

**CEN 4010 Principles of Software Engineering**

Summer 2020

**Group 2 “The Fam” Evaluation:**

-Oscar Aquino (25%)

-Ryan Bates (25%)

-Elizabeth Garcia (25%)

-Jesse Kelly (25%)



Revision 2.1 7/14/2020



**CEN 4010 Principles of Software Engineering**

Summer 2020

**Milestone 3 Detailed Description Vertical Prototype**

**Vertical Prototype – “What 2 Watch”**

Oscar Aquino - [oaquino2017@fau.edu](mailto:oaquino2017@fau.edu)

Elizabeth Garcia - [elizabethgar2017@fau.edu](mailto:elizabethgar2017@fau.edu)

Ryan Bates – [batesr2013@fau.edu](mailto:batesr2013@fau.edu)

Jesse Kelly – [jkelly2019@fau.edu](mailto:jkelly2019@fau.edu)



A website dedicated to the establishment of quality choices in what to watch established by the consumers, and how to find them.

**EXECUTIVE SUMMARY**

Do you often find yourself wanting to watch a movie, and then scrolling through the feed only to be discouraged after failing to identify with any of the titles? “What 2 Watch” will be launched with an effort to organize your options into an environment that provides you with a list of options that will resonate with you. One of the key advantages this platform will provide is the ability to have genre related discussions which ultimately should produce a large quantity of suggestions within the comment section alone. This will then create the opportunity for our users to copy and paste whatever might sound interesting to them in the search bar.

The user will have the option of saving any movies that spark their interest into a list of favorites. Being able to create this list will eliminate the annoying task of trying to store a friend’s suggestion in your head for some odd portion of the day. One of our main goals is to provide a comfortable platform where the user can keep their portfolio of favorites all in one place, but we also want to create an environment that ensures a quick solution to the common problem of taking too long when deciding what to watch. Our user interface will include an easy to use navigation system that will allow for quick browsing and simple solutions.

We want to encourage our users to enjoy the process of deciding what to watch, so in an effort to include the folks that like to take their time browsing, we will offer a section that allows the user to flip through a list of suggested titles two at a time. This flip section can be filtered as generally as sorting through all available titles, or as specifically as flipping through a genre that aligns with the user’s preference. Underneath each title we will present the user with the movie cover as well as a short description of the film.

Overall, what we value the most is how our users will enjoy their experience on our platform. This inspires our team to create something that everyone can relate with. Ultimately, it seems not enough thought is put into a plan of attack when it comes to deciding on what to watch, so we’ve decided to do the thinking for you.

**COMPETITIVE ANALYSIS**

What 2 Watch offers you a better and simple way to find, organize and select movies to watch. Here are our key features in comparison with our competition:

|  |  |  |  |
| --- | --- | --- | --- |
| Features | What 2 Watch | Rotten Tomatoes | IMDb |
| User-friendly |  |  |  |
| Effective navigation |  |  |  |
| Web compatibility |  |  |  |
| Social networking |  |  |  |
| Rating system |  |  |  |

What 2 Watch offers a user-friendly platform designed for people to organize their watch list. Our product makes it simpler to search and find names of movies of your liking. Unlike our competitor, we will provide you with the ability to join to discussions related to the genre of movies you are interested in and give you the opportunity to share different titles with the community on the discussion. The rating system will rank the movies with a rating in numbers and not only “good or bad” as our competitors, making our product unique.

**DATA DEFINITION**

“What2Watch” refers to the name of the website.

“User” refers to a registered visitor.

“Visitor” refers to a non-registered visitor.

“Administrator” refers the special user in charge of moderation.

“API” refers to the host, from which movie entity information is aggregated from.

“Movie Search” refers to the function (search bar) responsible for the aggregation of movie meta data.

“Registration” refers to the function of registering for our website and collection of personal information.

“Movie entity” refers to the results presented by the API. E.g. Title, synopsis, etc.

“Movie(s)” a film title from which a user or visitor can find specific meta data via movie search.

“Store link” refers to the hotlinks of vendors that support that title.

“User chat” refers to the subpages reserved for user discussion or the chat box from which the user can join to discuss.

“Collection” the user(s) personal library of favorite movies.

“User list” the function which aggregates all the user favorite movies into a collection.

“Rating” refers to the rating given by the user in a movie entity page.

**OVERVIEW, SCENARIOS, USE CASES**

A visitor visits What2Watch in need of a one stop shop of meta data for any movie(s), to participate in user discussion, and/or to be able to add their favorite movies in a collection from which the user can then filter as he wishes.

**Initial Assumption:** The user/visitor have one or more movies they want to search, discuss, or in need of wanting to form movie collection. The user/visitor have had a chance to be exposed to Watch2Watch features via our design, advertising, or interaction.

**Normal:**

* The user/visitor has a movie in mind, using the movie search they are re-directed to a page showing the user/visitor that specific movie entity. The are also exposed to store link(s). If they are a **user** then they can add that movie entity into their **collection** or have a **user discussion**.
* The user/visitor wants to sign up for What2Watch. They must go to the sign-up page and type in their personal information such as their email and password. They should also be given the option to sign-in if they are already a user.
* The user wants to view his collection, under the user home page they can view their movie collection. All the movies are ordered by the user list and each movie entry can take them to that specific movie entity.

**Concurrent events:**

* The user wants to purchase a movie then they can use the store link from the movie entity page and get re-directed to an external vendor. This vendor is not a part of What2Watch.
* The user wants to participate in a discussion a specific movie genre, from the movie entity page they can join a user chat and participate.
* The user wants to rate the movie, then they can leave a rating on the movie entity page, changing the global score that all users see

**What could go wrong:**

The user chat can get hijacked and spam the chat. The rating system can get hijacked resulting in non-organic ratings. No administrator is present so chat can be spammed and hijacked.

The API can go out of service resulting in no entities being aggregated.

The store link can expire and need to be reupdated.

**Ending state:**

Upon completion of the signup page the user will go to the homepage.

Upon entering a movie entity, the ending state is the aggregated page for that movie entity. Meta data such as title, artwork, trailer, and synopsis are visible.

Upon leaving What2Watch movie entity page, via store link, they are directed to an external vendor.

Upon entering user collection, their movie entries are displayed, and the user list is functionable. Titles are visible and ready to be clicked on and taken to that movie entity.

Upon using the chat system, the user remains in the movie entity genre, being able to see their comment within the user chat.

Upon entering their rating for that movie entity, they can see the updated global score for that movie entity while remaining in the entity. No refreshing is required.

**HIGH-LEVEL FUNCTIONAL REQUIREMENTS**

1. Search - A user shall be able to search the list of movies by title or genre.
   1. The search section must have the ability to find the movies by using key words like a word on the movie title or genre from the catalog. A list of movies will show on the screen for the user to choose which is the one they have looked for.
2. Register - Each registered user shall be uniquely identified by his/her username and password
   1. The register section must have different text input boxes asking the new user their information. User will be asked to insert full name, email, create a password and date of birth.
   2. User will be prompted to log in adding the email and password. After log in the user will have access to their unique profile and page. User will be able to update personal information as well as username.
3. List - Each registered user shall have the ability to create and save their list of favorite titles.
   1. The list feature must have the movies selected by the user to be saved as favorite or watch later.
   2. the list will have an option for the user to delete the title if the user does not want to have it on the list anymore
4. Comment - The system shall allow the user to post comment sections under the movie genre.
   1. The comment or discussion section must have a text input box for the user to share opinions and/or suggestions on movies.
   2. The user will be able to see and reply to other users comments, the comment will have the date when the comment was posted.
5. Rate - The website shall be able to collect the users rating of each movie title.
   1. User will have the option of giving a thump up or down as the rating system. Thumps up will equal to “like” and thumps down will equal to “dislike”.
   2. The amount of like and dislike will help the user to decide if the movie is worth to be watched.

**LIST OF NON-FUNCTIONAL REQUIREMENTS**

*Operational capabilities and System Constraints*

-

**Performance/Scalability** – A measure of how the software will be in respond to different workloads. We want to make sure the software is returning results in a responsive matter under all circumstances.

**Compatibility** – We want to make sure that our software will be able to run on desktops, laptops, and any smart devices that feature a browser.

**Reliability/Availability/Maintainability** – In the event of a crash or failure of some sort, what sort of availability is being offered?

**Security** - How the software and its data is protected as well as the a guarantee of the user’s safety when entering personal information.

**Usability** – We must consider the level of difficulty that comes across when a user meets our site. We want to keep it simple, with a sleek and easy to use design.

**Constraints**

**Response time** – The software will only retrieve outside data from one API containing all the necessary information as an outside source. Flags can be used for delays after 0.5 seconds, 5 seconds, and 10 seconds where if it reaches an unresponsiveness that lasts for 10 seconds the software should provide some solution based on the exception thrown.

**Scalability** – We will define the capacity of the software in terms of users to be 3,000 users for the first version that we put out in order to protect the responsiveness of the site.

**Hardware** – Low level hardware specifications will be required in order to welcome as many devices as possible

**Software** – Software will support new operating systems without the consideration of outdated versions.

**Reliability** – Based on the capacity limit of the users, if the software reaches a 10 second delay, this information can be stored in a count. If the count reaches more than 3 crucial delays, we will temporarily have the site under construction for 24 hours to fix the issue.

**Security** – Only a select few members (developers) will have a management status for on site issues. Software should be compliant with regulation user security. Heavy focus on protecting user information as no payment information will be required.

**Usability** – How easy is it to sign up? How easy is the search function?

How does the site appeal to the user from both a visual and technical perspective? User must feel satisfied and they must be able to reach their goals efficiently.

**HIGH-LEVEL SYSTEM ARCHITECTURE & DATABASE ORGANIZATION**

**PLATFORMS**  
-Google Chrome  
-Internet Explorer

Development will focus on implementation to the above listed platforms primarily because one comes standard with one of the most widely used operating systems on the market, and the other is the most widely used browser on the market.  
  
**DEVELOPMENT LANGUAGES**  
-html <https://bootstrap4.com/hotflix-online-movies-tv-shows-cinema-html-template/>  
-CSS <https://bootstrap4.com/hotflix-online-movies-tv-shows-cinema-html-template/>  
-bootstrap <https://bootstrap4.com/hotflix-online-movies-tv-shows-cinema-html-template/>

Primary web development languages we will be using while designing the website. Html, CSS, and bootstrap will be designed and implemented in a means to give the site an appealing look that makes users want to use the website.  
  
**INTERFACING/DATABASE LANGUAGES**  
-php <https://www.php.net/license/index.php>  
-JavaScript <https://bootstrap4.com/hotflix-online-movies-tv-shows-cinema-html-template/>  
-firebase <https://firebase.google.com/terms>

Database languages will be implemented to give a secure means of connectivity. Internal Chat will be implemented allowing users to communicate their ideas regarding movies and shows. Firebase is the primary means by which we wish to use to accomplish the setup. If the interface proves to not allow full capability the aged, but well established php is a backup plan.

**ORGANIZATION**

Landing Page with multiple subsections containing information regarding all movies listed on the site. The subsections will contain the chats and information detailing the movie for users to interact with.

Database Information pertaining to Login information will be contained as listed in the HIGH-LEVEL UML Diagrams.

**MEDIA STORAGE**

Most media will be stored on outside servers as the website is dedicated to finding quality shows to watch. So when the user decides they have found a show they wish to watch be it based off a trailer linked on the page or based off the chat taking place on the website. They can then follow one of the links to various other locations hosting the streaming media.

**SEARCH**

The website has a search feature for finding the media you wish to locate upon our site. A quick means by which to find a show you are after.

**SIGNIFICANCE**

Users can participate in up or down voting of shows they watch to show their view to others regarding the media. It will permit others to more easily decide if they wish to watch the show. A lower rating will mean less people enjoyed the show so users can skip it. At the same time a higher vote could mean the movie or show is worth watching.

(UML Diagrams moved from this section to HIGH-LEVEL UML Diagram section that was added)

**HIGH-LEVEL UML DIAGRAMS**

**USER ENTRY**



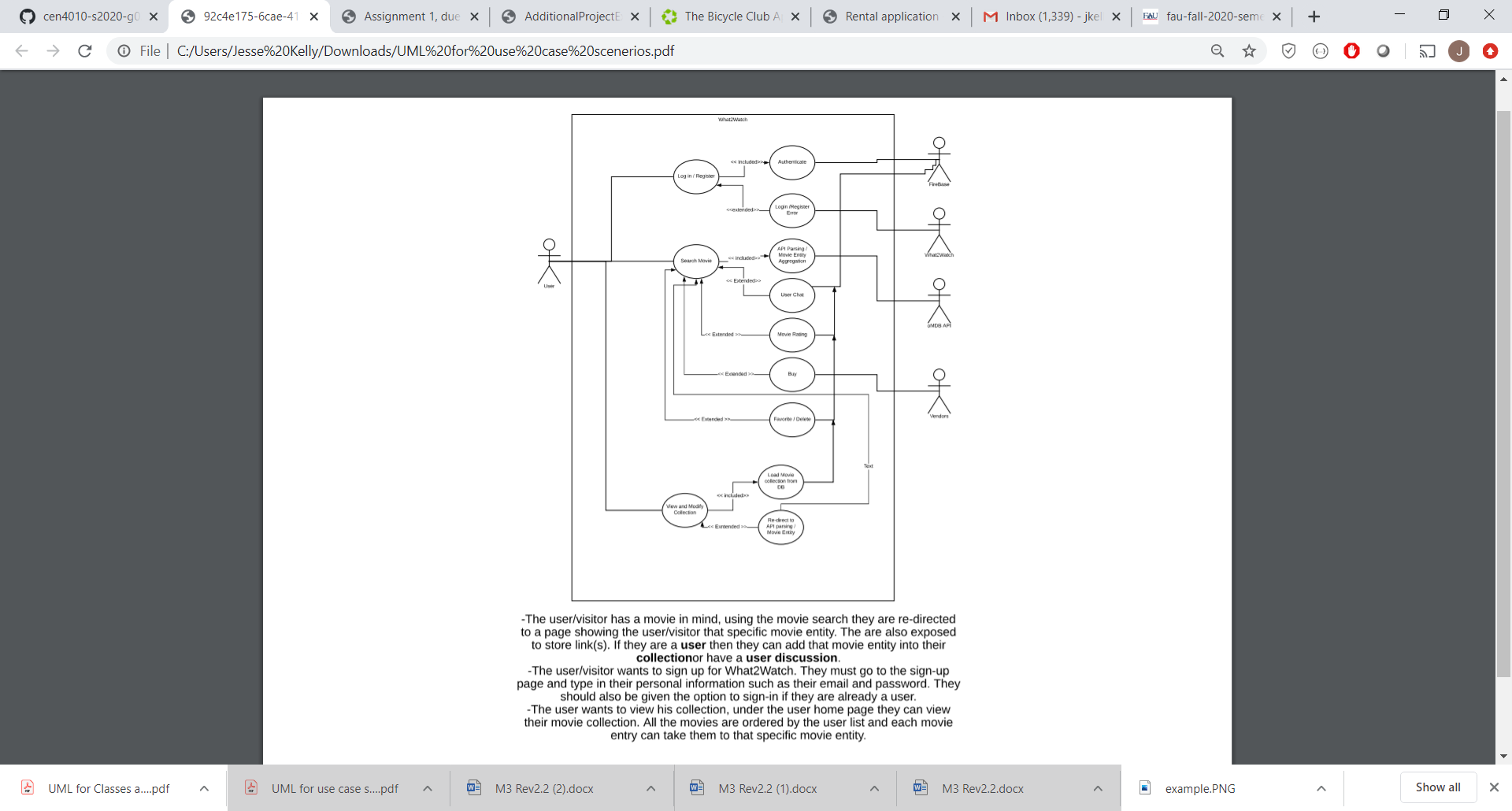
Users will load the browser then have the option of logging into the site to have higher website functionality associated with their account. They can then use features such as chat.

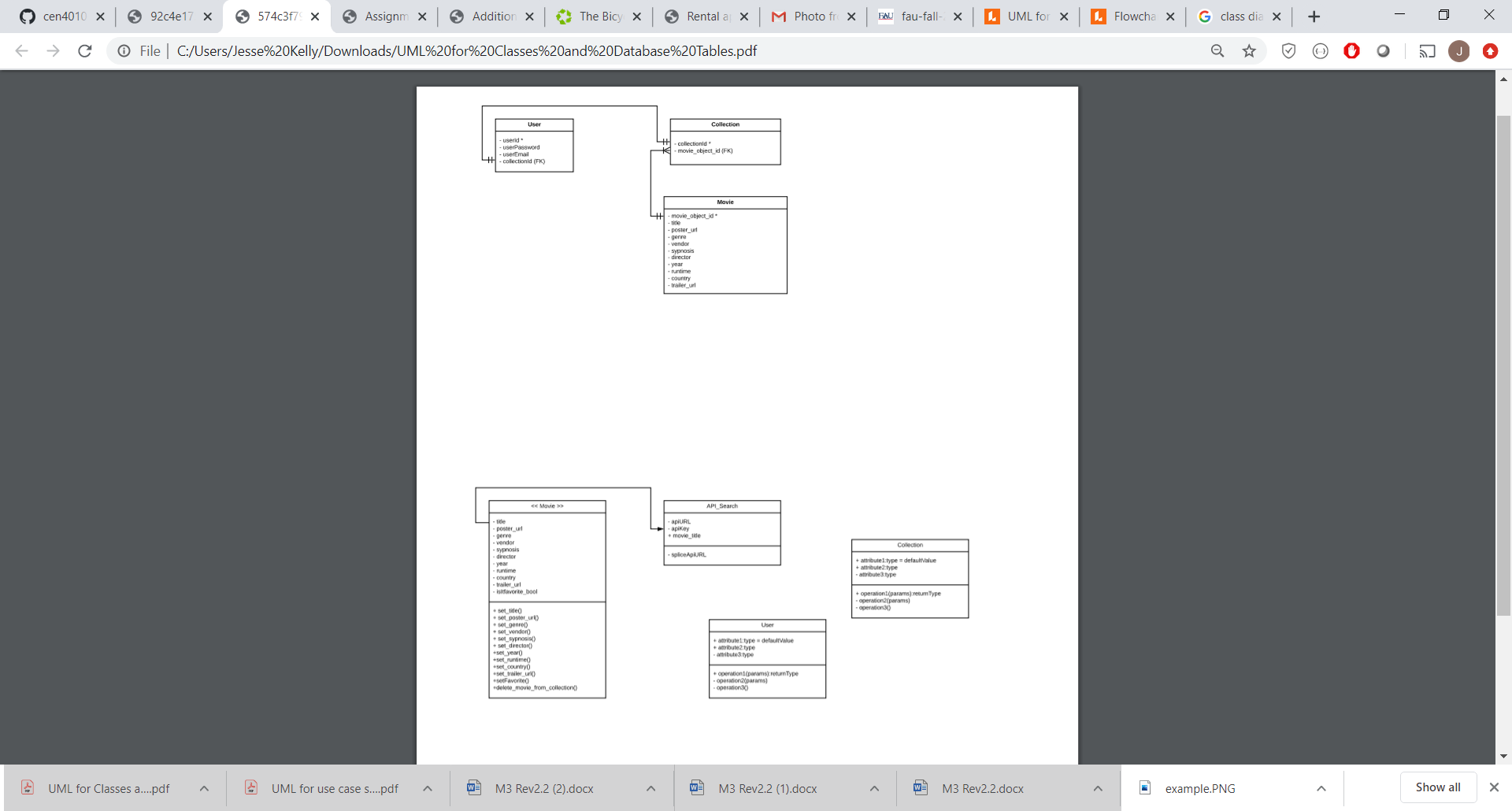
**USER CHAT**



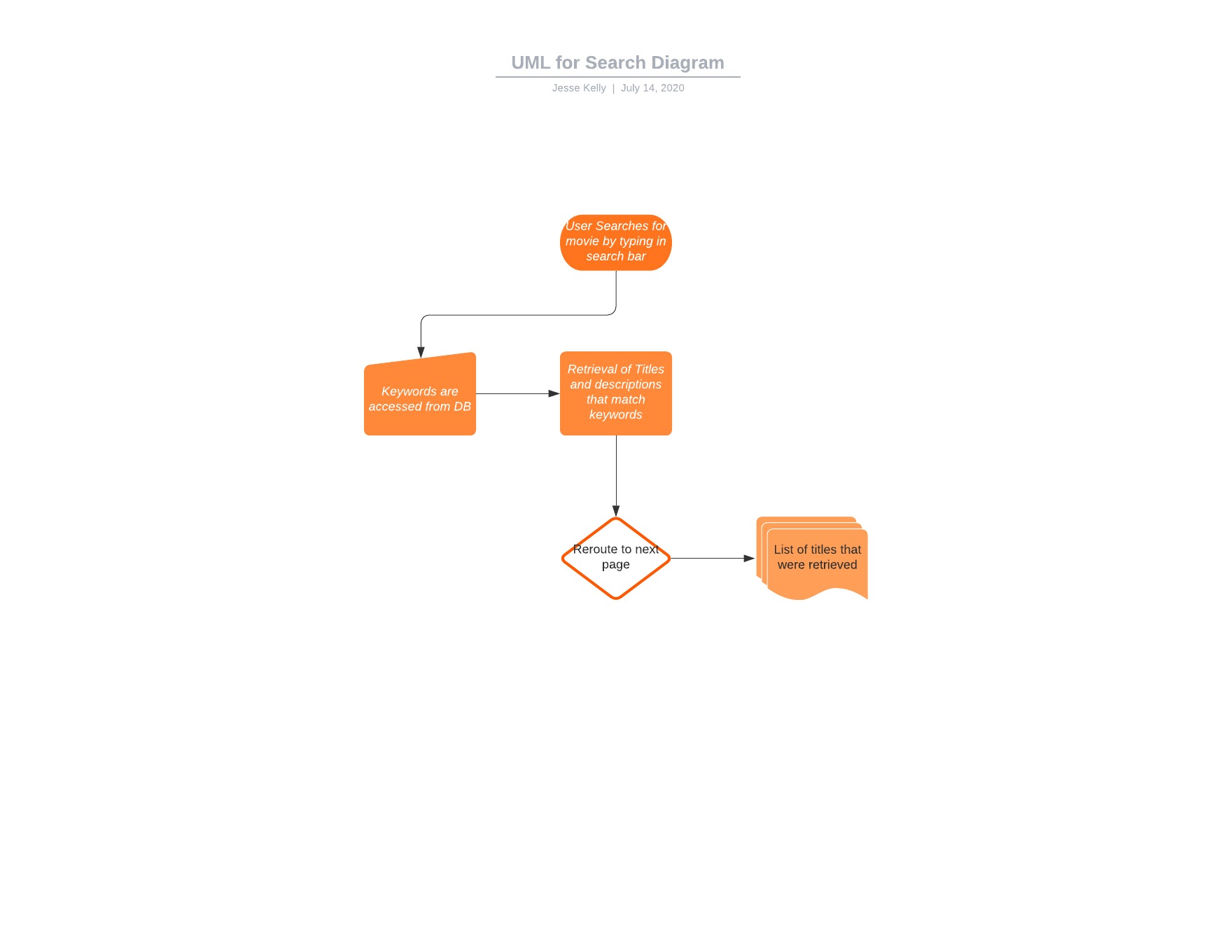
Multiple users can talk with each other discussing details about their preferences, and anything else they wish to discuss revolving around the movie subsection they are currently in.

**UML for use case scenarios:**



**UML for Classes and Database Tables: (in progress)**

**UML for Search Engine Functionality:**

****

**ACTUAL KEY RISKS FOR PROJECT**

1. Skills risks (do you have the right skills)
2. Schedule risks (can you make it given what you committed and the resources),

Setting up a time for us to get together has been the challenging part, as we all have full time jobs and different shifts as well as other classes and responsibilities. Whatsapp has been a great tool for us to communicate. We have set up times for each of us to work on the project as well as the work that needs to be done.

1. Technical risks (any technical unknowns to solve),
2. Teamwork conflicts

As many teams we have disagreements regarding the final product.

1. Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright).

We have looked into the options to avoid copyright problems. As our website will show the user movies and where to watch, we will have to make sure to give credits to the companies who created the movie as well as the website where the user will be able watch it.

**TEAM**

Front End Team Leader/Scrum Master: Ryan Bates  
Back End Team Leader: Oscar Aquino  
Github/Trello Master: Elizabeth Garcia  
Product Owner: All Software Developed and owned by Group 2  
Software Developers: Oscar Aquino, Ryan Bates, Elizabeth Garcia, Jesse Kelly