Jackson DuBart
Nick Spooner
Finbar Forward

BockBluster

Status Summary:

While beginning our project we were able to accomplish a few goals of ours. We were able to successfully get the IMDB API working and search for movies within our store class. We were able to create a clean and appealing user interface, using a display ui function to make a clean and neat effect. Then we were able to get a working customer menu, being able to select between the different options a customer would experience in our store.   Finbar was able to successfully implement the Factory pattern with the run simulation function. Run simulation successfully creates a customer object and an employee object. We set ourselves up with lots of planning, preparation and well structured thoughts, and we were able to avoid many major issues and design flaws. So far we haven't had too much change from Project 5 other than some minor changes.

Plan For Next Iteration :

For the next iteration of Bockbluster, we hope to maximize use with the IMDB API. We hope to implement a few functions for the user in our application. With the movies we hope to make them rentable, purchasable, and the ability to find reviews on them.   We will have some review functionality where the users are able to store their reviews based on the movie, and give users recommendations on movies to watch. Next, we hope to implement a poster section, where we can display movie posters, and sell them to the customer. We will implement an open store function that will initialize our store with some movies and posters for inventory. For the strategy pattern, there will be a search method for a customer and a search method for an employee. The customer's search method will add the movie to their cart and the employee's search method will add the movie to the store's inventory. The decorator pattern will add different functionality to store events. Finally, if we have enough time in the first week of December, we hope to add a database using Spring.

Class Diagram :

Store

moviesInStock: Array<Movie>

moviesOnShelf: Array<Movie>

postersInStock: Array<Poster>

DisplayCustomerUI(): none

DisplayEmployeeUI(): none

APISearch(String): Movie

Will use Singleton Pattern

User

Name: String
searchStrat: SearchStrategy

setSearchStrategy(SearchStrategy)

Poster

Title: String
Year: Int
URL: String

Movie

Title: String
Year: String
Rating: String
Genre: String
imdbID: String
filmRating: String
runtime: String
plot: String
director: String
actors: String
country: String

Employee

Name: String

addToCart(Movie): string

checkout(Array<Movie>): string

Customer

Name: String

orderMovie(Movie): none

orderPoster(String): Poster

stockShelves(Array<Movie>): none

displayPoster(Poster): String

UserFactory

getUser(): User