



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
CATEDRA CALCULATOARE**

MOVIE HOUSE

-documentație-
-proiect nr.17-

Student:

Pop Ruxandra Maria

Anul 3, grupa 30236

1. Enunțul problemei

Scopul proiectului este dezvoltarea unei aplicații desktop care poate fi utilizată de o casă de producție filme. Aceasta aplicație va dispune de 2 tipuri de utilizatori: angajat și administrator, fiecare putând efectua diferite operații în funcție de rolul lui.

Obiectivul principal al acestei teme este familiarizarea cu șablonul arhitectural Model-View-Presenter.

2. Instrumente utilizate

Înainte să ne apucăm de proiectarea și implementarea propriu zisă a aplicației, este necesar să cunoaștem ce cerințe trebuie acesta să îndeplinească, pentru a descoperii ce trebuie să facă sistemul, cum trebuie să funcționeze, pentru a ști ce obiective dorim să atingem la sfârșitul implementării.

Astfel ca și primă etapă în realizarea temei, am construit diagrama de use case, care cuprinde actorii aplicației și operațiile pe care le pot face. Diagrama de use case a fost realizată cu ajutorul instrumentului software **starUML**, tot prin intermediul acestui instrument am proiectat și diagrama de clase, care respectă arhitectura MVP.



În continuare o să atașez câteva imagini care să vă familiarizeze cu conceptele acestui tool.

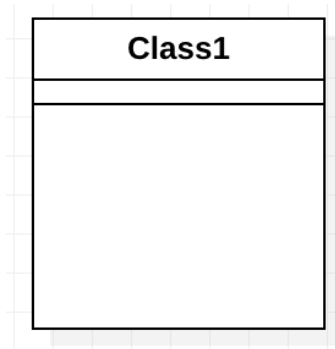


Fig2.1

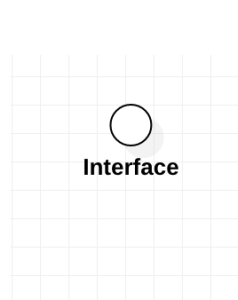


Fig2.2

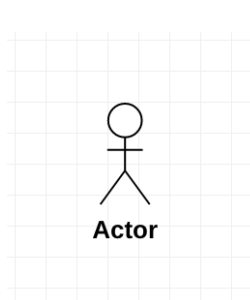


Fig2.3

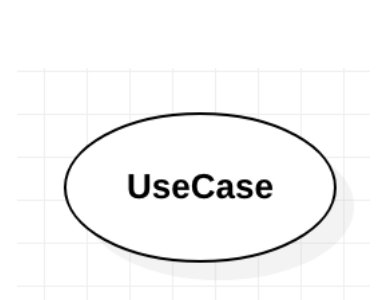


Fig2.4

Aplicația a fost implementată cu ajutorul mediului de dezvoltare IDE IntelliJ IDEEA 2021.2.2, iar ca limbaj de programare am ales să folosesc Java.



Fig2.5 IntelliJ icon

Java este un limbaj de programare orientat pe obiecte, a fost conceput de James Gosling la Sun Microsystems la începutul anilor 90 ,fiind lansat în 1995.

Am ales să folosesc acest limbaj ,pentru că mi se pare extrem de simplu să realizezi aplicații de tip GUI ,deoarece există framework-ul Java Swing ,care permite să realizezi interfețele într-un mod mai prietenos,având o privire de ansamblu asupra interfeței . Totodată , am vrut să îmi îmbunătățesc cunoștințele despre acest limbaj de programare, deoarece în viitor îmi doresc să lucrez la proiecte de dimensiuni mai mari .

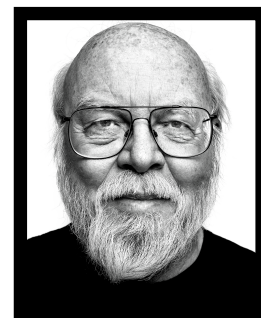


Fig2.6 James Gosling

3.Descrierea diagramelor UML

3.1 Diagrama de use case

Diagrama de use case prezintă o colecție de cazuri de utilizare și actori.

În cazul acestei teme :

- actorul este utilizatorul(administratorul/angajarul)
- iar cazurile de utilizare sunt:

Pentru administrator

În caz de succes:

- Username-ul și parola sunt corecte și preluate din fișierul staff.xml
- Adaugă un nou utilizator completând toate field-urile necesare și apasă butonul Add
- Selectează utilizatorul a cărui date vrea să le modifice,completând field-urile necesare,toate detaliile despre utilizator înafară de username pot fi modificate.
- Totodată poate să steargă utilizatorul selectat prin apăsarea butonului Delete
- Poate să părasească contul prin apasarea butonului LogOut

În caz de nesucces:

- Username-ul sau/și parola nu sunt corecte, nu se găsesc în fisierul staff.xml
- Nu selectează produsul pe care dorește să-l modifice
- Încearcă să modifice username-ul unui utilizator selectat
- Încearcă să adauge un nou utilizator care are același username ca un utilizator care există deja
- În toate situațiile se vor genera eroriile/avertizarile necesare prin interfața grafică

Pentru angajat

În caz de succes:

- Username-ul și parola sunt corecte și preluate din fișierul staff.xml
- Adaugă un nou film completând toate field-urile necesare și apasă butonul Add,
- Selectează filmul a cărui date vrea să le modifice, completând field-urile necesare, toate detaliile despre film în afară de titlu pot fi modificate.
- Totodată poate să steargă filmul selectat prin apăsarea butonului Delete
- Poate să caute un film pe baza titlului acestuia
- Poate să filtreze filmele după numeroase criterii: tip film, categorie film, anul realizării
- Poate să salveze rapoarte cu informații despre filme în mai multe formate: csv, json.

În caz de nesucces:

- Username-ul și parola sunt corecte și preluate din fișierul staff.xml
- Încearcă să adauge un nou film care are același titlu ca un alt film care există deja, iar filmul nu este de tip serial. Dacă filmul are același nume cu alt film, dar ambele filme sunt de tip serial, dar anul realizării este același.
- Încearcă să modifice titlul unui film selectat
- În toate situațiile se vor genera eroriile/avertizarile necesare prin interfața grafică

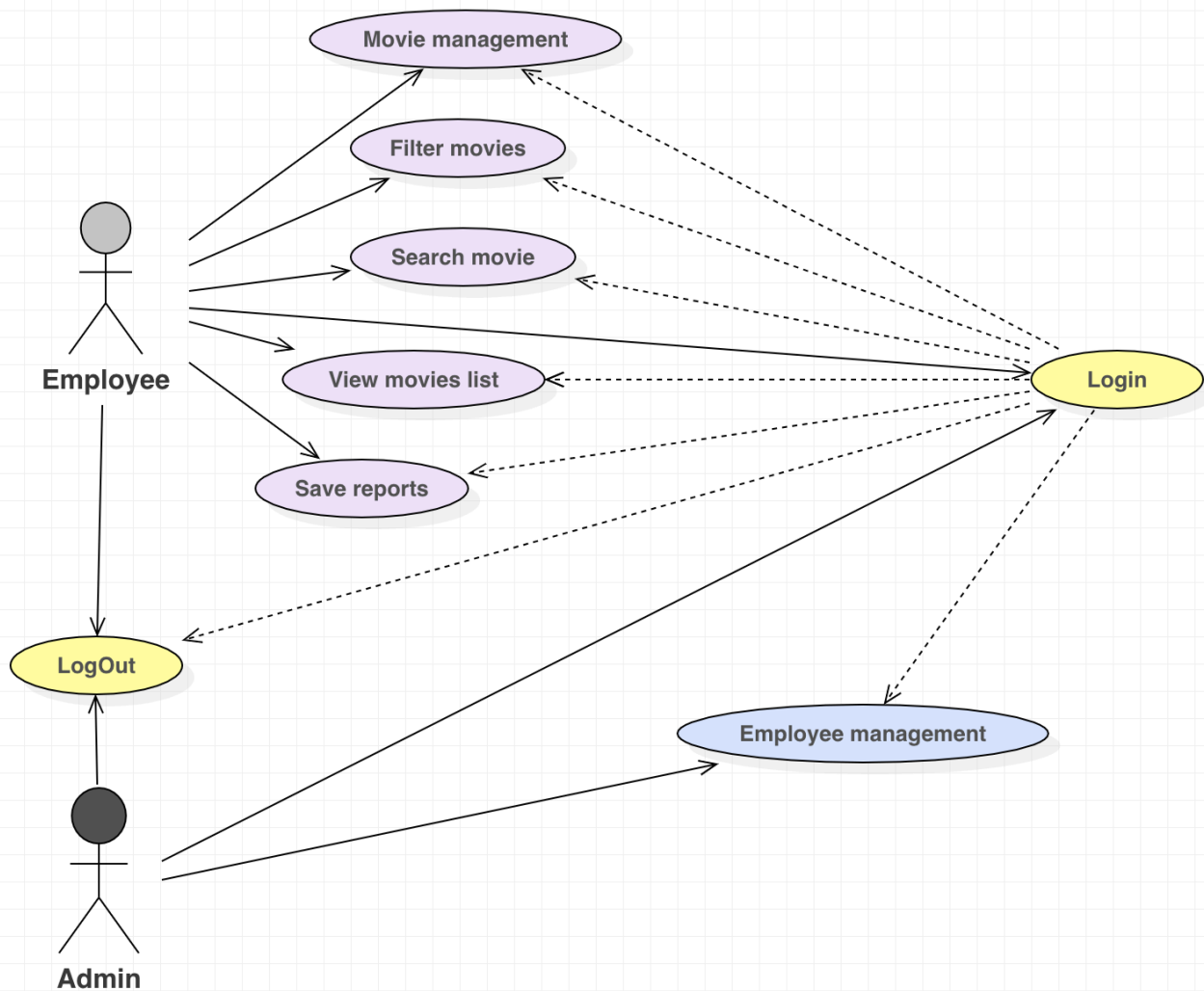


Fig 3.1.1 Diagrama de use case

3.3 Diagrama de clase

Reprezintă un set de clase, interfețe, colaborări și alte relații.

Diagrama de clase a fost realizată ,respectând modelul arhitectural Model-View-Controller.Astfel fiecare pachet ,clasa ,are un rol prestabilit.

MVP este un model arhitectural conceput pentru a facilita testarea automată a unităților și pentru a îmbunătăți separarea preocupărilor în logica prezentării.

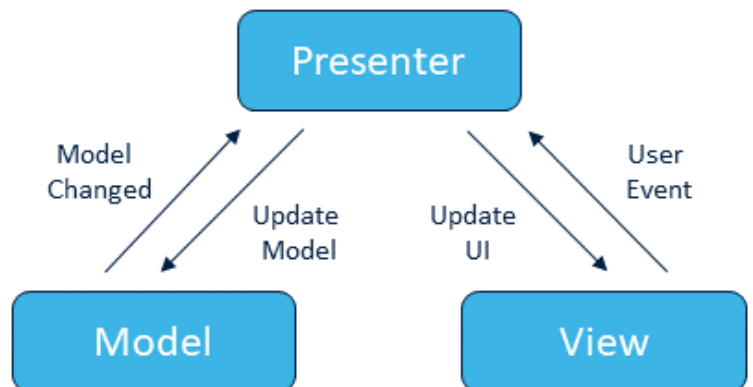
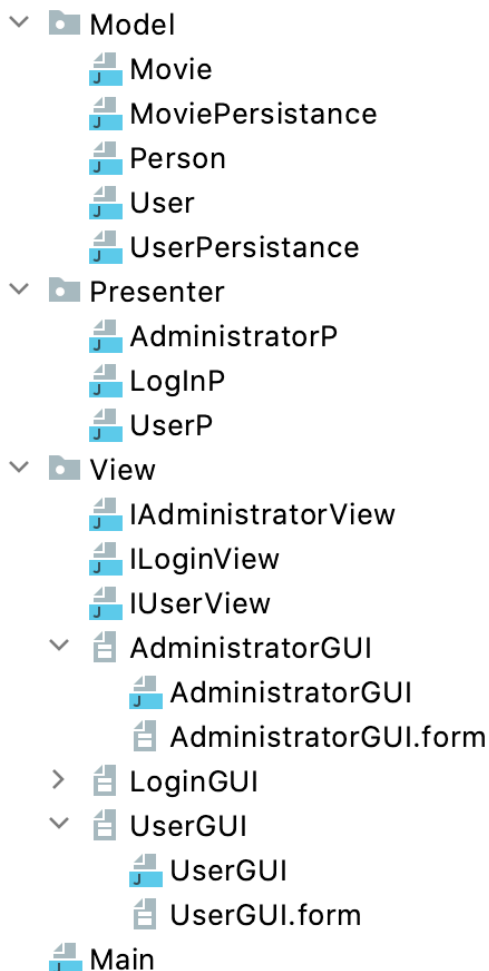


Fig 3.2.3 Modelul arhitectural MVP

Fig 3.2.1 Pachetul cu clasele corespunzătoare

- **Model** – definește datele care trebuie afișate sau asupra cărora se acționează în alt mod în interfața cu utilizatorul.

Curpinde următoarele clase:

- Movie : cuprinde detalii despre film(titlu,categorie,tip,anul realizari)
- MoviePersistence : curpinde o lista de filme asupra careia se vor realiză diferite metode care au ca scop scriere/citirea/modificarea unui fisier de tip xml
- Person : cuprinde detaliile despre o persoană din viața reala(numa și prenume)
- User : cuprinde detaliile despre utilizator(username,password,role)
- UserPersistence : curpinde o lista de utilizatori asupra careia se vor realiză diferite metode care au ca scop scriere/citirea/modificarea unui fisier de tip xml

- **View** - interfața pasivă care afisează date(modelul) și direcționează comenzile utilizatorului (evenimente) către prezentator pentru a acționa asupra acestor date.

Curpinde următoarele clase:

-IAdministratorView : o interfața care cuprinde metodele de get/set pentru pagina administratorului

-IUIView : o interfața care cuprinde metodele de get/set pentru pagina angajatului

-ILoginView : o interfața care cuprinde metodele de get/set pentru pagina de login

-AdministratorView: aici se proiectează interfața pt administrator prin numeroasele attribute prezente, și se gasesc metodele pentru butoanele care realizează comunicarea cu componenta din presenter corespunzatoare administratorului

-UIView: aici se proiectează interfața pt angajat prin numeroasele attribute prezente, și se gasesc metodele pentru butoanele care realizează comunicarea cu componenta din presenter corespunzatoare angajatului

-LoginView: aici se proiectează interfața pt pagina de login prin numeroasele attribute prezente, și se gasesc metodele pentru butoanele care realizează comunicarea cu componenta din presenter corespunzatoare pagini de login

- **Presenter** - preia datele din depozit(model) și le formatează pentru afișare

Curpinde următoarele clase:

-UserP : realizeaza interacțiunea dintre modelul corespunzator pentru film și dintre interfața pentru utilizator

-AdministratorP : realizeaza interacțiunea dintre modelul corespunzator pentru utilizator și dintre interfața pentru admin

-LoginP : realizeaza interacțiunea dintre modelul corespunzator pentru utilizator și dintre interfața pentru pagina de login

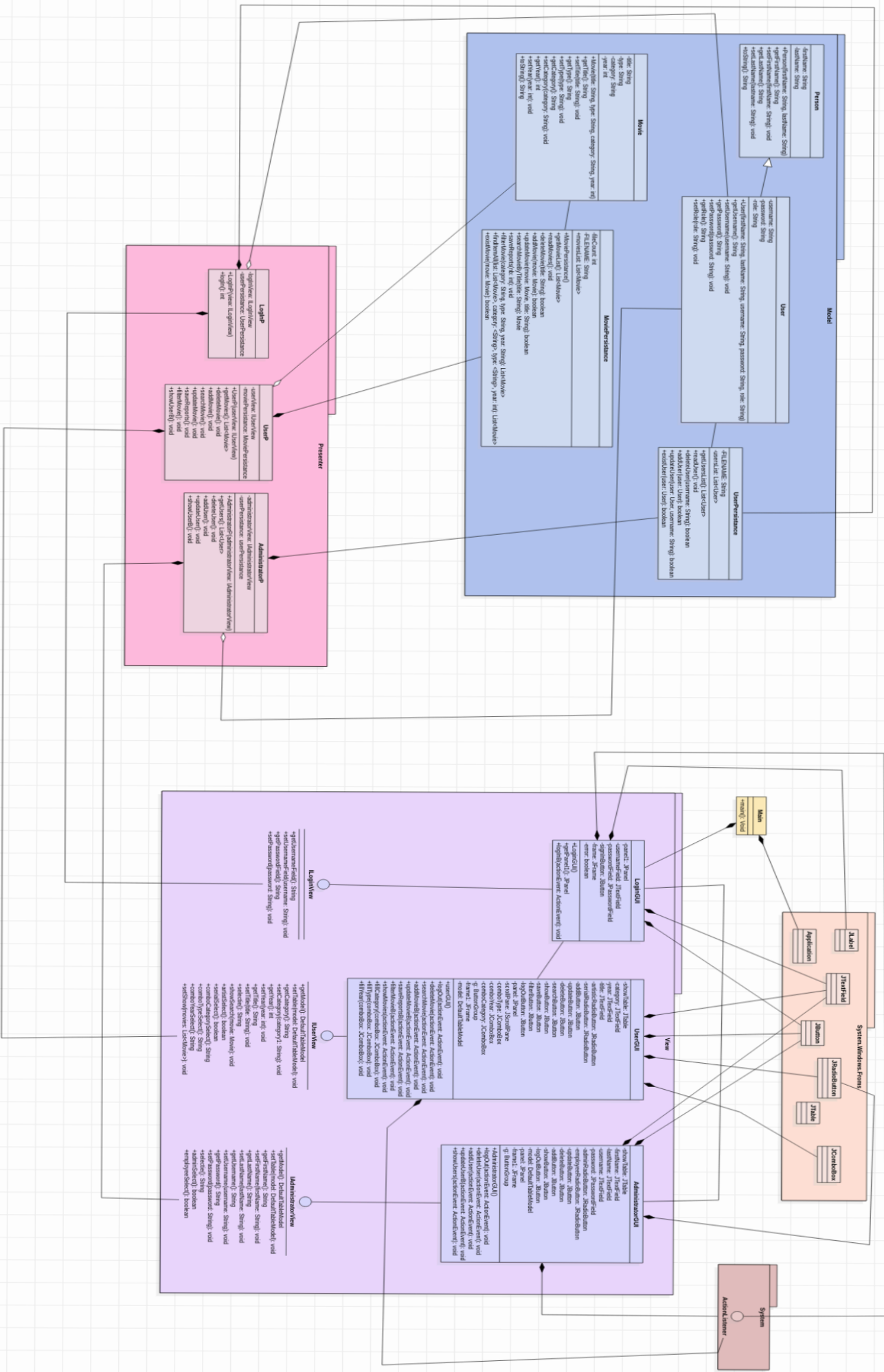


Fig 3.2.4 Diagrama de clase

4.Descrierea aplicației

Primele filme au revoluționat viața culturală a lumii. Prin îmbunătățiri constante aduse camerei de filmat și tehnicilor de proiecție, filmul a devenit cel mai important mediu de comunicare al secolului al XXI-lea, ajungând să modeleze lumea așa cum nu a mai făcut-o nicio altă artă până la apariția sa.

Tehnologia ,în ultimi anii , a evoluat drastic,datorită apariției noului "Coronavirus" ,iar marile case de producție filme ,au fost nevoite sa își mute activitatea în online ,pentru a nu pierde interacțiunea dintre administratori și angajații.

Suntem actorii unei lumi aflate în continuă ascensiune , tocmai datorită acestui fapt și noi trebuie să ne menținem potrivit așteptărilor . Așadar , pentru a eficientiza munca diferitelor case de productie de filme ,propun următoarea aplicație denumită "Movie House", pentru angajați și administrator ,care are următoarele beneficii:

- Vizualizarea listei care conține filmele
- Filtrarea filmelor după anumite criterii: tip film,categorie film,anul realizării
- Operații CRUD în ceea ce privește persistența filmelor
- Căutarea unui film după titlu
- Salvarea rapoarte/liste cu informații despre filme în mai multe formate
- Operații CRUD pentru informațiile legate de utilizatorii aplicației.

Codul care stă în spatele funcționalității acestei aplicații este unul destul de simplu și basic,am încercat să scriu cât mai clar și simplu toate operațiile menționate mai sus,am împărțit aplicația în clase specifice, am respectat modelul arhitectural MVP , respectiv modelul SOLID, pentru a fi mai simplu și eficient , a modifica sau a adăuga alte funcționalități ,fără să se producă modificări majore liniilor de cod existente , ci doar să adăugăm alte linii.

De asemenea ,am încercat să creez o interfață unică,care să îmbine simplitatea cu stilul meu, și a ieșit ce puteți vedea mai jos,eu zic că am făcut o treabă destul de buna.~)

Pentru filtrările după anumite categorii am folosit expresiile lambda ,doarece mi s-a părut o modalitatea mai ușoară și mai eficientă decât a face numeroase comparații între diferite filme.

Aplicația a fost concepută să fie utilizată de orice categorie de persoană ,atât de una care are cunoștințe în tainele programării cât și de una care acum aude pentru prima dată cuvântul "programare".Prin urmare, folosesc o interfață User-Friendly pentru a permite utilizatorilor să se simtă mai familiarizați cu programul chiar înainte de a-l fi utilizat. Printr-o interfață utilizatorul poate înțelege mai bine cum funcționează programul,poate învăța mai repede modul de utilizare a acestei aplicații.Aceasta interfață cuprinde 3 frame-uri,unul principal pentru logare,iar altele două pentru angajat ,repectiv administrator.

The image shows a login page for a film studio. The window title is "Login page". The background is a solid purple color. At the top, the word "Spectators" is written in a light blue, cursive font. Below it, the word "PASSIONS" is written in large, bold, white, sans-serif capital letters. Underneath "PASSIONS", the text "FILM STUDIO SINCE 2001" is written in a smaller, white, sans-serif font. A horizontal dotted line separates the header from the login fields. There are two input fields: the first is labeled "username" in a cursive font and has a user icon to its left; the second is labeled "password" in a cursive font and has a padlock icon to its left. Below the input fields is a large white button with the text "Log in" in a cursive font. At the bottom of the page, there are three icons: a clapperboard, a movie camera, and a film reel.

Fig4.1 Interfața pentru pagina de login

Admin

LogOut

USERS

-The best-

Username:lohanmihai

Password:mihai

First Name

Last Name

Username

Password

☐ Admin
 ☐ Employee

Add

Update

Delete

Show

Fig4.2 Interfața pentru pagina administratorului

Employee

LogOut

MOVIES

-Produced-

Select

Select

Select

Username:silviaoros

Password:ana

Title

Category

Year

☐ Artistic
 ☐ Serial

Filter

Search

Add

Update

Delete

Show

Save

Fig4.3 Interfața pentru pagina angajatului

