|  |
| --- |
| PROIECT – INGINERIA SISTEMELOR SOFTWARE  AN UNIVERSITAR 2022-2023  SEMESTRUL 2 |
| **Aplicatie**  **Rezervari locuri** |
|  |

|  |  |
| --- | --- |
| **Nume si prenume student**  Voinopol Cosmin-Marian  Grupa 227 semigrupa 2  Informatica | **Coordonator:**  Prof. Sima Ioan |

**Prezentarea cerintei**

Romania este o tara in care cultura este apreciata de multe persoane. De aceea, mersul la spectacole reprezinta o activitate practicata de oameni in timpul liber. Totusi, gestionarea spectacolelor si a salilor este o sarcina ingreunata de procesele manuale folosite in prezent, motiv pentru care se resimte nevoia digitalizarii si automatizarii. Aplicatia dezvoltata de noi aduce acest beneficiu in prim-plan, fiind o cale de a usura munca angajatilor in contextul rezervarii de locuri pentru o sala de spectacole.

O institutie teatrala pune la dispozitia spectatorilor un sistem pentru rezervarea locurilor la spectacole. În fiecare zi, institutia are o singura reprezentatie, la care spectatorii pot rezerva locuri începand de dimineata. Pentru un loc în sala, sunt memorate informatii referitoare la identificarea locului.

Terminalele puse la dispozitia spectatorilor afiseaza întreaga configuratie a salii, precizând pentru fiecare loc numarul si starea (liber sau rezervat). Folosind un astfel de terminal, spectatorul îsi poate introduce datele personale, poate selecta unul sau mai multe locuri si poate declansa un buton pentru rezervarea lor.

După fiecare rezervare, toate terminalele vor afisa situația actualizata referitor la ocuparea sălii.

**Functionalitatile**

**Administrator:**

* Se poate conecta la sistem printr-un cont asociat, cu care poate mai departe sa indeplineasca alte sarcini
* Odata conectat, poate sa configureze sala de spectacol cu locurile existente care sa fie luate in considerare pentru viitoare spectacole
* Odata conectat, poate sa creeze urmatorul spectacol, care se va desfasura in urmatoarea zi, devenind valabil incepand cu orele diminetii

**Spectator:**

* Poate afisa si observa configurarea curenta a salii de spectacol, aferenta acelei zile; in fiecare zi, dimineata, acestea se schimba.
* Odata ce vizualizeaza sala de spectacol pentru acea zi, poate selecta locuri care inca sunt valabile (nerezervate); Odata selectate locurile, poate introduce date personale (nume si varsta), pe baza carora se poate actiona butonul de rezervare pentru a confirma o rezervare. In urma actionarii butonului, va primi confirmare pentru rezervare, sau va fi informat in legatura cu introducerea unor date gresite sau neselectarea de locuri

**Modelul functional**

**Diagrama Cazurilor de Utilizare**

**O imagine care conține text, diagramă, captură de ecran, cerc

Descriere generată automat**

**Descrierea Cazurilor de Utilizare + Scenarii alternative**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | **UC-1**: Show room configuration | | |
| Primary actor | Spectator / client | Secondary actors |  |
| Description | A **spectator** accesses the reservation system through an access terminal (on-site through intranet) and visualizes the room configuration. | | |
| Trigger | A **spectator** indicates the wish to show the room configuration. | | |
| Preconditions | PRE-1. Current time is before the start of the show (but in that morning)  PRE-2. The **Spectator** can access a terminal. | | |
| Postconditions | POST-1. The **Spectator** the current state of the room configuration. | | |
| Normal flow | 1. **Show room configuration** 2. **The Spectator** accesses SR to see the room. (1.0.E1, 1.0.E2) 3. **SR** shows the current state of the room. | | |
| Alternative flows | - | | |
| Exceptions | **1.0.E1 SR is not currently working**  1. SR shows standard error message.  **1.0.E2 The show has already begun / has ended (cannot show the current state of the room)**  1. **SR** informs the **Spectator** that they cannot see the room anymore for that day. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | **UC-2**: Reserve seat(s) | | |
| Primary actor | Spectator / client | Secondary actors |  |
| Description | A **spectator** who has accessed the system (intranet) inputs personal data into the system, after which they select one/more seats and action a button to confirm the reservation. | | |
| Trigger | Un **spectator** with access to SR indicates the wish to reserve one or more seats. | | |
| Preconditions | PRE-1. **Spectator** has completed **UC-1**. (has accessed the system and has shown the current state and configuration of the room)  PRE-2. There are some available seats left for the show. | | |
| Postconditions | POST-1. **Spectator** has successfully reserved one or more seats.  POST-2. **SR** has registered the reservation and has updated the current state, both internally and visually (on the screen). | | |
| Normal flow | 1. **Seat reservation** 2. **Spectator** accesses the window for reservation. 3. **SR** shows the reservation window 4. **Spectator** writes personal data to identify the reservation (2.0.E1) 5. **SR** confirms personal data and enables the spectator to choose seats. 6. **Spectator** selects 1 seat (2.0.E2) 7. **SR** updates the window to reflect the choice. 8. **Spectator** presses reservation button, or alternative flow **1.1**. 9. **SR** shows success message. (2.0.E3) | | |
| Alternative flows | **1.1 Rezervare locuri multiple**   1. **Spectator** selects multiple seats. 2. Repeat steps 1.0.3-4 | | |
| Exceptions | **2.0.E1 Invalid input data**  1. **SR** shows error message. Go back to 1.0.1  **2.0.E2 The chosen seat is already reserved**  1. **SR** shows specific error message. Go to 1.0.3  **2.0.E3 Final verification: The seat has ben reserved in the mean-time (race-conditions)**  1. **SR** shows corresponding error message. Go back to 1.0.3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | **UC-3:** Authenticate / login (admin) | | |
| Primary actor | Administrator | Secondary actors |  |
| Description | An administrator logs into the system | | |
| Trigger | An administrator declares the intention to log in. | | |
| Preconditions | PRE-1. The administrator has access to a terminal (on-site or remote) | | |
| Postconditions | POST-1. The administrator has access into the system.  POST-2. The system has verified the administrator’s credentials. | | |
| Normal flow | 1. **Administrator login** 2. **The administrator** accesses SR in order to log in 3. **SR** shows the login window 4. **The administrator** inputs their usename and password , after which they press the login button. (1.0.E1) 5. **SR** confirms authentication. | | |
| Alternative flows | - | | |
| Exceptions | **1.0.E1. SR** shows an error message “invalid credentials”. Go back to **1.0.3** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | **UC-4**: Room configuration | | |
| Primary actor | **Administrator** | Secondary actors |  |
| Description | An administrator creates a room configuration | | |
| Trigger | An administrator indicates the wish to create a room configuration | | |
| Preconditions | PRE-1. The administrator is authenticated after **UC-3**. | | |
| Postconditions | POST-1. The new room configuration is created & loaded. | | |
| Normal flow | 1. **Room configuration creation** 2. **The administrator** accesses SR in order to create a new room configuration 3. SR shows the corresponding room configuration creation menu 4. The administrator creates a seat. 5. SR shows the updated state of the configuration 6. The administrator presses the finalize button, or alternative flow **1.1** 7. **SR** confirms the creation of the new configuration | | |
| Alternative flows | **1.1 Multiple seats creation 1.** Go back to 1.0.3 | | |
| Exceptions |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | **UC-5**: Schedule show | | |
| Primary actor | **Administrator** | Secondary actors |  |
| Description | An administrator schedules a new show | | |
| Trigger | An administrator indicates the wish to schedule a new show | | |
| Preconditions | PRE-1. The administrator is authenticated after **UC-3**.  PRE-2. There is at least 1 room configuration (**UC-4**). | | |
| Postconditions | POST-1. The new show is scheduled. | | |
| Normal flow | 1. **Room configuration creation** 2. **The administrator** accesses SR in order to schedule a new show 3. SR shows the corresponding show scheduling window 4. The administrator introduces the date and the configuration for the show 5. SR shows the updated state of the window 6. The administrator presses the finalize button (1.0.E1) 7. **SR** confirms the show scheduling. | | |
| Alternative flows | **-** | | |
| Exceptions | **1.0.E1** A show already exists for that date. Go back to **1.0.3** | | |

**Modelul conceptual**

**Diagrama de clase a modelului**

**O imagine care conține text, diagramă, Plan, schematic

Descriere generată automat**

Modelul conceptual contine termenii ce tin de domeniul problemei, precum si relatiile formate intre acestia. Mai departe, solutia este rafinata si apar entitati ce tin de domeniul solutiei.

Complexitatea diagramei rafinate ne obliga sa impartim vizualizarea in 2 parti, insa fisierul proiecului atasat contine toate informatiile necesare pentru o inspectie atenta.

**O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automatDiagrama de clase finisate**

**O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automat**

**Diagrama bazei de date**

**O imagine care conține text, captură de ecran, diagramă, Software multimedia

Descriere generată automat**

**Shows**

Entitatea ce tine cont de spectacolele care vor avea loc. Spectacolul curent este ales pe baza zilei in care are loc.

**RoomConfigurations**

Entitatea ce defineste configurarea unei sali de spectacol. Contine o lista de locuri.

**Seats**

Entitatea ce defineste un loc. Fiecare loc are asociat un nume pentru identificare de catre utilizator si un id intern pentru sistem.

**Spectators**

Entitatea ce defineste spectatorul. Fiecare are asociat un nume, o varsta si un id intern.

**Admins**

Entitatea ce defineste un administrator. Acestia se conecteaza prin username si parola.

**Reservations**

Entitatea ce defineste o rezervare. Acestea sunt asociate unui spectator si unui anumit spectacol. Contin si locurile rezervate de catre spectator.

**Modelul dinamic**

**Diagrame de secventa**

O imagine care conține text, diagramă, Paralel, număr

Descriere generată automat

O imagine care conține text, diagramă, Paralel, număr

Descriere generată automat

O imagine care conține text, captură de ecran, număr, diagramă

Descriere generată automat

O imagine care conține text, diagramă, număr, Paralel

Descriere generată automat

O imagine care conține text, linie, diagramă, număr

Descriere generată automat

O imagine care conține text, chitanță, captură de ecran, linie

Descriere generată automat

**Diagrame de comunicare**

O imagine care conține text, diagramă, linie, Plan

Descriere generată automat

O imagine care conține diagramă, linie, text, Interval

Descriere generată automat

O imagine care conține text, diagramă, linie, Paralel

Descriere generată automat

O imagine care conține diagramă, text, linie, Interval

Descriere generată automat

O imagine care conține text, linie, diagramă, Interval

Descriere generată automat

**Documentatie tehnica**

**Cerinte non-functionale**

Aplicatia trebuie sa fie disponibila pe desktop. Este necesara o interfata grafica pentru a facilita interactiunea cu utilizatorul. Pentru a asigura persistenta datelor, este necesara si o baza de date si o metoda de conectare cu aceasta.

Din punct de vedere al dezvoltarii, este nevoie de un mediu de dezvoltare integrat, precum si de unelte ce usureaza munca: metoda de a transforma entitatile din baza de date in entitati din memorie si o arhitectura potrivita ce permite scalare ulterioara.

**Tehnologii folosite**

Pentru a asigura interfata grafica si disponibilitatea pe desktop, am ales limbajul de programare Java impreuna cu un framework numit JavaFX. Impreuna, reprezinta o alegere ce permite acces la o comunitate imensa de programatori pentru ajutor online, permit dezvoltare ulterioara in alte directii si ofera performanta potrivita pentru acest tip de aplicatie.

Persistenta datelor este asigurata de catre PostgreSQL ca baza de date, iar metoda de comunicare intre ea si aplicatie este conectorul JDBC corespunzator, impreuna cu Hibernate ORM care permite legatura dintre entitati in baza de date si entitati in memorie in aplicatie.

Ca mediu de dezvoltare am ales Intellij IDEA pentru multitudinea de unelte oferite si usurarea muncii.

Pentru cladirea interfetei cu utilizatorul am ales SceneBuilder. Ofera viteza si usurinta.

Arhitectura aleasa este MVC, pentru ca permite impartirea clara a sarcinilor entitatilor din diagrama de clasa, astfel ca aplicatia poate scala fara efort exagerat in viitor.

**Help**

Aplicatia prezinta 2 perspective: cea a spectatorului si cea a administratorului. Odata pornita aplicatia, orice utilizator va intalni meniul principal:

**O imagine care conține text, captură de ecran, Dreptunghi, afișaj

Descriere generată automat**

**O imagine care conține text, captură de ecran, Dreptunghi, afișaj

Descriere generată automat**De aici, administratorul are optiunea de a se conecta:

Si de acolo va aparea ecranul de conectare, unde trebuie introduse datele de conectare, sau poate reveni la meniul principal prin butonul „Go back”:

**O imagine care conține text, captură de ecran, afișaj, Dreptunghi

Descriere generată automat**

Daca alege sa mearga mai departe si sa se conecteze, urmatoarea fereastra afiseaza ceea ce administratorul poate modifica in legatura cu spectacolul din ziua urmatoare. In urmatorul caz, consideram situatia in care spectacolul din ziua curenta tocmai a inceput, iar administratorul nu mai poate modifica nimic pe moment:

**O imagine care conține text, captură de ecran, afișaj, Dreptunghi

Descriere generată automat**

**O imagine care conține text, captură de ecran, Dreptunghi, afișaj

Descriere generată automat**Revenind la meniul principal, acum din perspectiva spectatorului, acesta poate alege butonul de rezervare locuri:

Urmatorul ecran ii permite sa observe configurarea curenta a salii de spectacol, precum si sa selecteze locuri pentru a le rezerva:

**O imagine care conține text, afișaj, captură de ecran, software

Descriere generată automat**

In situatia in care doreste sa faca o rezervare, va putea alege locurile dorite, dupa care introduce datele persoanele si apasa pe butonul de rezervare:

**O imagine care conține text, afișaj, captură de ecran, software

Descriere generată automat**

**O imagine care conține text, afișaj, captură de ecran, software

Descriere generată automat**Pentru a indica locurile alese curent, culoarea locurilor se va modifica:

**O imagine care conține text, afișaj, captură de ecran, software

Descriere generată automat**In ultima instanta, odata ce butonul de rezervare este actionat si rezervarea este confirmata printr-un mesaj popup, starea salii se va modifica pentru a reflecta valabilitatea locurilor:

In acest punct, spectatorul este pregatit pentru a se bucura de locurile alese la spectacol.

**Bibliografie**

Materiale Curs Ingineria Sistemelor Soft

Materiale Seminar Ingineria Sistemelor Soft

https://chat.openai.com

https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Alert.html

https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/FlowPane.html

https://stackoverflow.com/questions/20924414/how-to-get-javafx-hboxes-to-move-items-to-the-next-line

https://docs.oracle.com/javafx/2/layout/builtin\_layouts.htm#CHDGHCDG

https://hibernate.org/orm/documentation/6.2/

https://www.javatpoint.com/hibernate-tutorial

https://www.digitalocean.com/community/tutorials/hibernate-tutorial-for-beginners

https://stackoverflow.com/questions/34858706/flowpane-and-positioning

https://docs.staruml.io/working-with-uml-diagrams/sequence-diagram

https://stackoverflow.com/questions/69547952/javafx-desktop-app-with-hibernate-entity-and-other-annotations-are-not-recogn