data with the movie information based on our requests. Here it is their website <http://www.omdbapi.com/>

**YouTube API:** this API will be in charge of providing us the trailers videos of our requested movies. this API is provided by the Google team. Here it is their website, <https://developers.google.com/youtube/v3>

**React JS:** this is a JavaScript library that will provide us with many useful tools which lead to a faster development process, also we decided to use it because it is one of the most growing trends in web development.

**Non-functional Requirements**

**Responsive design:** in a few words, responsive design allows adjusting the design of our website depending on the screen`s size of the device used, in our project, this is essential since our focus is to get a fast review of a movie in order to take the decision of spending your time on it.

**Safe data storage/Security:** for the security of the data provided by the users, we are going to use the integrated authentication process provided by MySQL, plus this we are going to add another layer of security provided by the Spring Java Framework, this is one of the main features of the Spring framework that called our attention. on the other hand, our group will be very meticulous with the use of our HTTP request, which leads to better encryption of the data sent by out front-end.

**Portability:** since our app is essentially a web app, the portability is something that comes by default since the running environment of our app is the web browser. we are also planning to create an API that returns JSON data, but this API will be created in Java, and we all know about java portability and its virtual-machine which means that our server could be even a Mac or Linux server, so our portability is on good hands.

**Cross-browser design:** as mentioned before, our web app will be mainly supported on the three main web browsers (Google Chrome, Mozilla Firefox, Microsoft edge) this might represent around 90% percent of the browser used in the world.This could be another key feature of our portability.

**Performance:** in the performance, we are using some external APIs and some frameworks and libraries, we understand that this is translate in an average performance. In contrast, we are 90% percent sure of using React JS, which improves the performance since the HTML code and even some small part of the CSS code is created dynamically from the source of the root JavaScript code integrated into React JS.

**Risk factors and Proposed Solutions**

**Absence of a teammate:** This issue may hurt our team in a worse way than other teams, since we are just 2 members the absence of 1 of us will put all the weight of the project just on one person. Our solution to this problem is to have a well-detailed list of tasks, and instructions which will lead to all the steps in the project development, so this way the project may be doable for one person. Our team has a strong commitment to this project, but we understand that anything could happen, and this is why we share between us numerous tutorials about all the technologies needed for this project.

**HTML, CSS, JavaScript, Java, MySQL no longer supported:** we grouped this answer for all these technologies because this is VERY unlikely to happen, but in the case that all our main technologies are no longer supported by the time of creating the project we have many other alternatives to achieve our end goal. for example (we can just create a local app in C# with the same goal and focus). This risk factor is very unrealistic but been a little skeptical sometimes helps.

**React-JS no longer supported:** in the case that Facebook discontinues React JS which is also unrealistic, we will do the project with just plain JavaScript, because the main point of using React is just to speed up the development process, therefore this library is not essential for our project.

**Java Spring Framework no longer supported:** in the case that we can not use the Spring Framework, we will just use plain Java this will also hurt our project timing since Spring will help to significantly reduce the backend development process, another key factor that we will lose is part of our security, but we can fix this with some security libraries of JavaScript. This risk factor is very unlikely to happen.

**OMDB API no longer supported or change on the user’s policies:** This risk factor is very probable since this API is just a third party software from a not well-known team, this may be one of the worse things that could happen to our team since most part the useful information is provided by this API. Our solution is to have other APIs which provide similar information, and indeed, we have several chooses.

**YouTube API no longer supported or change on the user’s policies:** we are using the YouTube API to get our movies trailers videos, in case that for some reason we cannot use the API anymore, we can extract the video directly from the IMDB web-site, even though this task add more complexity level it is doable.

**Use Cases**

**Project: Movie Wise** -**Movie Information Website**

**Project Team:** Movie Team

**Use Case:** See Movie Information

**Description:** See a certain movie’s information

**Primary Actor:** User

**Secondary Actor(s):** None

**Trigger:** Left mouse button click on movie title/thumbnail

**Pre-Condition:** The website must be loaded first

**Post Condition:** Success. Movie information is displayed to user

**Main Sequence:**

1. User clicks on a movie title/thumbnail on the homepage
2. Site loads the page dedicated to movie
3. User is able to scroll through page to see the movie’s info

**Alternative Sequence(s):** Click the search bar, and proceed to Find a movie using the search function(which is the next use case)

**Exceptions:** The movie information could not be found, due to lack of information in the API

**Author:** L. Cespedes

**Date:** 23 September 2020

**Project: Movie Wise** -**Movie Information Website**

**Project Team:** Movie Team

**Use Case:** Search for a movie

**Description:** Find a movie using the search function

**Primary Actor:** User

**Secondary Actor(s):** None

**Trigger:** Entering a string input in search bar

**Pre-Condition:** The website must be loaded, and the search bar must be clicked first

**Post Condition:**

Success. User will be taken to movie’s page if an exact match was found, or taken to page that lists closest matches.

**Main Sequence:**

1. User clicks on the search bar to start typing
2. User inputs the movie title they are searching for
   1. As user types, search bar can drop down to show suggestions
3. User finds the movie they are looking for on the drop down suggestions and clicks on it
4. User is taken to that movie’s info page

**Alternative Sequence(s):**

1. User does not find movie they are looking for on the search’s drop down suggestions
2. User presses search button
3. User is taken to search result page of the input’s closest matches

**Author:** S. Saldua

**Date:** 23 September 2020

**Object/ Class list**

**Movies:** this class will hold all the information provided by the OMDB API, also this class will perform the proper adjustments to the data in order to use it in our app.

**SearchBar:** this class would be in charge to perform the queries and to sort the responses from the Movie-class whether it matches or not with the typed request from the user

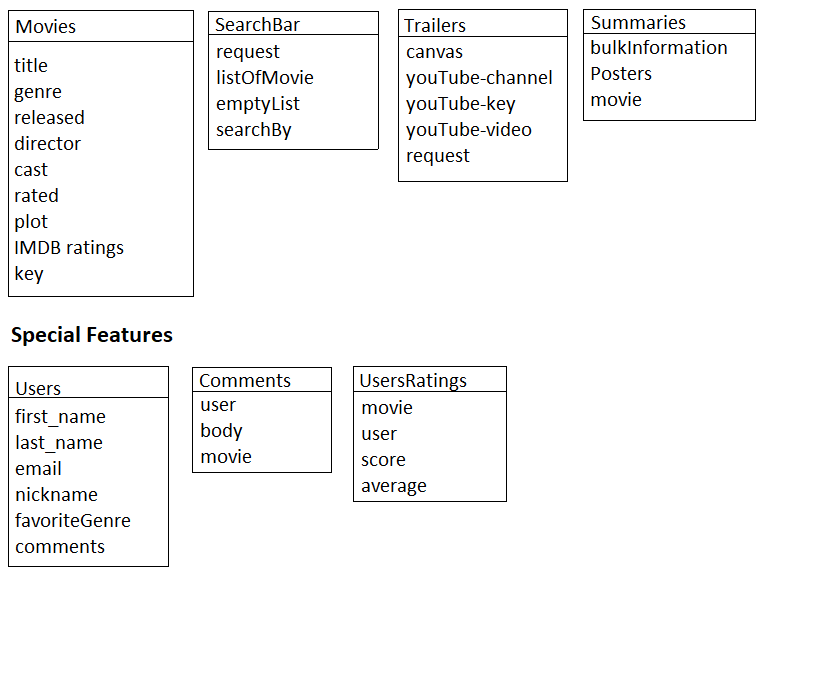
**Trailers:** this class will search for the movie requested, and perform queries on the YouTube API, in order to get the appropriate trailers for the movie

**User:** this class will be the one in charge to deal with the MySQL database in order to hold all the information from the user, the user information will be requested whenever the user decides to sign up on our app. (special feature)

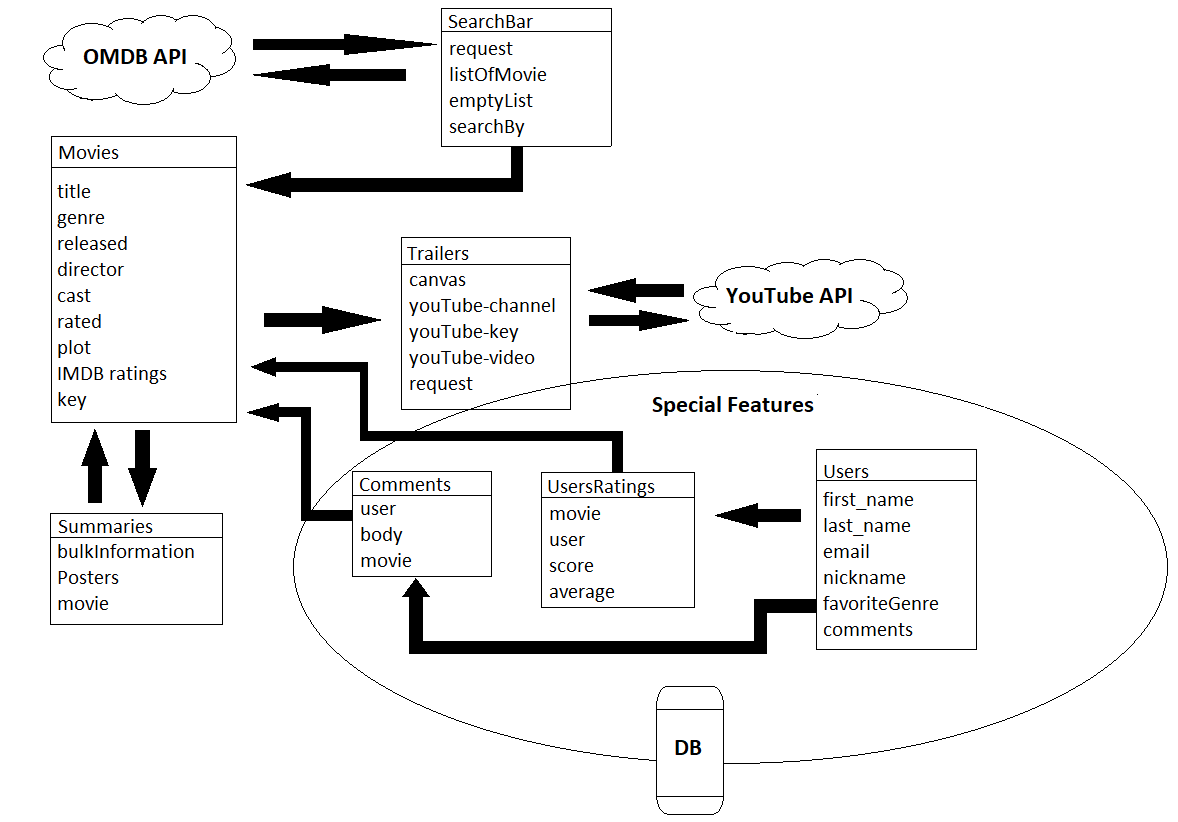
**Comments:** the comments class will basically hold the structure of the comments section, this class will store all the comments on our MySQL database, also relate them with the appropriate movie and user. (special feature)

**Rantings:** this will be the system where the user can vote for the movies, this class will be the one that generates all the custom scored for the movies based on the user's votes. (special feature)

**Classes Diagram (with possible field and methods)**

****

**Classes Relationship Diagram**

****