Freddie Birr

Notes on my page + things for improvement

selectedMovie.jsp.

* Uses a javaBean to get all the information of the movie selected.
  + Uses this javaBean to display all the information about the selectedMovie using the methods from the Movie class.
* Download Button
  + Uses a servlet
    - Gets all the information from the movie, the subscriber, and the user that application provides
    - Performs the code that accesses/updates the database
  + Cases:
    - No subscriber,
      * Check if sub == null, and if so, redirect the user to the signup.jsp
    - No user
      * Check if user == null, and if so, redirect the user to selectedProfile.jsp
    - Movie already downloaded
      * Use movieCheckedOut method in QueueDB to see if the movies been downloaded, then redirect to the download confirmation page.
    - Different Subscription level
      * Capture the subscription level by accessing the level from the database, and using the corresponding silver =1, gold =2, platinum =3 scheme set up to decide if the user can download the item requested
* Favorites Button
  + Uses a servlet
    - Gets all the information from the movie and the user
    - Performs the code/ accesses the database provided
  + Cases:
    - If a movie is already a favorite by the user, redirect to the selectedMovie.jsp. Use a for loop to check the three movies in the list with the movie passed in.
  + Garbage collection
    - Have to manually delete the 4th instance of the list when a new favorite is added to avoid mismanagement of memory
* Ratings Button
  + Uses a servlet
    - Gains the value the user selects
    - Updates the database with the correct values for the ratingSum, rating Average, etc
    - Updates the webpage with the new Rating so the user can see where the application now sits.

Issues: One small thing I would have preferred was to have used an Array of fixed size instead of a list for both recents and favorites. Since both of these are of a fixed size of 3, I believe an array would have been more efficient from a memory standpoint.

Another issue I saw was the Crew class: At least from the code I worked on, The Crew class just slowed me down. When retrieving the directors name, I needed to go through the Crew class, which had a minor issue retrieving the data for the first movie in the list. This problem was small, but I spent a decent amount of time looking at other places where the code could be breaking before diving into the CrewDB class to find the issue. If the code for the director was in the Movie class, I would have been saved some time. I can’t speak for the rest of the project as it may have been very helpful to the rest of the team, but I felt the Crew classes methods could have been implemented in the movie class.