Tiffany Allen

Josh Chang

Mounish Savier

HW 4 Findings Report:

Guide to HW4 changes:

Login.html implements user authentication. Home page has working logout and loads the movie list from the all_movies.json files.

Challenges:

Development:

There were quite a few development challenges this time around, mainly around organization. As we progressed through the project, we found that we needed some structures while others were unused. For example, when creating our new Thread element. We had organized the rest of the application and data structures to include the name, image, video, text, rating, movie and privacy setting. Our code and data structure was consistent with this. However, upon implementation of adding a new thread and keeping track of threads of characters, we realized that we neglected to include a field to keep track of all the threads per character. So, we had all the threads organized per movie, but we didn't have an option to organize by character (which is something that we wanted to implement). So, we had realized that we had a major design flaw and had to restructure our entire data base to include the character association with a thread.

We are also running into a design flaw of having two json databases (one for all movies and one for all characters). Initially we had thought that this would be a good idea to avoid nesting within the JSON file (to decrease time spent parsing the data structure), but we have found that it is proving difficult to cross reference both data bases at the same time (because there are characters listed in each movie and movies listed for each character). As a result, maintaining each database correctly and accurately is proving to be tedious and prone to errors. Also, since we started the project with two data bases in mind, we are unable to use the firebase database right now (since we are limited to one data base in fire base). This is making implementing CRUD more difficult and more time consuming than anticipated. We eventually implemented CRUD feature using jQuery and Javascript, storing the User's input Data in JSON. Our Users CRUD page shows a demo of how users can enter, edit, and delete their information. In the finalization of our project, we will make sure to implement the CRUD features fully along every aspect of our website, but for now we have a running example of how the features would essentially take effect.

Furthermore, we had general technical problems just with integrating JavaScript into our html. We are quite unfamiliar with both languages, and have had some difficulty accessing JSON objects, adding click listeners correctly, and just general integration of both smoothly. We had problems because if we ran into a bug, it was harder to troubleshoot the problem because we were unfamiliar with both languages. For example, when trying to load a JSON object, we were stuck for quite a while on why the getJSON() method was not executing (our code was hitting the JavaScript method, but the get method was just not being executed). It took way to long to find the solution to this problem (adding a documentReady

check). Little technical inexperience such as these greatly added to our total development time and restricted the amount of implementation we were able to accomplish.

Performance:

Building upon the output from HW3, we had testing performance difficulties. We did not have an older Android phone to test (we used a moderately new one) and our application ran smoothly. So we were caught off guard that our application took around 8 seconds to load the home page. We mitigated this problem by resizing our images. From there, it was more difficult feeling secure in our testing because our testing environment for performance proved to be different than the graders for the previous homework assignment.