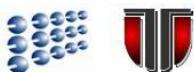


Universitatea Tehnică din Cluj-Napoca
Departamentul de Calculatoare



Inginerie Software

*Proiect: Focus Cinema – Aplicație pentru
gestiunea unui lanț de cinematografe*

*Studenți: Bozdog Raluca-Delia, Lazea Dragoș-Bogdan
Grupa: 30233*

An universitar: 2021 – 2022



Cuprins

<i>1. Specificații și analiza sistemului.....</i>	<i>3</i>
<i>2. Precizarea limbajului de programare ales și a sistemului de operare sub care se face implementarea.....</i>	<i>3</i>
<i>3. Proiectarea aplicației.....</i>	<i>4</i>
<i>3.1.Descrierea arhitecturii sistemului.....</i>	<i>4</i>
<i>3.2.Identificarea funcționalităților aplicației.....</i>	<i>5</i>
<i>3.3.Diagrama de clase.....</i>	<i>7</i>
<i>3.4.Diagrame de interacțiune.....</i>	<i>8</i>
<i>3.5.Detalii de implementare (diagrama bazei de date, diagrama de clase cu atribute și metode, diagrama de module și componente).....</i>	<i>13</i>
<i>3.6.Cazuri de testare.....</i>	<i>16</i>
<i>4. Testarea propriu-zisă a aplicației.....</i>	<i>17</i>
<i>5. Manual de instalare și utilizare.....</i>	<i>23</i>
<i>6. Concluzii și dezvoltări ulterioare.....</i>	<i>32</i>
<i>7. Bibliografie.....</i>	<i>32</i>

1. Specificații și analiza sistemului

Se dorește proiectarea și implementarea unei aplicații web de tipul „Software as a Service” (SaaS) responsabile de gestiunea sistemului informatic asociat unui lanț de cinematografe. Aplicația este destinată utilizării atât de către clienții lanțului de cinematografe, cât și de către personalul administrativ al acestuia, disponând de o interfață grafică ce facilitează interacțiunea sistemului cu utilizatorii de toate tipurile și fiind supusă unui set de specificații de proiectare. Din perspectiva clientului, aplicația trebuie să permită crearea unui nou cont de utilizator, conectarea folosind un cont deja existent, vizualizarea programului, vizualizarea unor detalii despre filmele prezente în program (descriere, trailer), crearea de rezervări, căutarea filmelor după diverse criterii și cumpărarea de bilete în sistem online, în timp ce administratorul sistemului informatic trebuie să poată actualiza și stabili programul fiecărui cinematograf, să administreze conturile clienților înregistrați și să stabilească prețul biletelor.

2. Precizarea limbajului de programare ales și a sistemului de operare sub care se face implementarea

Limbajul de programare ales este Java, întrucât acesta reprezintă un limbaj de programare orientat pe obiecte fapt ce facilitează crearea aplicațiilor cu aplicabilitate în sistemele din lumea reală prin modelarea obiectelor reale (în cazul aplicației de față: film, cinema, client, rezervare etc.) prin intermediul principiilor programării orientate pe obiecte (POO). De asemenea, un alt avantaj al utilizării acestui limbaj de programare îl reprezintă faptul că Java este platform-independent, iar astfel codul Java compilat poate fi rulat pe orice sistem de calcul care suportă Mașina Virtuală Java (JVM), indiferent de sistemul de operare, fiind totodată un limbaj de programare ce dispune de un nivel ridicat de securitate prin evitarea folosirii explicite a pointerilor și prin legarea împreună a datelor și a funcțiilor (un obiect este definit prin attribute și metode), dar și de un sistem automat de administrare a memoriei utilizate și de eliberare a memoriei ocupate de obiectele care nu mai sunt utilizate de niciun program (Java Garbage Collection).

Interfața cu utilizatorul și dezvoltarea aplicației web se va realiza prin intermediul Vaadin, o platformă open-source de dezvoltare a aplicațiilor web pentru Java. Acesta reprezintă un framework ce permite folosirea limbajului de programare Java pentru dezvoltarea interfețelor grafice cu utilizatorul fără a necesita utilizarea directă a HTML sau JavaScript, disponând de o arhitectură de tipul server-side ce implică faptul că majoritatea logicii asociate interfeței grafice funcționează securizat la nivelul serverului, fiind astfel redusă expunerea la atacuri de securitate. Din perspectiva clientului, Vaadin este construit pe baza componentelor standard de dezvoltare Web.

Sistemul de operare sub care se face implementarea este Windows 10, standardul pe 64 de biți (x86-64).

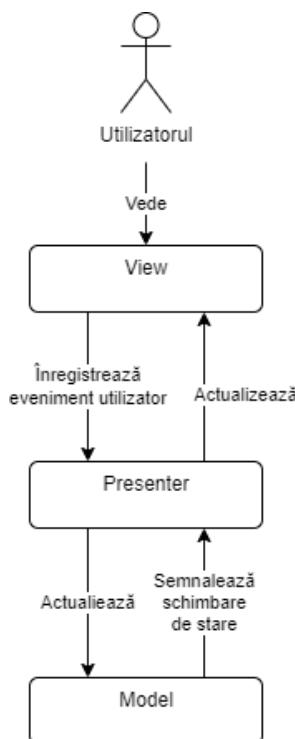
3. Proiectarea aplicației

3.1. Descrierea arhitecturii sistemului

Arhitectura sistemului urmărește pattern-ul arhitectural Model-View-Presenter, o variație a cunoscutului pattern Model-View-Controller care este însă aplicabilă și în domeniul aplicațiilor web implementate folosind limbajul de programare Java și framework-ul Vaadin. Astfel, aplicația are la bază un model de date descris prin intermediul obiectelor definite de clase precum Film, Cinema, Bilet, Client, pe care îl expune utilizatorului sub formă grafică (Graphical User Interface). Conectarea informației din modelul de date la informația aflată la nivelul grafic, de prezentare, al aplicației este realizat folosind informații de prezentare (control).

- Modelul conține informația brută, referitoare la obiectele care interacționează într-o astfel de aplicație din sfera cinematografiei (Film, Client, Cinema, Rezervare etc.), comportamentul și stările posibile ale acestora, fără a conține detalii despre modul de expunere al acesteia către utilizatorul sistemului.
- View-ul are responsabilitatea de a prezenta informația utilizatorului aplicației, știind cum să o acceseze, dar nefiind capabilă să o interpreteze și să o manipuleze corespunzător, conținând doar elementele legate de interfață cu utilizatorul (UI).
- Presenter-ul realizează legătura dintre model și view, ascultând și înregistrând evenimentele de la nivelul view-ului și traducându-le în acțiuni concrete executate de către model (cumpără bilet, evaluează film etc.), ca mai apoi să poată actualiza view-ul.

Diagrama arhitecturii sistemului poate fi vizualizată în figura de mai jos:



Figură 3.1. Diagrama arhitecturii sistemului

3.2. Identificarea funcționalităților aplicației (diagrama de cazuri de utilizare)

S-au identificat, conform specificațiilor de proiectare, patru tipuri de utilizatori: client neînregistrat, client înregistrat, administrator și administrator general.

Din punctul de vedere al clientului, aplicația implementată trebuie să suporte următoarele cerințe funcționale:

- Crearea unui cont de utilizator;
- Conectarea la sistemul informatic utilizând un cont deja existent;
- Vizualizarea filmelor disponibile;
- Vizualizarea programului zilnic al fiecărui cinematograf;
- Vizualizarea unor detalii despre fiecare film din program (trailer, scurtă descriere);
- Căutarea filmelor după anumite criterii (nume, data difuzării, cinematograf);
- Crearea de rezervări (dacă clientul este înregistrat);
- Cumpărarea de bilete în sistem online;
- Modificarea datelor personale asociate contului de utilizator;
- Ștergerea contului de utilizator.

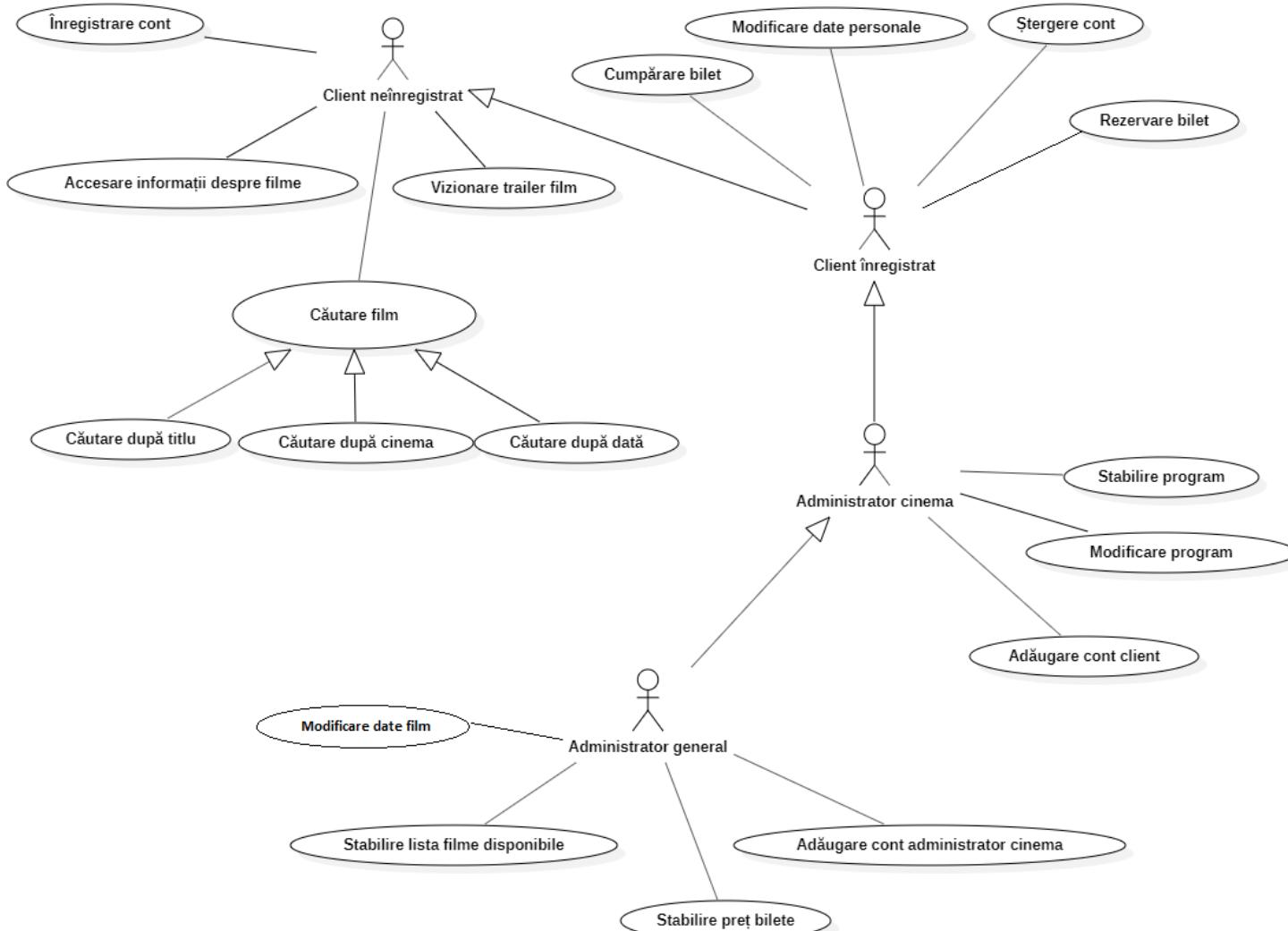
Suplimentar, personalul administrativ al lanțului de cinematografe va avea asociat următorul cadru de cerințe funcționale:

- Adăugarea de filme noi la program;
- Eliminarea filmelor din program;
- Modificarea și stabilirea programului fiecărui cinematograf;
- Adăugarea unor noi conturi de clienți;
- Ștergerea conturilor existente de clienți;
- Stabilirea prețului biletelor.

Cadrul de cerințelor nefuncționale ale aplicației de implementat este următorul:

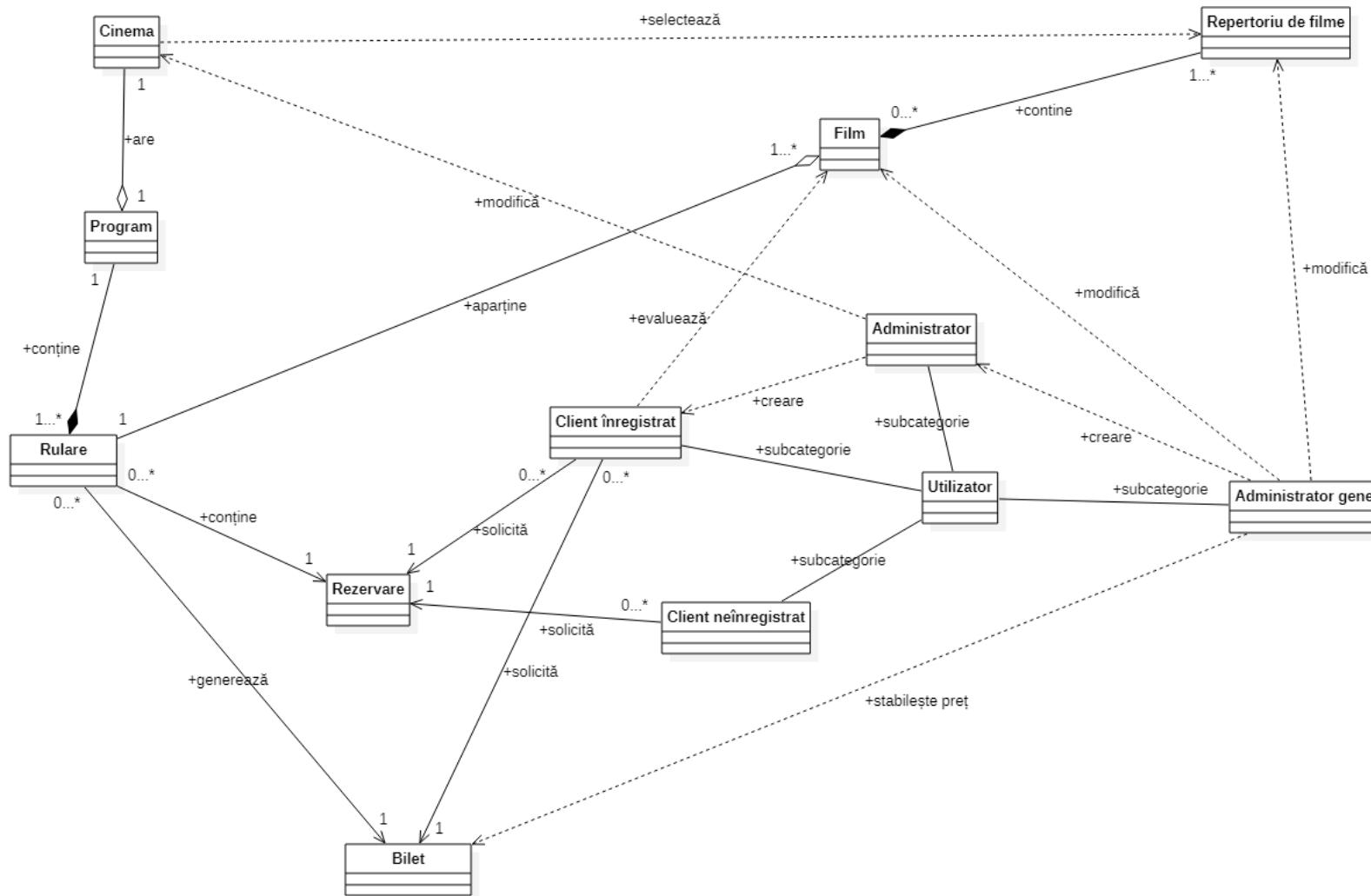
- Aplicația trebuie să disponă de o interfață grafică intuitivă, care să determine o interacțiune prietenoasă cu utilizatorul;
- Aplicația trebuie să fie ușor de utilizat pentru orice tip de utilizator, indiferent de domeniul de activitate al acestuia.

Cazurile de utilizare identificate sunt sintetizate în următoarea diagramă de cazuri de



Figură 3.2. 1 Diagrama cazurilor de utilizare

3.3. Diagrama de clase

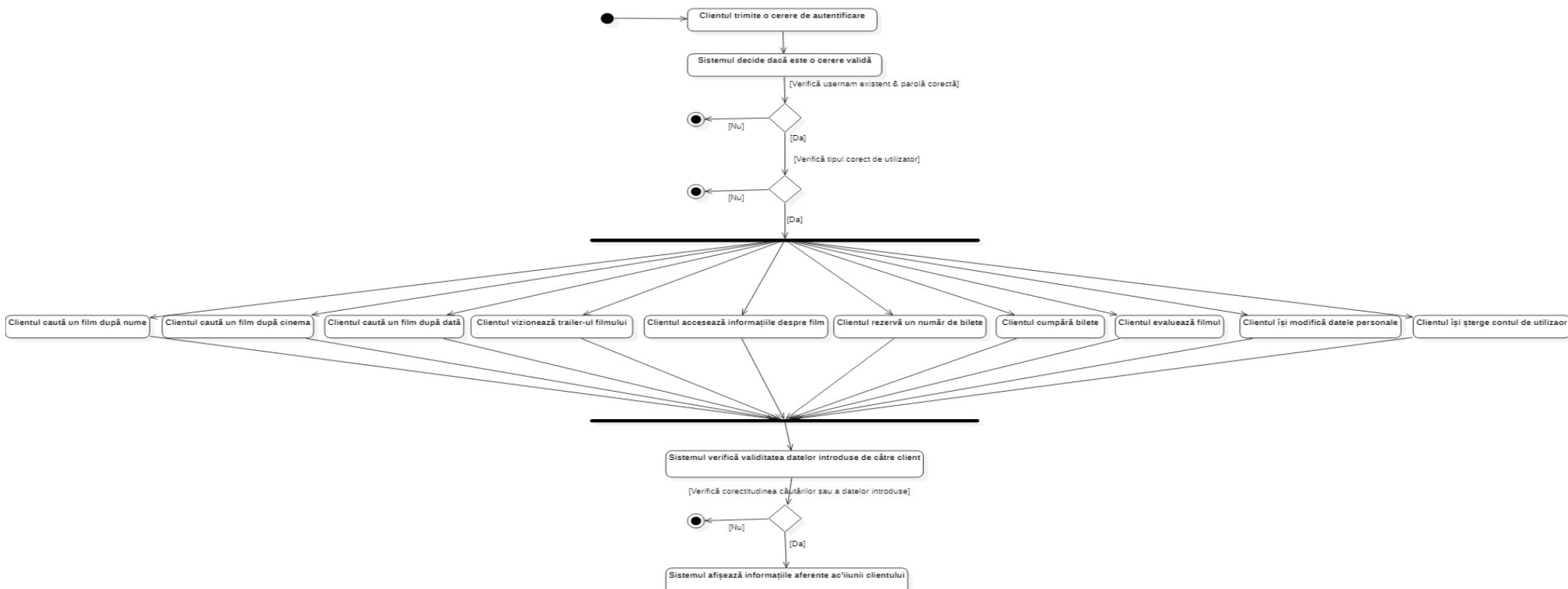


Figură 3.3. 1 Diagrama UML de clase

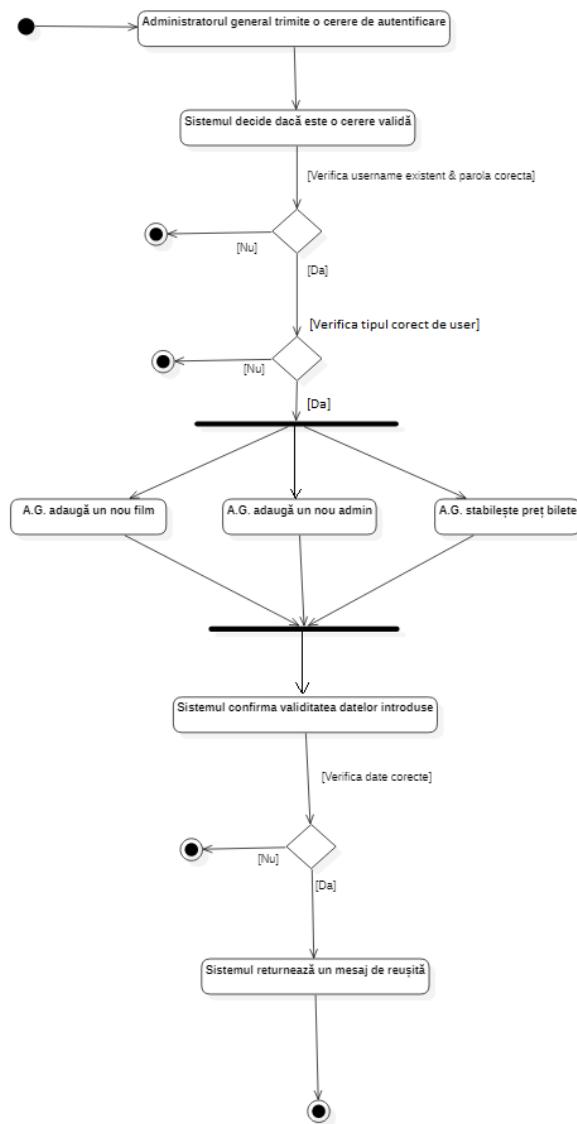
3.4. Diagrame de interacțiune

Diagramele de interacțiune descriu comportamentul aplicației proiectate și succesiunea pașilor pe care aceasta o urmează pentru a îndeplini sarcinile necesare realizării funcționalităților de care aceasta trebuie să dispună. În cele ce urmează se vor expune multiple diagrame de interacțiune pentru fiecare tip de utilizator și asociate diferitor cazuri de utilizare.

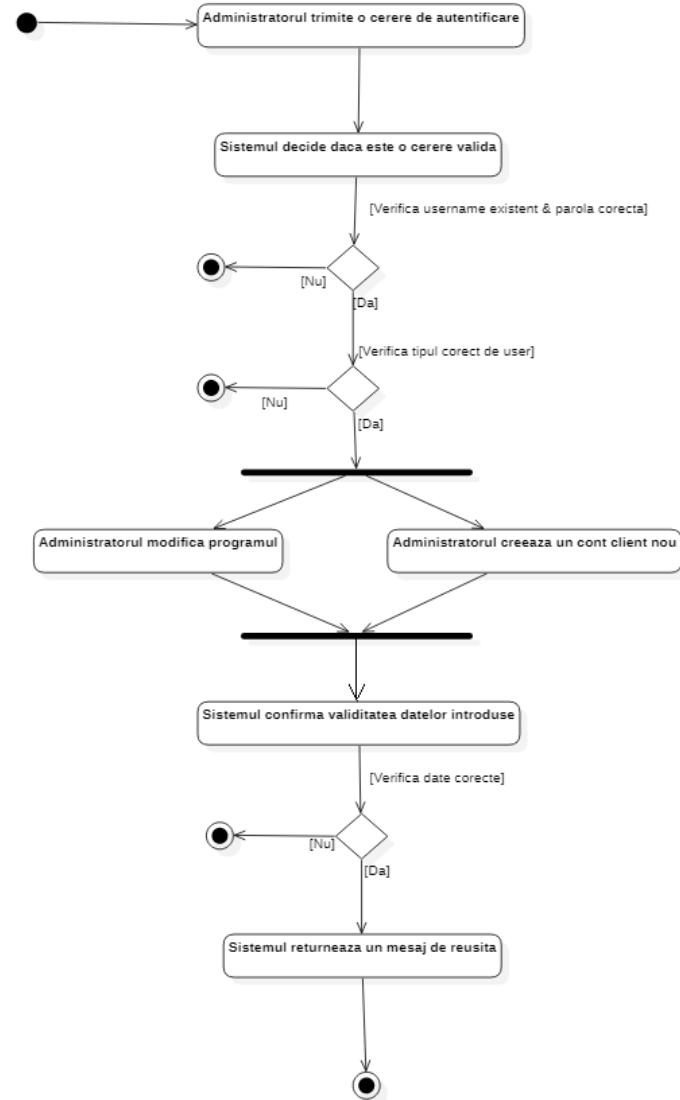
i. Diagrame de activitate



Figură 3.4. 1 Diagrama de activitate pentru client înregistrat

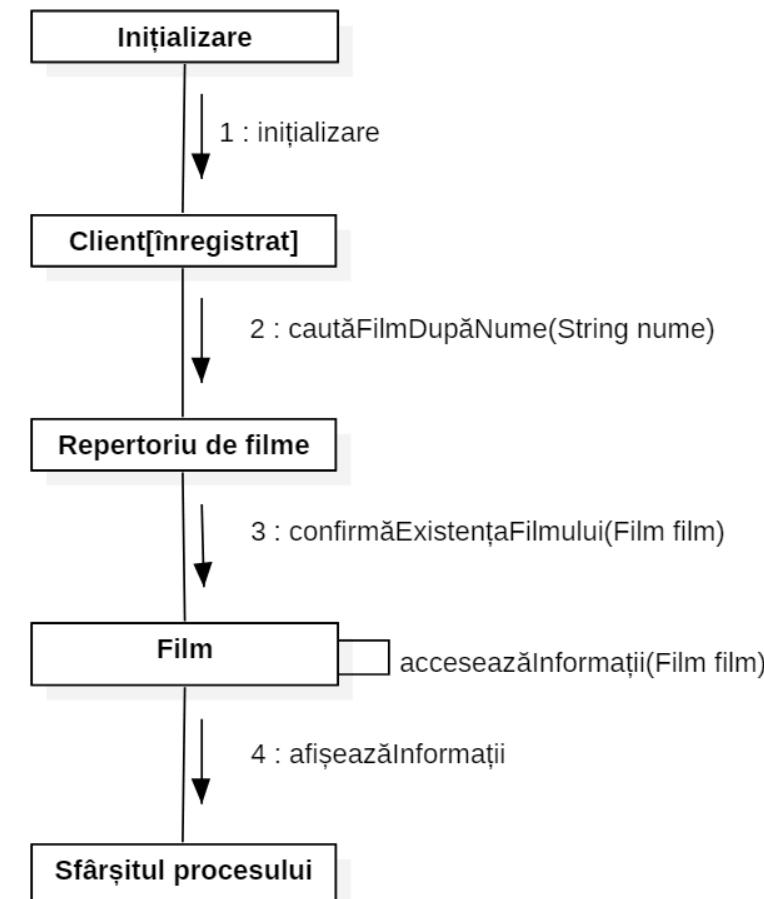
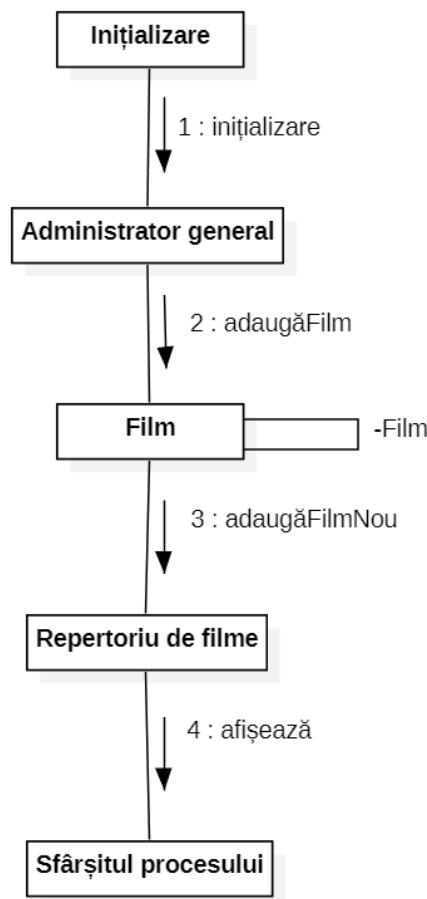
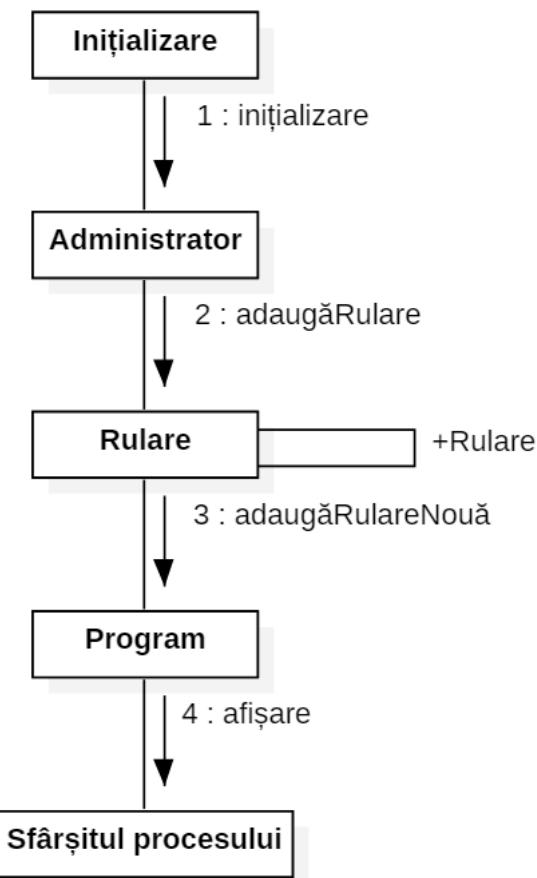


Figură 3.4. 2 Diagrama de activitate pentru administrator general



Figură 3.4. 3 Diagrama de activitate pentru administrator

ii. *Diagrame de colaborare*

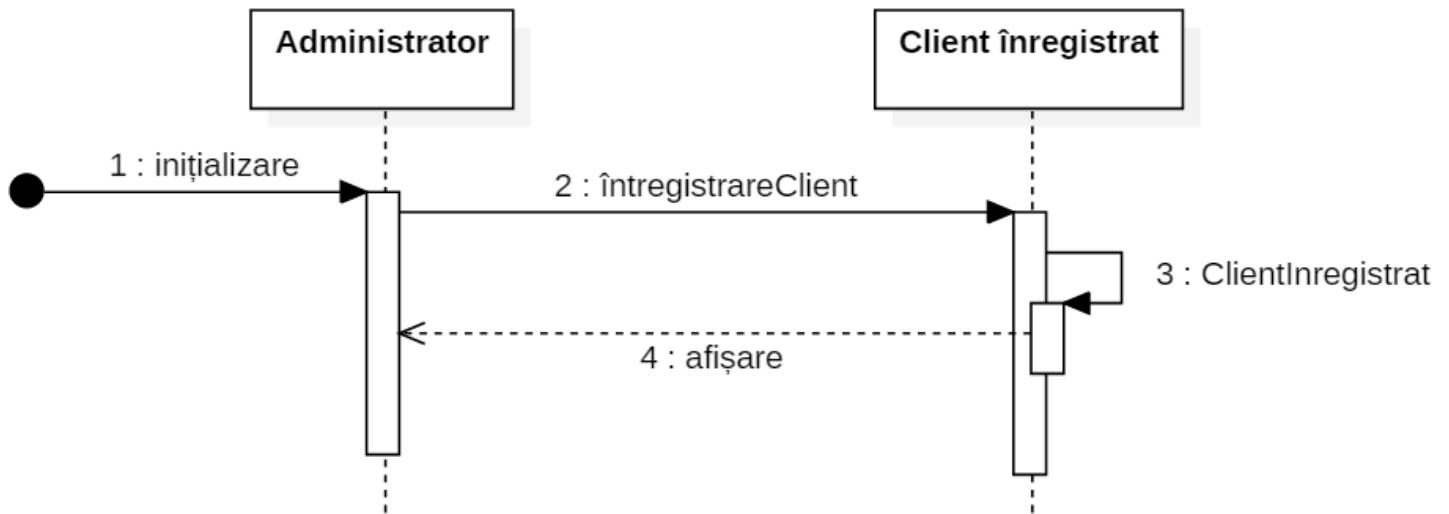


Figură 3.4. 4 Diagrama de colaborare pentru administrator

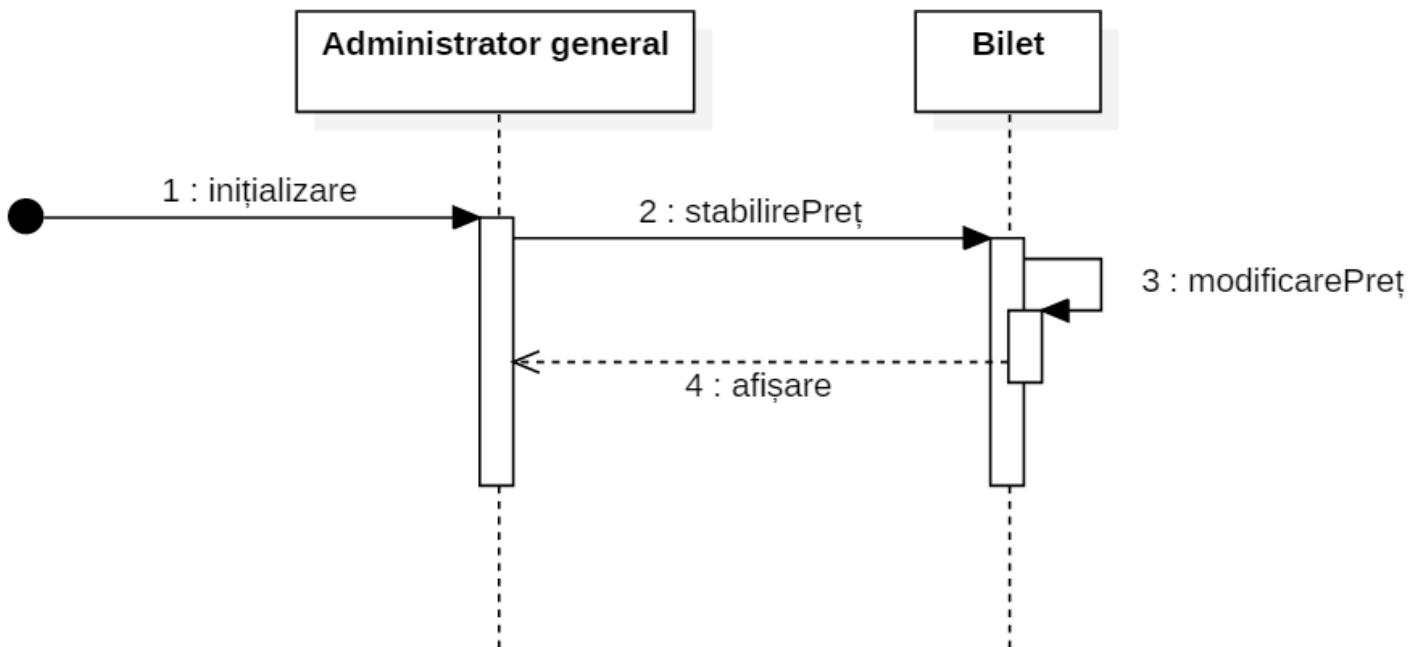
Figură 3.4. 5 Diagrama de colaborare pentru administrator general

Figură 3.4. 6 Diagrama de colaborare pentru client (înregistrat)

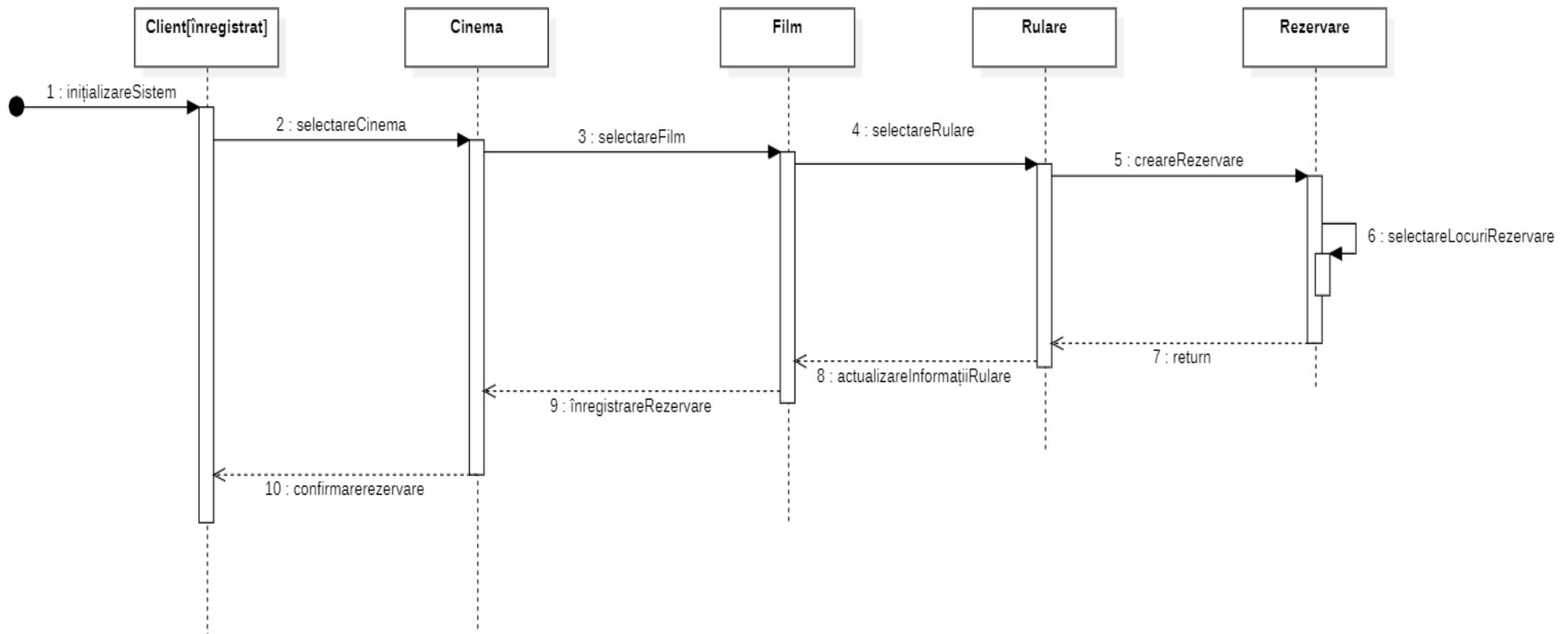
iii. Diagrame de secvențiere



Figură 3.4. 7 Diagrama de secvențiere pentru administrator pentru cazul de utilizare Adăugare cont client



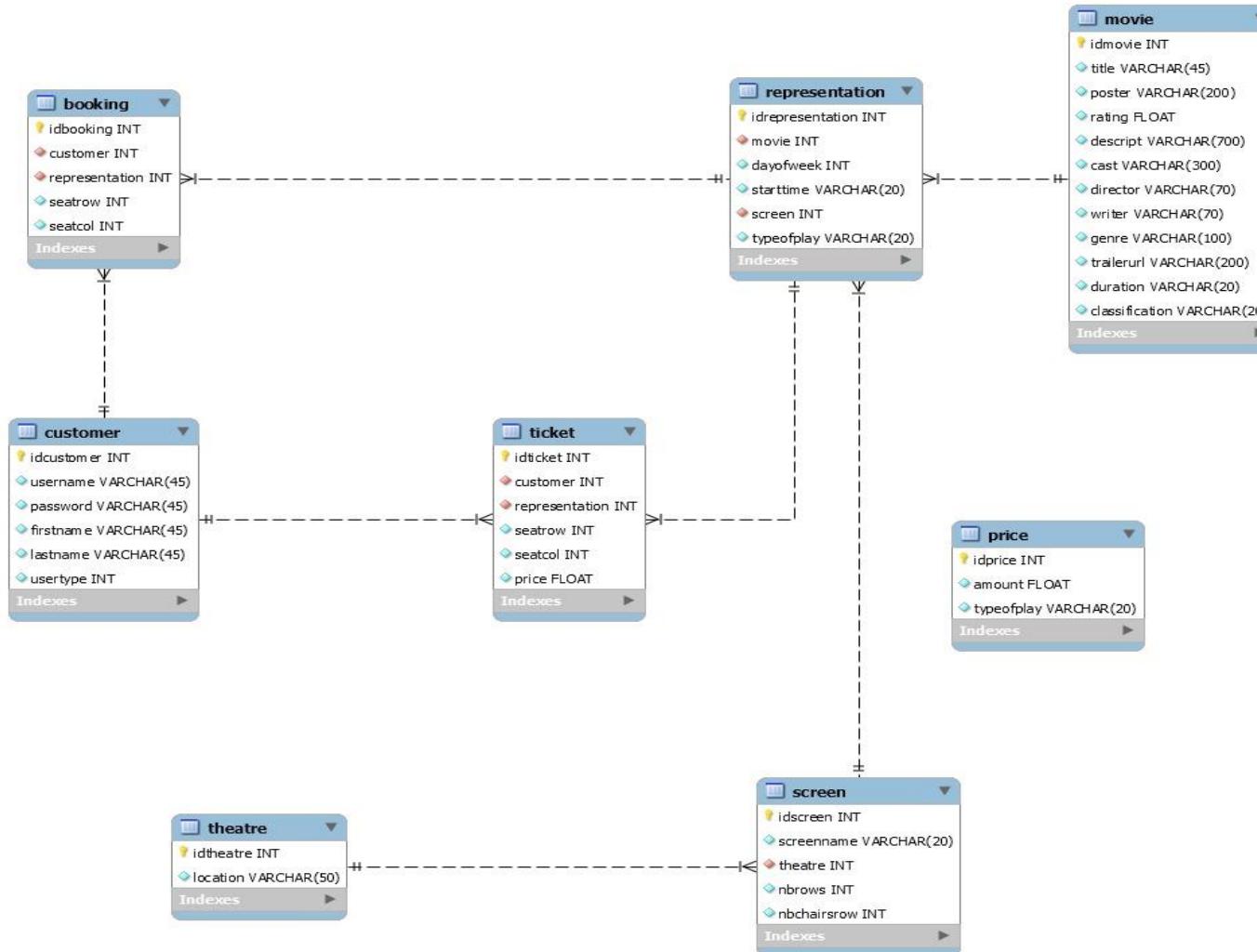
Figură 3.4. 8 Diagrama de secvențiere pentru administrator general pentru cazul de utilizare Stabilire preț bilete



Figură 3.4. 9 Diagrama de secvențiere pentru client înregistrat pentru cazul de utilizare Rezervare bilet

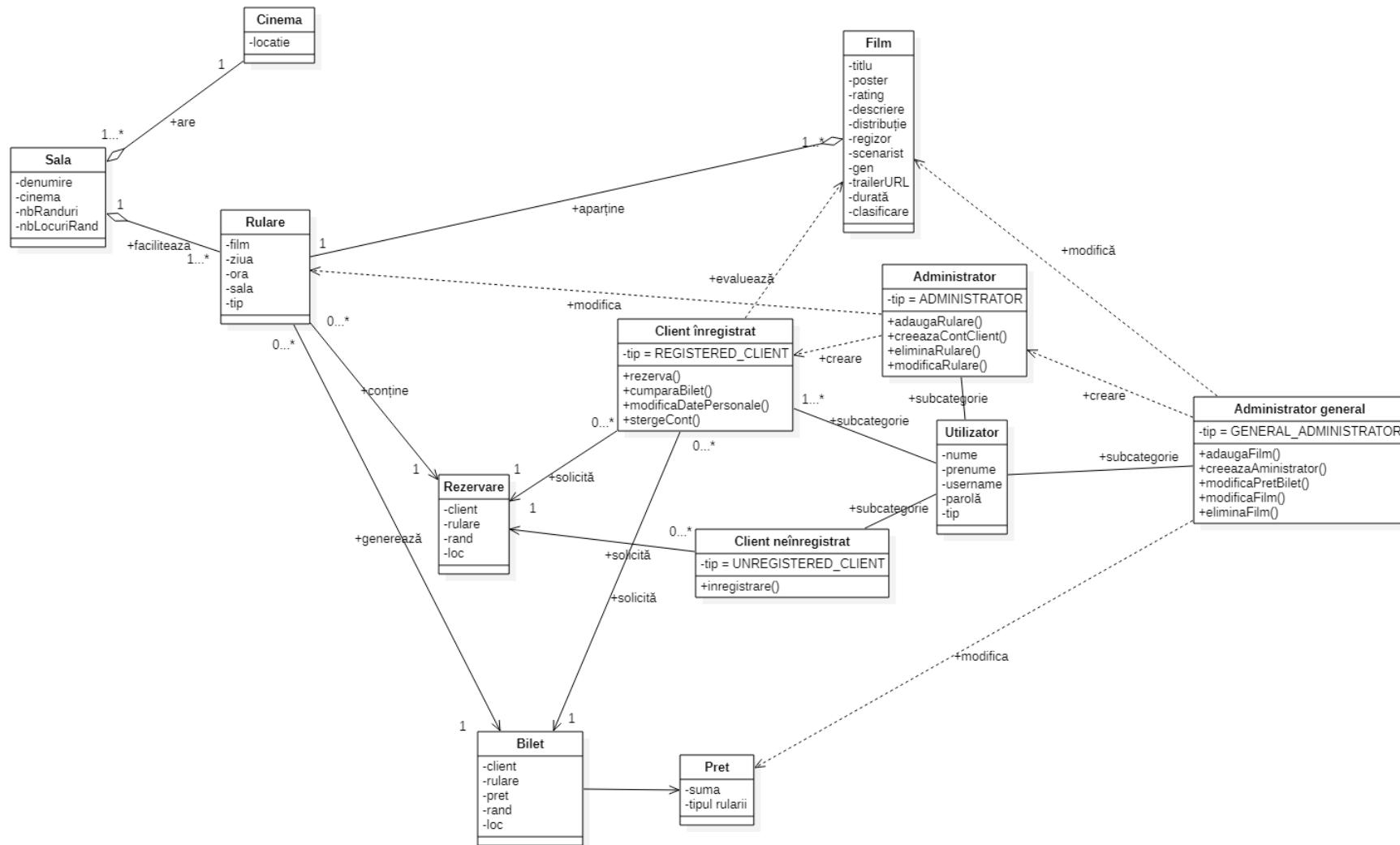
3.5. Detalii de implementare

i. Diagrama bazei de date



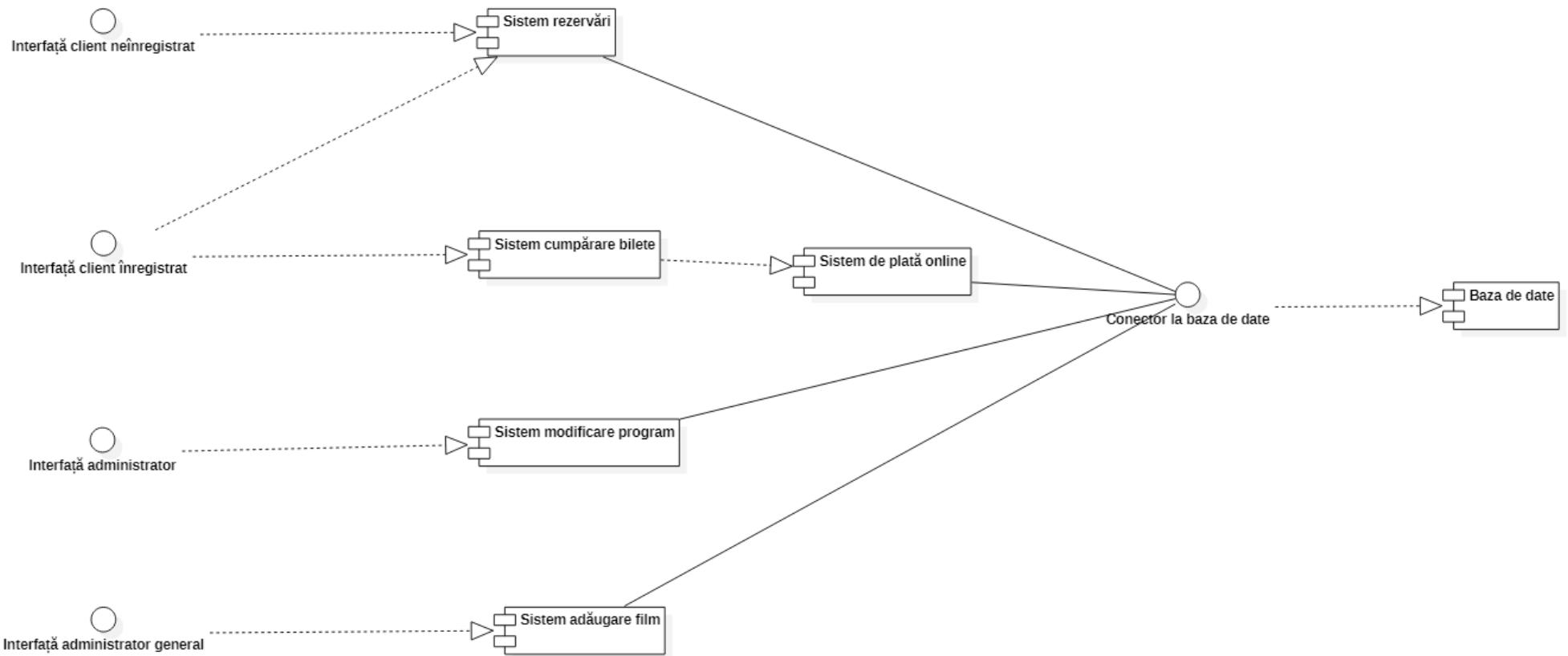
Figură 3.5. 1 Diagrama bazei de date

ii. Diagrama de clase cu atribute și metode



Figură 3.5. 2 Diagrama de clase cu atribute și metode

iii. Diagrama de module și componente



Figură 3.5. 3 Diagrama de module și componente

3.6. Cazuri de testare

Testarea aplicației se va realiza astfel încât să poată surprinde eventualele erori și situații excepționale ce pot fi întâlnite în timpul rulării acesteia. Astfel, au fost imaginate următoarele cazuri de testare:

- a. Crearea unui nou cont de utilizator

Actorul principal: Utilizatorul

Scenariul de succes:

- i. Utilizatorul introduce corect toate informațiile necesare creării unui nou cont de utilizator.
- ii. Se creează un nou cont de utilizator, prin apăsarea butonului Înregistrează-te din interfața grafică.
- iii. Contul de utilizator a fost creat, aplicația permitând accesul noului utilizator la facilitățile oferite.

Scenarii alternative (posibile erori): Utilizatorul nu introduce toate informațiile necesare înregistrării, introduce un nume de utilizator deja existent sau confirmarea parolei nu coincide cu parola inițial introdusă.

- i. Utilizatorul este notificat prin apariția în interfața grafică a unui mesaj de eroare corespunzător excepției generate.
- ii. Scenariul se întoarce la pasul de introducere a informațiilor necesare creării unui nou cont de către utilizator.

- b. Conectarea clientului la aplicația web

Actorul principal: Utilizatorul

Scenariul de succes:

- i. Utilizatorul introduce corect numele de utilizator și parola.
- ii. Utilizatorul se conectează la sistem, prin apăsarea butonului conectează-te din interfața grafică, aplicația permitând accesul la utilitățile disponibile.

Scenarii alternative (posibile erori): Utilizatorul nu introduce toate informațiile necesare conectării sau le introduce greșit.

- i. Utilizatorul este notificat prin apariția în interfața grafică a unui mesaj de eroare corespunzător excepției generate.
- ii. Scenariul se întoarce la pasul de introducere a informațiilor necesare conectării la sistem.

- c. Crearea unei rezervări

Actorul principal: Utilizatorul

Scenariul de succes:

- i. Utilizatorul selectează opțiunea de creare a unei noi rezervări din secțiunea corespunzătoare filmului pe care dorește să îl vizioneze.

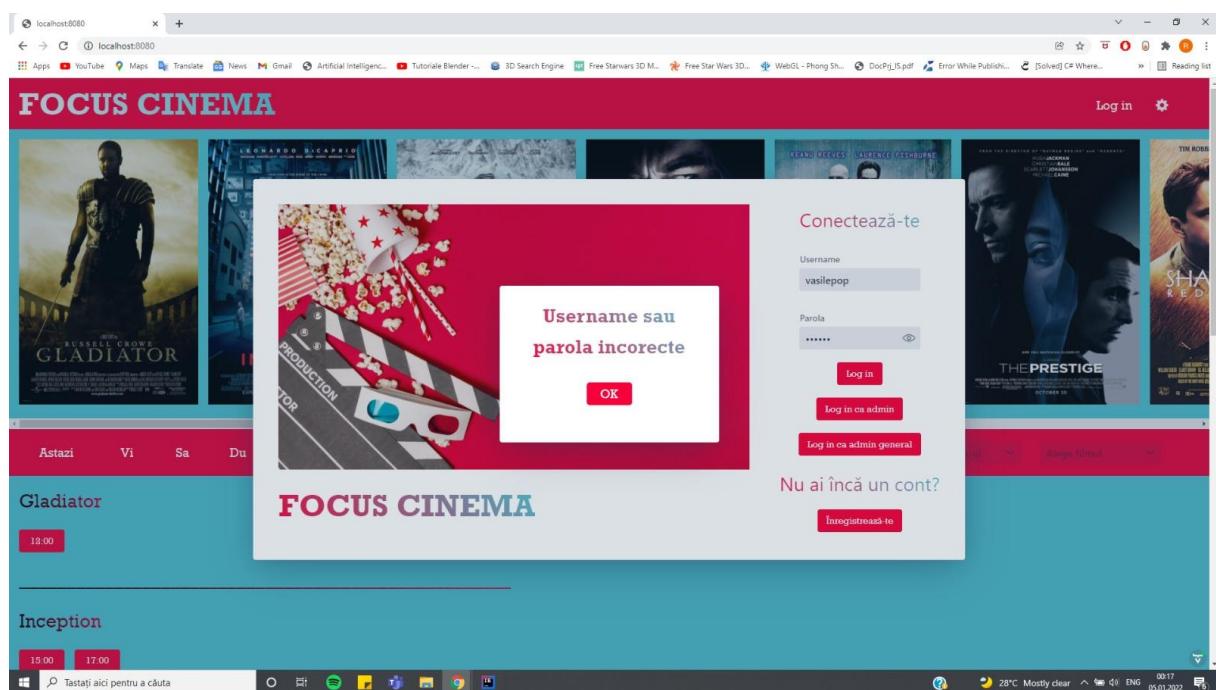
- ii. Utilizatorul selectează data și ora la care rulării la care dorește să participe.
 - iii. Utilizatorul selectează locul/locurile pe care dorește să le rezerve.
 - iv. Cererea de rezervare a locurilor este înregistrată.
- Scenarii alternative (posibile erori): Utilizatorul selectează un loc indisponibil (care a fost rezervat anterior).
- i. Utilizatorul este notificat prin apariția în interfață grafică a unui mesaj de eroare corespunzător excepției generate.
 - ii. Scenariul se întoarce la pasul de selectare a locurilor pe care utilizatorul dorește să le rezerve.

4. Testarea propriu zisă a aplicației

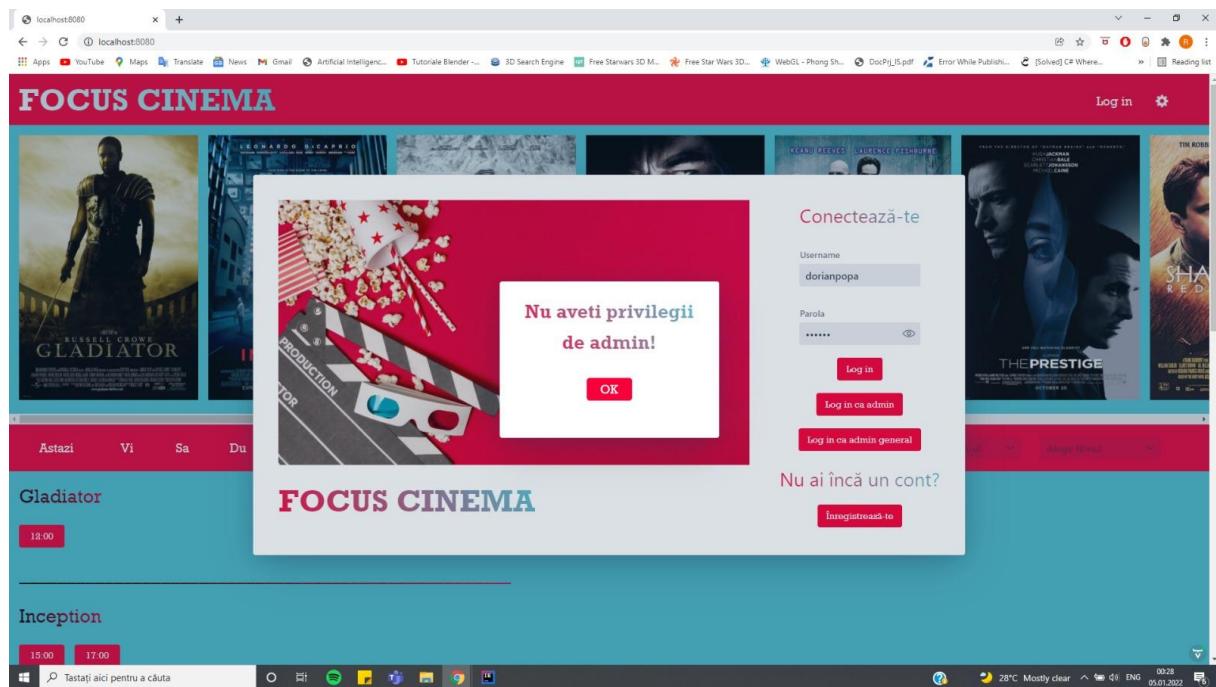
Testarea propriu zisă a aplicației a fost efectuată astfel încât să acopere atât toate cazurile de testare proiectate în cadrul etapei de proiectare, cât și alte funcționalități suplimentare, auxiliare, având drept scop verificarea validității rezultatelor și a comportamentului pe care sistemul îl are în timpul efectuării operațiilor solicitate de către utilizator, pentru satisfacerea funcționalităților implementate.

În urma testării, aplicația a reacționat pozitiv la toate solicitările, având comportamentul așteptat și furnizând rezultate corecte pentru fiecare dintre funcționalitățile implementate, atât în ceea ce privește cazurile de testare mai sus menționate, cât și celealte funcționalități din lista de cerințe funcționale.

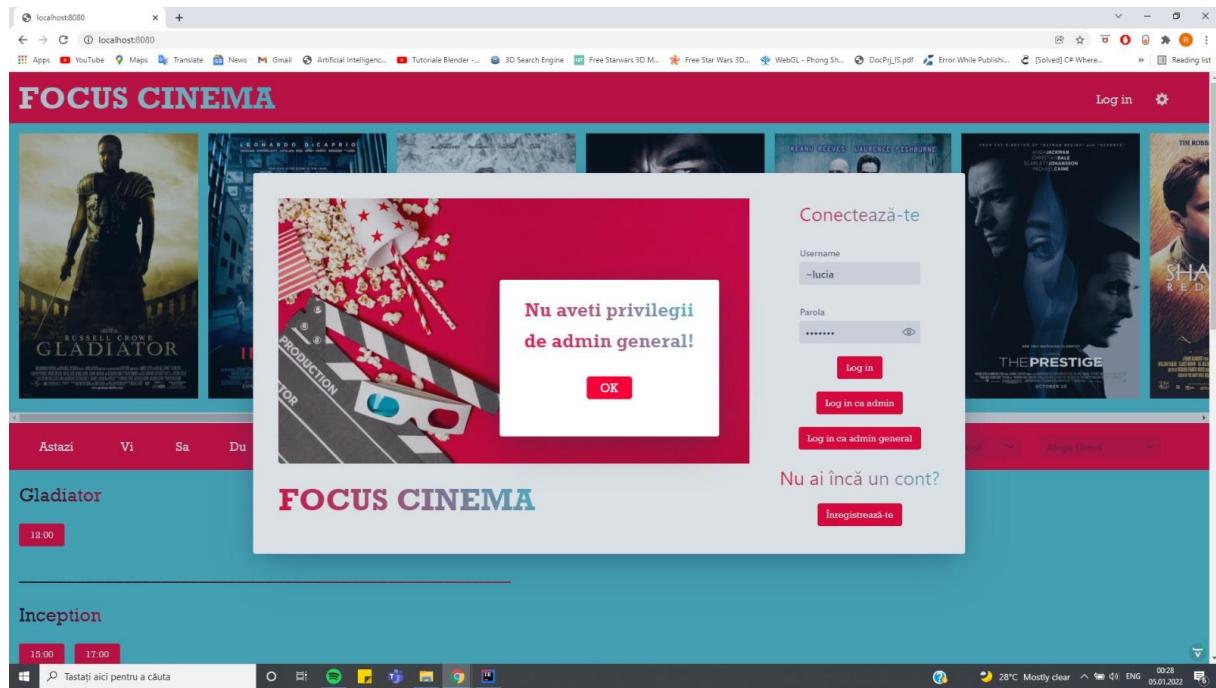
În cele ce urmează, se pot vizualiza câteva rezultate obținute în urma testării aplicației software.



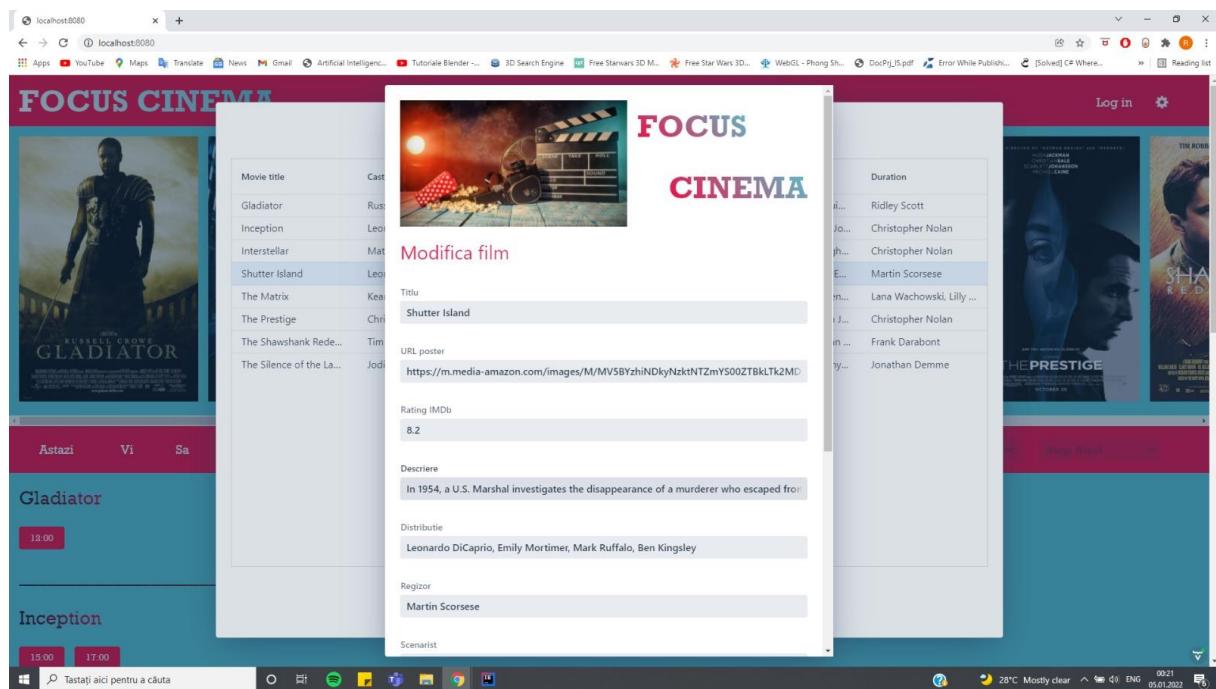
Figură 4 1 Testare log in cu username sau parolă invalide



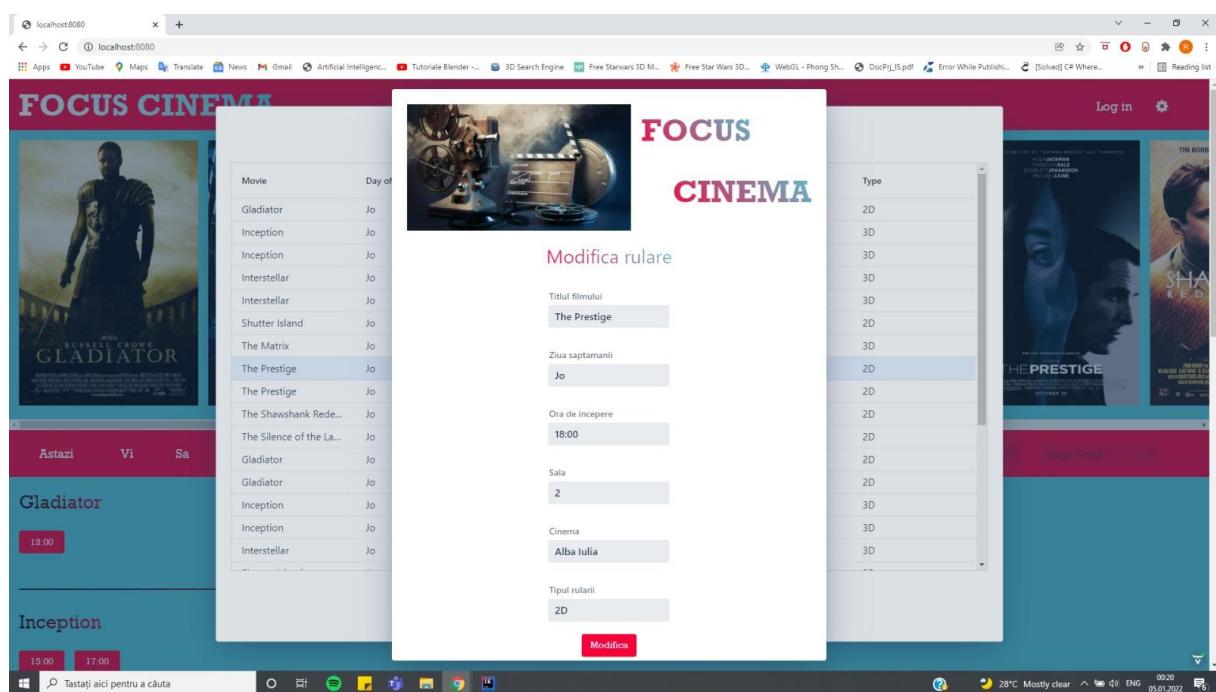
Figură 4 2 Testare conectare cu tip de utilizator necorespunzător (admin)



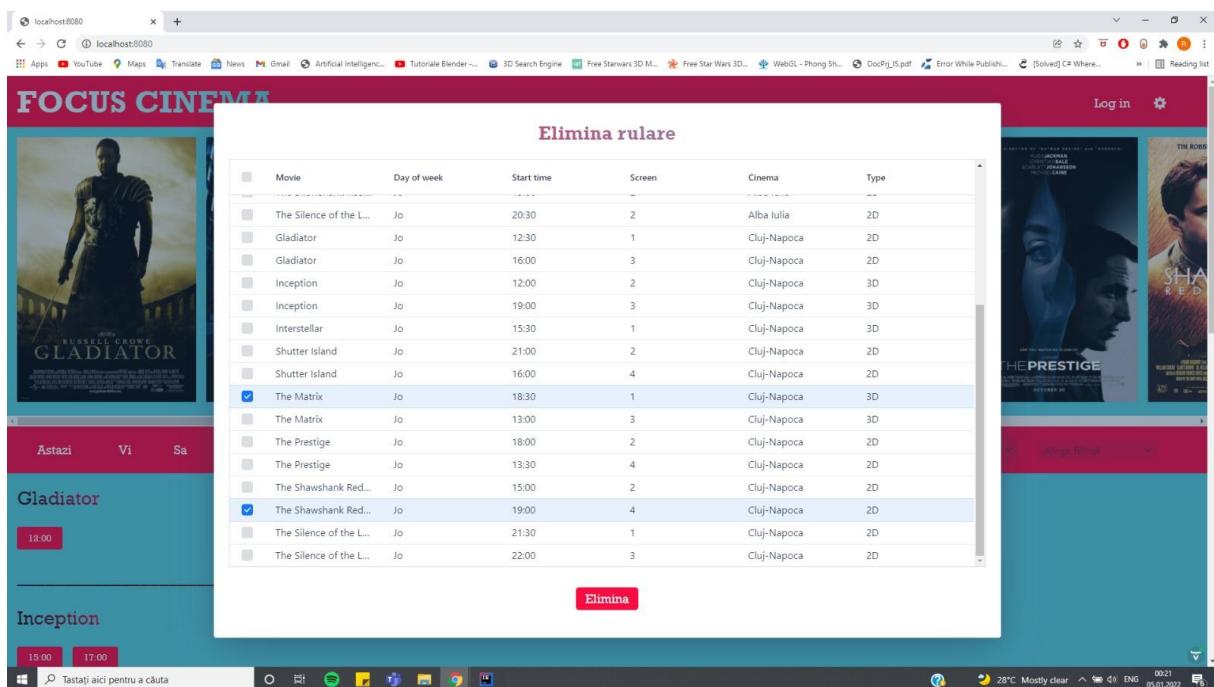
Figură 4 3 Testare conectare cu tip de utilizator necorespunzător (admin general)



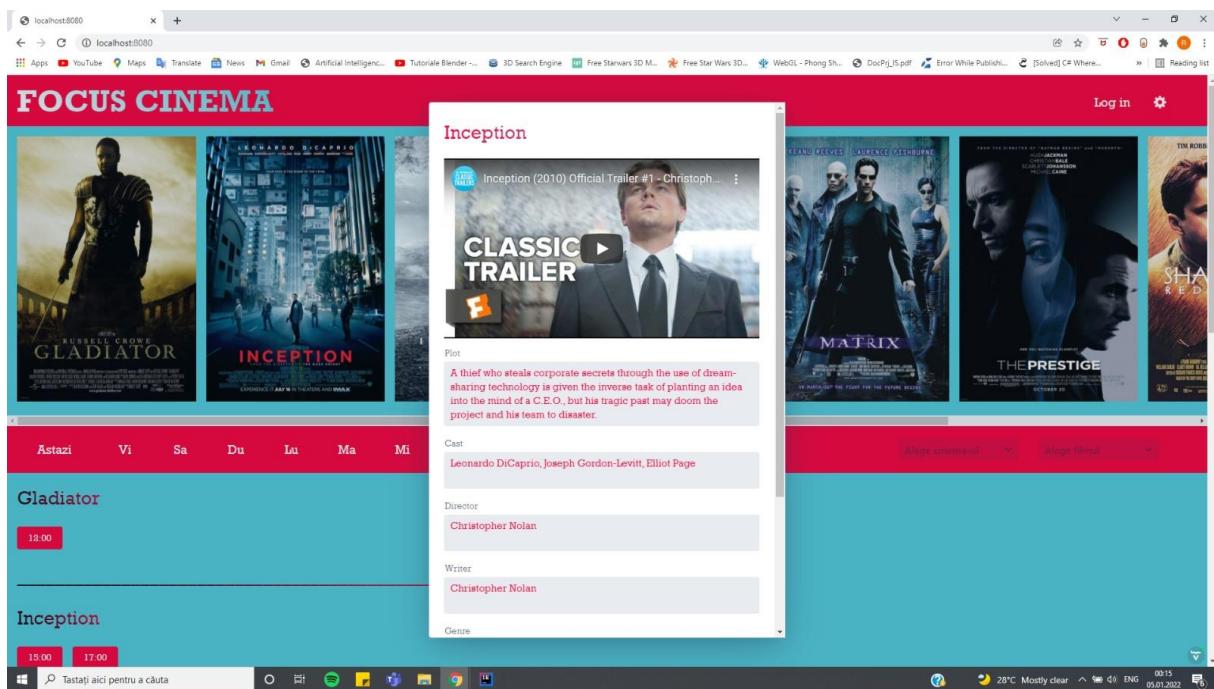
Figură 4 4 Testare modificare informații despre film



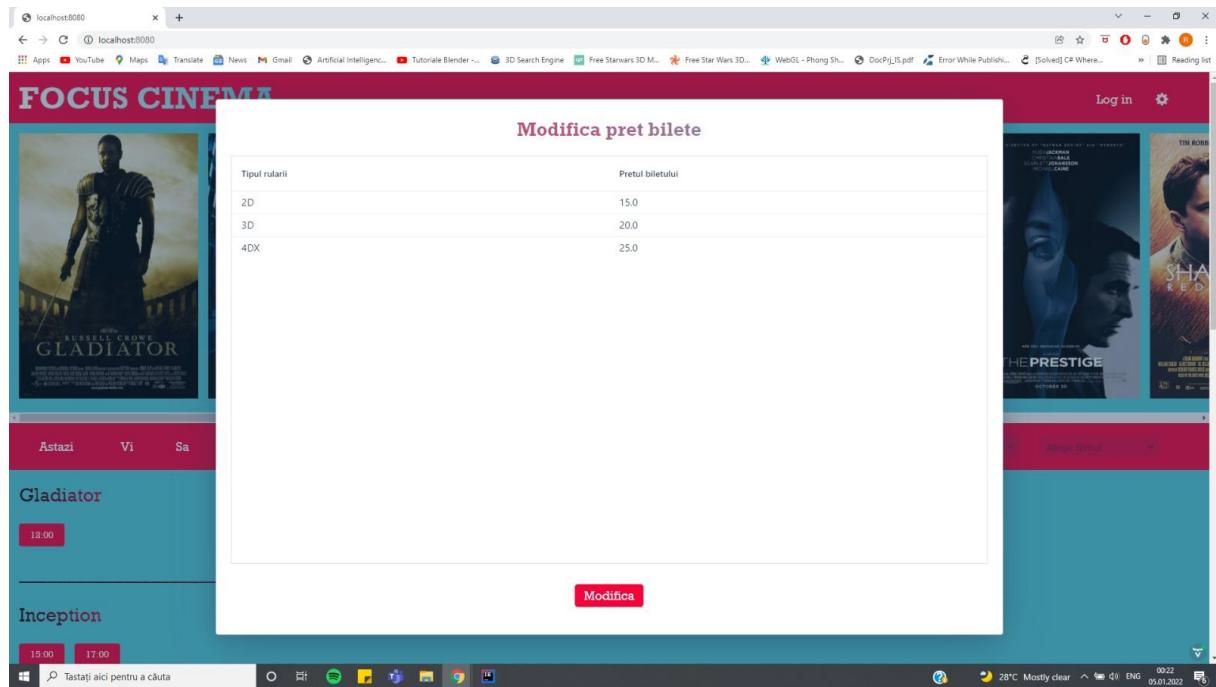
Figură 4 5 Testare modificare informații despre rularea unui anumit film



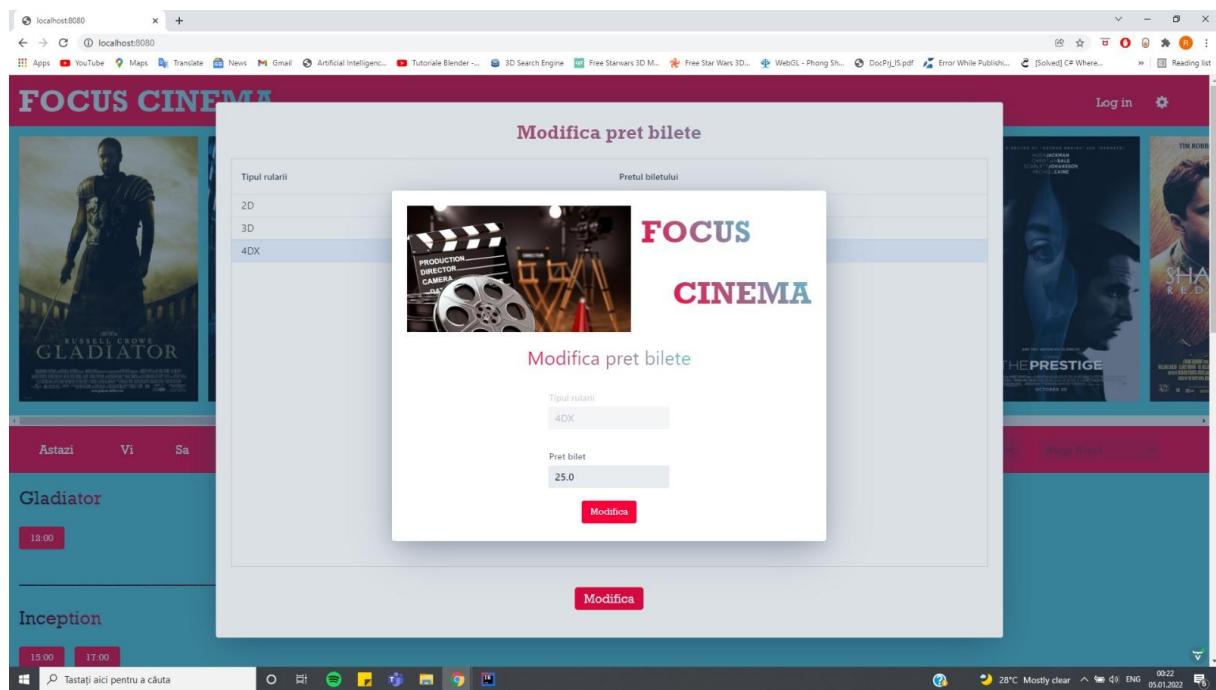
Figură 4 6 Testare eliminare rulare pentru un anumit film



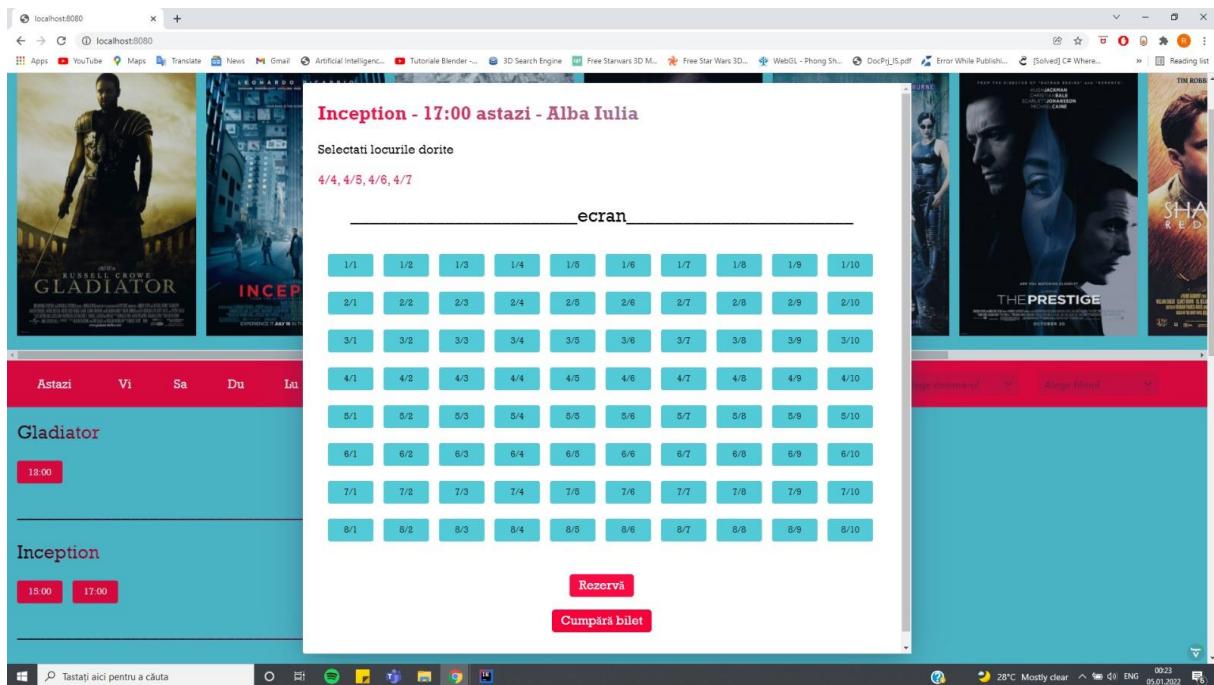
Figură 4 7 Testare vizualizare informații despre film



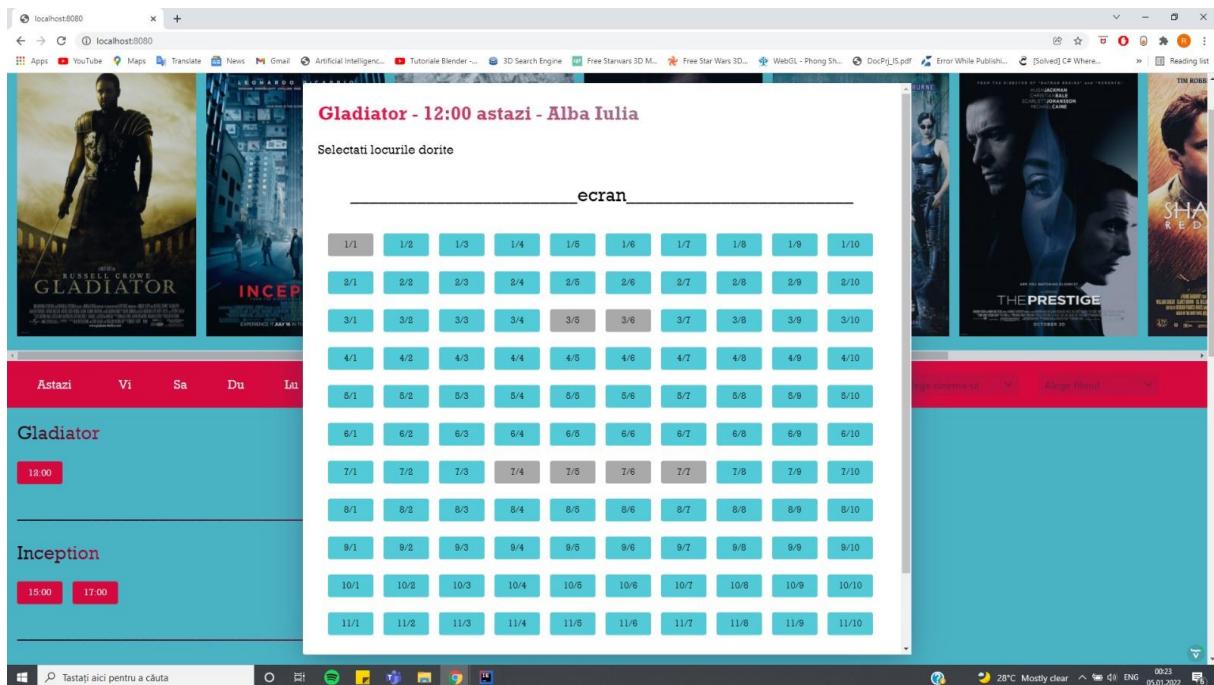
Figură 4 8 Testare selectare tip bilet pentru modificare preț



Figură 4 9 Testare modificare preț bilet



Figură 4 10 Testare selectare locuri pentru rezervare



Figură 4 11 Testare rezervare locuri

5. Manual de instalare și utilizare

i. Instalare

Întrucât sistemul implementat reprezintă o aplicație web, acesta nu necesită o instalare propriu zisă pe calculatoarele pe care urmează să fie lansată în execuție, însă fiind dezvoltată folosind mediul de dezvoltare IntelliJ, prezența acestuia pe calculatorul pe care se dorește pornirea sistemului reprezintă o cerință fundamentală pentru rularea aplicației. De asemenea, pentru lansarea în execuție a acesteia este necesară prezența unui browser web, care să permită și să asigure rularea efectivă, recomandat fiind Google Chrome întrucât acesta reprezintă browser-ul în care a fost efectuată testarea progresivă, pe parcursul implementării aplicației. Nu în ultimul rând, pentru rularea efectivă este necesară conexiunea la o bază de date, definită conform diagramei din *Figura 3.5. 1*, astfel încât funcționalitățile implementate să aibă o finalitate concretă, iar rezultatele să poată fi vizualizate corespunzător.

Astfel, pentru lansarea în execuție a aplicației se va rula, folosind IDE-ul IntelliJ, clasa Application definită în cadrul proiectului, clasă care utilizează framework-ul Spring Boot pentru a porni serverul atașat aplicației. După pornirea serverului, în browserul de web Google Chrome se va introduce adresa: <http://localhost:8080>, 8080 reprezentând identificatorul portului ales pentru realizarea conectării la Web.

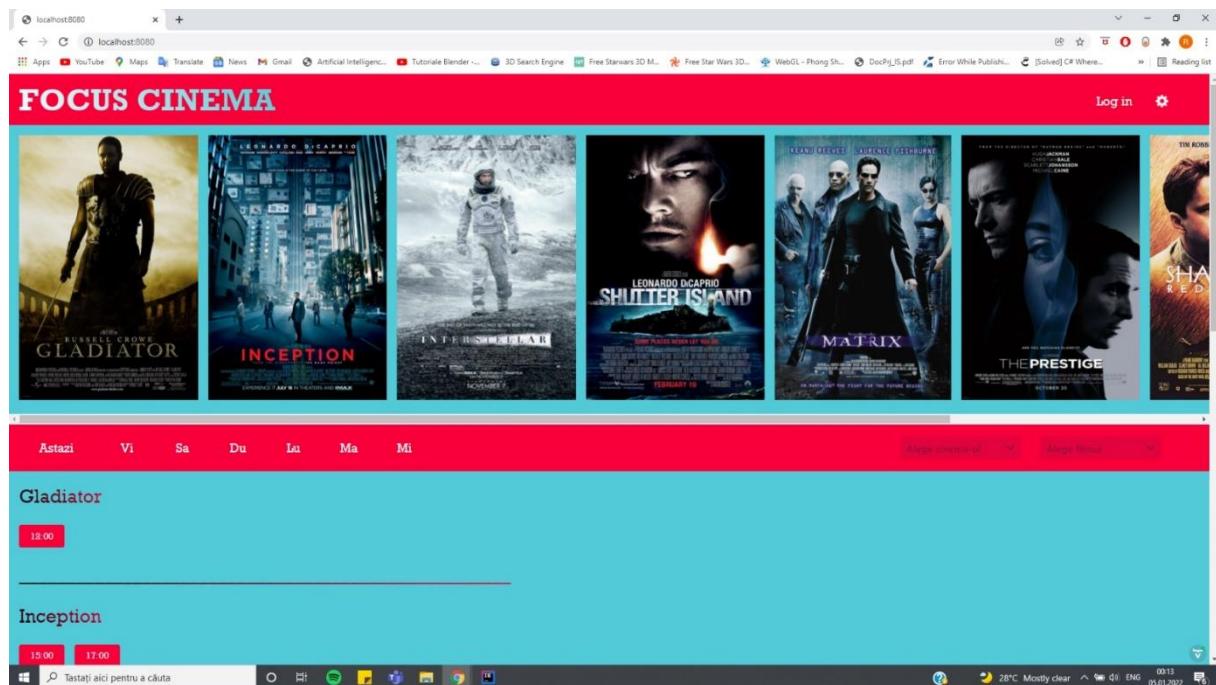
ii. Utilizare

Utilizarea aplicației este facilitată de interfață grafică intuitivă, care permite realizarea, cu ușurință, a funcționalităților, de către orice tip de utilizator, indiferent de sfera sa de activitate și de pregătirea sa profesională.

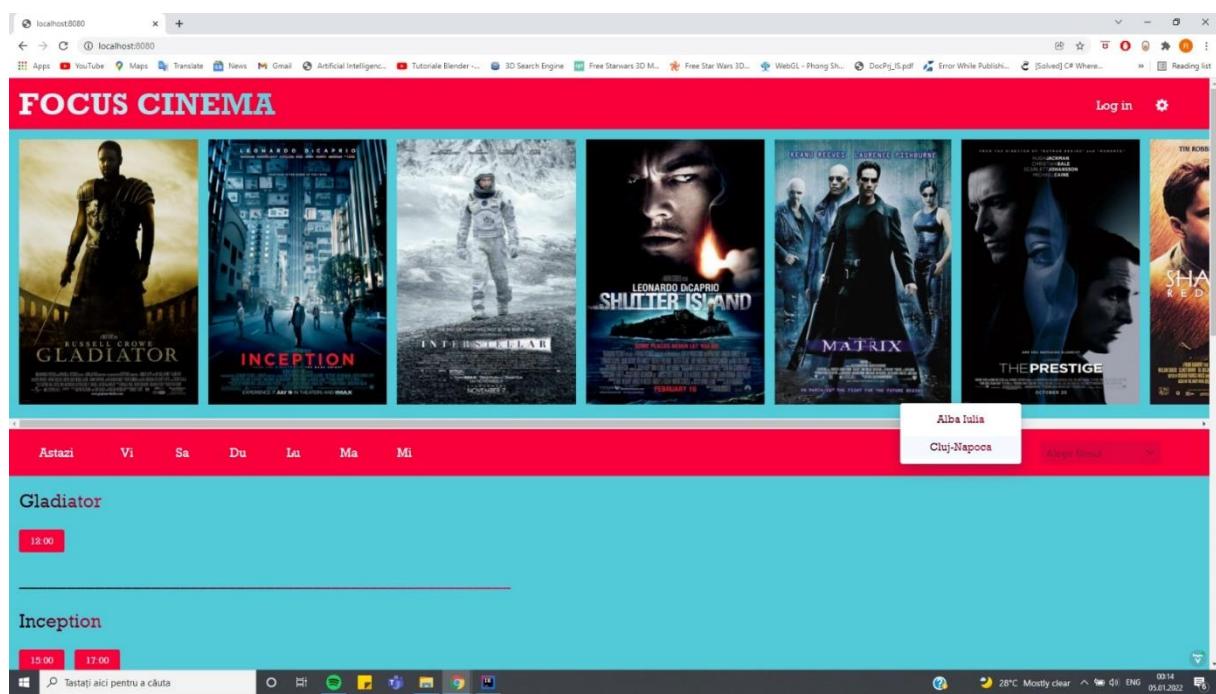
Aplicația dispune de 23 de pagini de interfață grafică, fiecare dintre ele indicând scopul propriu și funcționalitățile pe care le prezintă, fie prin descrieri sugestive, fie prin imagini, butoane sau câmpuri de text ce dispun de toate indiciile necesare utilizării corespunzătoare a sistemului implementat.

Fiind o aplicație de tipul Software as a Service (SaaS), aceasta prezintă o pagină de pornire, de unde se poate realiza conectarea (log in), în urma conectării utilizatorul având acces la funcționalitățile care îi revin, conform categoriei din care acesta face parte: client sau personal administrativ.

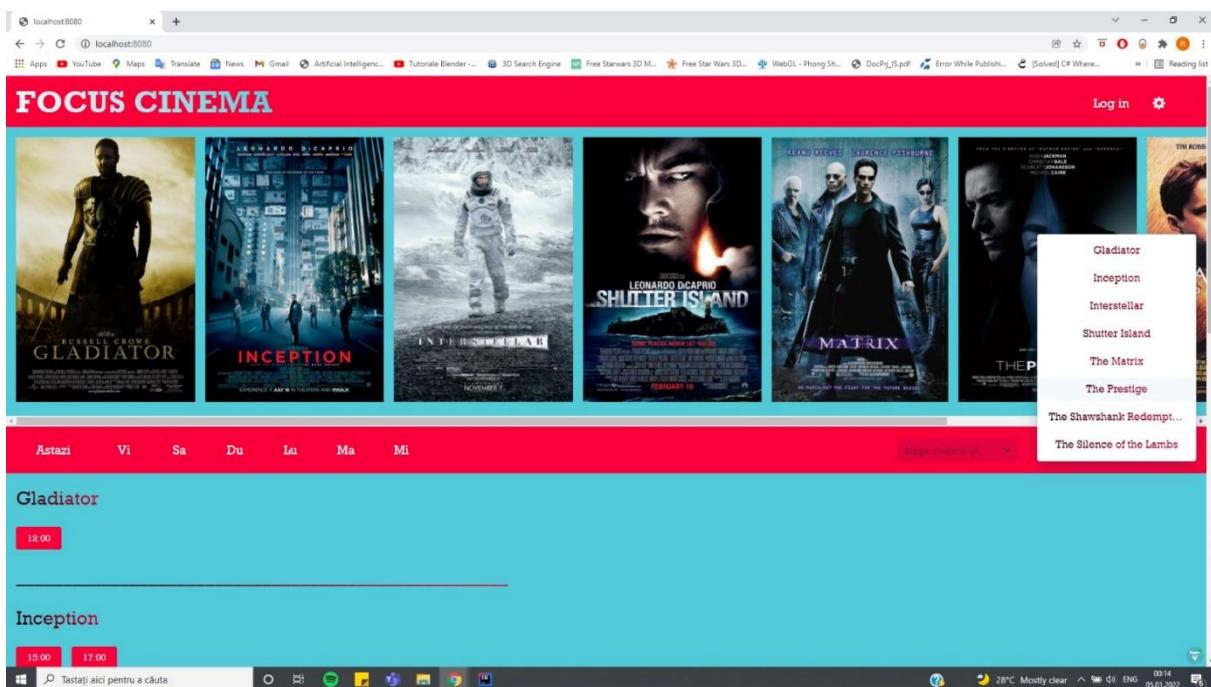
În cele ce urmează pot fi vizualizate principalele pagini de interfață grafică cu utilizatorul, implementate conform specificațiilor de proiectare și cadrului de cerințe funcționale și nefuncționale, detaliate în capitolele anterioare:



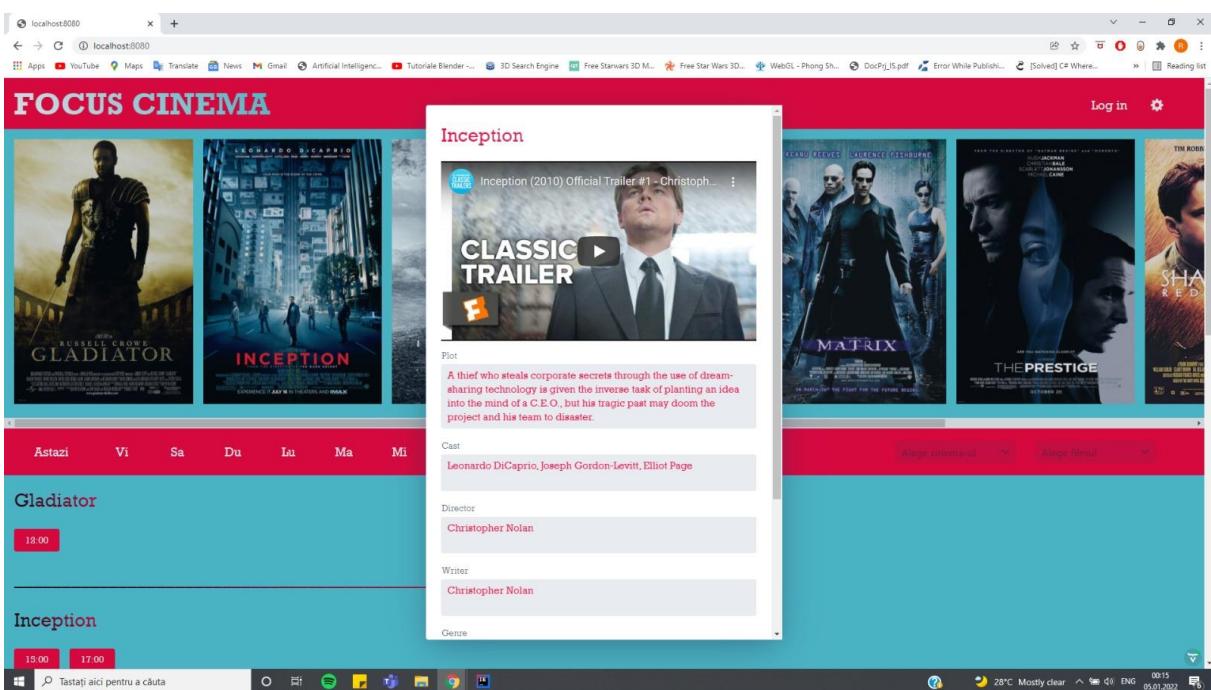
Figură 5 1 Pagina de pornire



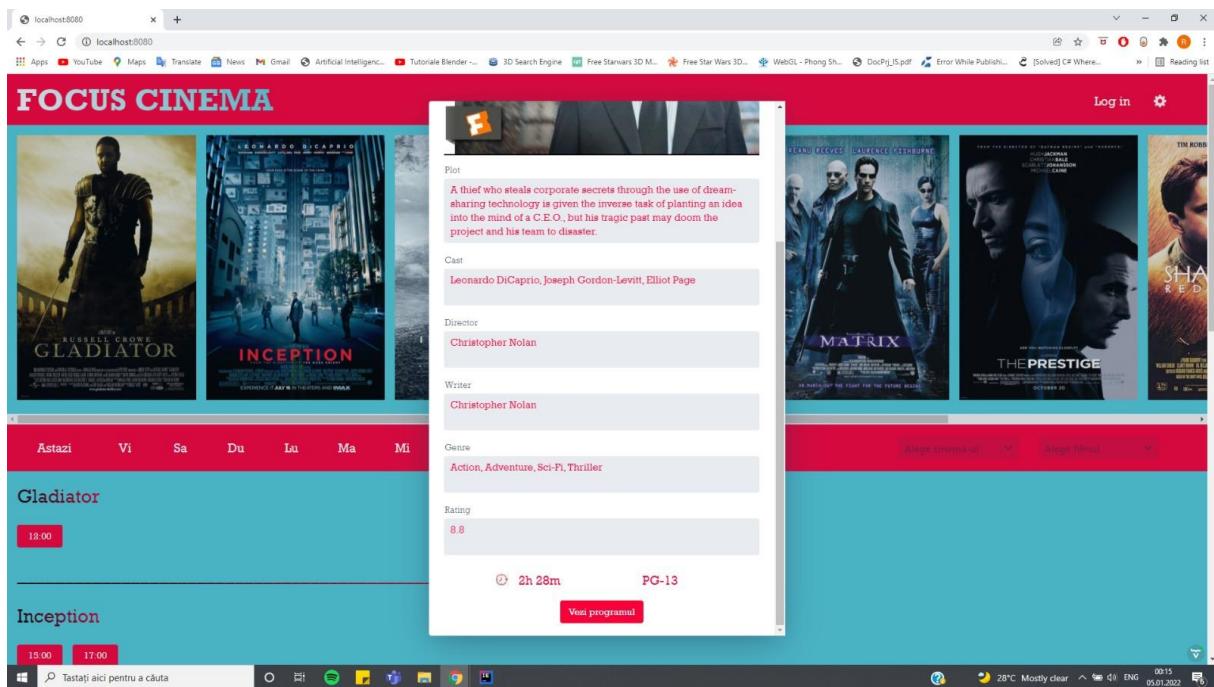
Figură 5 2 Pagina de pornire cu opțiunea de selectare a cinematografului



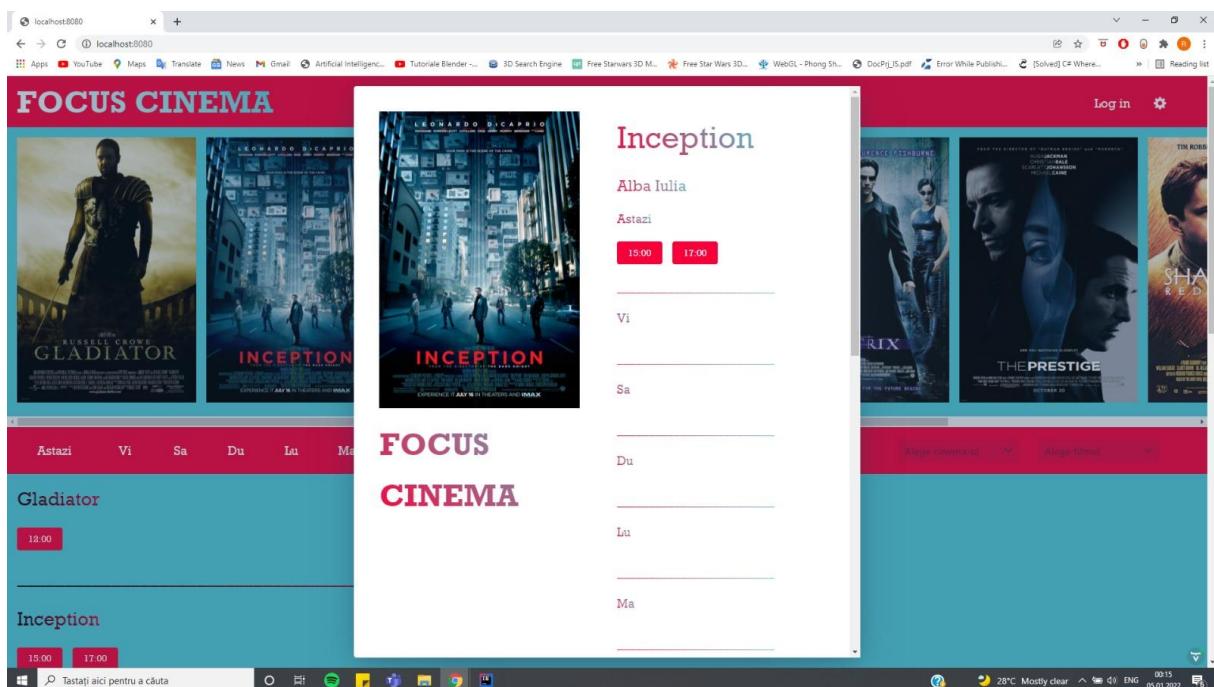
Figură 5 3 Pagina de pornire cu posibilitatea selectării filmului despre care se dorește afișarea de informații



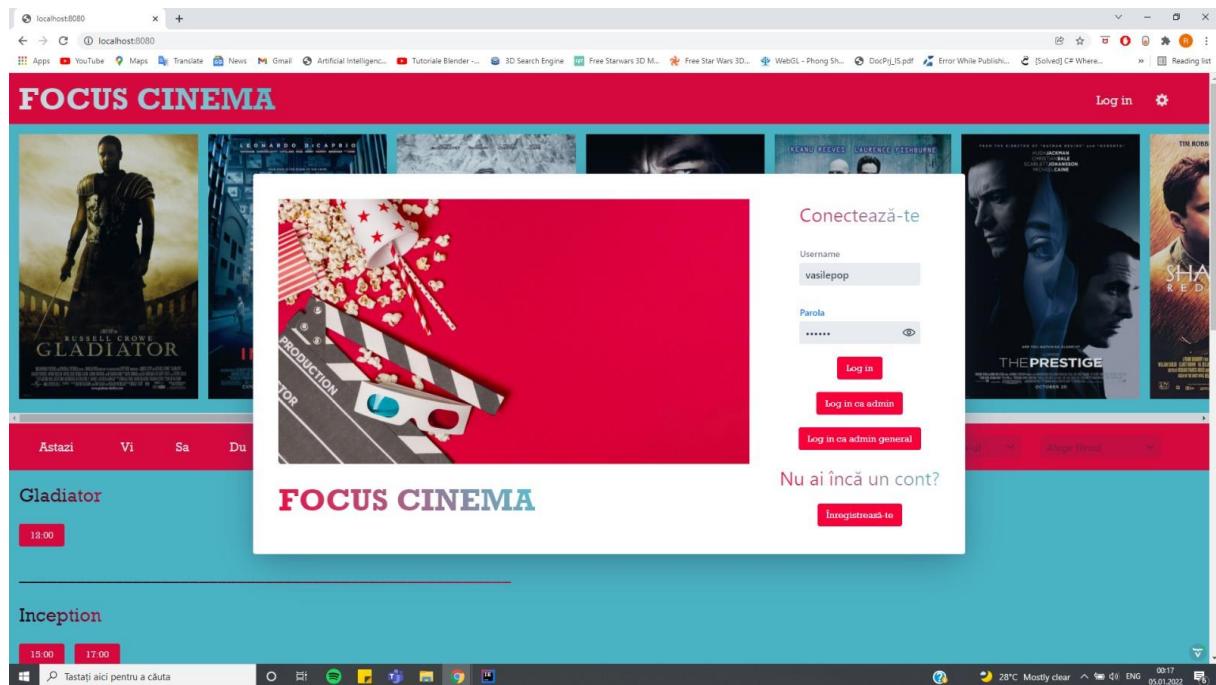
Figură 5 4 Pagina cu informații despre un anumit film



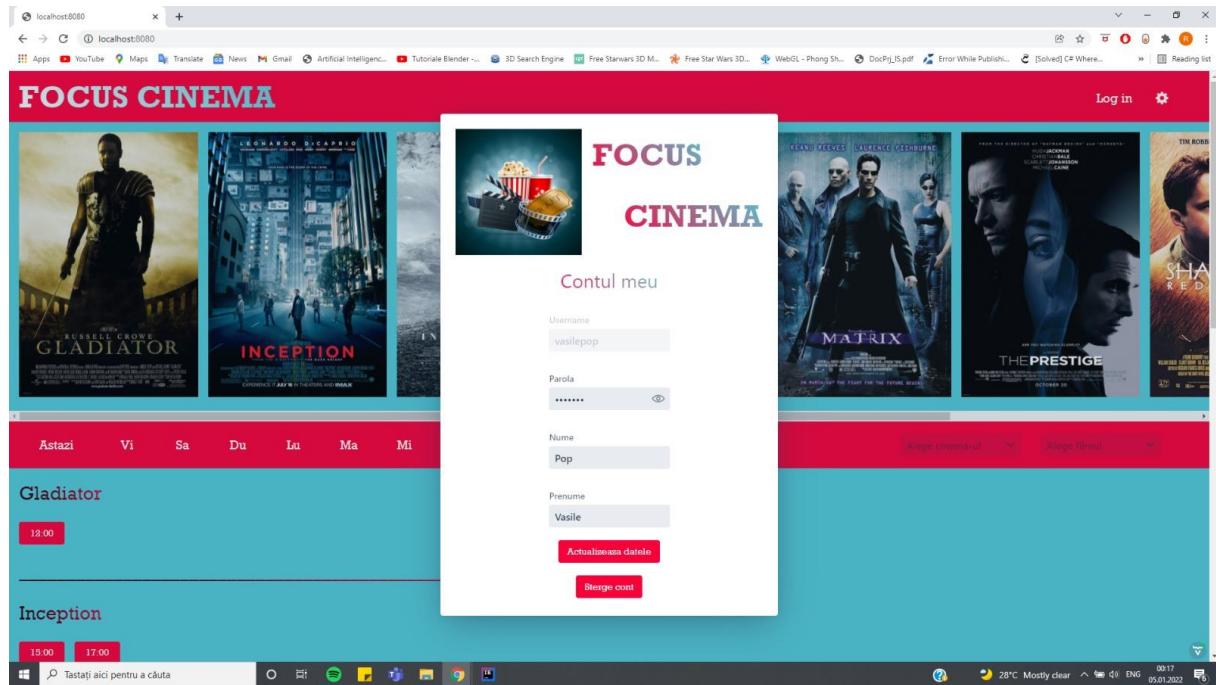
Figură 5 5 Pagina cu informații specifice filmului și posibilitatea vizualizării programului pentru filmul curent



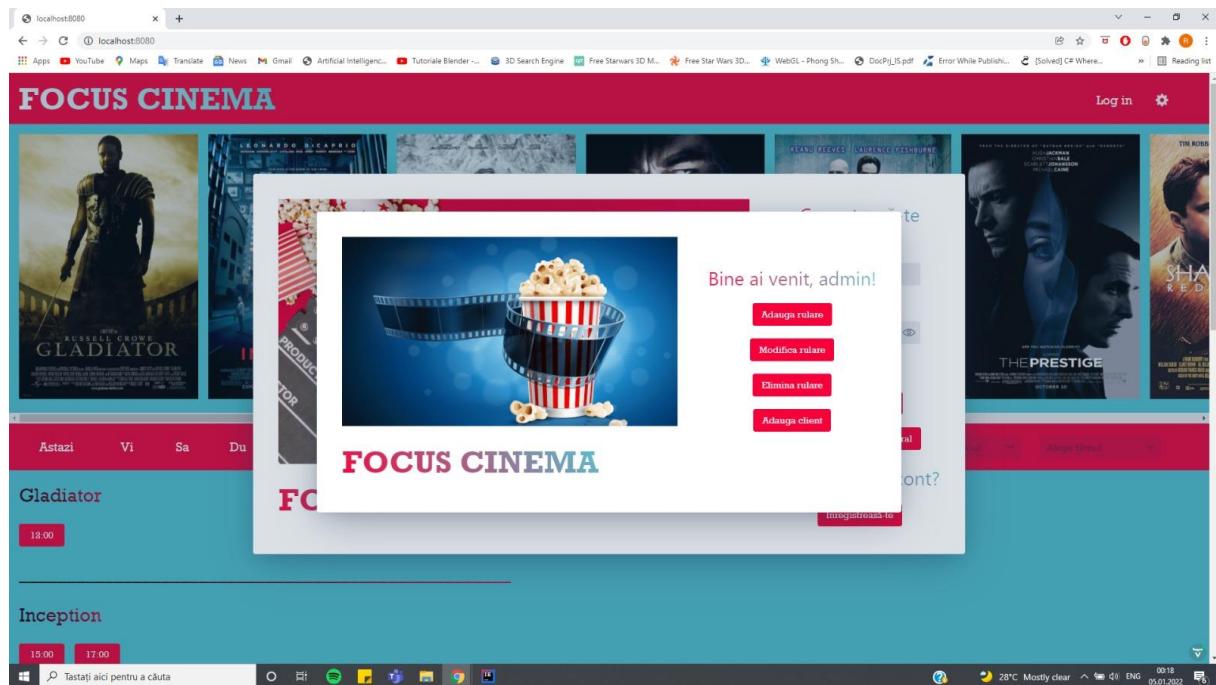
Figură 5 6 Pagina cu programul pentru un film pentru toată săptămâna, în toate cinematografele



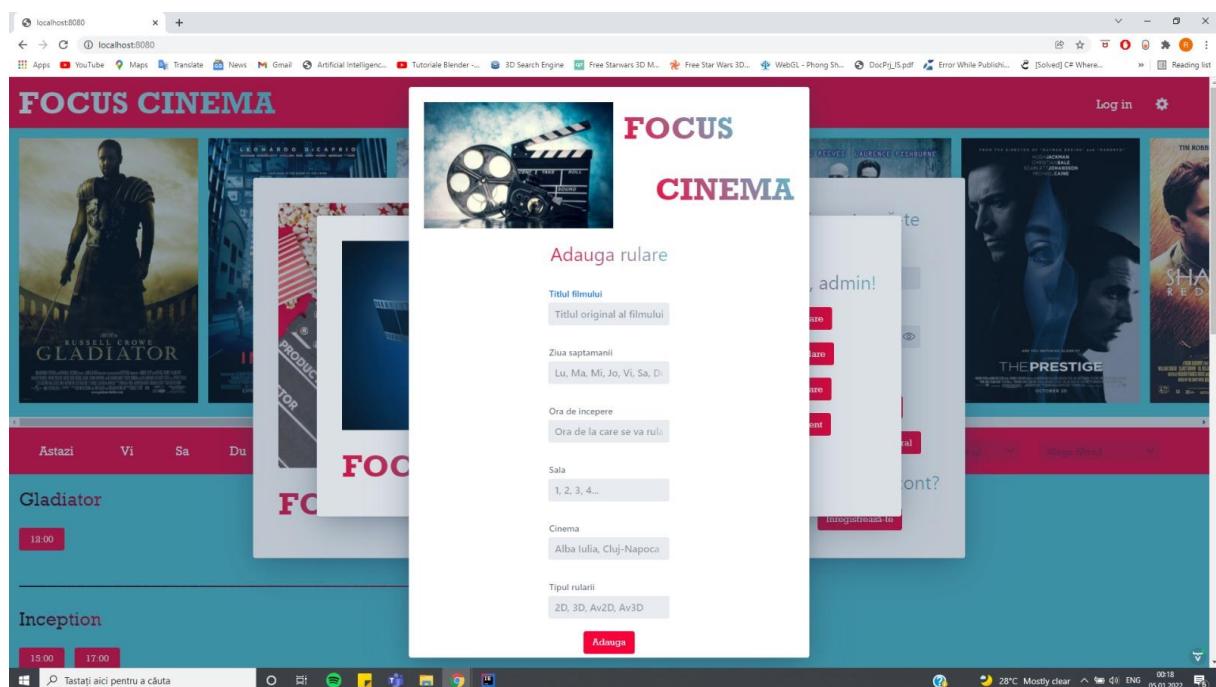
Figură 5 7 Pagina de log in, cu opțiunile tipurilor de utilizatori



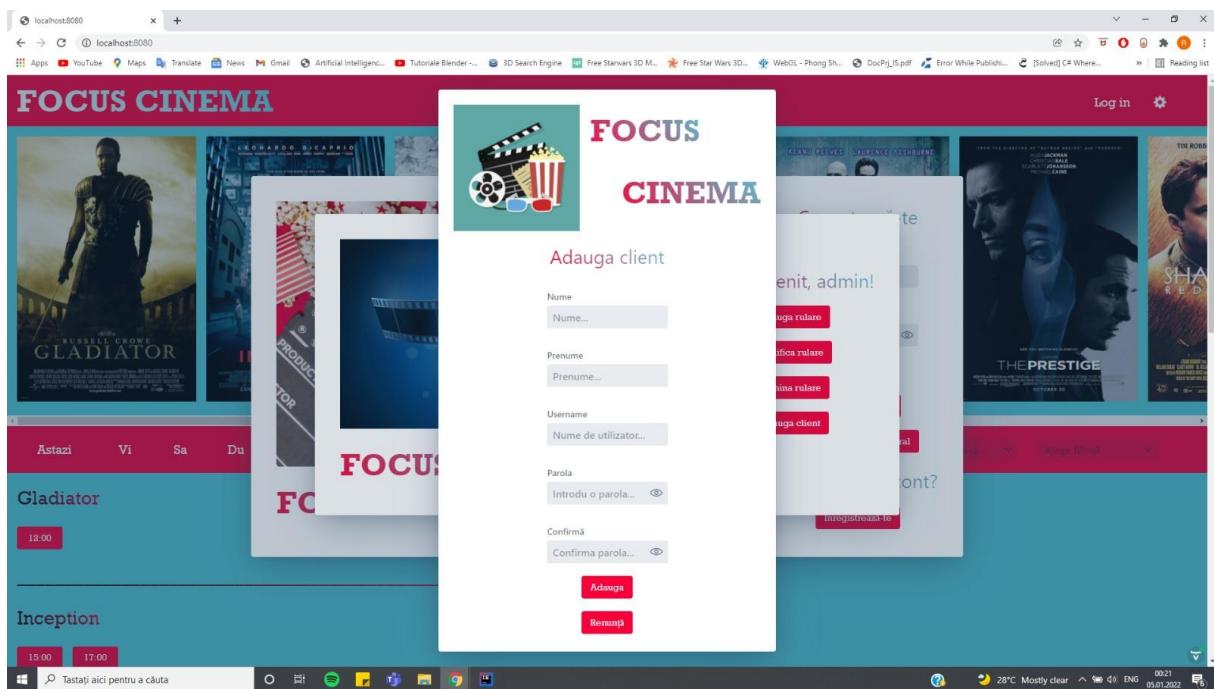
Figură 5 8 Pagina pentru editare date personale și stergere cont de utilizator



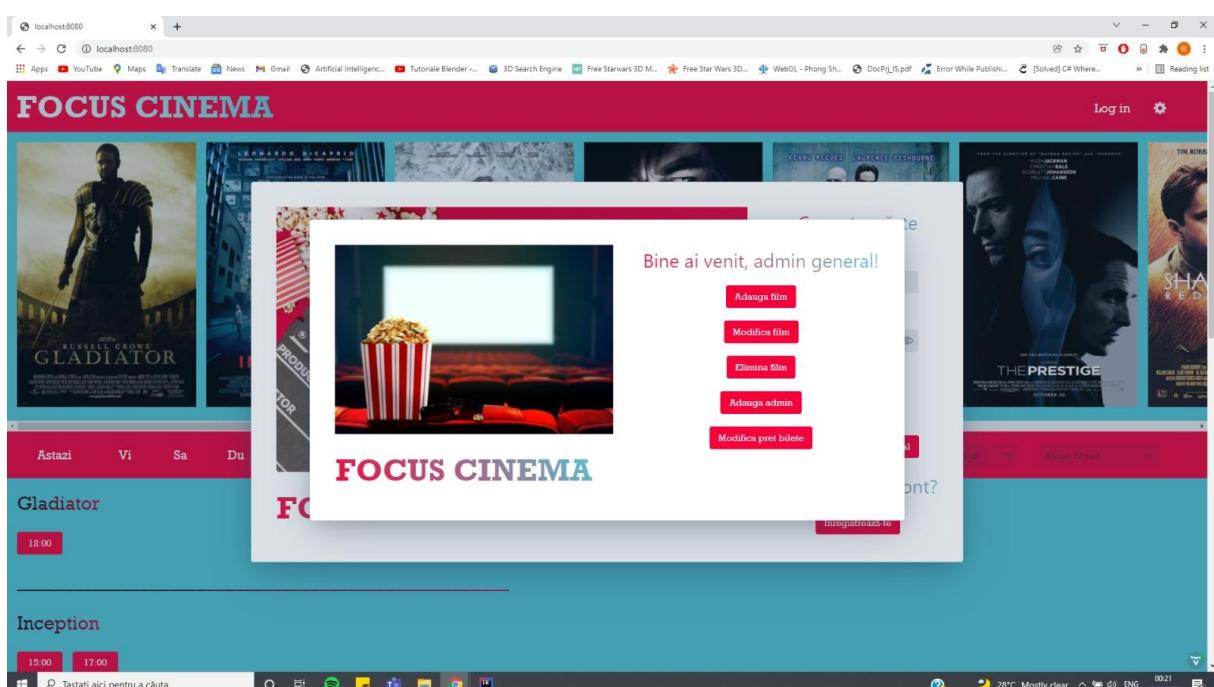
Figură 5 9 Pagina de bun venit pentru administrator



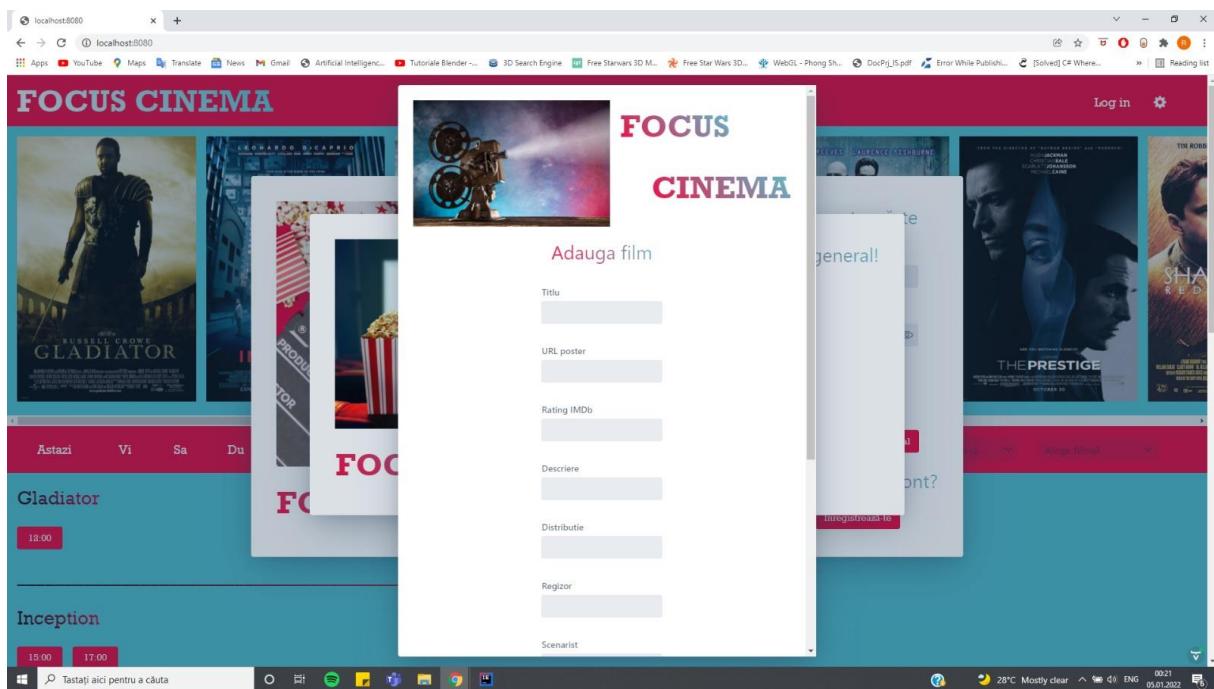
Figură 5 10 Pagina pentru adăugare a unei noi rulări pentru un film



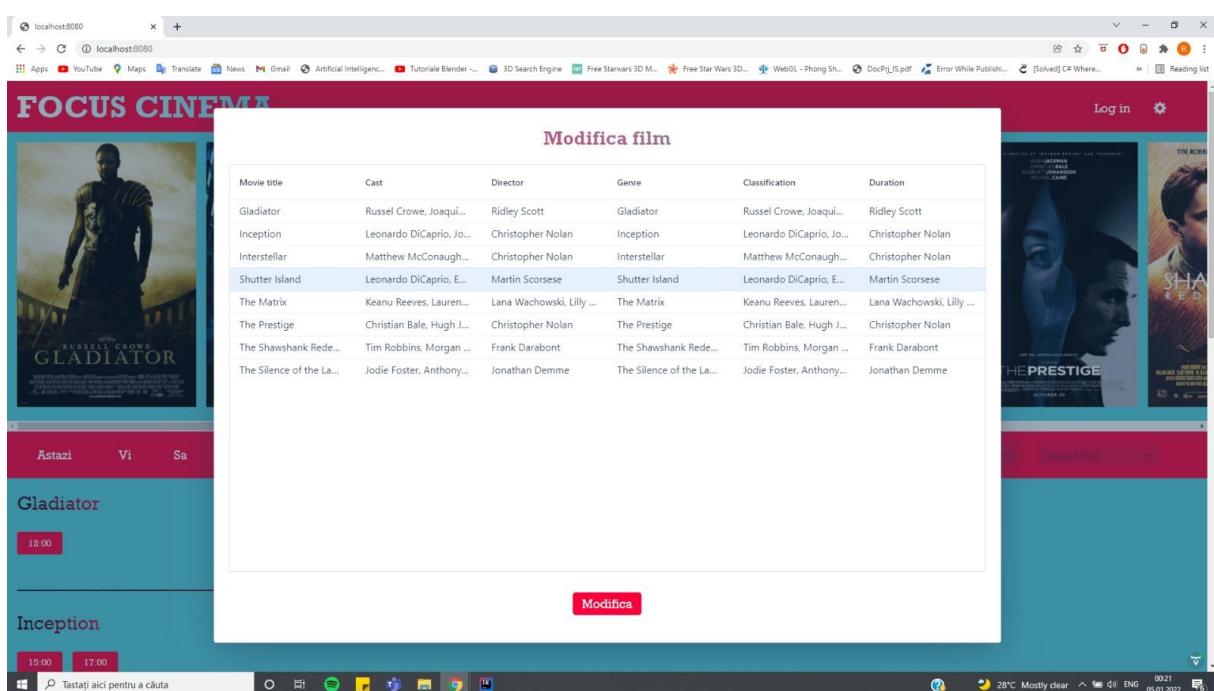
Figură 5 11 Pagina pentru adăugarea de către administrator a unui nou client



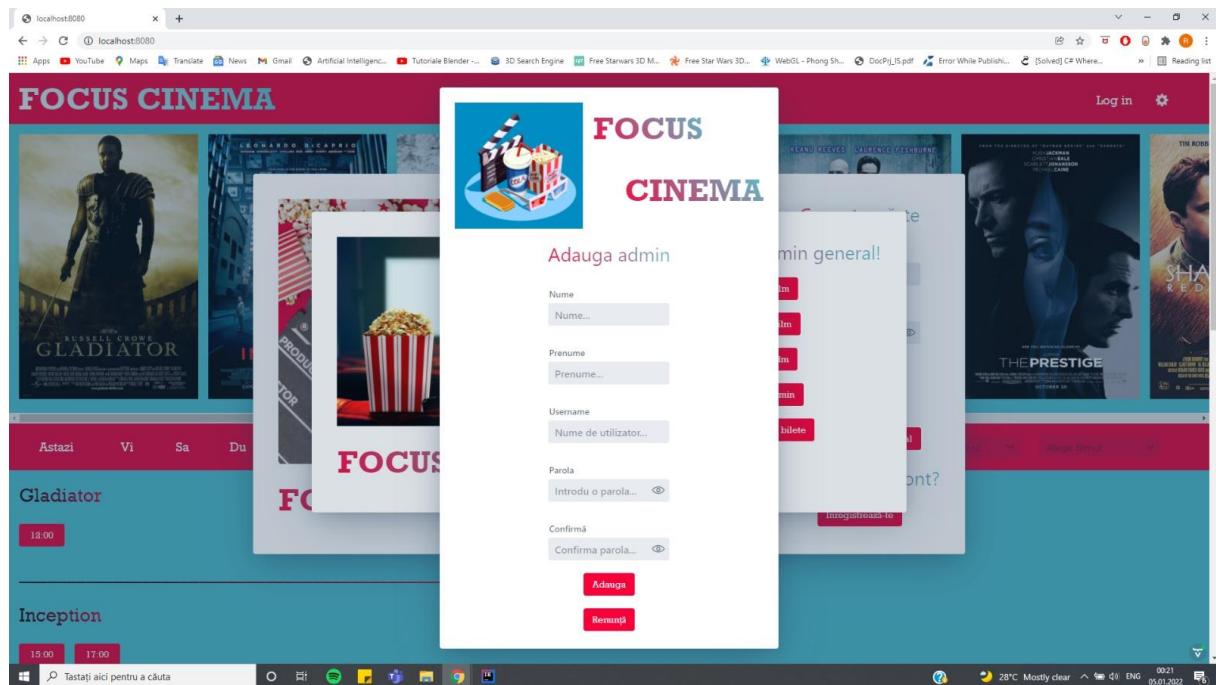
Figură 5 12 Pagina de pornire pentru administrator general



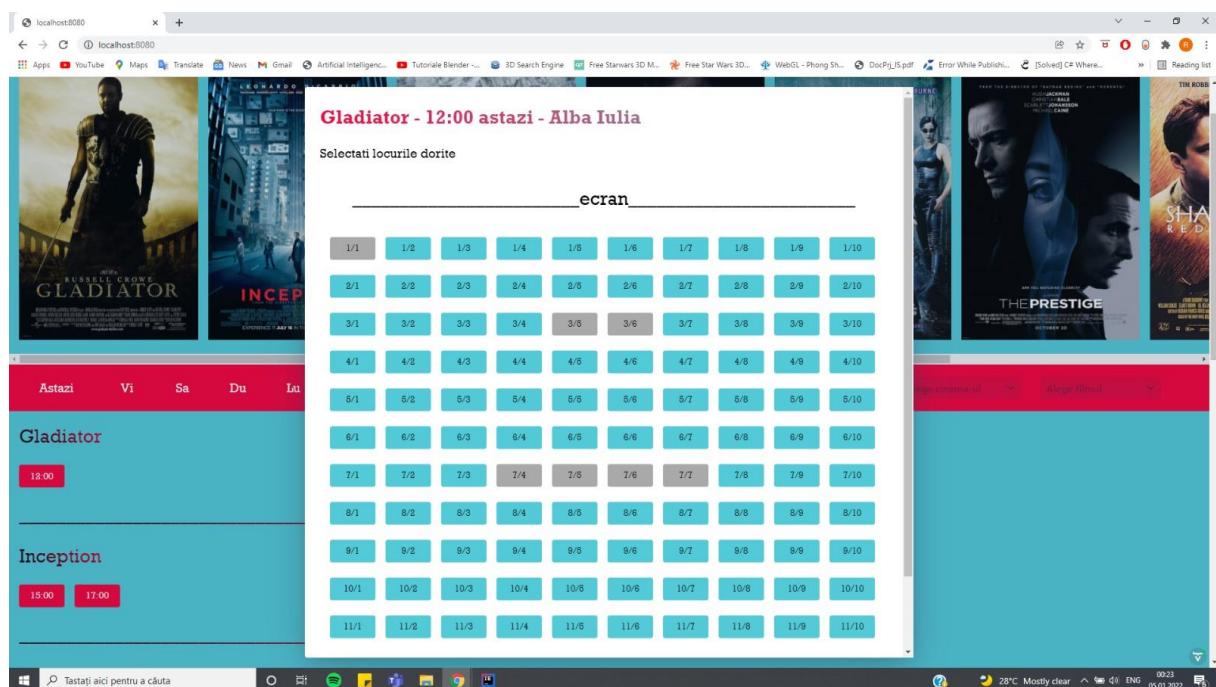
Figură 5 13 Pagina de adăugare a unui nou film în repertoriu



Figură 5 14 Pagina pentru selectarea filmului a cărui informații vor fi editate



Figură 5 15 Pagină pentru adăugarea unui nou film în repertoriu



Figură 5 16 Pagina cu locurile disponibile spre a fi rezervate, cu posibilitate de selecție

6. Concluzii și dezvoltări ulterioare

Concluzionând, se poate afirma că aplicația implementată vine spre a oferi flexibilitate și acces imediat la funcționalitățile de care un lanț de cinematografe dispune, atât în rândul clienților, cât și în cel al personalului administrativ, făcând posibilă desfășurarea activității cinematografelor în mediul virtual prin intermediul unui sistem software.

În urma implementării proiectului s-a ajuns la concluzia că limbajel de programare orientate pe obiecte Java, folosit împreună cu platforma Spring și framework-ul Vaadin, facilitează dezvoltarea de aplicații web de tipul Software as a Service (SaaS) ce au drept punct de plecare date stocate într-o bază de date.

Pentru dezvoltatorii proiectului, implementarea aplicației pentru sistemul informatic al lanțului de cinematografe Focus Cinema s-a dovedit utilă, având drept rezultat aprofundarea cunoștințelor din domeniul programării orientate pe obiecte și, în particular, a elementelor specifice limbajului de programare Java, dar și o bună însușire a etapelor de proiectare și implementare a aplicațiilor web prin folosirea framework-ului Spring Boot, împreună cu Vaadin, o tehnologie intuitivă, de viitor, în plină dezvoltare.

Printre dezvoltările ulterioare ale aplicației s-ar putea număra includerea unei pagini pur informative, dedicate meniului asociat fiecărui cinematograf, conținând produsele alimentare (popcorn, nachos, băuturi răcoritoare) disponibile, împreună cu prețul fiecărui, dar și realizarea unei secțiuni de tipul „În curând”, unde se vor putea vizualiza postere și scurte descrieri ale filmelor ce urmează să fie incluse în lista de rulări și difuzate.

7. Bibliografie

- [1] *Spring Boot*, Disponibilă online: <https://spring.io/projects/spring-boot>
- [2] *Spring Boot – Introduction*, Disponibilă online:
https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm
- [3] *Vaadin - The modern web application platform for Java*, Disponibilă online:
<https://vaadin.com/>
- [4] *Spring Initializr*, Disponibilă online: <https://start.spring.io/>
- [5] *Association vs. Dependency vs. Aggregation vs. Composition*, Disponibilă online:
<https://nirajrules.wordpress.com/2011/07/15/association-vs-dependency-vs-aggregation-vs-composition/>
- [6] *UML - Activity Diagrams*, Disponibilă online:
https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
- [7] *UML - Interaction Diagrams*, Disponibilă online:
https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm