

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

Cinema4U

Document de analiza si design

Student(i): Pop Alexandra

Grupa: 30235

MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

Contents

I Specificatia proiectului	3
1.1 Diagrama Domain Model	3
II Modelul Use-Case	4
2.1 Utilizatori si stakeholderi	4
2.2 Identificarea Use-Case-uri	4
2.3 Diagrama UML Use-Case	6
III Design arhitectural	7
3.1 Arhitectura conceptuala	7
3.2 Diagrama de pachete	8
3.3 Diagrama bazei de date	9
3.4 Diagrame de secventa	10
3.5 Diagrame de activitati	12
IV Specificatii suplimentare	13
4.1 Specificatii non-functionale	13
4.2 Constrangeri de design	14
V Testare	14
5.1 Testarea functionalitatii aplicatiei	14
5.2 Dezvolatari ulterioare	14
VI Bibliografie	15

MINISTRY OF EDUCATION



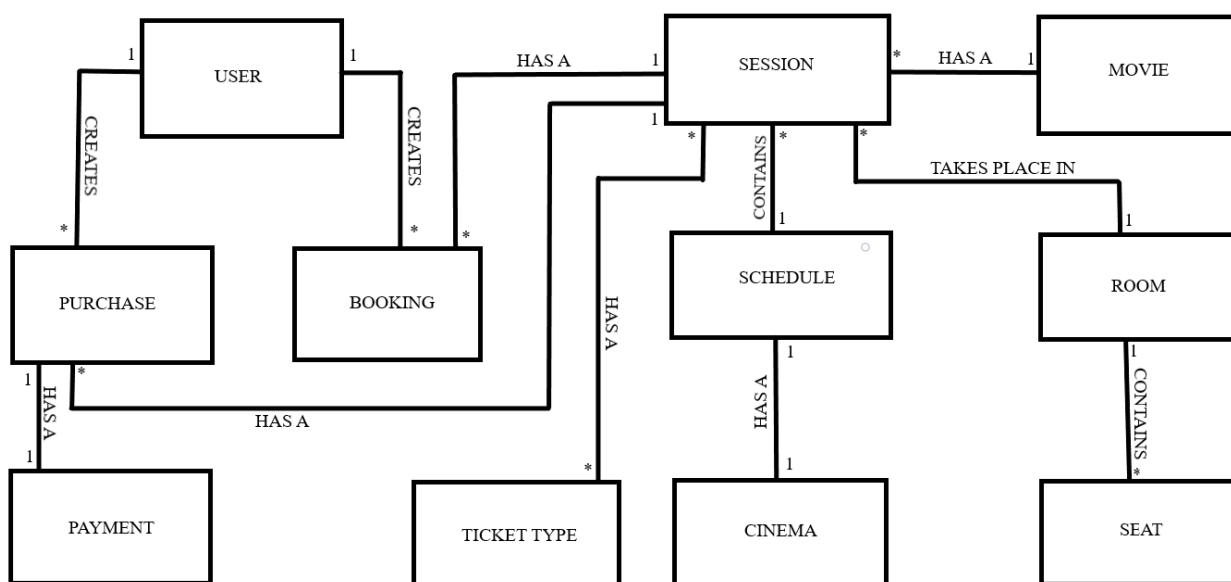
TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

I Specificatia proiectului

Proiectul Cinema4U presupune o aplicație ce permite utilizatorilor să vizualizeze filmele disponibile la cinema-urile înregistrate în aplicație, să își rezerve sau să cumpere bilete la un anumit film, într-o zi și la o oră din cele listate ca fiind disponibile. Pe lângă acestea, administratorul poate să adauge noi cinema-uri în lista de cinema-uri vizibile utilizatorilor și noi filme pentru cinema-uri.

1.1 Diagrama Domain Model



Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

II Modelul Use-Case

Aplicația prezintă două tipuri de utilizatori principali: user-ul obișnuit ce folosește aplicația în scopul cumpărării sau rezervării de bilete de cinema, și administratorul care folosește aplicația atât ca un utilizator obișnuit, cât și ca un maintainer, ocupându-se de lucruri administrative precum adăugare de cinema-uri în listă, adăugare/modificare program cinema.

2.1 Utilizatori si stakeholderi

Utilizatorii aplicației sunt oamenii de orice vârstă care doresc cumpărarea/rezervarea unor bilete la cinema-urile din zona lor.

Stakeholderii aplicației sunt managerii/directorii de cinema-uri din lista care va fi prezentă în aplicație.

2.2 Identificarea Use-Case-uri

Nume Use case: View cinema schedule.

Nivel: User-Goal.

Actor principal: User, Admin.

Scenariul principal de success: User-ul/Admin-ul deja logat, aflat în pagina principală a site-ului, după alegerea cinema-ului și a datei, apasă butonul de “show schedule”, iar apoi va fi redirecționat către o pagină unde îi vor fi listate toate filmele de la acea dată, cu toate orele la care sunt disponibile.

Extensie: Dacă user-ul/admin-ul alege o dată la care cinema-ul ales nu are adăugat până la acel moment un orar, atunci se va afișa un mesaj “Orar indisponibil”.

MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

Nume Use case: Select seats.

Nivel: User-Goal.

Actor principal: User, Admin.

Scenariul principal de success: User-ul/Admin-ul deja logat, aflat în pagina de cumpărare/rezervare bilete, după alegerea numărului de bilete dorit, este redirecționat către o pagină unde poate selecta, dintr-o reprezentare a sălii de cinema, locurile dorite, fiind limitat de numărul ales anterior, iar locurile alese vor fi colorate în portocaliu.

Extensie: Dacă user-ul/admin-ul alege un loc indisponibil se va afișa un mesaj de warning “Loc indisponibil”.

Nume Use case: Add cinema.

Nivel: User-Goal.

Actor principal: Admin.

Scenariul principal de success: Admin-ul deja logat, aflat pe pagina de adăugare a unui cinema, va insera detaliile cinema-ului și va apăsa butonul de “add”.

Extensie: Dacă admin-ul adaugă date neconforme cu cerințele se va afișa un mesaj de warning în care va fi atenționat legat de ce date trebuie modificate pentru a avea succes.

Nume Use case: Pay.

Nivel: User-Goal.

Actor principal: User, Admin.

Scenariul principal de success: User-ul/Admin-ul deja logat, aflat în pagina de cumpărare bilete, după realizarea ultimului pas (alegere locuri), va fi redirecționat către o pagină de plată, unde va adăuga detaliile card-ului utilizat și va apăsa pe butonul de “pay”.

Extensie: Dacă user-ul/admin-ul nu adaugă datele unui card valid, sau plata nu a fost acceptată, se va afișa un mesaj de warning “Plată eșuată”.

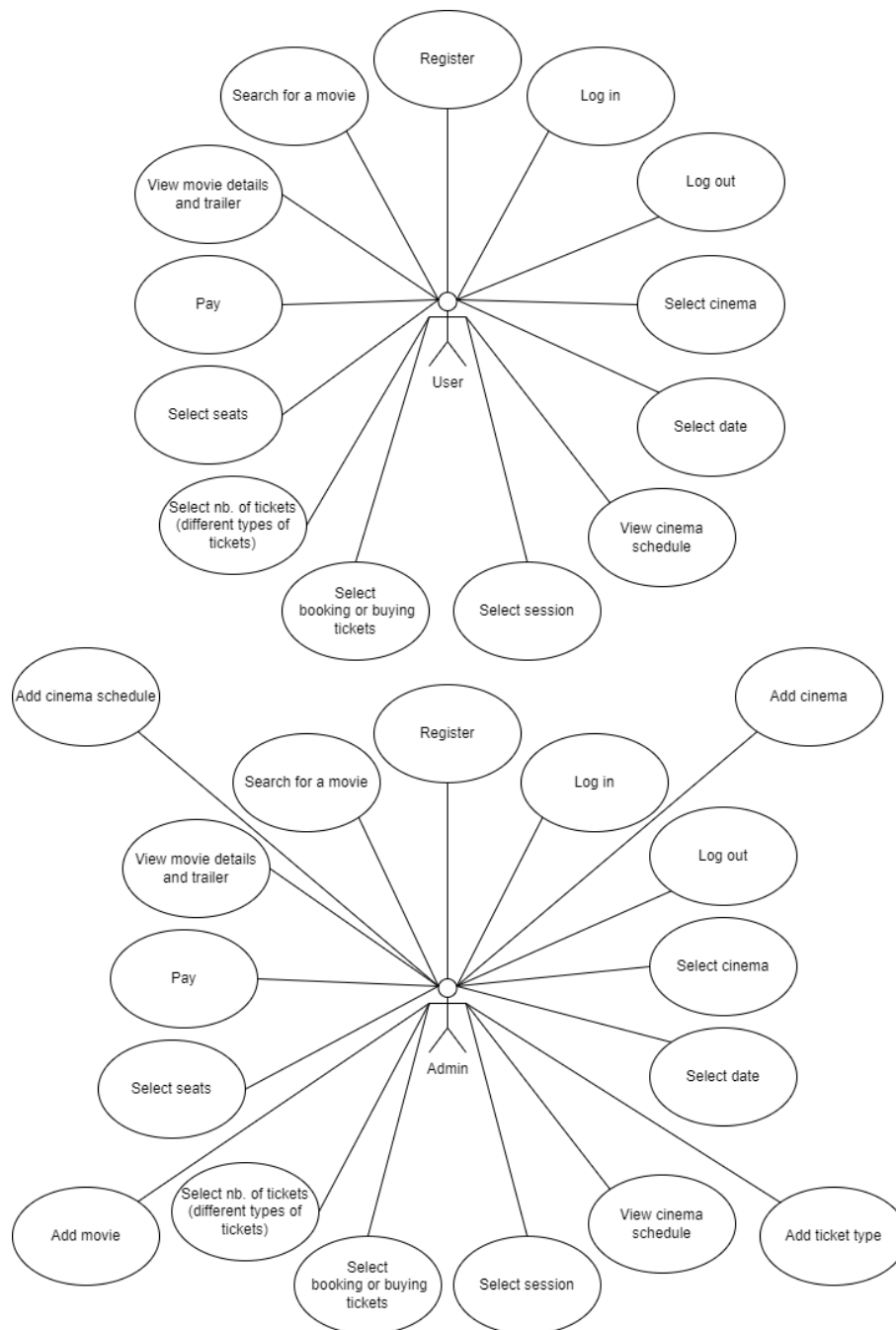
MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

2.3 Diagrama UML Use-Case



MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

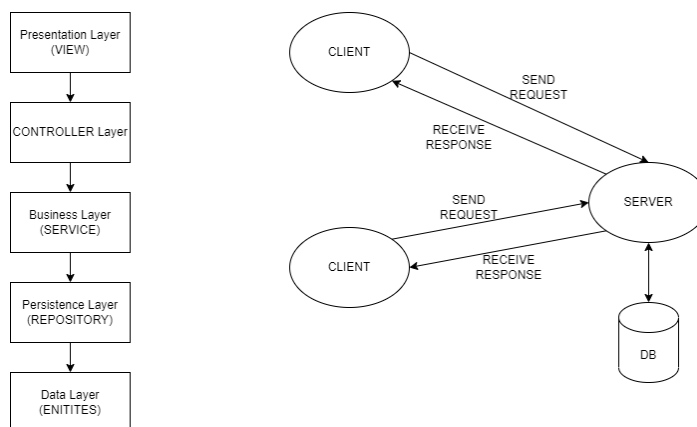
III Design arhitectural

Din punct de vedere arhitectural, aplicația presupune existența mai multor clienți care se conectează la un server pentru accesarea datelor dorite. Clienții vor fi cei care trimit request-uri către server, iar acesta le pune la dispoziție anumite servicii din cele conținute în funcție de request-ul primit.

3.1 Arhitectura conceptuala

Pattern-urile arhitecturale principale ale acestei aplicații vor fi “Layered Architecture” și “Client-Server Architecture”.

Figura diagrame “layered architecture”
si “client-server architecture”



“Layered Architecture” este un pattern potrivit deoarece astfel descompunem atât clientul, cât și server-ul în părți independente, fiecare având ca și scop un subtask al serviciilor puse la dispoziție. Avantajul este că modificarea unui layer nu va duce la modificarea logicii celorlalte layere.

“Client-Server Architecture” va fi folosit deoarece, fiind o aplicație web și având mai mulți clienți care pot să se conecteze simultan, avem nevoie de un mecanism prin care request-urile acestora să fie procesate. Astfel, prin arhitectura client-server, avem un server principal ce conține toată logica ce se aplică pe datele aplicației, și mai mulți clienți care cer server-ului acele date. Server-ul se va afla pe computer-ul principal, și va avea acces la baza de date a aplicației.

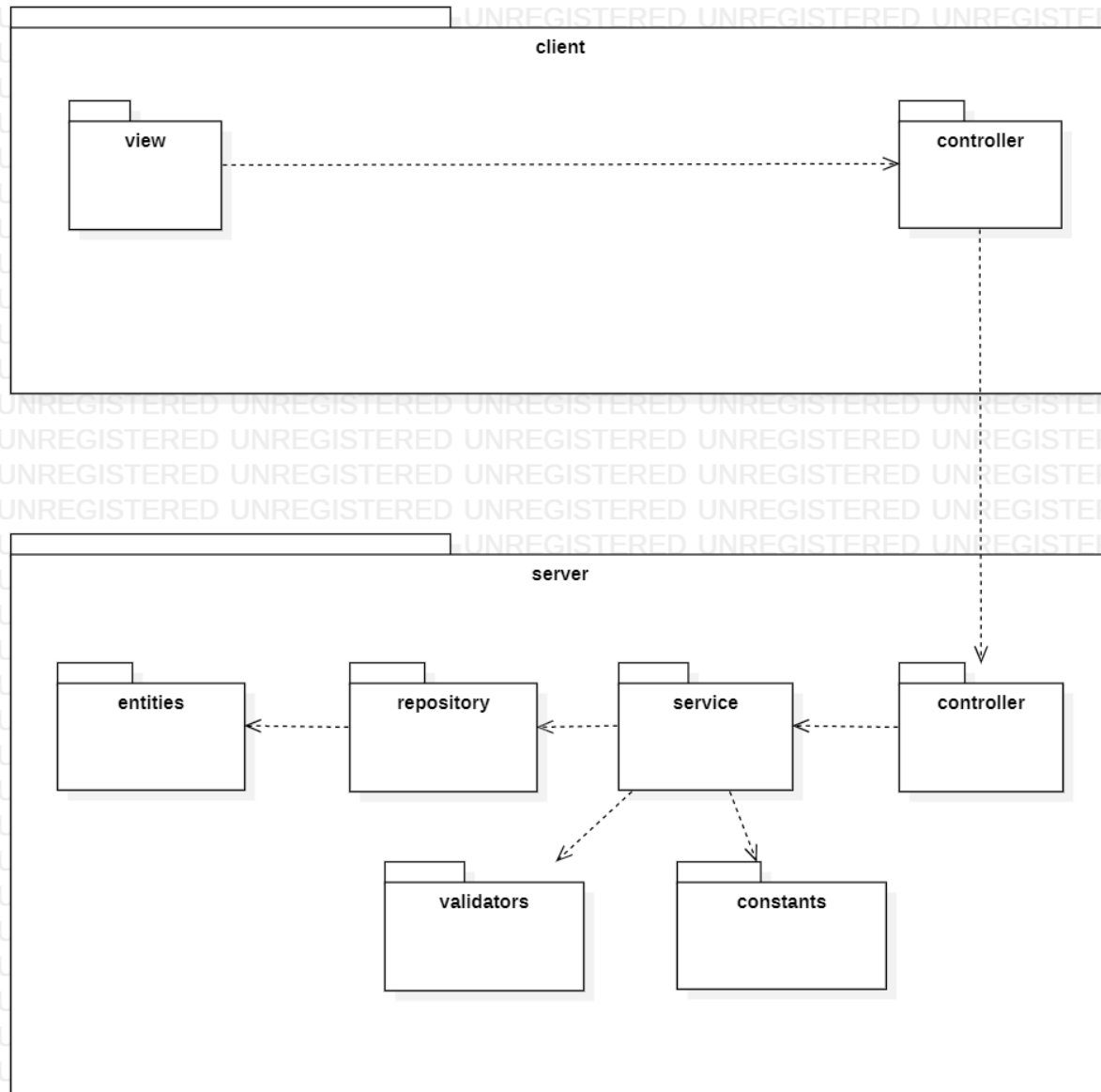
MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

3.2 Diagrama de pachete



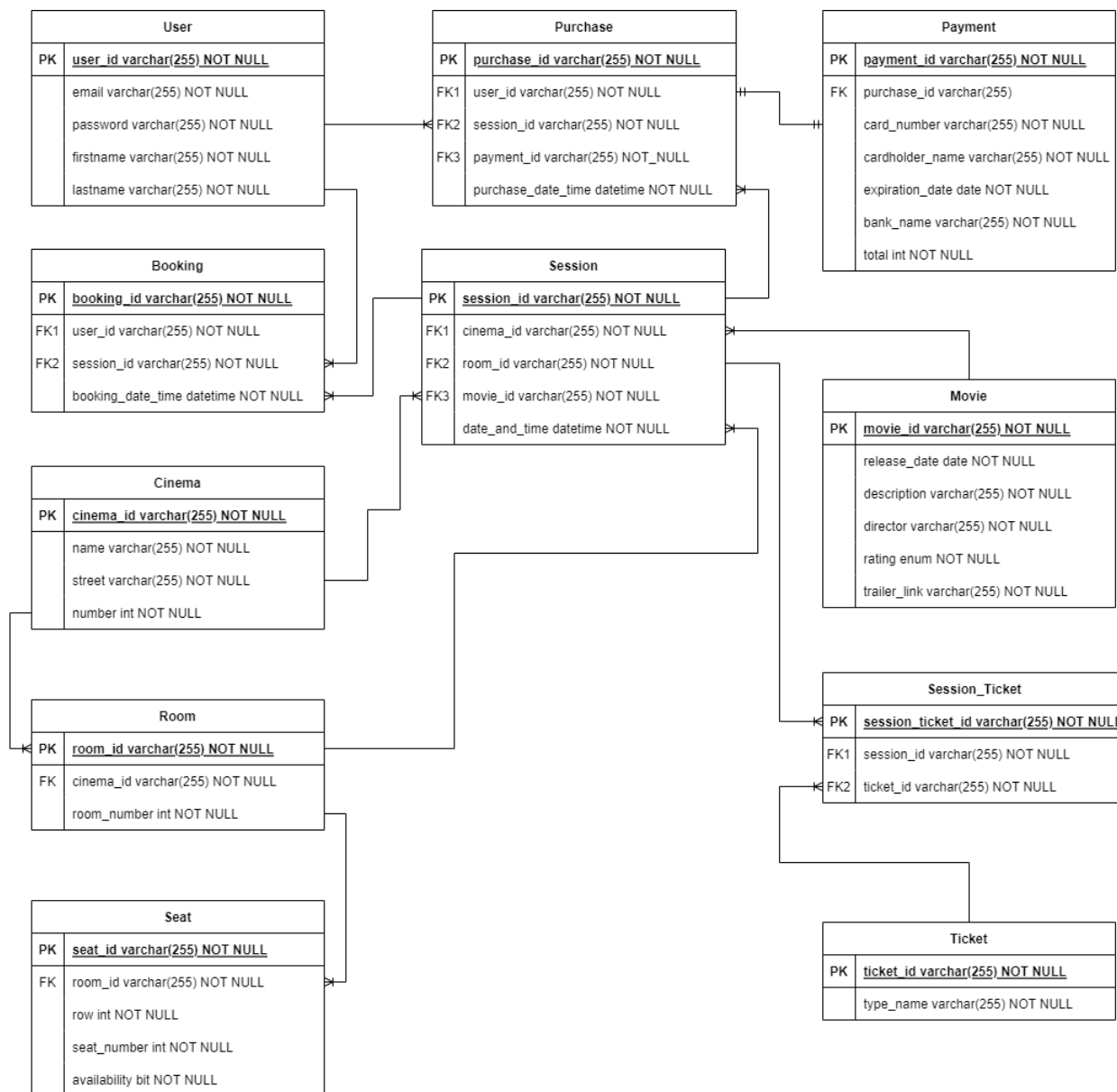
MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

3.3 Diagrama bazei de date



MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

3.5 Diagrame de secventa

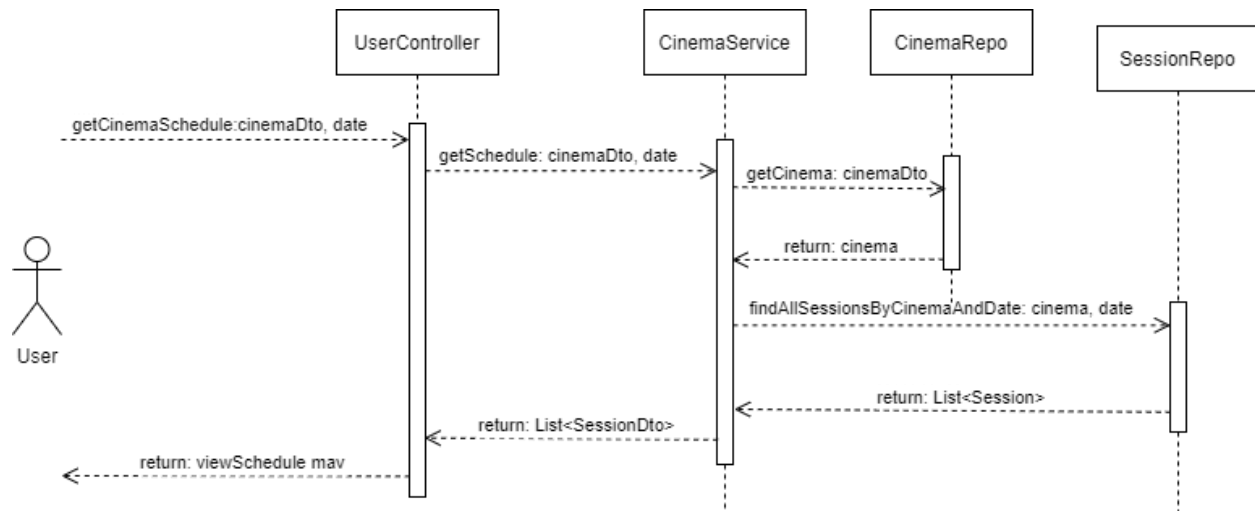


Figure 1 Diagrama de secventa pentru use case-ul "View Cinema Schedule"

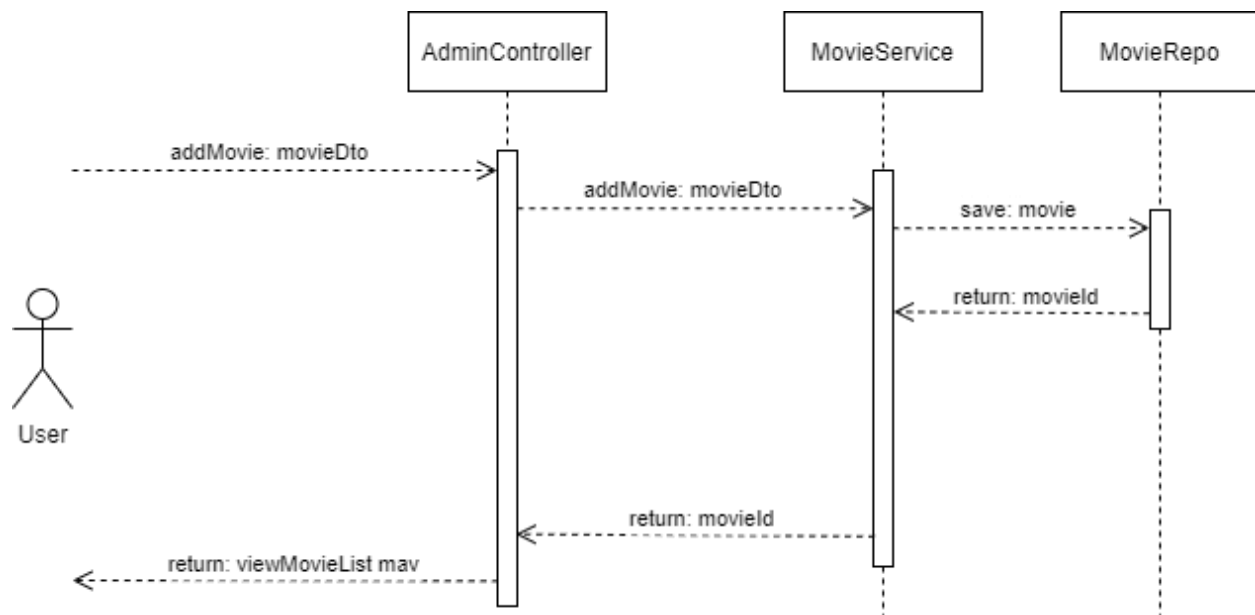


Figure 2 Diagrama de secventa pentru use case-ul "Add movie"

MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

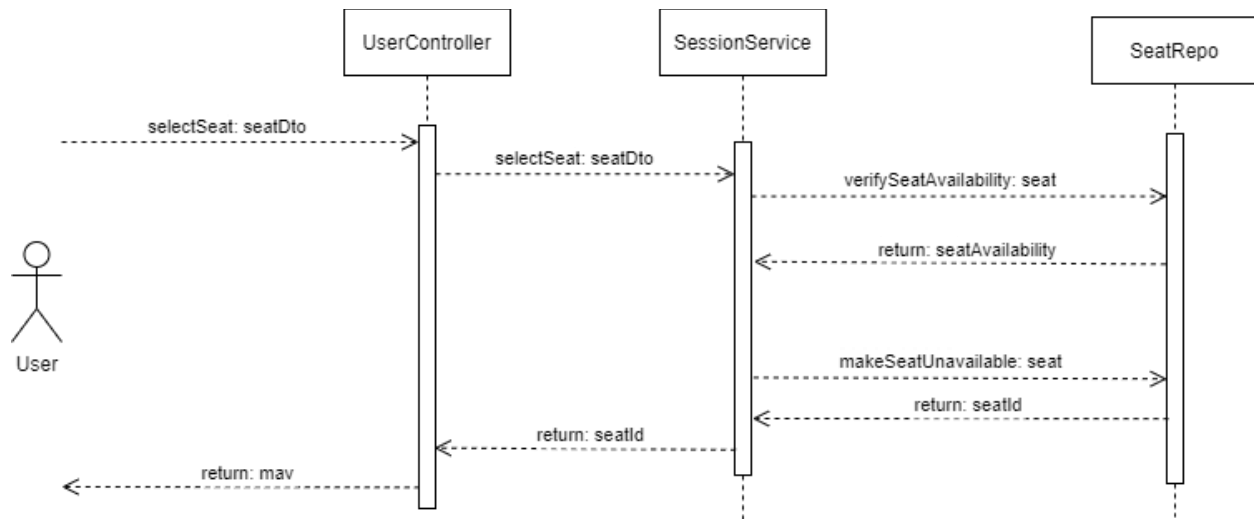


Figure 3 Diagrama de secventa pentru use case-ul "Select Seats"

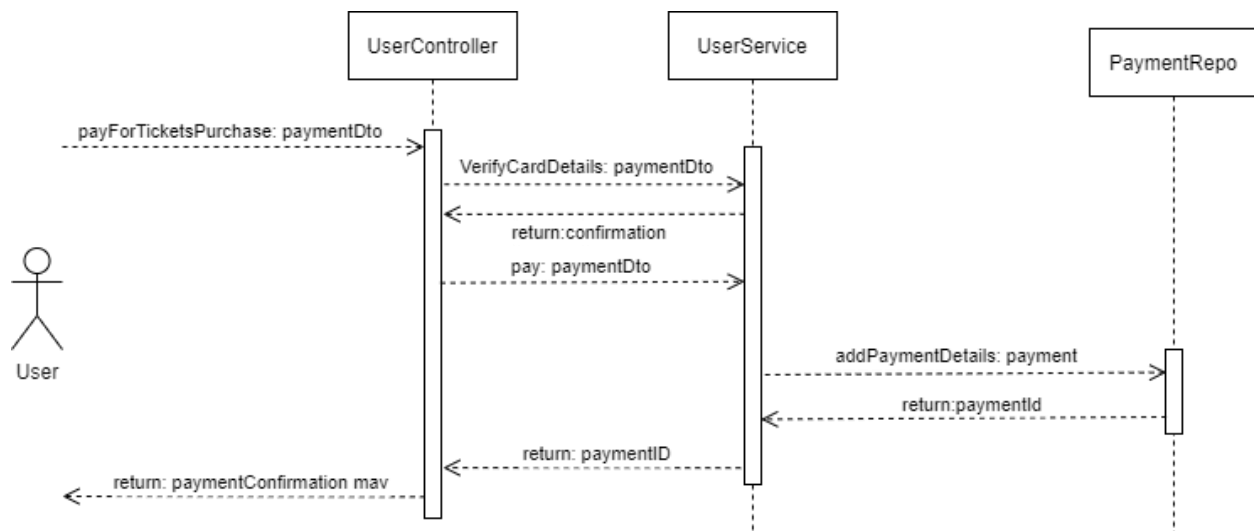
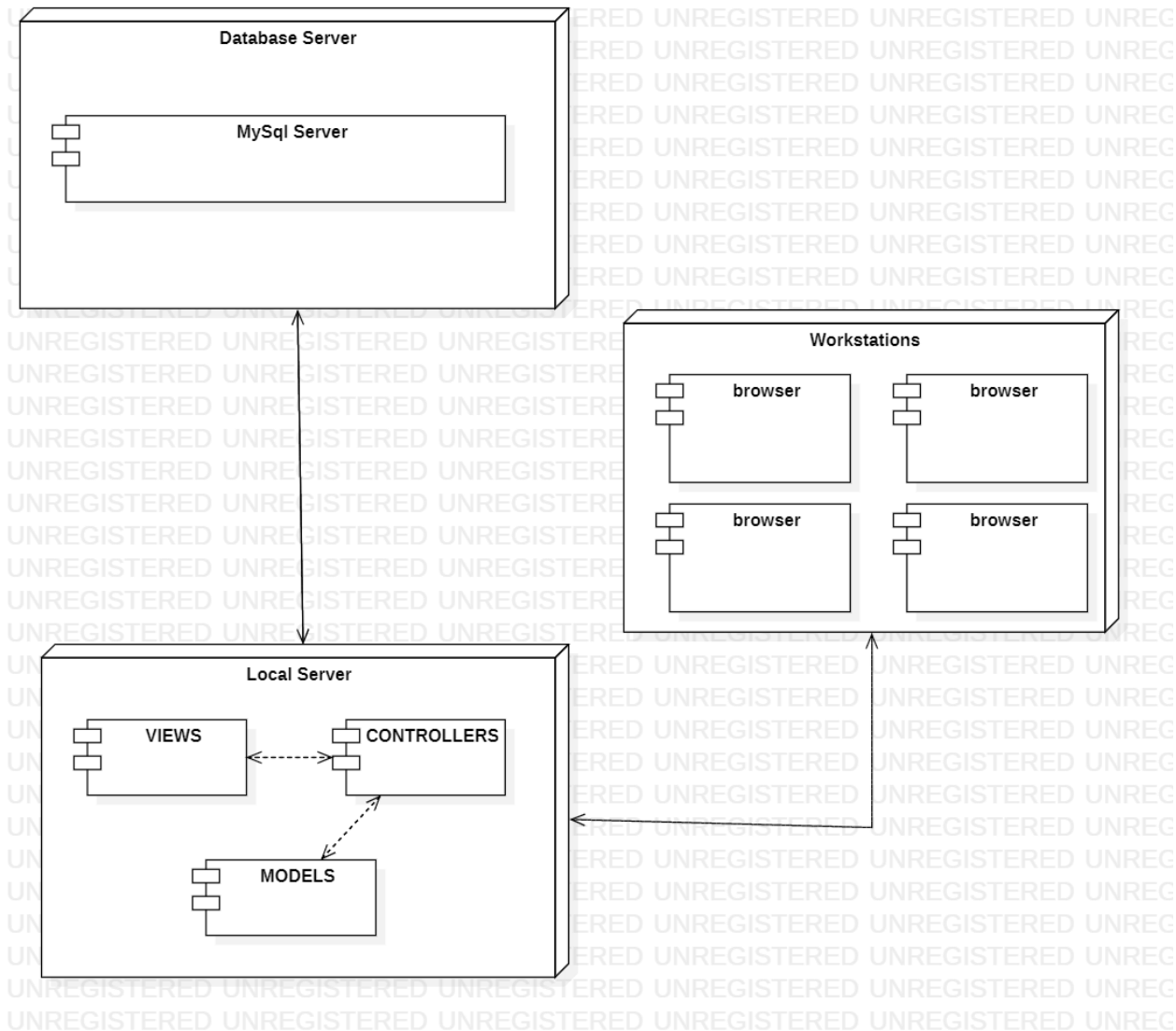


Figure 4 Diagrama de secventa pentru use case-ul "Pay"

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

3.6 Diagrame de deployment/component



MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

IV Specificatii suplimentare

Până acum au fost descrise în principal specificațiile funcționale ale aplicației „Cinema4U”, adică toate constrângerile și funcționalitățile pe care trebuie să le respecte aplicația pentru a avea un comportament corespunzător cerințelor clientului. Pe lângă acestea, și specificațiile non-funcționale au o importanță mare în ciclul de viață al aplicației, deoarece ele decid eficiența și calitatea sistemului.

4.1 Specificatii non-functionale

Principalele specificații non-funcționale ale aplicației sunt:

- Cross-platform: aplicația trebuie să fie funcțională pe mai multe platforme, cum ar fi pe desktop-uri/laptop-uri, android, iOS, pentru ca un număr mai mare de utilizatori să poată avea acces la cinema-urile din zona lor, pentru o utilizare mai facilă și plăcută.
- Extensibility: sistemul trebuie să permită extinderea funcționalităților și adăugarea unor funcționalități în viitor, deci ar trebui să aibă o structură care să fie deschisă schimbărilor. Funcționalități care ar putea fi adăugate în viitor ar fi: notificarea utilizatorilor la adăugarea unor filme noi pe anumite genuri (dacă aceștia aleg să fie notificați), adăugarea unei liste de „favorite movies” pentru utilizatori, adăugarea unei liste de „must watch movies” pentru utilizatori.
- Response time: timpul în care se procesează acțiunea unui utilizator trebuie să fie mai mic de 1 secundă pentru ca utilizatorii să nu părăsească site-ul din motive de timp.
- Backup: aplicația trebuie să aibă un server suplimentar pentru a ține datele stocate, în care să existe cel puțin o copie a datelor din server-ul principal. În caz de eșuare a server-ului principal, se va opera cu datele din server-ul de backup.

MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

4.2 Constrangeri de design

Aplicația trebuie postată într-un repository online la care să aibă acces atât developerii cât și clientul. Fiecare developer va avea un branch propriu în repository, iar codul va fi pus pe branch-ul master doar după ce fiecare dintre developeri verifică codul celorlalți colegi, și se oferă status-ul de „accepted” din partea tuturor. Aceste verificări ale codului se vor face odată la două săptămâni, perioadă în care developerii lucrează la anumite funcționalități alese de client ca având o importanță mai mare. La începutul fiecărei perioade de două săptămâni, clientul și developerii vor avea o întâlnire în care clientul va scoate în evidență aspectele care dorește să fie atinse în funcționalități, iar developerii își vor asigura între ei task-urile. După punerea codului pe branch-ul master, clientul va utiliza demo-ul aplicației câteva zile pentru a face un review.

V Testare

< Se va discuta la laborator./>

5.1 Testarea functionalitatii aplicatiei

5.2 Dezvoltari ulterioare

MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Cinema4U	Version: 2.0
<i>Document de analiza si design</i>	Date: 29/03/2022

VI Bibliografie

1. Deck, P. (2016). Spring MVC: A Tutorial (Second Edition) 2nd Edition. Brainy Software
2. Kuchana, P. (2004). Software Architecture Design Patterns in Java. CRC Press
3. McLaughlin, B. (2006). Head First Object-Oriented Analysis and Design: A Brain Friendly Guide to OOA&D. O'Reilly Media
4. Grzegorz Ziemoński. „Layered Architecture Is Good”,
<https://dzone.com/articles/layered-architecture-is-good>. 19.04.2022 [online]
5. Unhelkar, B. (2017). Software Engineering with UML. CRC Press
6. Spence, I. Bittner, K. (2002). Use Case Modeling. Addison-Wesley
7. Steven Haines. Java persistence with JPA and Hibernate, Part 1: Entities and relationships. <https://www.infoworld.com/article/3373652/java-persistence-with-jpa-and-hibernate-part-1-entities-and-relationships.html>. 20.03.2022 [online]
8. Duckett J. (2011). HTML & CSS: Design and Build Web Sites. Wiley
9. Gamma E., Helm R., Vlissides J., Johnson R. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley
10. Bass L., Clements P., Kazman R. (1997). Software Architecture in Practice. Addison-Wesley

MINISTRY OF EDUCATION



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA