Aplicație pentru rezervare de bilete la filarmonică

Document de analiză și design

Student: Adrian Bischin

**Grupa: 30238**

**Contents**

[I Specificatia proiectului](#_heading=h.gjdgxs) **3**

[1.1 Diagrama Domain Model](#_heading=h.30j0zll) 3

[II Modelul Use-Case](#_heading=h.1fob9te) **4**

[2.1 Utilizatori si stakeholderi](#_heading=h.y9rd2x440h7k) 4

[2.2 Identificarea Use-Case-uri](#_heading=h.3znysh7) 4

[2.3 Diagrama UML Use-Case](#_heading=h.2et92p0) 4

[III Design arhitectural](#_heading=h.tyjcwt) **5**

[3.1 Arhitectura conceptuala](#_heading=h.3dy6vkm) 5

[3.2 Diagrama de pachete](#_heading=h.1t3h5sf) 5

[3.3 Diagrama de clase](#_heading=h.4d34og8) 6

[3.4 Diagrama bazei de date](#_heading=h.2s8eyo1) 6

[3.5 Diagrame de secventa](#_heading=h.17dp8vu) 6

[3.6 Diagrame de activitati](#_heading=h.3rdcrjn) 6

[IV Specificatii suplimentare](#_heading=h.26in1rg) **6**

[4.1 Specificatii non-functionale](#_heading=h.lnxbz9) 7

[4.2 Constrangeri de design](#_heading=h.35nkun2) 7

[V Testare](#_heading=h.1ksv4uv) **7**

[5.1 Testarea functionalitatii aplicatiei](#_heading=h.44sinio) 7

[5.2 Dezvolatari ulterioare](#_heading=h.2jxsxqh) 7

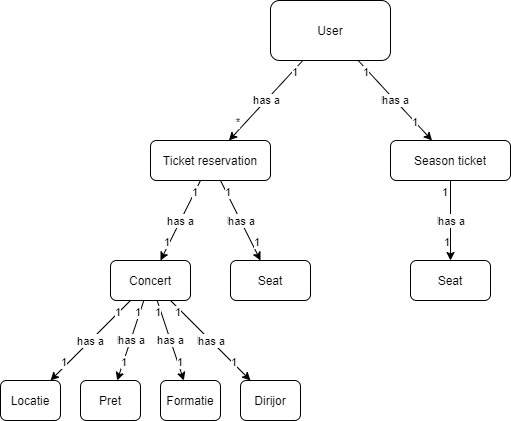
[VI Bibliografie](#_heading=h.z337ya) **8**

# I Specificația proiectului

Proiectul de față reprezintă o aplicație web pentru rezervarea de bilete și pentru abonarea la filarmonică. Sistemul conține o bază de date în care vor fi stocate informații despre stagiunea de concerte, despre locurile disponibile și despre clienții care au cumpărat bilete sau care sunt abonați. Utilizatorii vor interacționa cu sistemul printr-o interfață grafică web.

Asemenea altor aplicatii de acest fel, cei care acceseaza aplicația vor putea găsi informații cu privire la instituția filarmonicii (dirijori permanenți, director, orchestră, cor, secretari și alți angajați) precum și informații cu privire la concertele din stagiunea curentă.

## 1.1 Diagrama Domain Model

**

# II Modelul Use-Case

Dat fiind contextual aplicației -rezervări de bilete la filarmonică- cazurile principale de utilizare sunt de rezervare de bilete, de abonamente, vizualizarea concertelor și a angajaților instituției, operații CRUD, înregistrare utilizatori/administratori. Detalii vor fi date în cele ce urmează.

## 2.1 Utilizatori si stakeholderi

Aplicația poate fi folosită de trei tipuri de utilizatori: vizitator, utilizator și administrator, fiecare având posibilități/atribuții diferite. Vizitatorul poate folosi aplicația fără a fi autentificat, în schimb administratorii și utilizatorii vor trebui să fie înregistrați ș autentificați.

## 2.2 Identificarea Use-Case-uri

*< Aici se vor prezenta 3-4 use-case-uri mai importante din applicatie dupa urmatorul model*

***Nume Use case: < Nume use-case>.***

***Nivel: < User-Goal, Subfunction, Summary > .***

***Actor principal: < Actorul scenariului> .***

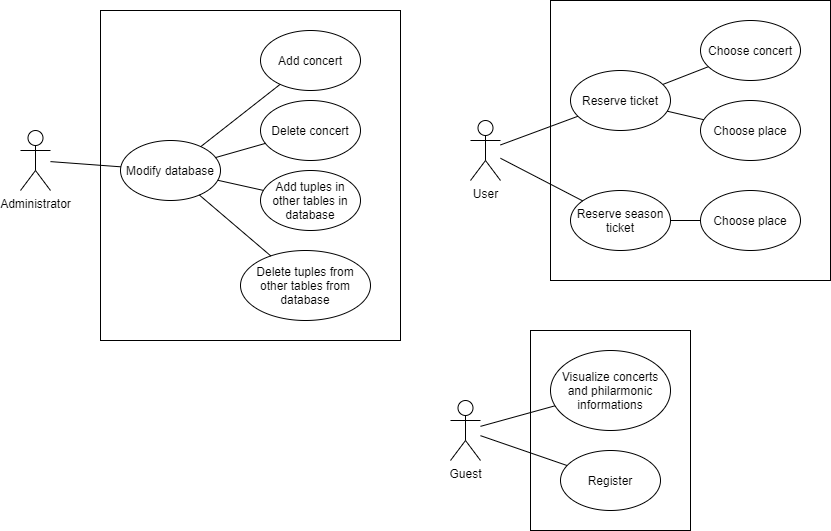
***Scenariul principal de success: <Descriere detaliata a scenariului>.***

***Extensie: <Un caz particular al scenariului, fie pozitiv sau negativ>***

*Se va discuta la laborator mai detaliat.*

* Use case vizualizare:
  + Actor principal: vizitatorul
  + Scenariul principal de succes: vizitatorul poate vizualiza concertele și informațiile aferente lor precum și informații despre angajații instituției.
  + Necesitate autentifiacer: NU
* Use case rezervare bilet:
  + Actor principal: utilizatorul
  + Scenariul principal de success: utilizatorul poate să aleagă un concert din listă, apoi să reserve un bilet alegând un loc în sală dintre cele rămase disponibile.
  + Necesitate autentificare: DA
* Use case rezervare abonament:
  + Actor principal: utilizatorul
  + Scenariul principal de success: utilizatorul poate să își facă abonament alegând un loc disponibil la toate concertele
  + Necesitate autentificare: DA
* Use case operații CRUD:
  + Actor principal: administratorul
  + Scenariul principal de success: administratorul poate să vizualizeze, să adauge și să șteargă într-un meniu specific lui: concerte, orchestranți, coriști, dirijori, alți angajați, utilizatori și bilete.
  + Necesitate autentificare: DA, în cont de administrator

## 2.3 Diagrama UML Use-Case

****

# III Design arhitectural

*< Se va scrie o mica introducere./>*

Ca arhitectura,, aplicatia va avea un design de tip layers combinat cu modelul Model View Controller.

## 3.1 Arhitectura conceptuala

*<In acest capitol se vor prezenta arhitectura proiectului. Este o aplicatie web sau desktop. Are baze de date? Cate componente are? Client-Server? etc*

Aplicatia va fi web si pentru asta vom lucra (cel mai probabil) cu framework-ul Spring. Aplicatia Se poate imparti in 3 mari componente: Baza de date, logica de business (back-end) si partea de web (front-end). Pentru baza de date vom folosi aplicatia MySQL Workbench, Hibernate, JPA (Java Persistence API). Componenta de back-end va avea o subcomponenta repository, care se va ocupa de interactiunea cu baza de date si de schimbul de informatii cu aceasta, cu ajutorul hibernate si JPA care fac interactiunea mai usoara.

*/>*

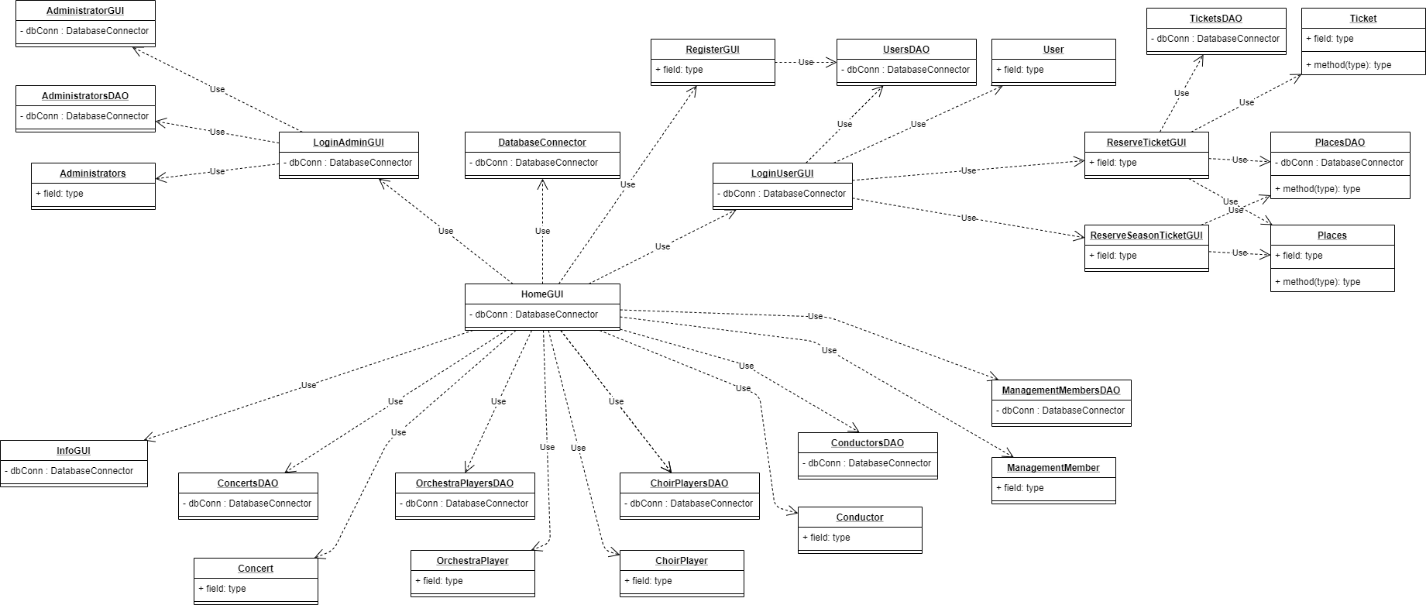
## 3.2 Diagrama de pachete

*< (Package Diagram)/>*

**

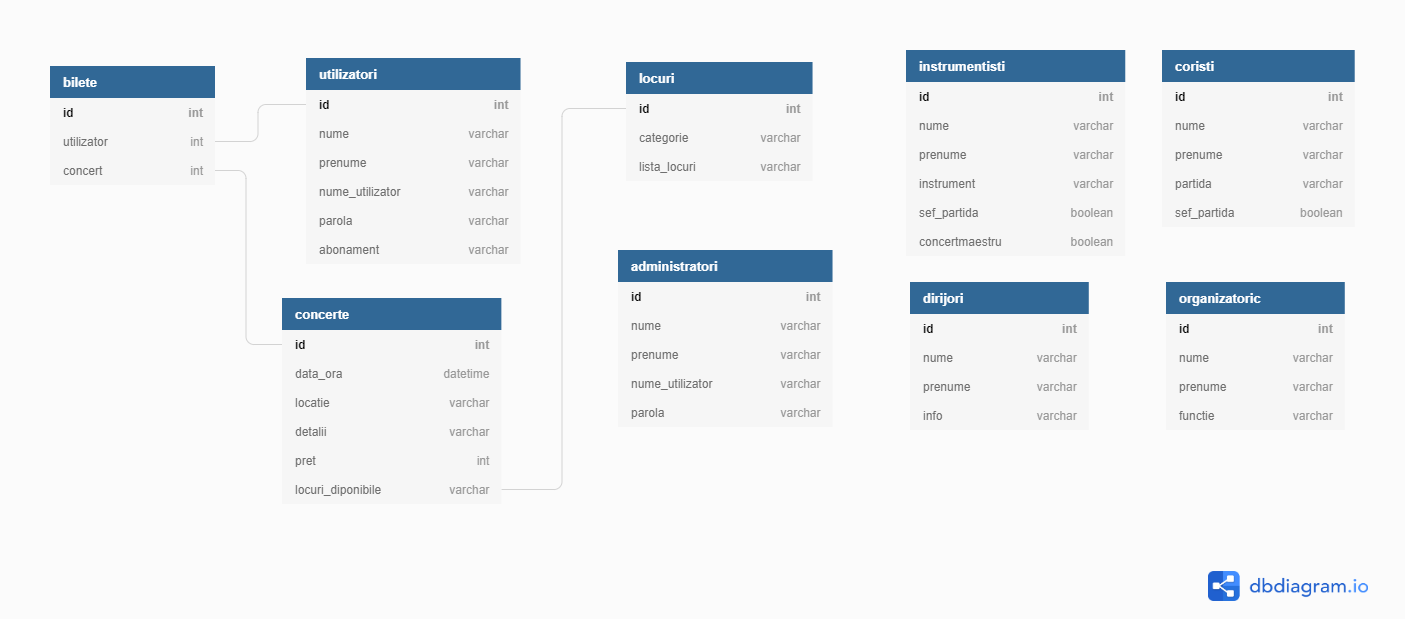
## 3.3 Diagrama de clase

*< (Class Diagram)/>*



## 3.4 Diagrama bazei de date

*< (Data Model)/>*

**

## 3.5 Diagrame de secventa

*< (Sequence Diagram)/>*

## 3.6 Diagrame de activitati

*< (Activity Diagram)/>*

# IV Specificatii suplimentare

*< Se va scrie o mica introducere./>*

## 4.1 Specificatii non-functionale

*< Specificatiile non-functionale ale aplicatiei. Se va discuta la laborator./>*

## 4.2 Constrangeri de design

*< Se va discuta la laborator./>*

# V Testare

*< Se va discuta la laborator./>*

## 5.1 Testarea functionalitatii aplicatiei

## 5.2 Dezvolatari ulterioare

# VI Bibliografie