

README

Mihai Alexia-Andra, 325 CC

Aceasta aplicatie ofera utilizatorilor posibilitatea de a gestiona productii cinematografice si actori. Proiectul cuprinde diverse functionalitati, inclusiv lucrul cu utilizatorii, vizualizarea informatiilor despre productii si a actori, un sistem de notificari, adaugarea unor recenzii si multe altele.

****Clase Cheie****

1. ****IMDB****

Clasa IMDB serveste ca nucleu al aplicatiei IMDB, unde sunt administrate toate detaliile sistemului. Ea extrage informatii din fisierele JSON si furnizeaza functionalitati esentiale pentru utilizatori.

****Attribute****

- Lista de utilizatori: Regular, Contributor si Admin.
- Lista de actori: Contine informatii despre toti actorii implicati in productiile cinematografice
- Lista de cereri (Request): Mentine o lista cu toate cererile efectuate in sistem, cum ar fi solicitari de adaugare sau modificare a datelor
- Lista de filme si seriale (Movies si Series): Include detalii despre productiile cinematografice disponibile in sistem.

****Metode Principale****

Metoda **run** se ocupa de incarcarea datelor parsate din fisierele JSON folosind Jackson, autentificarea utilizatorului si pornirea fluxului principal al aplicatiei in functie de rolul utilizatorului.

Metoda **authenticateUser** solicita introducerea credentialelor utilizatorului (email și parolă) și verifică autentificarea acestuia. După autentificare, utilizatorul este redirecționat către meniul specific rolului său.

Metoda **performActions** oferă meniuri specifice pentru utilizatorii cu diferite roluri (REGULAR, CONTRIBUTOR, ADMIN) și le permite să aleagă și să efectueze diverse acțiuni.

Exemplu pentru Utilizator REGULAR:

- Vizualizare detalii producție

- Vizualizare detalii actor
- Vizualizare notificări
- Căutare actor/film/serial
- Adăugare/Ștergere actor/film/serial la/de la favorite
- Creare/Retragere cerere
- Adăugare/Ștergere evaluare pentru o producție/actor
- Logout

Exemplu pentru Utilizator CONTRIBUTOR:

- Vizualizare detalii producție
- Vizualizare detalii actor
- Vizualizare notificări
- Căutare actor/film/serial
- Adăugare/Ștergere actor/film/serial la/de la favorite
- Creare/Retragere cerere
- Adăugare/Ștergere actor/film/serial din sistem
- Actualizare Detalii Film
- Actualizare Detalii Actor
- Rezolvare cerere
- Logout

Exemplu pentru Utilizator ADMIN:

- Vizualizare detalii producție
- Vizualizare detalii actor
- Vizualizare notificări
- Căutare actor/film/serial
- Adăugare/Ștergere actor/film/serial la/de la favorite
- Adăugare/Ștergere utilizator
- Adăugare/Ștergere actor/film/serial din sistem
- Actualizare Detalii Film
- Actualizare Detalii Actor
- Rezolvare cerere
- Logout

Alegeți opțiunea corespunzătoare și urmați instrucțiunile pentru a executa acțiunile dorite. Pentru opțiunile invalide, sistemul va oferi un mesaj de avertizare.

2. ****User****

Clasa User reprezintă diferite tipuri de utilizatori, cum ar fi Regular, Contributor și Admin. Utilizatorii pot efectua acțiuni diverse, precum adăugarea de recenzii, crearea de cereri și gestionarea contribuțiilor lor, în funcție de tipul contului lor.

Metode:

`addFavoriteActor(Actor actor)`: Adaugă un actor la lista de actori favoriți ai utilizatorului.

`deleteFavoriteActor(Actor actor)`: Elimină un actor din lista de actori favoriți ai utilizatorului.

`addFavoriteProduction(Production production)`: Adaugă o producție la lista de producții favorite ale utilizatorului.

`deleteFavoriteProduction(Production production)`: Elimină o producție din lista de producții favorite ale utilizatorului.

`getProductionsContribution()`: Returnează lista de titluri de producții la care utilizatorul a contribuit.

`getActorsContribution()`: Returnează lista de nume de actori la care utilizatorul a contribuit.

`getFavoriteProductions()`: Returnează setul sortat de producții favorite.

`getFavoriteActors()`: Returnează setul sortat de actori favoriți.

`getUserRequests()`: Returnează lista de solicitări ale utilizatorului.

`getInformation()`: Returnează informațiile utilizatorului.

`getAccountType()`: Returnează tipul de cont al utilizatorului.

`getUsername()`: Returnează numele de utilizator al utilizatorului.

`getExperience()`: Returnează nivelul de experiență al utilizatorului.

`getNotifications()`: Returnează lista de notificări primite de utilizator.

`setExperienceStrategy(ExperienceStrategy experienceStrategy)`: Setează strategia de experiență a utilizatorului.

`updateExperience()`: Actualizează nivelul de experiență al utilizatorului pe baza strategiei de experiență asigurate.

`getObservers()`: Returnează lista de observatori care monitorizează utilizatorul.

`logoutmethod(IMDB imdb)`: Deconectează utilizatorul din aplicația IMDb.

`addReview(User user, Production production, int rating, String comment, IMDB imdb)`: Adaugă o recenzie pentru o producție, anunțând utilizatorii relevanți și actualizând notificările acestora.

`addReviewActor(User user, Actor actor, int rating, String comment, IMDB imdb)`: Adaugă o recenzie pentru un actor, anunțând utilizatorii relevanți și actualizând notificările acestora.

`update(String notification)`: Înlocuiește metoda `update` a interfeței `Observer` pentru a

gestiona notificările pentru utilizator.

Clasa: Information

Metode: Diverse metode de obținere a informațiilor despre utilizator.

Clasa: Credentials

Metode: Metode de obținere a adresei de email și parolei.

3. ****Production****

Clasa Production reprezinta filmele sau serialele din sistem, incluzand detalii precum titlu, regizori, actori, genuri, recenzii, descrierea intrigii si anul de lansare.

Metode:

displayInfo(): Metoda abstractă care trebuie implementată în clasele derivate pentru a afișa informații specifice producției.

toString(): Override pentru metoda toString care afișează informații despre producție.

compareTo(Production other): Override pentru metoda compareTo din interfața Comparable, comparând producțiile după titlu.

Diverse metode de acces și modificare pentru câmpurile producției.

4. ****Actor****

Clasa Actor reprezinta actorii din sistem, continand detalii precum nume, performante si biografie.

Metode:

compareTo(Actor other): Override pentru metoda compareTo din interfața Comparable, comparând actorii după nume.

addRating(Rating rating): Adaugă o evaluare la lista de evaluări și recalculează ratingul mediu.

deleteRating(Rating rating): Elimină o evaluare din lista de evaluări și recalculează ratingul mediu.

displayInfo(): Afișează informații despre actor, inclusiv biografie, roluri și evaluări.

calculateAverageRating(): Calculează ratingul mediu al actorului.

toString(): Override pentru metoda toString, returnând o reprezentare sub formă de șir a obiectului actor.

*****Operarea cu Utilizatori*****

Utilizatorii pot fi adaugati in sistem prin metoda `addUser`. Sistemul genereaza un nume de utilizator unic, o parola puternica si gestioneaza introducerea detaliilor de utilizator.

`deleteUser` permite utilizatorilor Admin sa stearga alti utilizatori din sistem, eliminandu-i din lista de utilizatori.

*****Calculul Experientei*****

Experienta utilizatorului este calculata dinamic in functie de modelul de strategie. Diverse actiuni declanseaza strategii diferite, influentand experienta generala a utilizatorului.

*****Notificari*****

Utilizatorii primesc notificari pentru diverse evenimente, precum cereri rezolvate sau respinse si noi recenzii pentru productiile la care au contribuit.

Design Patterns în Proiect

În cadrul proiectului, am adoptat mai multe design patterns pentru a asigura o structură coerentă și ușor extensibilă. Fiecare pattern a fost aplicat strategic pentru a rezolva anumite probleme și pentru a facilita dezvoltarea și mentenanța sistemului.

1. Singleton Pattern:

- **Utilizare:** Am implementat Singleton Pattern pentru clasa IMDB, asigurând că există o singură instanță a acesteia în întregul sistem.
- **Motivație:** Această abordare previne crearea mai multor instanțe IMDB, asigurând eficiență în gestionarea resurselor și centralizarea funcționalităților globale.

2. ****Factory Method Pattern:****

- ****Utilizare:**** Am integrat Factory Method Pattern în clasa UserFactory, furnizând o modalitate flexibilă și extensibilă de creare a obiectelor User în funcție de tipul de cont specificat.
- ****Motivație:**** Permite adăugarea simplă de noi tipuri de utilizatori fără a modifica în mod semnificativ codul existent, încurajând extensibilitatea și menținerea codului.

3. ****Observer Pattern:****

- ****Utilizare:**** Am aplicat Observer Pattern în gestionarea notificărilor către utilizatori.
- ****Motivație:**** Acest pattern permite comunicarea eficientă între diferite părți ale sistemului, astfel încât schimbările într-o componentă să poată afecta dinamic alte componente fără a le cunoaște structura internă.

4. ****Strategy Pattern:****

- ****Utilizare:**** Am implementat Strategy Pattern pentru calculul experienței utilizatorului în funcție de acțiunile efectuate.
- ****Motivație:**** Oferă flexibilitate prin schimbarea dinamică a strategiilor de calcul, permițând extensibilitate și adaptabilitate la schimbările în cerințele sistemului.

5. ****Builder Pattern:****

- ****Utilizare:**** Am folosit Builder Pattern pentru construirea obiectelor Information din clasa User.
- ****Motivație:**** Acest pattern oferă o modalitate clară și extensibilă de a construi obiecte complexe pas cu pas, gestionând parametrii opționali într-un mod coerent și ușor de înțeles.

Adoptarea acestor design patterns în proiectul meu a condus la o structură mai modulară, ușor de înțeles și extensibilă. Fiecare pattern a fost aplicat cu un scop bine definit, contribuind la crearea unui cod robust și eficient. Aceste abordări ajută la menținerea și dezvoltarea ulterioară a sistemului într-un mod sustenabil și coerent.

****Concluzie****

Sistemul IMDb ofera o platforma robusta pentru interactiunea utilizatorilor cu datele despre filme si seriale. Implementarea valorifica principiile orientarii obiectelor, precum mostenirea si modelele de strategie, pentru a oferi extensibilitate si flexibilitate.

Durata implementare : o saptamana;

Grad de dificultate: mediu, dar necesita mult timp deoarece este complexa.

Observatii:

->am adaugat functionalitati bonus

->am inceput implementarea interfetei grafice, dar am apucat doar sa fac putin din ea