6. REZERVARI LOCURI 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 urmatoarele informatii: pozitie (rândul x, loja y etc.), numar si pret.

Terminalele puse la dispozitia spectatorilor afiseaza întreaga configuratie a salii, precizând pentru fiecare loc pozitia, numarul, pretul 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.

**Subst:**

Institutie teatrala

Spectator

Sistem

Loc

Spectacol

Reprezentatie

Sala

Terminal

**Verbe:**

Afiseaza configuratie (terminalul, spectatorului) – pozitie nuar pret stare

Introduce datele (spectatorul) – nume

Selectare 1+ loc(uri) (spectator)

Declanseaza (buton) – pentru rezervare

**Actori:**

* Spectator
* Sistem

**Functionalitati (use case):**

* UC1 = Afisare locuri din configuratia salii
* UC2 = Rezervare locuri (prin selectarea lor) – necesar ca pas introducerea datelor personale; include UC1

UC pentru adaugare si schimbare configuratii

Administrator ca actor

Cerinte non-functionale:

* Aplicatie de Desktop
* Java cu JavaFX
* Postgresql
* Intellij Idea ca IDE
* Ne trebuie un ORM (care foloseste un conector)
* Trebuie aplicatie MVC (putem folosi un framework in care doar ne punem modelele, service-urile si view-urile)

JDBC ca ex. e conector, dar nu ORM

Framework – Spring with Hibernate as ORM (pare ca va lucre frumos cu postgresql)

* Login trebuie implementat (daca fol framework zice ca e gata-facut) Ex. Jumbo / Laravel

Clasele desenate

Baza de date complet facuta, bagam noi niste entitati dar nu trebuie facut CRUD

* In aplicatie (in Java adica), trebuie sa facem doar

Ce trebuie specific pentru problema de Sali de spectacol:

------------------------------

* Lista de spectacole poate fi vazuta indifferent
* Login pentru administrator
* Cumparare / Rezervare

Pentru data viitoare facem alta diagrama, ne tragem entitatile, adaugam controller service repo ce mai e

Deci diferenta dintre modelul conceptual si diagrama de clase e ca in modelul conceptual avem doar entitatile ce tin de domeniul problemei

Si in diagrama de clase includem si tot ce tine de domeniul solutiei

**In final cand vom avea codul scris si cu diagramele gata, trebuie sa ne asiguram ca ce e in cod corespunde cu ce e in diagrama (deci daca ceva e gresit, refac diagrama de clase mai intai si apoi revin la cod)**

Diagrama de secventa , what is it