

Movie Tracker

OOAD CSCI 5352 Project 7

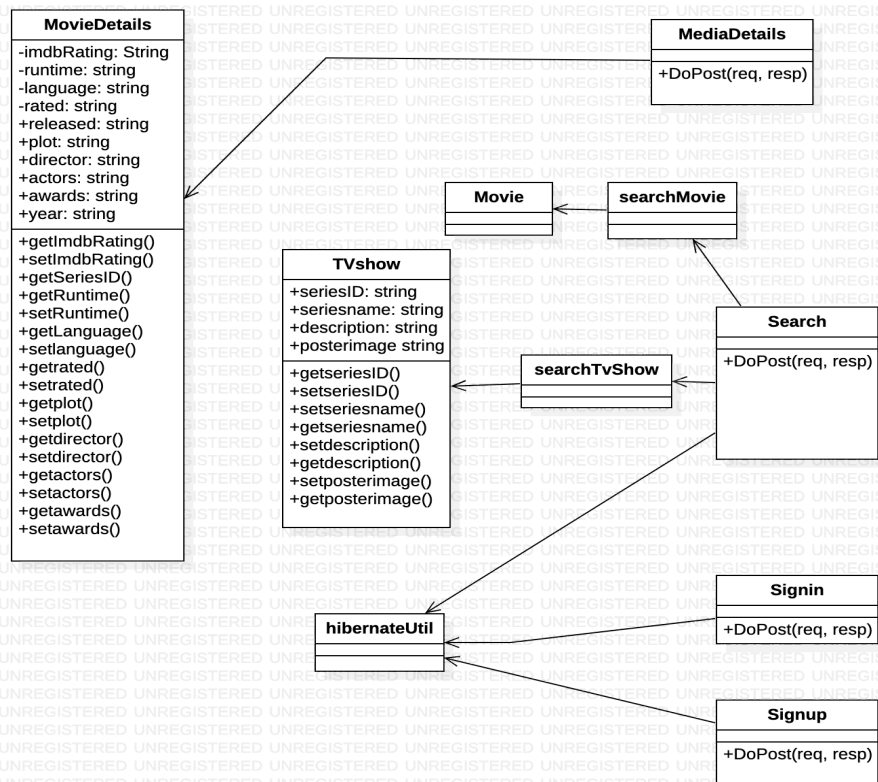
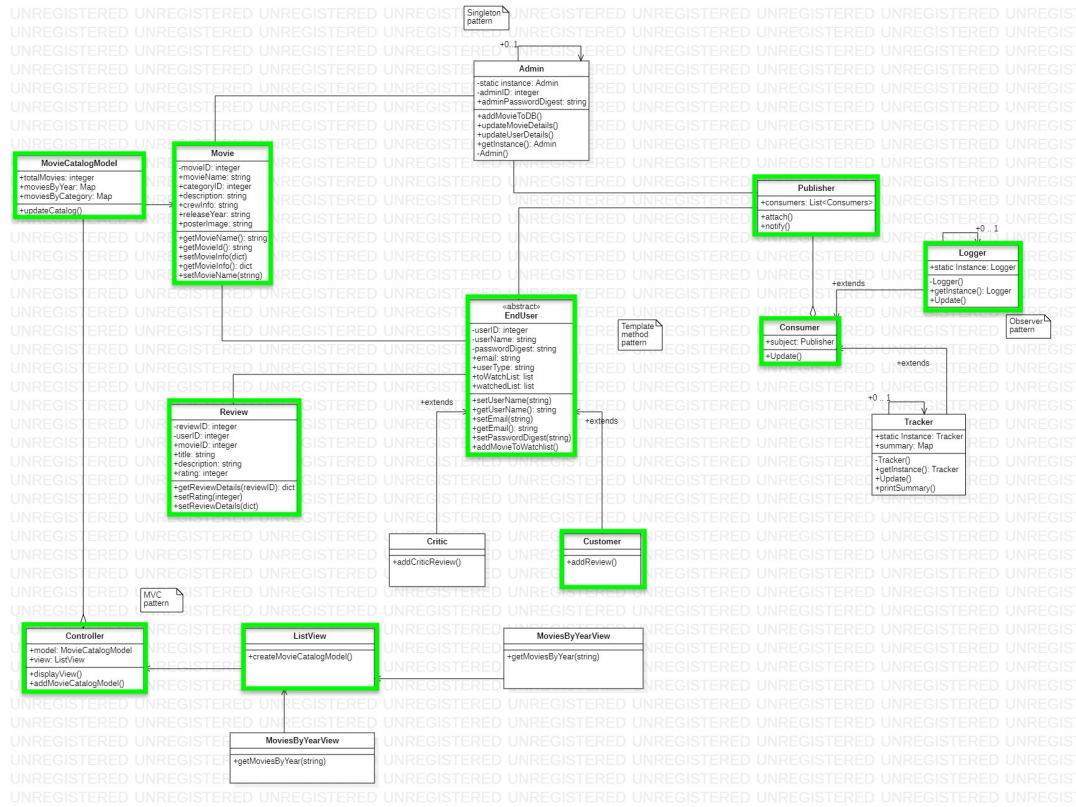
Team: Niraj Gupta, Tejas Kaushik, Joshua Sun

2. Final State of System Statement:

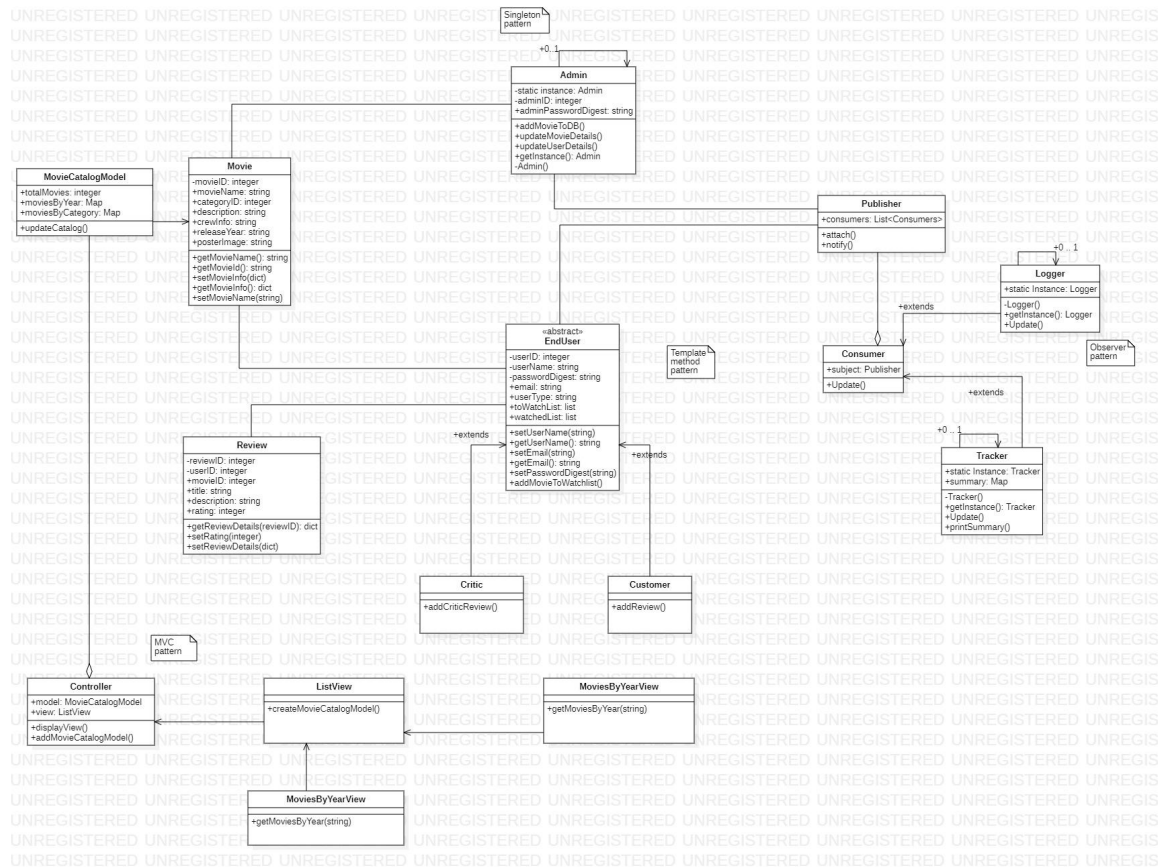
Our movie tracker has implemented the signup, login & logout features, as well as movie search for the OMDb and IMDB API. The user is able to see a list of movies. These features are kept from Project 6, and we also added support for querying and adding TV shows. For Project 7, we are able to add a searched movie/TV show to a user's watchlist, which is saved in the database. Using the observer pattern, we implemented a logger feature for the backend. We were also able to implement the feature of users adding reviews to a movie/tvshow and list reviews from other users. These were implemented as the core functionality of the movie tracker app. We were not able to add an admin account, although we still used the singleton pattern in observer package. This was left behind because of time constraints.

3. Final Class Diagram and Comparison Statement:

Final UML's:



Project 5 UML:



Key changes:

Green in the first UML diagram represents what we have in the final project. The second diagram shows the new classes that were added. One primary change was getting rid of the admin object, as we have discussed above. We did not use the template method pattern differentiating between critic and customer. While we did not necessarily have all the details the same as in the original diagram, we kept patterns such as MVC, Observer, and Singleton (in Logger) & ORM.

4. Third-Party code vs. Original code Statement

We used the [Node.js](#) framework for the frontend along with [Bootstrap](#). Most of the HTML components are from the Bootstrap site and examples, but the Node.js apis are implemented by us. We used [Hibernate](#) and [Jetty](#) for the backend, along with [Maven](#) for build automation.

We used this [HappyCoding.io](#) tutorial for setting up the servlets, with many of our own modifications. We also modified this [Hibernate tutorial](#). Other code, especially for our core functionality was original.

5. Statement on the OOAD process for your overall Semester Project

1. The use of patterns and object oriented programming was helpful, but also required us to think differently than we normally would have. The backend was therefore somewhat challenging to design, especially with the ORM database design.
2. We decided not to implement the admin feature from the original design, but instead we added support for TV shows and the details modal.
3. The UML diagram and database design was quite helpful in getting started on the project, and being able to see our design before writing any code helped the process along.