Shows

Student: Victor-Petru Almasan

**Group: 30235**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 4

3. System Architectural Design 5

4. UML Sequence Diagrams 7

5. Class Design 7

6. Data Model 8

7. System Testing 8

8. Bibliography 9

1. Requirements Analysis

# Assignment Specification

Aceasta aplicatie este de tip client-server si ajuta la gestionarea facila a unor show-uri cum ar fi: filme, evenimente sportive sau piese de teatru.

# Functional Requirements

Aplicatia are *trei* tipuri de useri: *administrator*, *premium* *user* si *regular user*.

*Administratorul* poate sa vizualizeze/editeze/adauge/stearga show-uri. De asemenea el poate sa vizualizeze/editeze/adauge/stearga useri tip premium sau regular.

*Regular* *user*-ul poate sa caute show-uri si sa vizualizeze detalii despre acestea. El poate sa vizulizeze si propriul istoric al show-urilor vazute. De asemnea acest tip de user poate sa adauge un comentariul si un rating la un show.

*Premium* *user*-ul poate sa caute show-uri si sa vizualizeze detalii despre aceastea. La fel ca si tipul regular user, premium user-ul poate sa vizualizeze lista cu show-urile vazute si sa adauge un comentariu si un rating la un show. De asemenea acest tip de user poate sa recomande un show unui prieten iar acesta va vedea recomandarea pe pagina lui.

# Non-functional Requirements

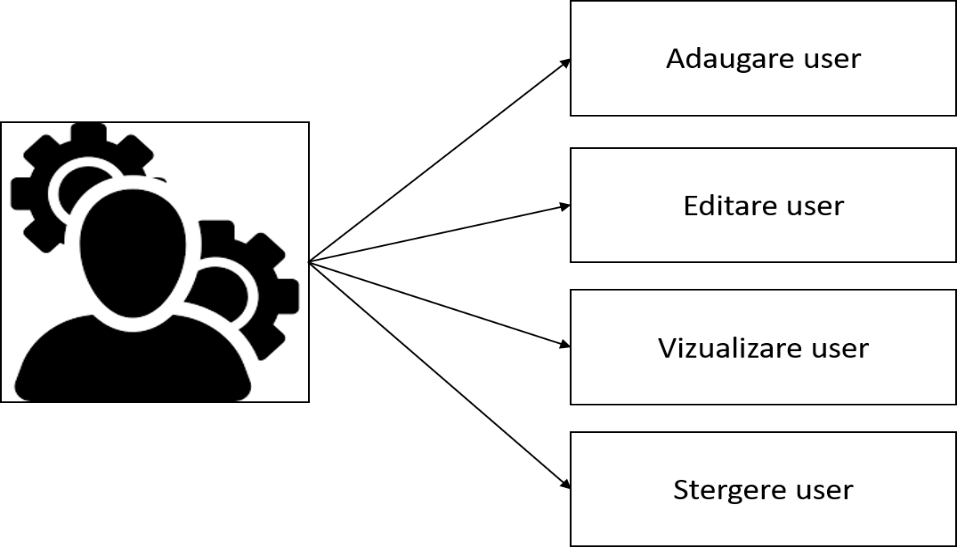
* *Actualizare lista useri in interfata*
  + Source of stimulus: Administrator
  + Stimulus: Interfata trebuie sa fie prietenoasa la adaugarea unui user
  + Environment: Logare ca administrator
  + Artifact: AdminController
  + Response: Lista tuturor userilor este actualizata
* *Adaugarea unui prieten*
  + Source of stimulus: Premium User
  + Stimulus: Interfata ofera vizual lista de useri care pot fi adaugati ca prieteni
  + Environment: Logare ca premium user
  + Artifact: PremiumUserController
  + Response: O fereastra cu confirmarea adaugarii ca prieten

2. Use-Case Model

Use case: Operatii CRUD pe useri

Level: summary level

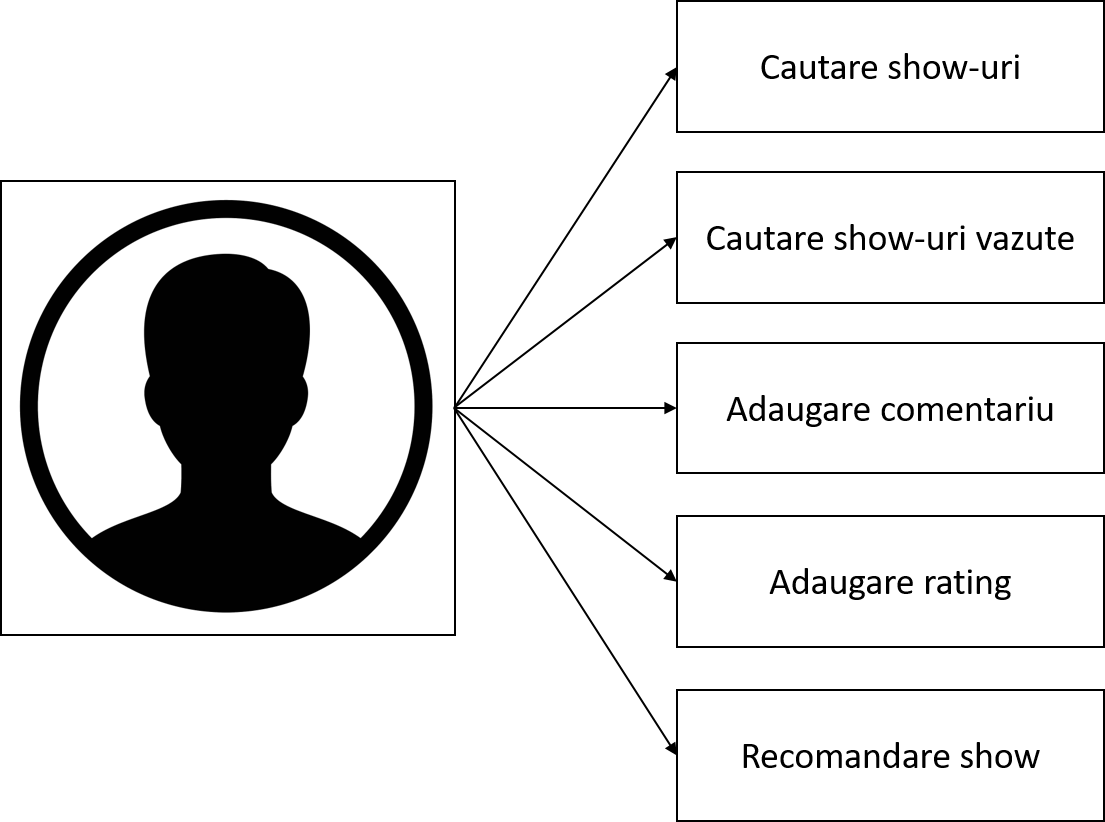
Primary actor: administrator



Use case: Operatii premium user

Level: user-goal level

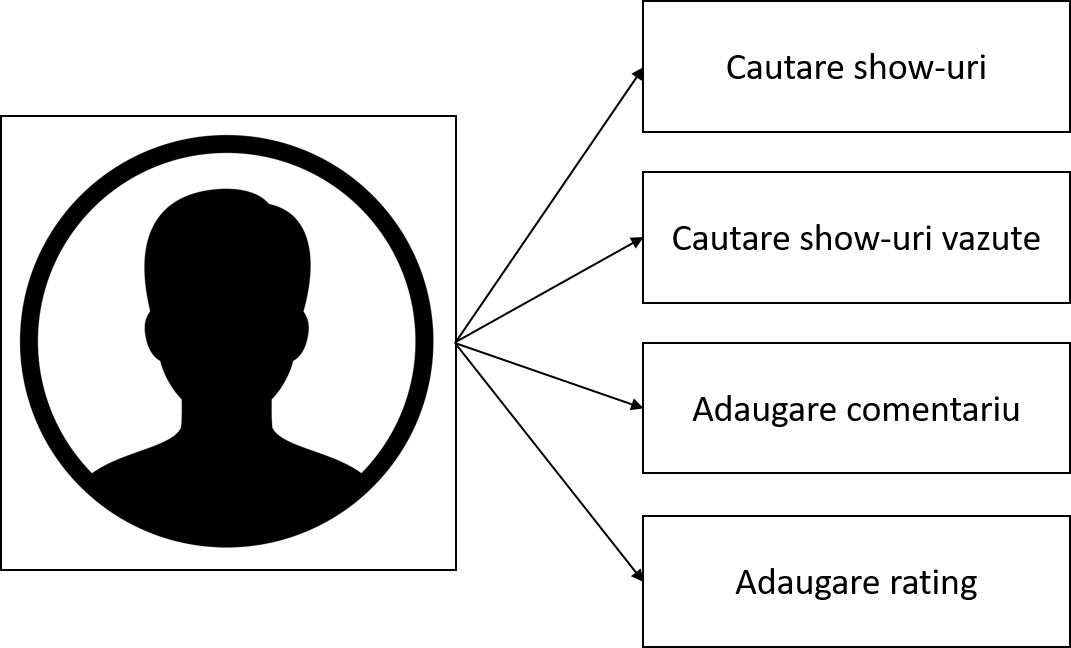
Primary actor: premium user



Use case: Operatii regular user

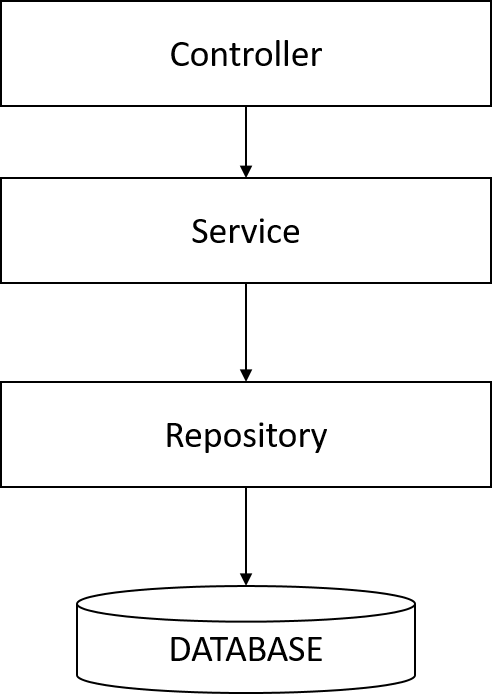
Level: user-goal level

Primary actor: regular user



3. System Architectural Design

**3.1 Architectural Description**

**

**3.2 Diagrams**

Diagrama de pachete:

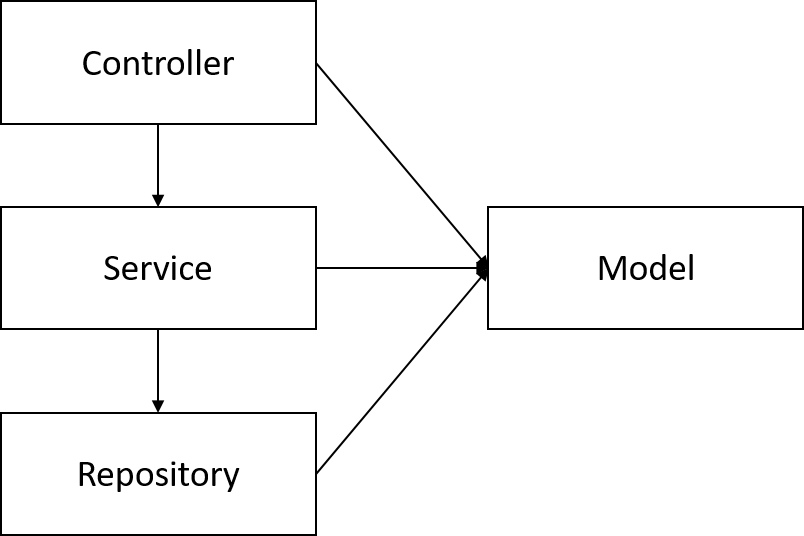


Diagrama de componente pentru PremiumUserController:

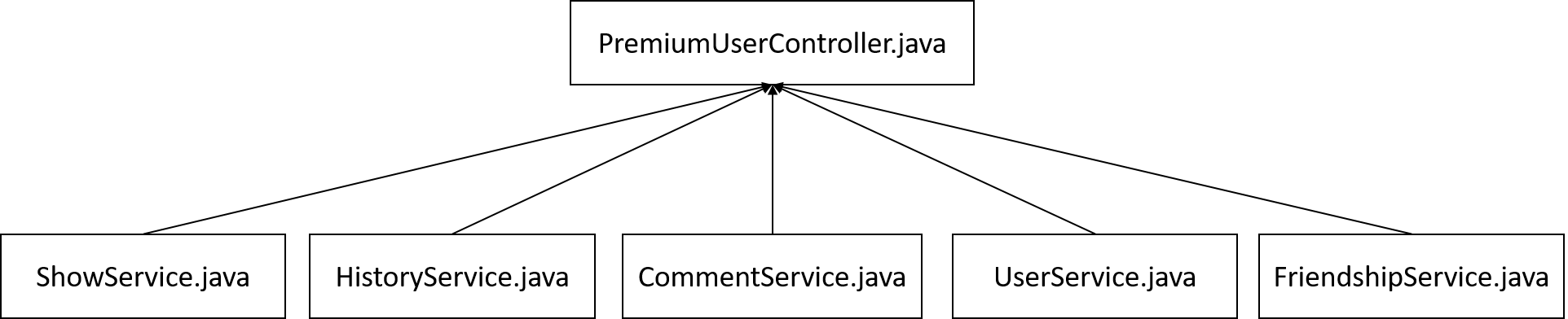
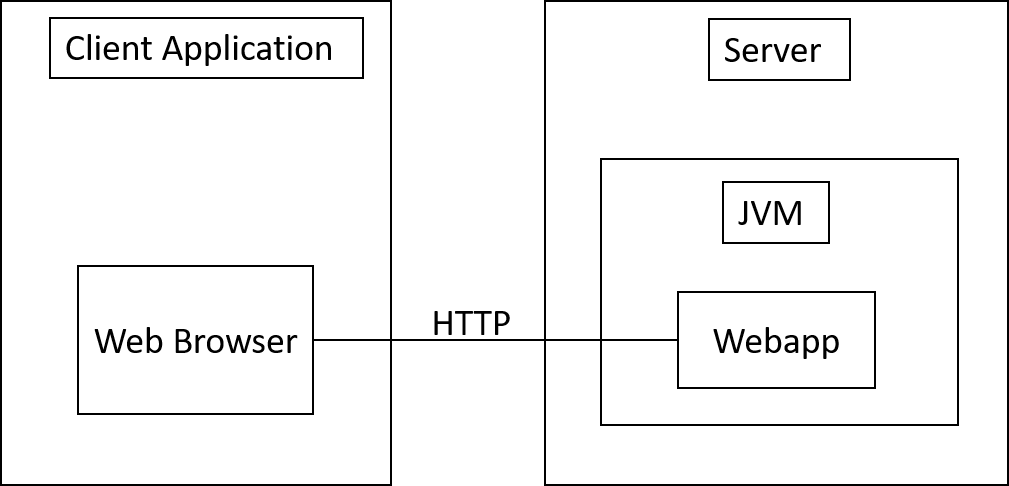
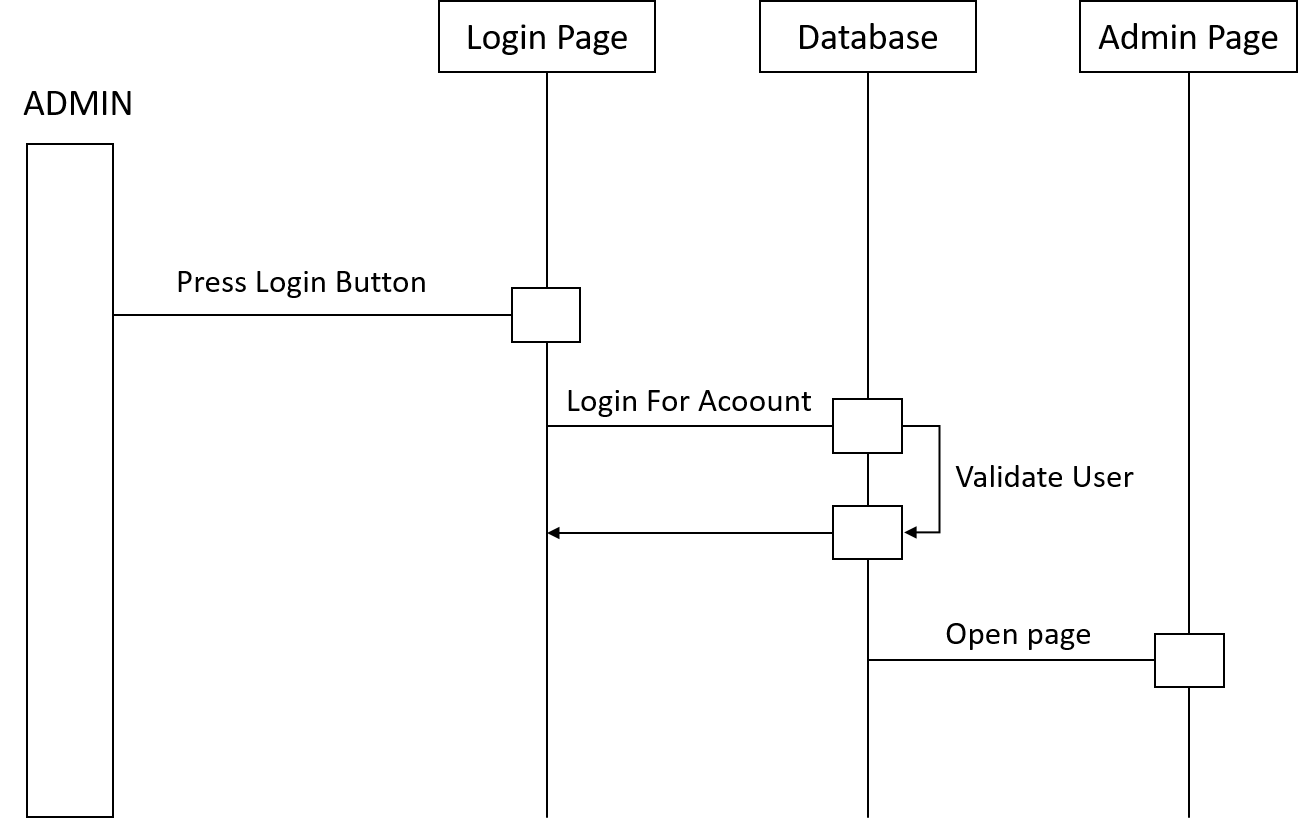


Diagrama Deployment:

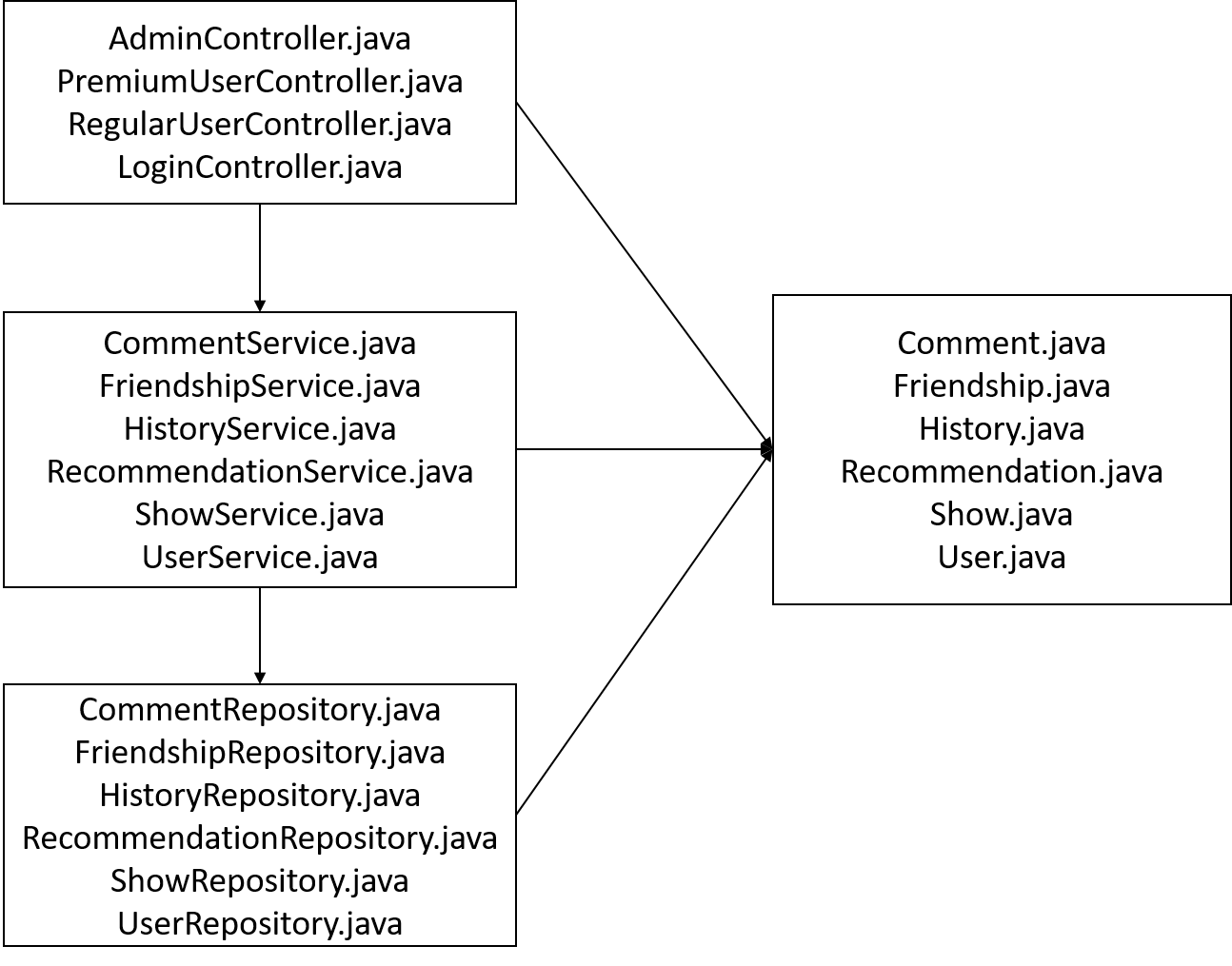


4. UML Sequence Diagrams

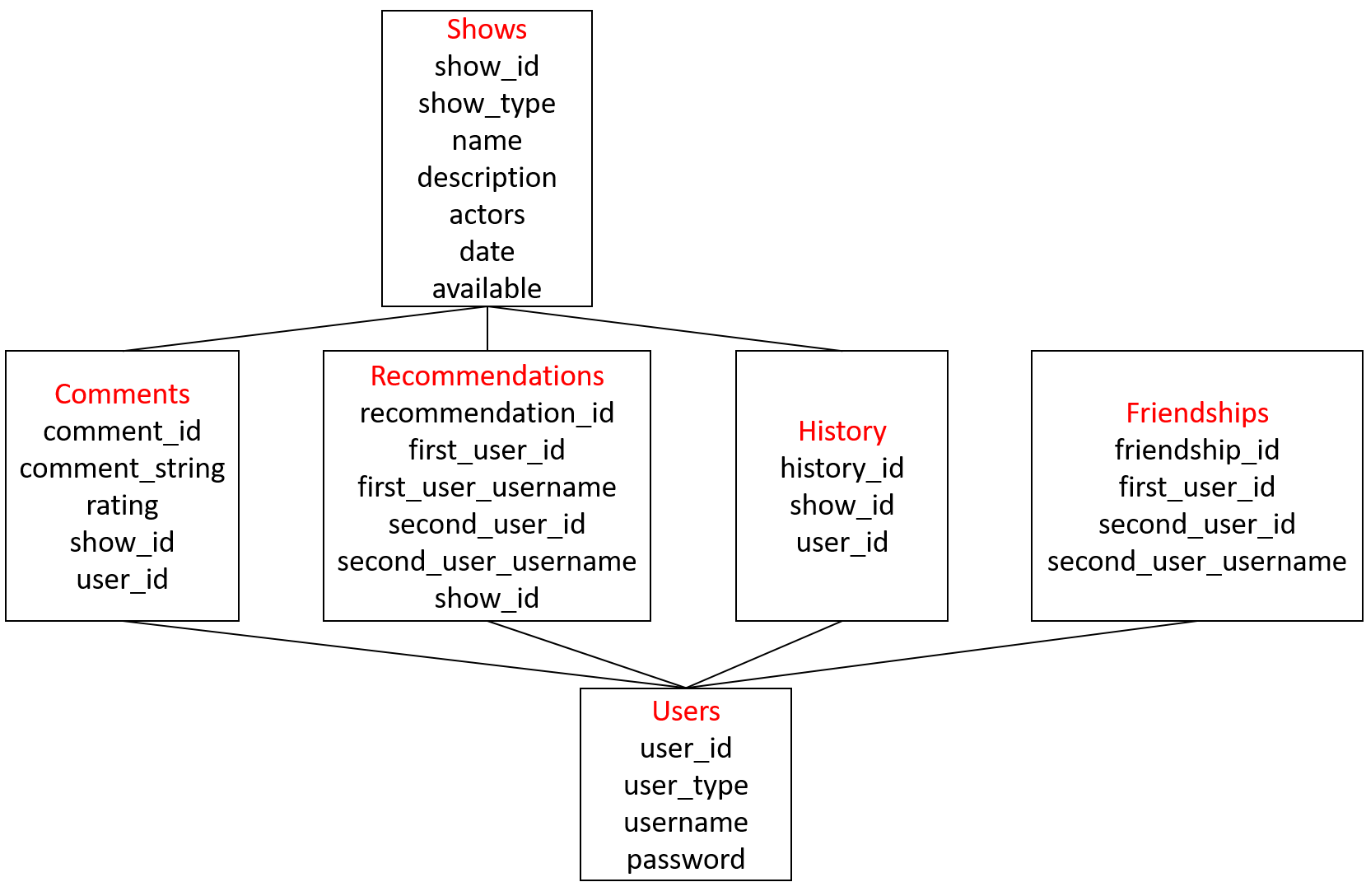
**

5. Class Design

**5.1 UML Class Diagram**



6. Data Model

**

7. System Testing

Aceasta aplicatie are validari. Spre exemplu atunci cand cineva incearca sa se logheze cu un username gresit sau cu o parola gresita el va fi redirectionat din nou spre pagina de login pentru a putea introduce datele corecte.

8. Bibliography

<https://docs.oracle.com/javase/8/docs/api>

<https://spring.io/guides/gs/spring-boot/>

<https://www.mkyong.com/spring-boot/spring-boot-hello-world-example-jsp/>

<https://spring.io/guides/gs/consuming-rest/>

<https://www.youtube.com/watch?v=UP-v1dTOT9g>

<https://www.youtube.com/watch?v=ijcce18bzng>