

Status Report 2

Group 1

Tristan Allen, Daniel Carter, Will Cox, Josiah Jackson

4/7/24

1. Introduction

1.1 Highlights

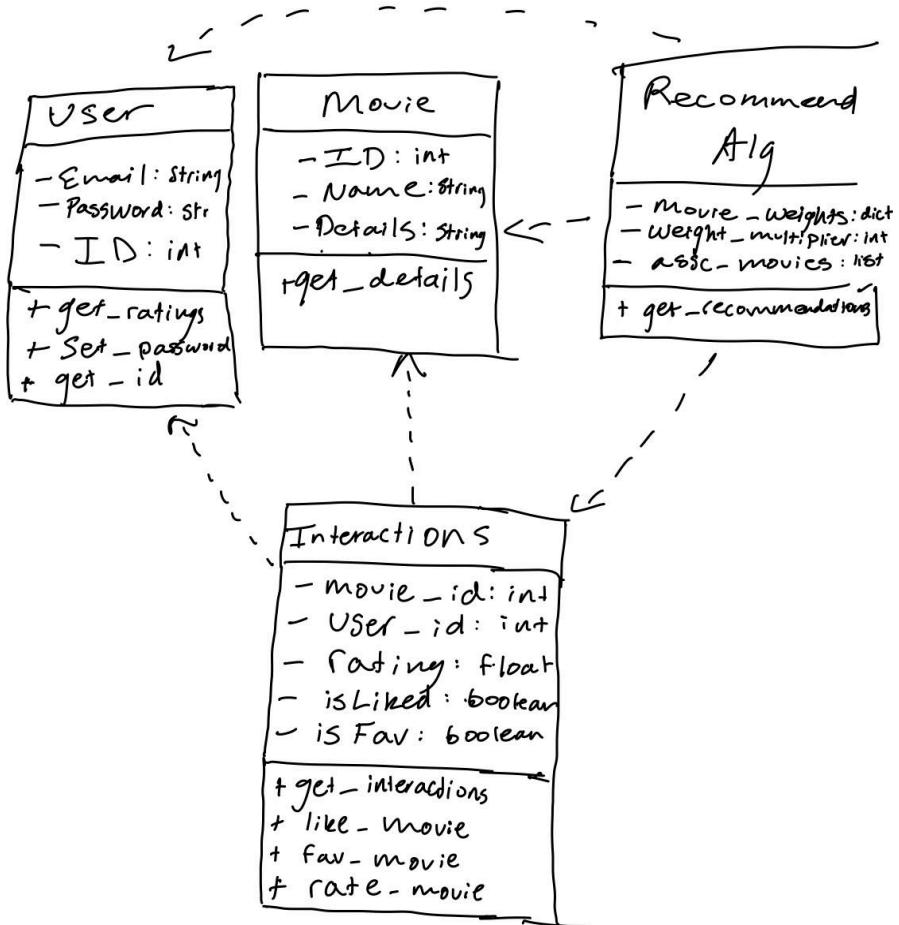
Our plan for this iteration was to implement all of our basic functionality this sprint. This included liking, rating, disliking, recommending, and searching for movies. Each team member completed work items that were integral to the implementation of these features. The following tasks were implemented and completed in our previous iteration:

<input checked="" type="checkbox"/>	implement favorite/like/dislike movie api resource	↓↑	>
<input checked="" type="checkbox"/>	implement rate movie api resource		
<input checked="" type="checkbox"/>	need to add error handling around sql calls in from sql_query		
<input checked="" type="checkbox"/>	Wireframes for home page and implement v2		
<input checked="" type="checkbox"/>	Wireframes for recommendation page and implement v2		
<input checked="" type="checkbox"/>	Wireframes for profile and implement v2		
<input checked="" type="checkbox"/>	replace old movie review if a movie is re-reviewed		
<input checked="" type="checkbox"/>	Add cascade on delete for users to delete all associated user data from database		
<input checked="" type="checkbox"/>	add recommendations table to database		
<input checked="" type="checkbox"/>	determine which crew to include in movie database		
<input checked="" type="checkbox"/>	Prevent sign-up with duplicate username		
<input checked="" type="checkbox"/>	add error handling for missing values in login POST request		
<input checked="" type="checkbox"/>	add error handling for missing values in signup POST request		

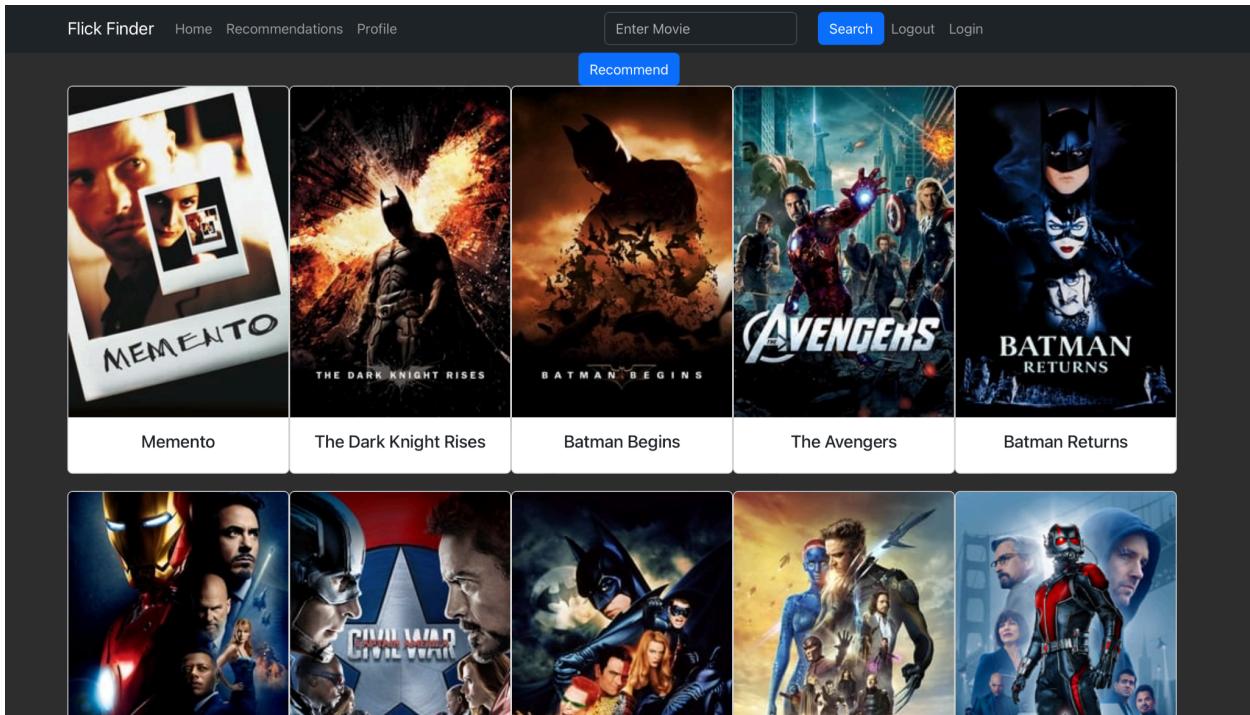
1.2 Changes

- Database Changes
 1. (3/26) Added a likes table to the SQL database so that we can store the movies that the users enjoyed and use their likes to better recommend movies that they would enjoy.
 2. (3/26) Added a recommendations table to the SQL database so that we can store movie recommendations for any movie given that the user has liked or given that movie a high rating. It is essentially just storing a list of movies associated with a single movie that are all similar so that we can provide recommendations faster and more efficiently.

2. UML: Class Diagram

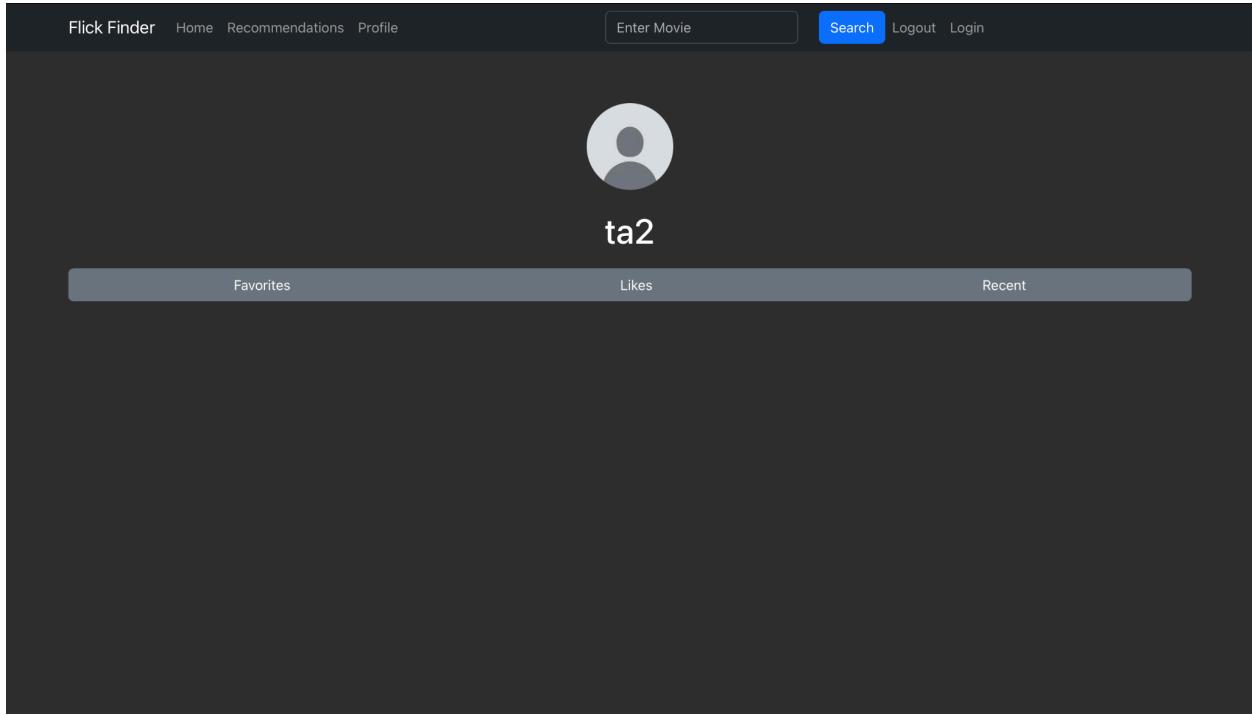


3. Current Status

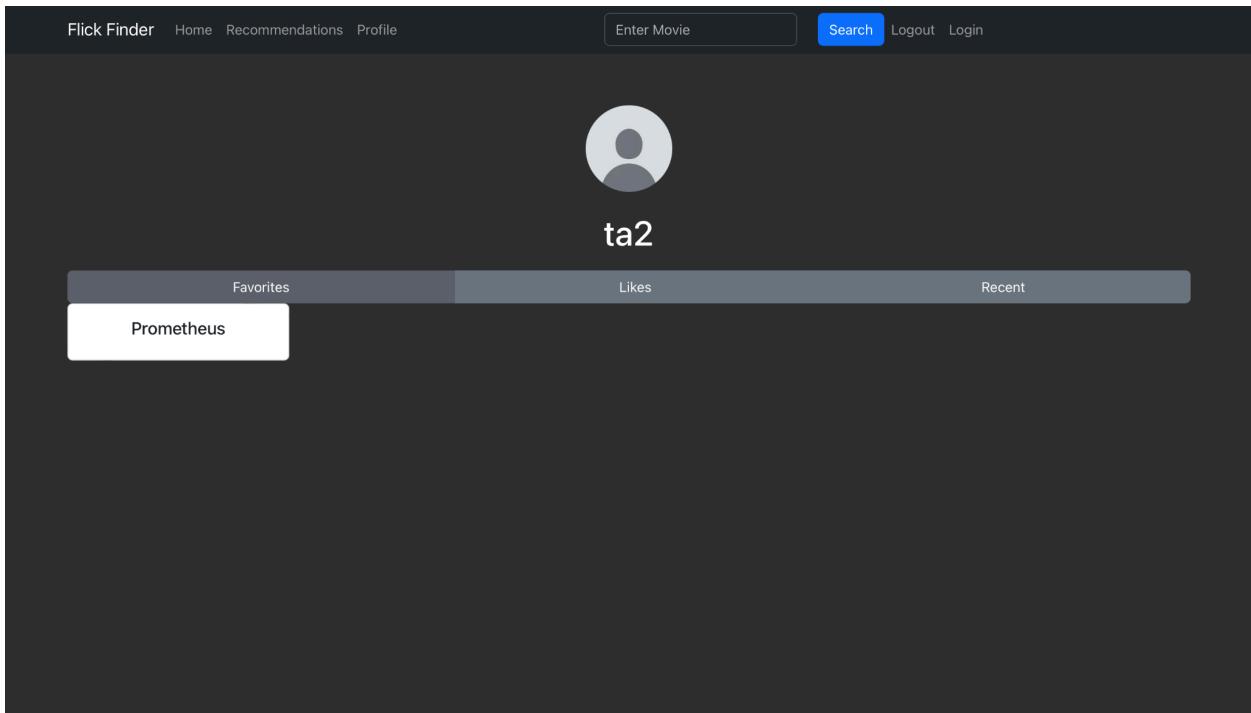


This screenshot shows our recommendation functionality that is apart of the recommendation class. Through our movie weights, we determine which movies should be recommended to a user by running each movie through a content based recommendation system, then weight the movies based on previous movies that the users has interacted with. This is the primary function of our

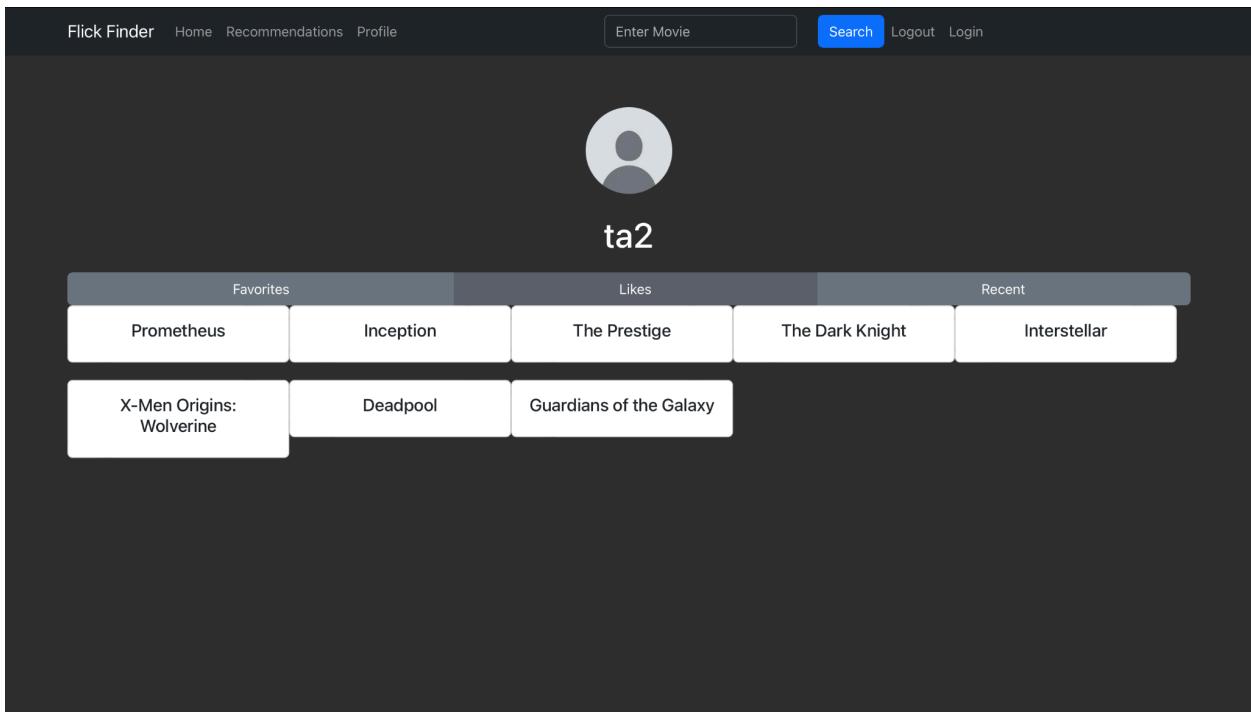
recommendation class.



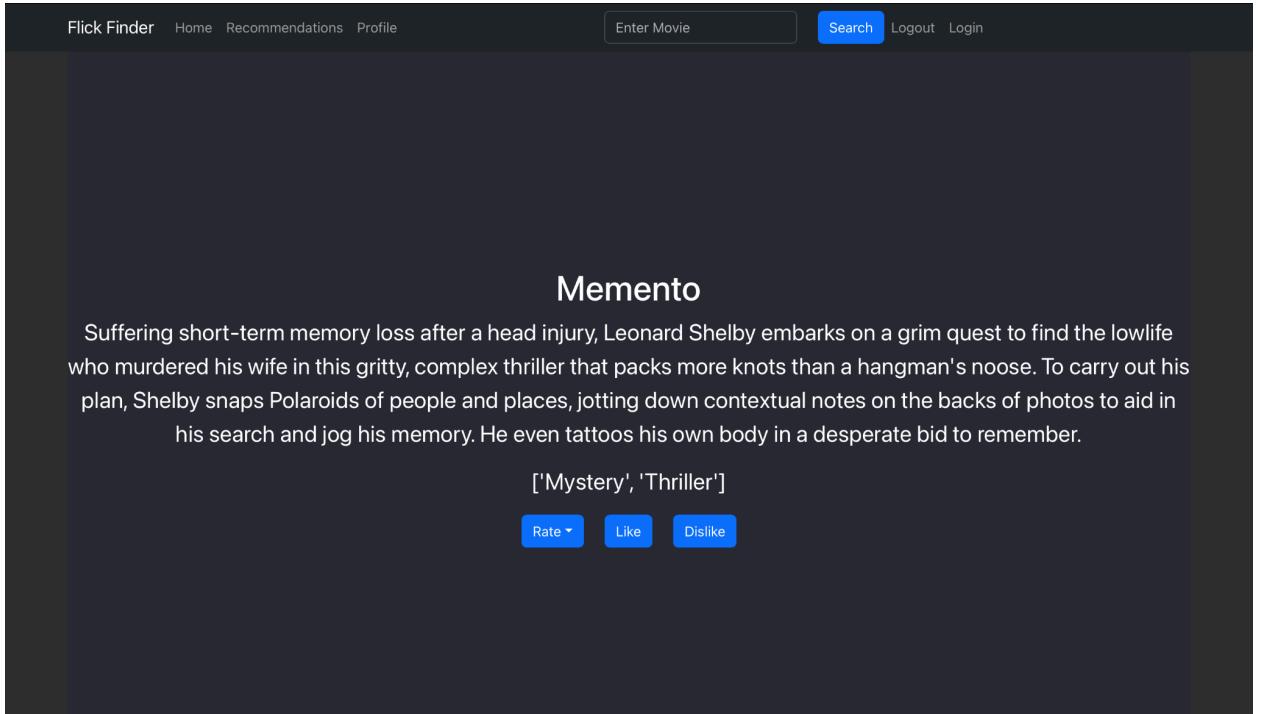
This screenshot shows the second implementation of our user profile page. This feature belongs to the interactions, movies, and user class. It is a profile showing the user's username, as well as the user's favorite movies and movies that have been liked by the user. The recent tab is not currently functional. Each movie of the movie class is loaded when the profile page is initially rendered. The `get_interactions()` operation is used from the interactions class to load the movies that a user has liked, disliked, or favorited.



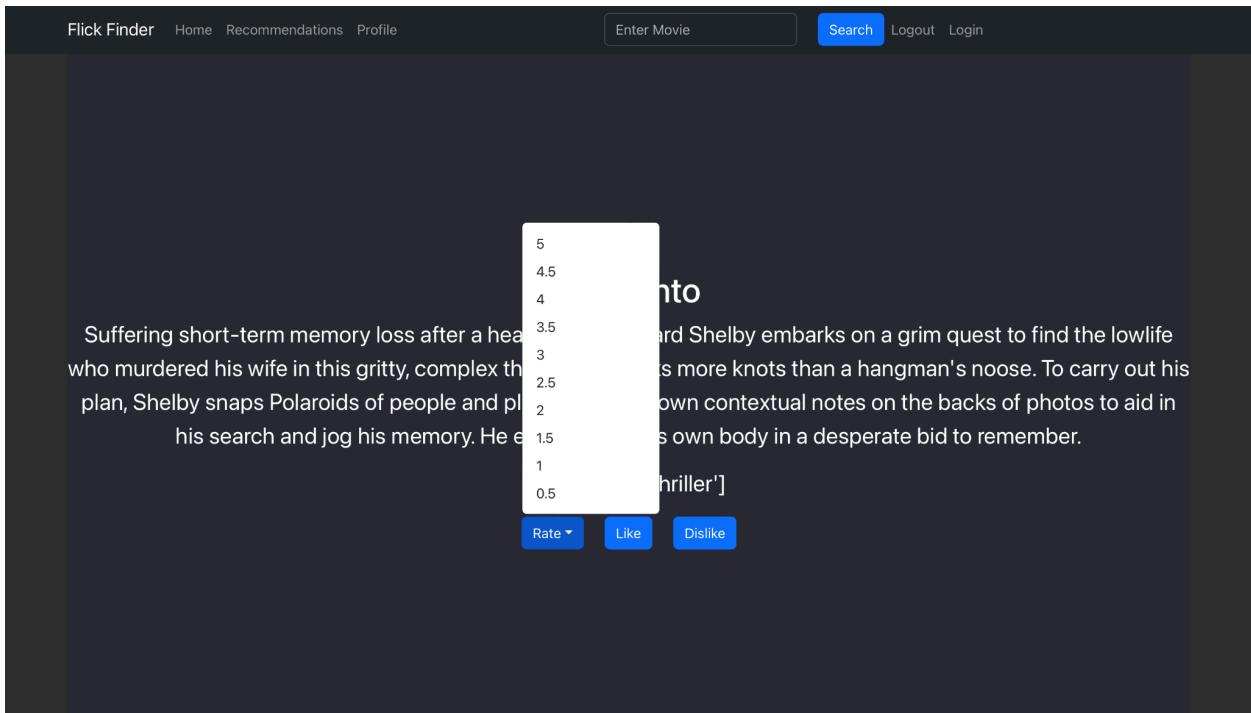
The above screenshot shows the profile page displaying movies that a user has set as a favorite. There is currently no frontend functionality to add favorites, so this movie was added using a curl call directly to the backend.



The above screenshot shows the profile page when the user has selected to display all liked movies. Note that favorite movies are included in the liked movies selection.



The above screenshot shows the movie page and the addition of components to like, dislike and rate a movie. This page belongs to the movie, users and interactions classes. Each movie's details are found for the movie using the `get_details()` operation in the movie class and then displayed in the frontend. For each component the interactions class uses `rate_movie()` and `like_movie()` to add the user's input to the database. The `user_id` from the user class is used in each interaction call to ensure that the information is saved for the correct user.



This movie belongs to the iteration class. The user can rate a movie from 0.5 - 5. Once the user rates the movie the rating is stored in a MySQL database. We can return the rating a user gave a movie through the `get_rating()` function in the iteration class.

4. Project Management

Date	Description
1. 3/10 2. 3/26 3. 3/26	1. Decided to implement user authentication in backend instead of frontend 2. Added a likes table to the SQL database to store user information on liked movies

	<p>3. Added a recommendations table to the SQL database to store similar movies so that the recommendation process is more efficient.</p>
--	---

5. Review and Retrospective

- This sprint we did a pretty good job distributing work so that each person was working on a separate piece of the system. That allowed for efficient merging at the end of the sprint with only minor merge conflicts.
- We couldn't really think of anything that went wrong this sprint. We finished designing the core functionality of our system and are ready to make some styling changes for the final push.
- We met all of our goals for the sprint.
- What could possibly be improved for the next sprint is more asking for help or clarity to potentially mitigate any further merge conflicts.
- In the next iteration we will try to have more short check ins instead of one long meeting to try and ensure the teams stays on the same page for the final push.

6. Team Management

- Team Roles:
 - Tristan Allen: Scrum Master/Developer
 - Will Cox: Product Owner/Developer
 - Daniel Carter: Developer

- Josiah Jackson: Developer
- Team Member Contributions:
 - Tristan Allen: Backend Development
 - Daniel Carter: Updated Database Schema/ Updated Data Importer
 - Will Cox: Frontend Development
 - Josiah Jackson: Exception handling, testing
- Our communication has improved as a group compared to our first iteration but there is still room for improvement. We all have rather busy schedules this semester but we were able to meet twice a week on Sunday and Wednesday for a stand-up meeting and sprint recap which helped with communication and assigning tasks for this iteration.
- We still plan to meet each Wednesday for a standup meeting and each Sunday for a sprint recap meeting. We are moving into the refining phase of our project with few functionalities left to add so this is plenty of time to communicate about the current state of our project and what needs to be done to finish in time for the final presentation.
- We would suggest continuing effective communication with partners when problems arise and when tasks are completed. We would also recommend that we provide better documentation for each file so that a group member can pick up right where one left off to more efficiently make progress.

7. Goals for the Next Iteration

Product Backlog

- Customizable Profile

- Home page for browsing movies
- Include all movies that have the searched keyword in their title
- Show movie rating
- Page to display all of users previously rated movies and their rating
- Remove movies that have previously been rated by a user from the users recommendations

Sprint backlog

- Create a sign up page for users to make accounts
- Implement forgot password/password process
- Create user accounts settings page
- Email verification
- Implement length and character requirements for password security
- Update frontend to log status responses for POST requests
- Add home page
- Add backend function to get all previous movie interactions (like, rate, favorite)
- Implement contains functionality in search bar (searching “cars” would return cars 1, 2, and 3.)
- Style all pages

This will be our main focus for chapter and if we have enough capacity we would like to finish everything in our product backlog. There is also a lot of room for continued development even after our final presentation.

Potential Challenges

- Outside of usual challenges, such as merge conflicts, a potential challenge we may face is the meeting conflicts which may arise due to end-of-year senior commitments. We can overcome this by having good communication and being flexible in our scheduling.
- Additionally, altering or adding UI changes which were not formerly present in the wireframes due to new ideas could present a challenge. We can create new wireframes to overcome this.