

Watch It

Title	<u>Watch It</u>
Description	<p>Entertainment application offering movies to the users such as the popular Watch It.</p> <p>The project will have 6 main entities (you can add additional entities but those are the main 6):</p> <p>The application should keep track of these entities.</p> <ol style="list-style-type: none"> 1. User: the user has (ID, Username, Password, First name, Last name, Email, Subscription, Added to list movies (List of movies that was marked to be watched by the user later), List of User watch record (List of movies that the user already watched)). 2. Subscription: The Subscription has (UserID, Plan which is one of three Plans (Basic, Standard, Premium), Price of Plan, Start date of subscription). 3. Movie: the movie has (Movie ID, Movie Title, Release Date, Running time (Duration), List of Cast, Genres, Director, languages, country , budget, Revenue , Poster (the image poster of the movie if exists)). 4. Cast: the cast has (First Name, Last Name, date of birth, gender, List of Movies, nationality, social media link (ex. instagram, twitter, ... etc)) 5. Director: The director has (First Name, Last Name, date of birth, gender, List of Movies, nationality, social media link (ex. instagram, twitter, ... etc)) 6. User Watch record: this entity has (User ID, Movie, Date of watching the movie, Rating (from 1 to 5 , The user may not enter the rating to the movie)).
Constraints	<p>For the subscription, it only lasts for 1 month (30 days) if the user is subscribed to the basic plan, he only can watch up to 5 movies per month, while if subscribed to the standard plan, he can watch up to 10 movies per month, however in the premium plan, he can watch up to 30 movies per month. If 30 days passed from the beginning of the subscription start date, he needs to subscribe again for one of the three plans or he can't watch any movie.</p>
Deliverables	<ol style="list-style-type: none"> 1. Class diagram for the project (Printed) 2. Java project includes all classes and functions in the description. 3. Documentation that contains <ol style="list-style-type: none"> a. your own system description b. input and output scenarios

Bonus	<ol style="list-style-type: none"> 1. Powerful GUI 2. Any non-trivial function
Required Functionalities	<p>Functionalities:</p> <ol style="list-style-type: none"> 1- Add /update /delete in all of the entities. 2- The user can display all the movies he already watched. 3- The movie rating should be updated by the new ratings given to a movie based on the users watch record. 4- Displaying the top-rated movies to the user. 5- The admin can see the most subscribed plan among (Basic, Standard, Premium). 6- The admin can see which month had the most revenue to the application by adding all the plan prices subscribed in that month. 7- The user can see the casts and directors' information and their movies. 8- The user can search for a movie/director/actor by his name. 9- The user can also search and display all movies by the genre of movie. 10- Display the top watched movies for the user. 11- Display the recent movies for the users. 12- Display movies that matches specific filters (The filters may be based on rating, movie duration and language) 13- Recommendation feature (That recommends movies to user based on the genres and cast that the user usually watch movies for)
Notes	<ol style="list-style-type: none"> 1. You should implement all concepts of OOP. 2. Each member MUST work on at least one of the required classes besides file processing or GUI. (Individual marks) 3. The evaluation will be mainly based on the student's ability to use and apply OOP concepts and the explanation of the code. 4. You must deliver the Class Diagram for the project. 5. You must apply exception handling. 6. Using Files is mandatory (Not Database) 7. Any project must have at least 8 classes 8. Regarding files: <ul style="list-style-type: none"> - You must have only two functions for file reading and writing.

	<ul style="list-style-type: none">- You should read data once at the beginning of your run then do your operations and access the code then save in files at the end of your program.
--	---