



**UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA**

CinemaABC

Inginerie software

Grupa: 30236

Alina-Maria Gyorgy
Bianca-Maria Negoi-Stoica
Catalin Golban

FACULTATEA DE AUTOMATICA
SI CALCULATOARE

PROFESOR COORDONATOR: ADRIAN GABRIEL BURZO

2025-2026

Cuprins

1 Descrierea proiectului	2
2 Funcționalităile aplicației	2
2.1 Selectarea cinematografului	2
2.2 Vizualizarea filmelor disponibile	2
2.3 Căutarea filmelor după titlu	2
2.4 Filtrarea filmelor după gen	2
2.5 Ascunderea și restaurarea filmelor	2
2.6 Vizualizarea filmelor programate pentru ziua curentă	3
2.7 Achiziționarea biletelor	3
2.8 Aplicarea reducerilor pentru studenți și copii	3
2.9 Vizualizarea locației cinematografului	3
3 Design pattern-uri utilizate	4
3.1 Factory Method (generarea tipurilor de bilete)	4
3.2 Strategy (sortare și filtrare filme)	4
3.3 Command Pattern (acțiuni și operații Undo/Redo)	4
4 Diagrame utilizate	5
4.1 Diagrame realizate de Alina	5
4.1.1 Diagrama de secvență – rezervarea biletului	5
4.1.2 Diagrama de comunicare – rezervarea biletului	5
4.1.3 Diagrama de stare – cumpărarea biletelor	5
4.2 Diagrame realizate de Bianca	6
4.2.1 Diagrama de secvență – selectarea genului filmelor	6
4.2.2 Diagrama de activitate – generarea codului QR	6
4.2.3 Diagrama de stare – plata fictivă	7
4.3 Diagrame realizate de Cătălin	7
4.3.1 Diagrama de stare – acțiuni rezervare/anulare	7
4.3.2 Diagrama de secvență – selectare și afișare pe hartă	8
4.3.3 Diagrama de activitate – interacțiunea utilizatorului	8

1 Descrierea proiectului

Cinema ABC este o aplicație web destinată gestionării și achiziționării biletelor de cinema. Aplicația permite utilizatorilor să selecteze cinematograful dorit și să vizualizeze lista de filme disponibile, fiecare film fiind însotit de informații relevante precum descrierea, genul, prețul și numărul de bilete disponibile.

Pentru a îmbunătăți experiența utilizatorului, aplicația oferă funcționalități de căutare, filtrare și sortare a filmelor, facilitând identificarea rapidă a conținutului dorit. Procesul de cumpărare a biletelor este simbolizat printr-un formular de plată, în care datele introduse sunt validate înainte de finalizarea comenzi.

Aplicația este construită pe o arhitectură de tip client-server, utilizând **React** pentru dezvoltarea interfeței grafice și **FastAPI** împreună cu o bază de date **SQLite** pentru gestionarea logicii aplicației și a datelor persistente. Structura proiectului permite extinderea ulterioară a funcționalităților și integrarea unor servicii suplimentare.

2 Funcționalitățile aplicației

2.1 Selectarea cinematografului

Aplicația permite utilizatorului să selecteze cinematograful dorit dintre mai multe locații disponibile. În funcție de cinematograful ales, sunt afișate doar filmele care rulează în locația respectivă. Selectia este reținută local, astfel încât utilizatorul să nu fie nevoie să aleagă din nou cinematograful la fiecare accesare a aplicației.

2.2 Vizualizarea filmelor disponibile

Pentru cinematograful selectat, aplicația afișează lista filmelor disponibile, fiecare film fiind prezentat sub forma unui card informativ. Acesta conține titlul filmului, genul, locația, prețul biletului și numărul de bilete disponibile. Filmele sunt afișate într-un mod clar și organizat, pentru o experiență de utilizare intuitivă.

2.3 Căutarea filmelor după titlu

Utilizatorul poate căuta filme după titlu prin intermediul unui câmp de căutare. Lista filmelor se actualizează dinamic pe măsură ce utilizatorul introduce text, fiind afișate doar filmele care conțin termenul căutat în titlu.

2.4 Filtrarea filmelor după gen

Aplicația oferă posibilitatea filtrării filmelor în funcție de genul acestora. Utilizatorul poate selecta un gen dintr-o listă, iar aplicația va afișa doar filmele corespunzătoare. Această funcționalitate ajută la o navigare mai rapidă și la personalizarea listei de filme afișate.

2.5 Ascunderea și restaurarea filmelor

Utilizatorul are posibilitatea de a ascunde temporar anumite filme din listă. Filmele ascunse nu mai sunt afișate, însă pot fi restaurate ulterior prin intermediul funcțiilor de *undo*,

redo sau prin restaurarea completă a tuturor filmelor ascunse. Această funcționalitate permite personalizarea afișării conținutului.

2.6 Vizualizarea filmelor programate pentru ziua curentă

Aplicația afișează separat filmele programate pentru ziua curentă, oferind utilizatorului o vedere rapidă asupra evenimentelor disponibile în aceeași zi. Această secțiune este actualizată automat în funcție de data curentă.

2.7 Achiziționarea biletelor

Utilizatorul poate achiziționa bilete pentru un film selectat prin intermediul unei pagini dedicate. Procesul de cumpărare presupune introducerea datelor personale și completarea unui formular de plată fictiv. Aplicația validează datele introduse pentru a simula un proces real de achiziție.

2.8 Aplicarea reducerilor pentru studenți și copii

Aplicația oferă posibilitatea aplicării unor reduceri speciale pentru anumite categorii de utilizatori, precum studenții și copiii. În cadrul procesului de achiziționare a biletelor, utilizatorul poate selecta tipul de reducere dorit, iar prețul final al biletului este recalculat automat în funcție de politica de discount aplicată.

2.9 Vizualizarea locației cinematografului

Pentru fiecare cinematograf, aplicația permite vizualizarea locației acestuia pe harta orașului Cluj-Napoca. Utilizatorul poate accesa o hartă interactivă care indică poziția exactă a cinematografului, facilitând orientarea și planificarea deplasării.

Această funcționalitate îmbunătățește experiența utilizatorului și oferă un suport vizual util, în special pentru utilizatorii care nu sunt familiarizați cu zona.

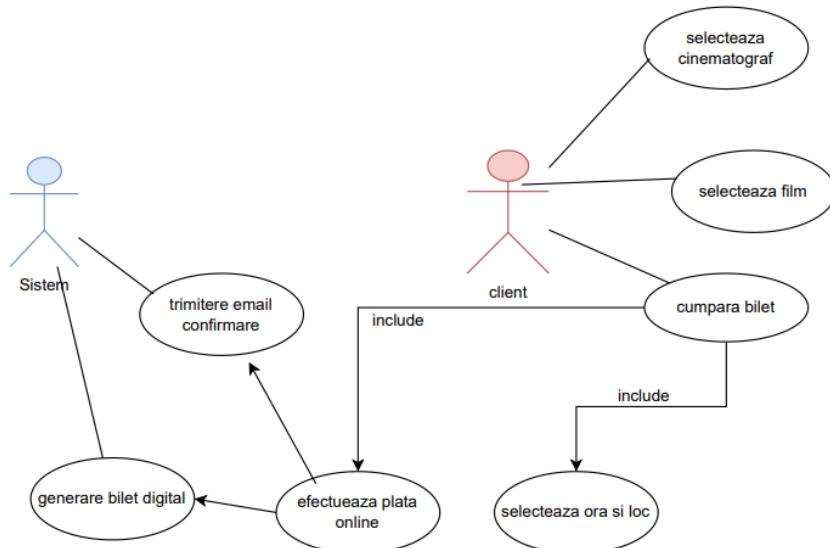


Figura 1: Diagrama funcționalităților aplicației Cinema ABC

3 Design pattern-uri utilizate

Pentru a obține o aplicație modulară, ușor de extins și de întreținut, în implementare au fost folosite mai multe *design pattern-uri*. Acestea separă clar responsabilitățile, reduc dependențele directe între componente și permit adăugarea de funcționalități noi fără modificări majore în codul existent.

3.1 Factory Method (generarea tipurilor de bilete)

În cadrul aplicației există mai multe tipuri de bilete (ex.: *Adult*, *Student*, *Copil*), fiecare având reguli și prețuri diferite (ex.: reducere pentru student sau copil). Pentru a evita instanțierea directă a acestor tipuri în mai multe locuri din cod și pentru a facilita extinderea (de exemplu, adăugarea unui nou tip de bilet *Senior*), a fost utilizat **Factory Method**.

Astfel, aplicația cere un tip de bilet, iar fabrica returnează obiectul corespunzător, împreună cu logica sa de calcul a prețului. Avantajul principal este că logica de creare este centralizată și poate fi extinsă fără a modifica restul aplicației.

3.2 Strategy (sortare și filtrare filme)

Filmele afișate în pagina principală trebuie:

- să poată fi **filtrate** în funcție de gen (*drama*, *acțiune*, *SF* etc.);
- să fie **sortate alfabetic** pentru o afișare coerentă și ușor de utilizat.

Pentru a implementa aceste reguli într-un mod flexibil, s-a folosit **Strategy Pattern**. Fiecare strategie reprezintă o regulă separată (de exemplu, *SortByTitleAscStrategy*, *GenreFilterStrategy*, *SearchFilterStrategy*), iar aplicația poate combina sau schimba strategiile fără să rescrie logica principală de afișare.

Acest pattern permite adăugarea rapidă a unor noi metode de filtrare/sortare (de exemplu, sortare după preț sau filtrare după numărul de bilete disponibile) fără a afecta codul existent.

3.3 Command Pattern (acțiuni și operații Undo/Redo)

Aplicația include acțiuni care pot fi **anulate** sau **refăcute** (*Undo/Redo*), de exemplu:

- adăugare film / ștergere film;
- rezervare bilet / anulare bilet;
- ascunderea unui film din listă și readucerea lui în listă.

Pentru a modela aceste operații într-un mod controlat, s-a utilizat **Command Pattern**. Fiecare acțiune este încapsulată într-o comandă care conține:

- metoda de execuție (*execute*);
- metoda de anulare (*undo*);

Avantajul acestui pattern este că aplicația poate păstra o istorie a acțiunilor și poate implementa ușor funcționalități precum *Undo/Redo*, fără să amestecă logica acțiunilor cu logica interfeței.

4 Diagrame utilizate

Pentru analiza și modelarea aplicației **Cinema ABC**, au fost utilizate mai multe tipuri de diagrame UML, fiecare având rolul de a descrie clar funcționarea sistemului, fluxurile de date și interacțiunea dintre utilizator și aplicație. Diagramele au fost realizate pentru a evidenția atât logica internă a aplicației, cât și experiența utilizatorului final.

4.1 Diagrame realizate de Alina

4.1.1 Diagrama de secvență – rezervarea biletului

Această diagramă descrie pașii parcursi în procesul de rezervare a unui bilet, de la selecțarea filmului până la confirmarea rezervării.

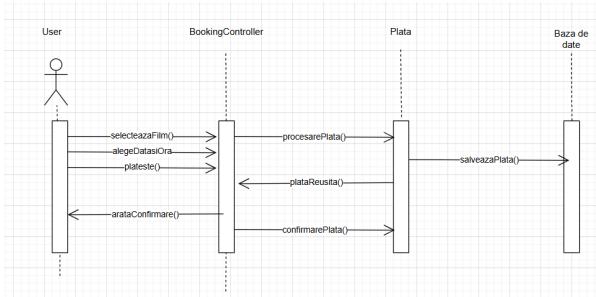


Figura 2: Diagrama de secvență pentru rezervarea biletului

4.1.2 Diagrama de comunicare – rezervarea biletului

Diagrama de comunicare prezintă schimbul de mesaje dintre componentele aplicației implicate în rezervarea unui bilet, punând accent pe relațiile dintre obiecte și ordinea comunicării.

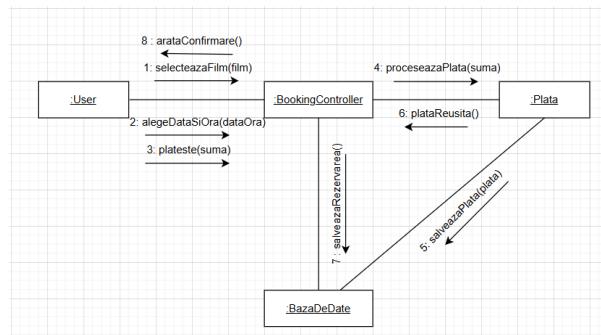


Figura 3: Diagrama de comunicare pentru rezervarea biletului

4.1.3 Diagrama de stare – cumpărarea biletelor

Această diagramă ilustrează stările prin care trece un bilet în procesul de cumpărare, de la inițiere până la finalizarea plății și confirmarea achiziției.

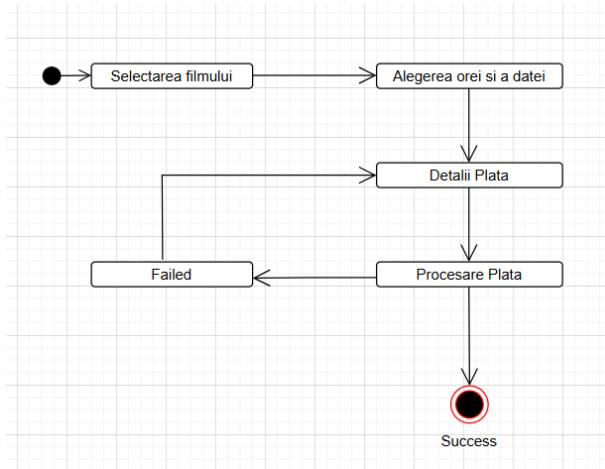


Figura 4: Diagrama de stare pentru cumpărarea biletelor

4.2 Diagrame realizate de Bianca

4.2.1 Diagrama de secvență – selectarea genului filmelor

Diagrama descrie pașii prin care utilizatorul selectează un gen de film și modul în care aplicația filtrează și afișează filmele corespunzătoare.

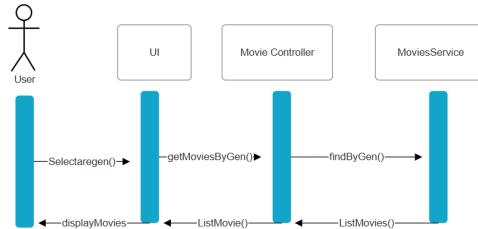


Figura 5: Diagrama de secvență pentru selectarea genului filmelor

4.2.2 Diagrama de activitate – generarea codului QR

Această diagramă prezintă fluxul de activități implicat în generarea codului QR asociat biletului, după finalizarea procesului de cumpărare.

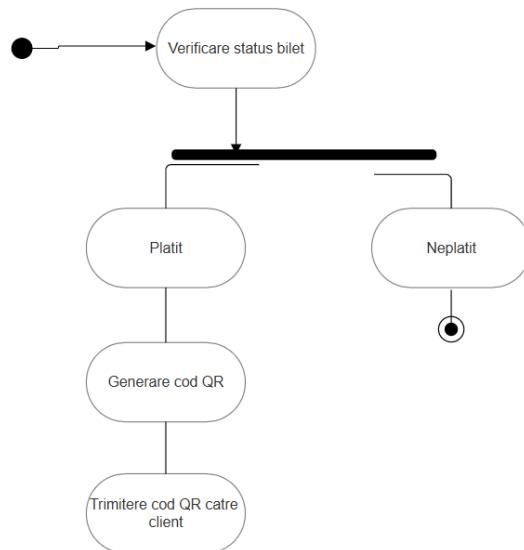


Figura 6: Diagrama de activitate pentru generarea codului QR

4.2.3 Diagrama de stare – plata fictivă

Diagrama evidențiază stările sistemului în timpul procesului de plată fictivă, incluzând validarea datelor și confirmarea tranzacției.

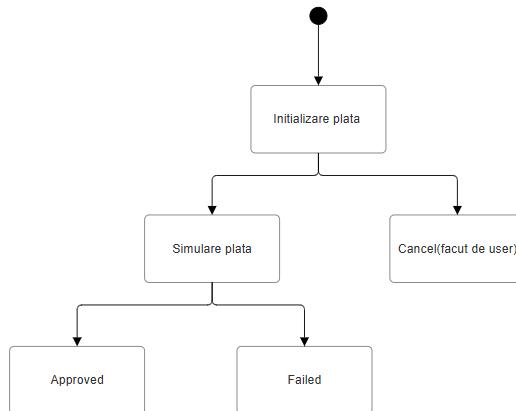


Figura 7: Diagrama de stare pentru plata fictivă

4.3 Diagrame realizate de Cătălin

4.3.1 Diagrama de stare – acțiuni rezervare/anulare

Această diagramă descrie stările posibile ale unei rezervări și tranzitiaile dintre acestea în cazul rezervării sau anulării unui bilet.

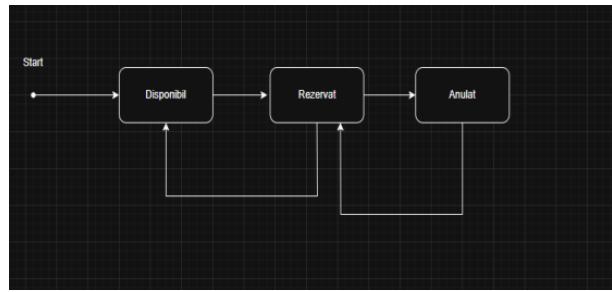


Figura 8: Diagrama de stare pentru acțiuni de rezervare și anulare

4.3.2 Diagrama de secvență – selectare și afișare pe hartă

Diagrama prezintă interacțiunea dintre utilizator și aplicație în momentul selectării unui cinematograf și afișării acestuia pe harta orașului.

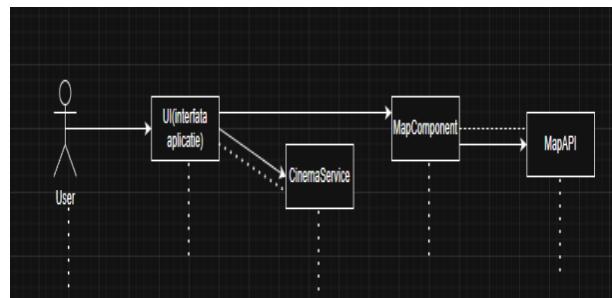


Figura 9: Diagrama de secvență pentru selectare și afișare pe hartă

4.3.3 Diagrama de activitate – interacțiunea utilizatorului

Această diagramă este centrată pe acțiunile utilizatorului în aplicație, ilustrând pașii principali parcursi de acesta de la accesarea aplicației până la cumpărarea unui bilet.

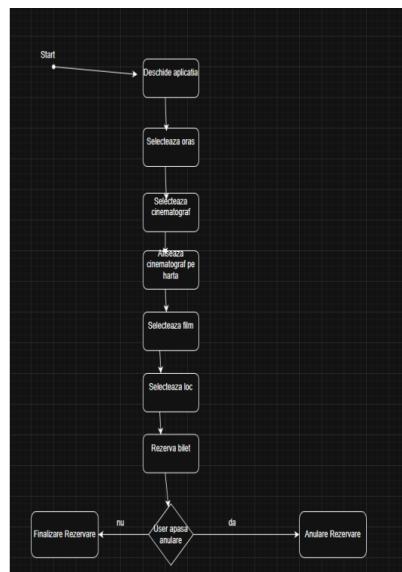


Figura 10: Diagrama de activitate bazată pe interacțiunea utilizatorului