

1. **Name:** Jacob Tran
2. **Project Description:** A movie rental service that allows for users to rent movies for a given period of time
3. **List the features that were implemented:**

ID #	Requirement	Actors
001	Users should be able to sign up for the service.	User
002	Users should be able to sign in/out for the service.	User
003	Users should be able to select what movie they want to rent out.	User
004	Users should be able to checkout a movie.	User
005	Users should be able to input their payment method into the checkout page.	User
006	Admins should be able to sign in.	Admin
007	Admins should be able to add a movie to the available movies to the client.	Admin
009	Admins should be able to delete a movie from the available movies to the client.	Admin
010	Admins should be able to ban users for breaking terms of use.	Admin

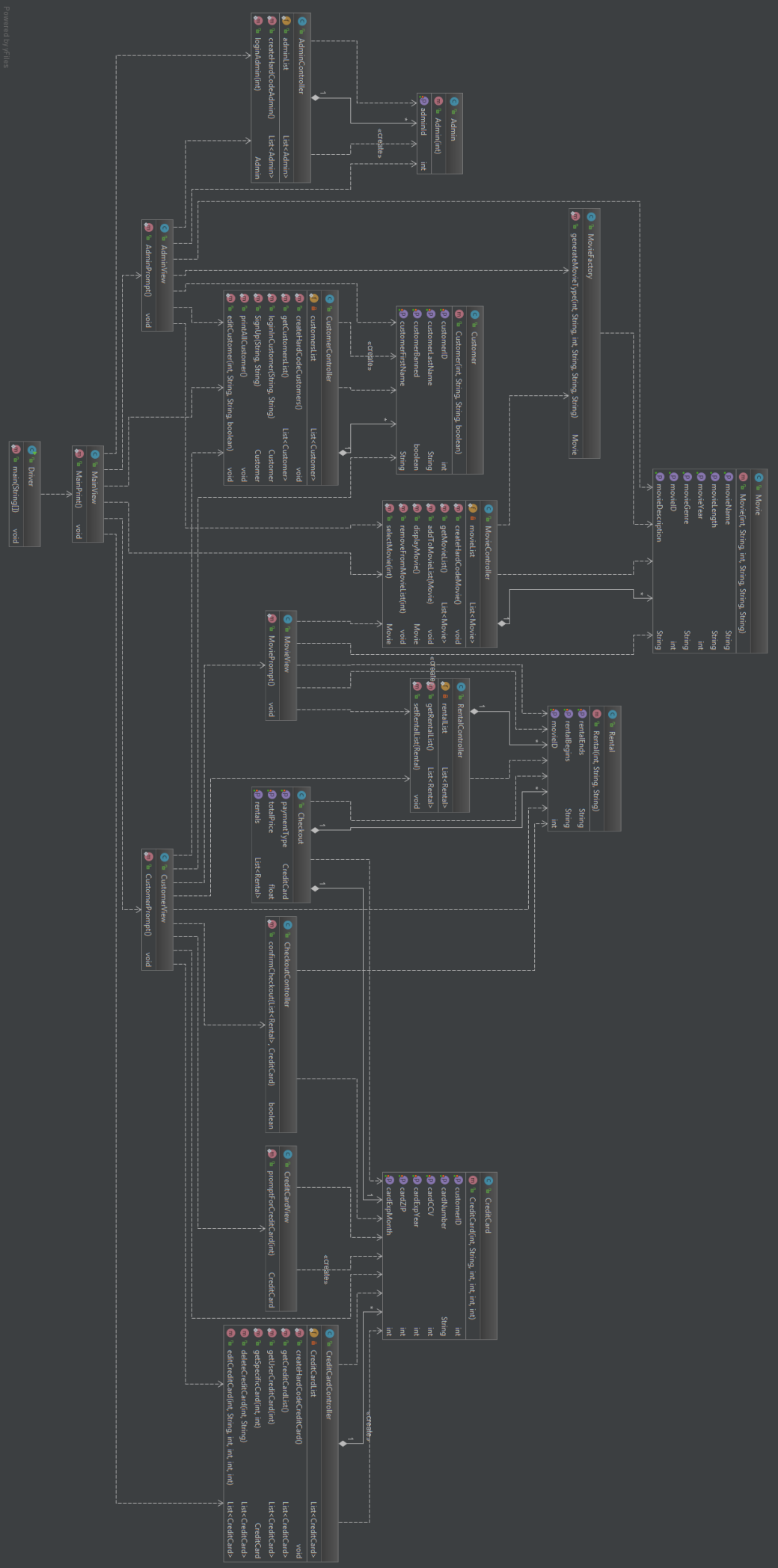
4. **List the features were not implemented:**

008	Admins should be able to view movie analytics and how it is doing.	Admin (Optional requirement)
-----	--	---------------------------------

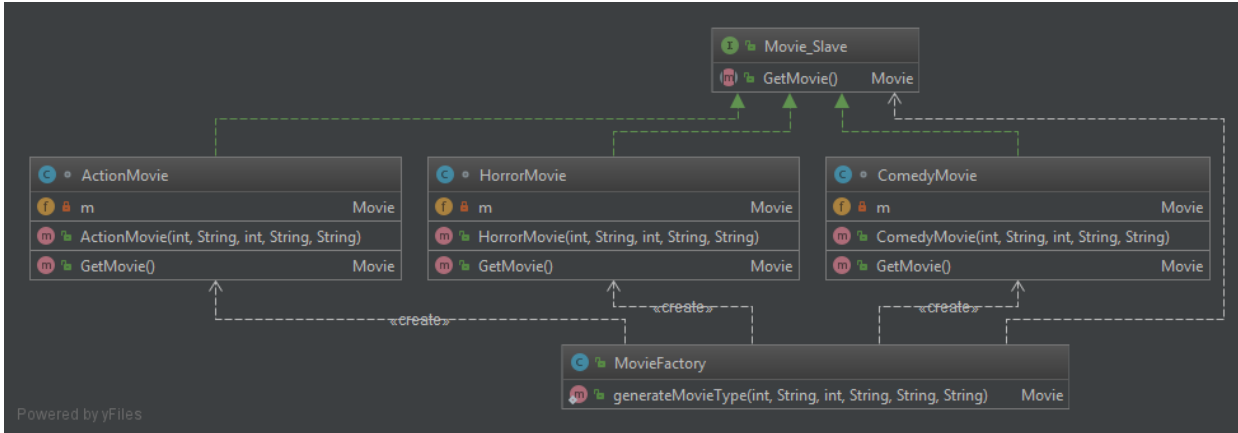
5. Show your final class diagram:

(refer to src/Factory.png in GitHub if needed)

At first, I didn't implement MVC properly so I had to create a brand-new UML diagram. Initially I wanted to use an API to fetch movies but I couldn't find one to satisfy my needs. I resorted to using array lists to store my movie data. After planning out my project with the intent to use MVC the entire system made sense, I was able to code up the models and views easily while focusing on the controller where all the magic happens.



6. I did a factory design to generate different types of movies based on genre. For the size of the project I wanted to experiment with a design pattern that I haven't used before. The **Movie Factory** constructs an **Action/Horror/Comedy Movie** and gets called with **Movie Slave** when we want to retrieve the movie instance. I choice to do a factory because if this system gets scaled up, it will benefit in runtime because the factory objects will be generated more effectively. This also allows for easy creation of a movie and allows for differences in subclasses. For example, a scary movie could have a scary meter while an action movie wouldn't need such characteristic.



7. **What have you learned about the process of analysis and design now that you have stepped through the process to create, design and implement a system?**

I learned that design patterns have their place in industry and helps automate the process of creating large scale products. Designing a system is by far the hardest part of programming. It is a great idea to plan your project out to help mitigate issues during the design process. I also learned that design patters can make coding easier and helps solve very specific problems when designing a system.