CS 3205
IMDb Haters
Celestine Nguyen haw3wk
Hanna Chong vvb6ab
Katie Mead fch9mr
Ryan Nguyen rtn8fwq

Evaluation Plan

Heuristic Evaluation: We will be using the Nielson Heuristic as criteria for this Evaluation. These include:

1. **Visibility of system status**: The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.
2. **Match between system and real world**: The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom**: Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.
4. **Error prevention**: Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.
5. **Consistency and standards**: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.
6. **Recognition > recall:** Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.
7. **Flexibility and efficiency of use**: Shortcuts — hidden from novice users — may speed up the interaction for the expert user so that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. **Aesthetic and minimalist design**: Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.
9. **Help users recognize, diagnose, and recover from errors**: Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.

10. **Help and documentation**: It's best if the system doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

User Descriptions:

Female college (non-CS) professor, age 60+

- General movie fan; Representative of older user group with little technical knowledge. Rarely uses IMDb

Female college (non-CS) student, age 21

- Movie buff; representative of a younger user group with advanced technical knowledge. Likes using IMDb to search up movies.

Male college student (non-CS), age 18

- General movie fan. Likes to use IMDb for quick casual searches. Represents young men with lots of familiarity with websites.

Male CS student, age 22

- General movie fan. Likes to search IMDb with friends. Represents users with technical website perspectives.

Think Aloud Protocol: What we are going to say to the person doing the evaluation (before, during, and after we give them a task). How we will tell them to "think aloud", what we will say if they stop talking, etc.

Before Giving the Tasks:

"Today, we'll be asking you to use a prototype of a redesigned IMDB website while speaking your thoughts out loud. Please share what you are thinking, noticing, or feeling, even if it seems obvious. This is not a test of your skills, so don't worry about making mistakes. We're interested in how the app works for you. If you go silent, I might prompt you to keep talking. Is that okay?"

During the Task:

If the participant goes silent, we will use non-intrusive prompts like:

- What are you thinking now?
- What do you expect to happen when you click that?
- How does this compare to what you expected?
- What are you looking for right now?

After the Task:
"Thank you for completing these tasks. Do you have any thoughts or suggestions about your experience that you didn't get a chance to mention earlier? Was there anything particularly frustrating or enjoyable?"

Usability Tasks:
Watchlist

- Task instructions
    1. Find the watchlist menu
    2. Create a new watchlist
    3. Add a movie to the new watchlist
    4. Finish creating the new watchlist
    5. Find the new list that was just created
    6. Share the new list with someone
- Follow-up questions
    - Was creating a new watchlist intuitive?
    - Were the buttons in places that you expected them to be?
    - What was the least difficult to figure out?
    - What was the most difficult to figure out?
    - Was there a big difference in difficulty?
    - Is it clear how to remove a film from the watchlist?

Search Page + User Reviews

- Task instructions
    1. Search for *example movie title*
    2. Select the movie to view its page
    3. Find the user reviews for the movie
    4. Select more user reviews
    5. Filter the reviews to view only critic reviews
    6. Search for *example critic name* in the reviews
- Follow-up questions
    - Did the layout of the movie page match your expectations?
    - How intuitive did the filtering process feel for narrowing down review types?
    - Would any additional filtering options be helpful when searching within the reviews?
    - Are there any additional features you think would improve the prototype's functionality?
    - What did you find easiest or most intuitive?

- Were there any moments that were confusing or frustrating?
- Were movies on the search page visible?
- How would you categorize the search page 1-10, 1 being overwhelming and 10 being focused?
    - If on the lower end, what felt overwhelming?