

The God We Trust



# 4th Assignment Report

<Mitra Dehghan> <98222038>

## Introduction:

هدف کلی این پروژه ساخت پلتفرمی مشابه نتفلیکس (پلتفرم تماشای فیلم و سریال) می باشد که این برنامه به کاربر این قابلیت را میدهد که فیلم مدنظر خود را بر اساس عنوان ، ژانر و سال پخش آن جستجو کند و فیلم مدنظر را به لیست علاقه مندی های خود اضافه کند و یا در میان لیست خود جستجو انجام دهد و یا حتی آن را مشاهده کند .

در حالت کلی به کاربرانی که از قبل اکانت داشته باشند اجازه ورود با نام کاربری و رمزعبور داده میشود و برای کاربران جدید هم این امکان وجود دارد تا اکانتی با نام کاربری و رمزعبوری که خودشان بخواهند را بسازند .(در صورتی که اطلاعات غیر تکراری باشند)

## Design and Implementation:

### کلاس TVShow :

در این کلاس متغیرهای `rating` , `duration` , `releaseYear` , `genre` , `title` , را به ترتیب برای عنوان فیلم ، ژانر آن ، سالی که پخش شد ، مدت زمان طول ساخت (بر اساس سال) و رتبه نهایی آن تعریف کردیم . `constructor` و `getter setter` های آنها را نیز ساختیم .

### کلاس User :

ابتدا متغیرهای `username` و `password` (نام کاربری و رمز عبور کتابدار) و دو لیست هم برای ذخیره سازی فیلم های مورد علاقه هر کاربر و تاریخچه فیلم های دیده شده توسط کاربر با نام های `favoriteShows` و `watchHistory` تعریف نمودیم .

بعد از تعریف متغیرهای موردنظر `constructor` و `getter setter` های آنها را نیز ساختیم . در مرحله بعد متود `searchByTitle` را تعریف کردیم که ورودی آن از جنس `String` میباشد که با کمک یک حلقه `while` مقدار ورودی را تک به تک با محتوای لیستی که شامل سریال های مورد علاقه کاربر بود قیاس میکنیم و زمانی که به تطابقی برسیم سریال مورد نظر با باقی اطلاعاتش (ژانر و سال ساخت و ..) چاپ و به نمایش گذاشته میشوند . مشابه با همین متود ، متودهای `searchByGenre` و `searchByReleaseYear` را هم در این کلاس تعریف کردیم .

گام بعدی به سراغ تعریف متود `addToFavorites` رفتیم که کاری که این متود انجام میدهد در اصل افزودن یک فیلم به لیست علاقه مندی های کاربر است .

در این کلاس متود دیگری با نام `viewFavorites` هم قرار دادیم با این هدف که کاربر هرزمان خواست بتواند لیست علاقه مندی های خود را مشاهده کند .

## کلاس `NetflixServer`:

در این کلاس سه لیست کلی با نام های `movieList` , `tvShowList` , `userList` برای ذخیره سازی کلیه فیلم های موجود و برنامه های موجود و کاربران موجود تعریف کردیم . و پس از آن مشابه کلاس های قبلی `constructor` و `getter setter` های آنها را نیز ساختیم .

پس از آن سه متود با هدف اضافه کردن برنامه ، فیلم و کاربر به لیست هر یک با عنوان های `addTVShow` و `addMovie` و `addUser` را قرار دادیم که جنس متغیر ورودی هر کدام هم باتوجه به کارکرد آن تعریف کردیم مثلاً برای اضافه کردن کاربری به لیست کاربران موجود در لیست پلتفرم ، ورودی از جنس `User` (کاربر) دریافت نمودیم .

در اینجا دو متود برای راحتی کار اضافه کردیم (در `ReadMe` ذکر نشده بودند) با نام های `usernameUniqueness` و `passwordUniqueness` که هدف از ساخت

این متود ها چک کردن نام کاربری و چک کردن رمزعبوری است که یک کاربر تازه وارد سایت میکند تا اکانت خود را بسازد و هدف از این چک کردن این است که اگر مشابه آن نام کاربری یا رمزعبور در دیتابیس ما وجود داشت از ساختن این اکانت جلوگیری شود و به کاربری متنی نمایش داده شود که باید نام کاربری یا رمز خود را عوض کند چرا که از قبل وجود داشته . بعد به سراغ ساخت دو متود دیگر برای ورود کاربر شدیم که یکی از آنها مختص به کاربریست که از ثبل اکانتی ندارد :

`createAccount` که در اینجا از کاربر یک نام کاربری و یک رمزعبور می گیریم و با استفاده از متودهایی که پیشتر توضیح دادیم یکتایی آنها را میسنجیم و اگر قابل قبول بودند اکانت مورد نظر ساخته میشود و اطلاعات این کاربر به لیست کاربران پلتفرم اضافه میشود . و متود دوم هم `login` میباشد که مخصوص کاربرانیست که از قبل اکانتشان ساخته شده و زمانی که نام کاربری و رمز خود را وارد میکنند این اطلاعات در میان لیست کاربران با کمک یک حلقه `while` جستجو میشود و در صورت تطابق ، ورود کاربر موفقیت آمیز است . در نهایت مشابه با متود های جستجوگری که در `User` تعریف کردیم ، اینجا هم سه متود `searchByTitle` , `searchByGenre` و `searchByReleaseYear` را مجددا تعریف کردیم .

## کلاس Main :

در گام آخر مربوط به نوشتن کلاس `Main` بود که در اینجا برای بهبود عملکرد برنامه و بررسی کارکرد صحیح آن ابتدا چند کاربر فرضی و چند برنامه فرضی تعریف و به لیست کاربران و برنامه هایمان در پلتفرم اضافه کردیم .

اولین پیامی که به کاربر نمایش داده میشود را اینجا با عنوان `WELCOME` قرار دادیم و بعد از خوشامدگویی از کاربر خواستیم تا هویت خودش را برای ما اینگونه تعیین کند که اگر کاربری جدید است عدد 1 و اگر کاربریست که از قبل اکانت دارد عدد 0 را وارد کند و بعد از آن باتوجه به عددی که کاربر وارد کرده یکی از متود های

login و یا creatAccount که در کلاس NetflixService تعریف شدند فراخوانی میشوند و بعد از مرحله ورود کاربر متود runMenu صدا زده میشود که محتوای این متود در اصل اجرای قابلیت هایبست که کاربر میتواند داشته باشد. در ابتدا از کاربر خواسته میشود که باتوجه به کاری که میخواهد در پلتفرم انجام دهد شماره مختص به آن را وارد کند

1. جستجو میان فیلم ها

2. اضافه کردن فیلمی به لیست علاقه مندی های خود

3. لیست علاقه مندی های خود را مشاهده کند

4. فیلم های پیشنهادی به کاربر نمایان شود

که در اصل یک switch case نوشتیم که هر case آن مربوط به یکی از موارد بالاست و کاربر هر شماره ای را بزند به آن case پاس داده میشود و کارمورد نظر باتوجه به متودهای مرتبط انجام میشود البته از آنجایی که جستجو در پلتفرم میتواند به دوصورت انجام شود ، اگر کاربری گزینه 1 را وارد کند دوباره پرسشی در صفحه نمایش نمایان میشود که از کاربر میخواهد انتخاب کند که قرار است جستجو میان تمام برنامه های پلتفرم باشد یا فقط میان آنهایی که در لیست علاقه مندی خودش وجود دارد و بعد از آن از او خواسته میشود که تعیین کند بر چه اساسی میخواهد جستجو کند نام برنامه / ژانر برنامه / سال پخش آن .

## : UML



