Project 7

Nick Spooner
Jackson Dubart
Finbar Forward

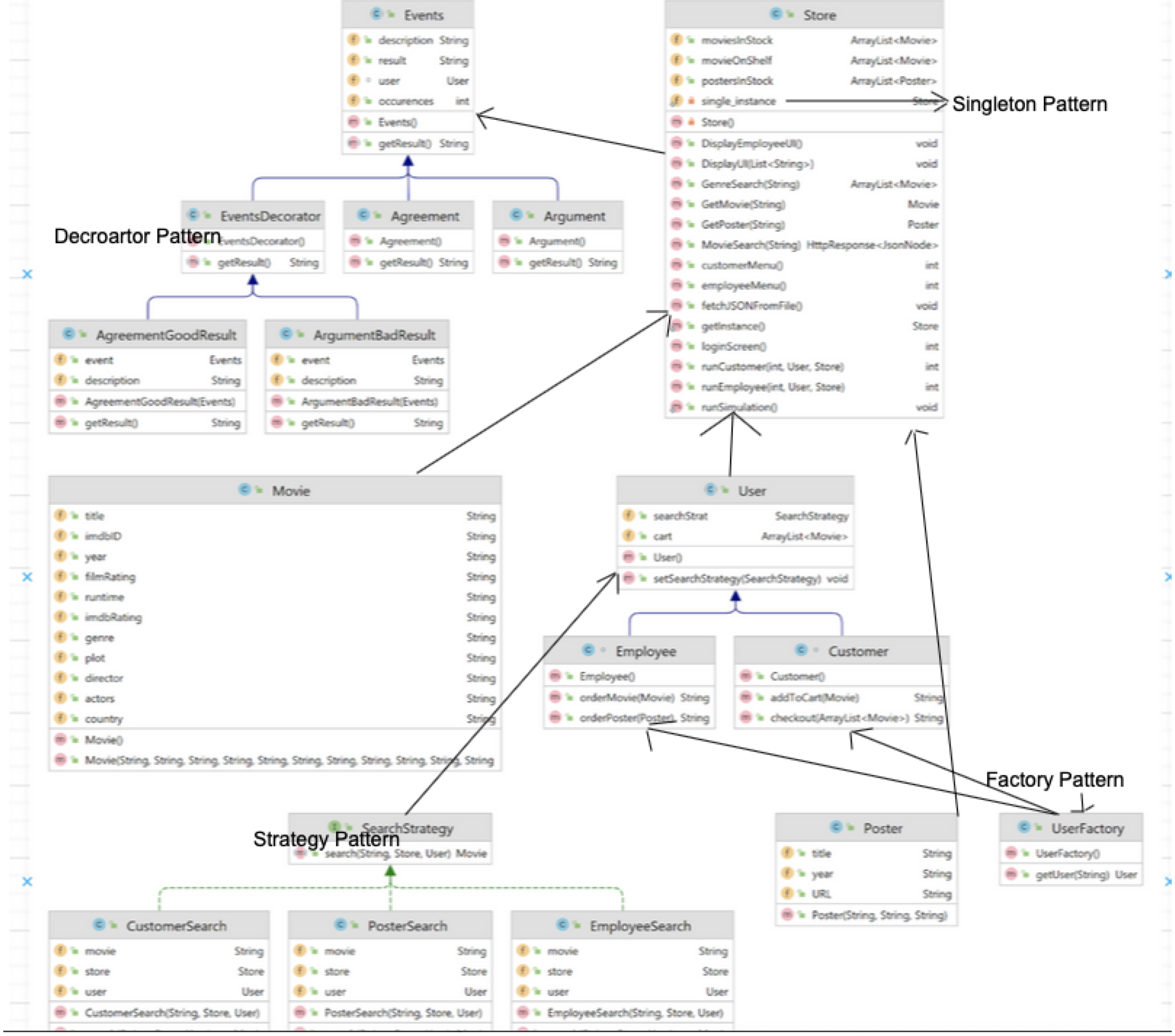
Final State of System Statement:

We finished all of the functionality and implemented all of the patterns except for giving users the ability to add reviews to movies. Other than that, we completed our goal of implementing login functionality, movie searches, movie checkout, giving recommendations to the user, and ordering and displaying movie posters, as well as storing and retrieving data from a MongoDB database. We successfully implemented the factory, singleton, strategy and decorator patterns. We were able to create the UI that we designed in project 5 using formatted text output to the terminal. We were able to successfully utilize the Movie Database API in order to search for movies and get JSON objects. We stored data about the movie and poster objects that were ordered into the database and then retrieved all of the data when the user wanted to see what the store had on the shelf (in this case what was in each collection in the database).

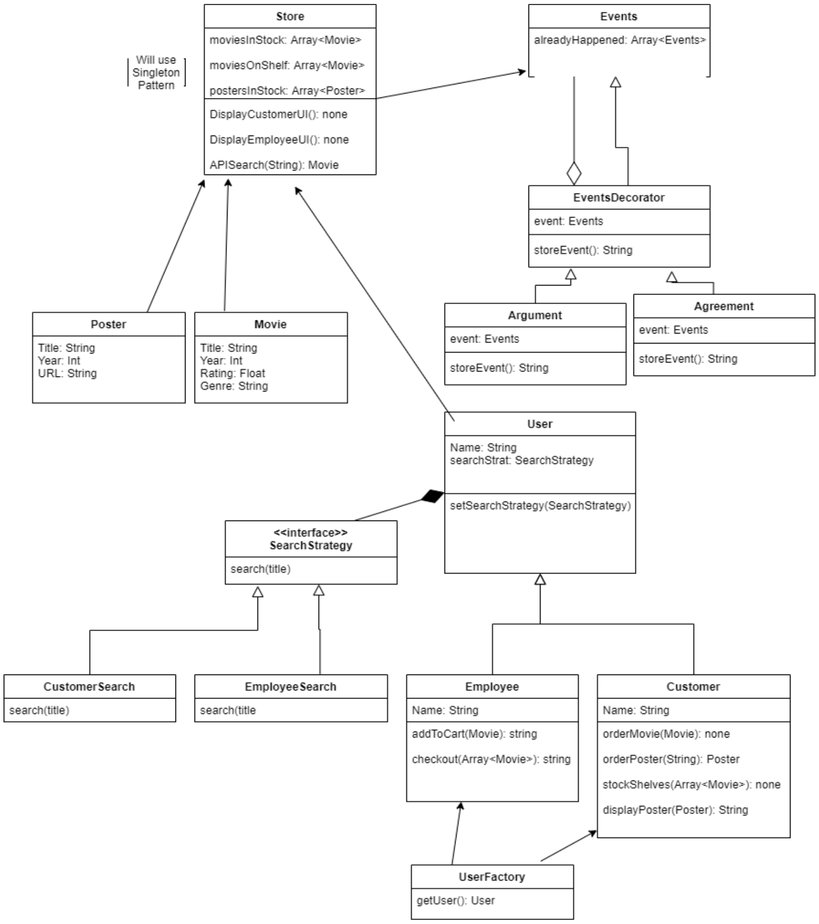
Final Class Diagram and Comparison Statement:

We used a few different design patterns when implementing BlockBuster! We were able to successfully use the decorator, factory, singleton and strategy design patterns. Some of the key changes in the system since projects 5 and 6 were in the store class and the decorator pattern. We added a lot of methods in the store class to take care of functionality related to searching for movies, posters, and getting recommendations. As part of the decorator pattern, we added the AgreementGoodResult class and the ArgumentBadResult class that handled the additional functionality added to the

Agreement and Argument base classes.



Project 5 Class Diagram :



Third-Party code vs. original code Statement

The java frameworks and libraries we used in our code were the Unirest library for REST API endpoints, the org.json library, as well as tutorialspoint.com for tutorials on java patterns, the documentation on RapidAPI for the Movie Database API, and a few other libraries for smaller functionality like java.io and java.util libraries. For connecting to the database, we used the com.mongodb library and followed this tutorial.

Statement on the OOAD process for your overall Semester Project

- One of the most important parts of the process for our group was utilizing Github to work together on the code. Github allowed us to all work on the code simultaneously and push it to the repo when we finished something, allowing the other group members to pull the new code and utilize features someone else had been working on

- Another positive aspect of our design process was planning out the design patterns we were going to use in advance and figuring out features of our project would be best suited for a certain pattern. We did this in our UML diagram initially, which made it much easier to implement the patterns into our code.

- One part of the process we could've done better is keeping track of who was working on which feature. There were times when two group members would accidentally work on the same thing. We could've used some kind of project tracking tool or even just a spreadsheet or google doc to help keep track of exactly what each person was working on at a given time.