# PROJECT PRESENTATION

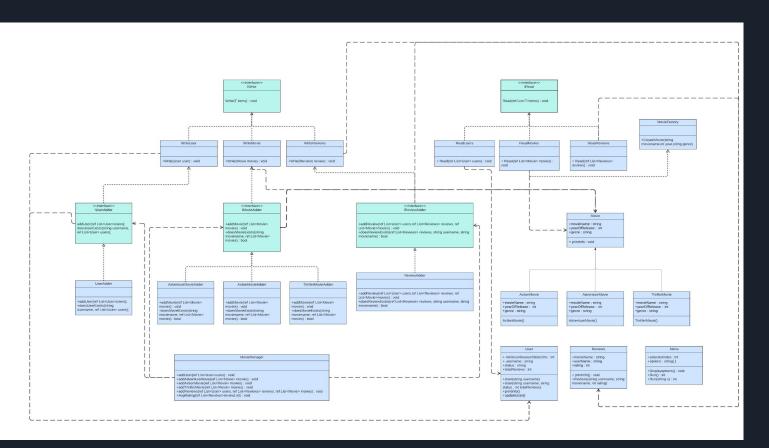## Object Oriented Concepts -II (SWE 4301-SWE 4302)

## Movie Review System

KHALID HASAN ADOR
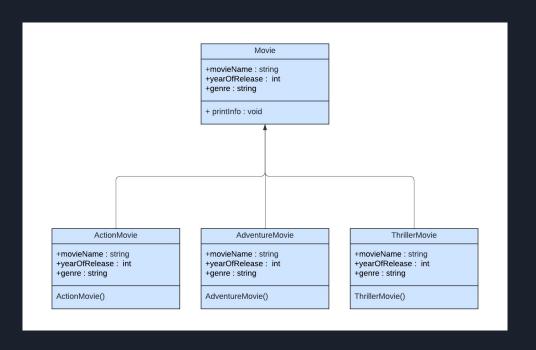210042102

1

# KEY FEATURES

- **Add New User**
- **Add New Movie**
- **Add Review**
- **Print Movies**
- **Print Users**
- **Print Reviews**
- **Average Rating**

# UML

# Single Responsibility Principle

In the project, each class is designed to fulfill a specific role. For example, the MovieAdder class only has one responsibility and that is to add Movies. It does so by the addMovie() method. There are no other methods in that class. So the class has only one reason to change.

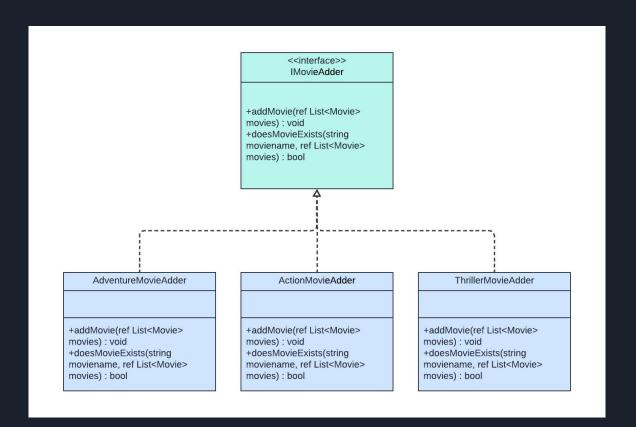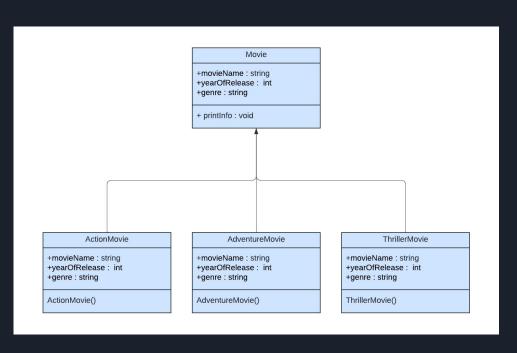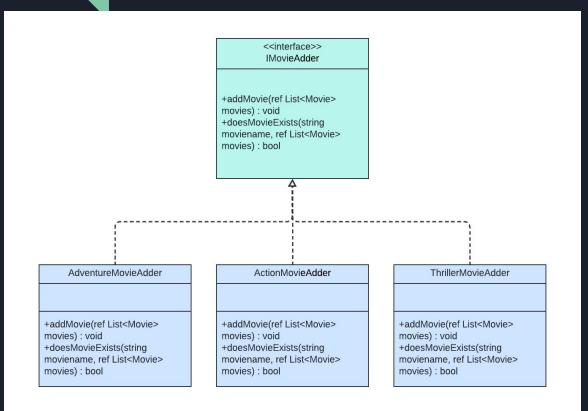# Open Close Principle

# Open Close Principle

# Liskov Substitution Principle

# Interface Segregation Principle



Classes like AdventureMovieAdder, ActionMovieAdder and ThrillerMovieAdder implements IMovieAdder. But they do use the methods of that interface. So the design follows ISP.

8

**IUserAdder** `<<interface>>`
addUser(ref List<User>Users);
doesUserExists(string username, ref List<User> users);

**IMovieAdder** `<<interface>>`
+addMovie(ref List<Movie> movies) : void
+doesMovieExists(string moviename, ref List<Movie> movies) : bool

**IReviewAdder** `<<interface>>`
+addReview(ref List<User> users,ref List<Reviews> reviews, ref List<Movie>movies) : void
+doesReviewExists(ref List<Reviews> reviews, string username, string moviename) : bool

**UserAdder**
+addUser(ref List<User>Users);
+doesUserExists(string username, ref List<User> users);

**AdventureMovieAdder**
+addMovie(ref List<Movie> movies) : void
+doesMovieExists(string moviename, ref List<Movie> movies) : bool

**ActionMovieAdder**
+addMovie(ref List<Movie> movies) : void
+doesMovieExists(string moviename, ref List<Movie> movies) : bool

**ThrillerMovieAdder**
+addMovie(ref List<Movie> movies) : void
+doesMovieExists(string moviename, ref List<Movie> movies) : bool

**ReviewAdder**
+addReview(ref List<User> users,ref List<Reviews> reviews, ref List<Movie>movies) : void
+doesReviewExists(ref List<Reviews> reviews, string username, string moviename) : bool

**MovieManager**
+addUser(ref List<User>users) : void
+addAdventureMovie(ref List<Movie> movies) : void
+addActionMovie(ref List<Movie> movies) : void
+addThrillerMovie(ref List<Movie> movies) : void
+addReviews(ref List<User> users, ref List<Reviews> reviews, ref List<Movie> movies) : void
+AvgRating(ref List<Reviews>reviewList) : void

9

# Thank you!