

Documentație proiect

1. Feature-urile aplicației

Aplicația pune la dispoziție 9 endpoint-uri REST pentru management-ul pacienților, medicilor, programărilor și așa mai departe din cadrul unei clinici medicale.

Funcționalitățile aplicației sunt:

- ✚ crearea de programări
- ✚ modificarea datei unei programări
- ✚ anularea unei programări
- ✚ adăugarea unui pacient
- ✚ adăugarea unui diagnostic pentru un pacient
- ✚ afișarea unui diagnostic
- ✚ adăugarea unui tratament pentru diagnosticul unui pacient
- ✚ adăugarea unui doctor
- ✚ listarea doctorilor din aplicație

2. Descrierea modelelor și a serviciilor

În cadrul aplicației sunt implementate următoarele modele și servicii:

✚ *Patient* – acest model reține informațiile necesare pentru un pacient înregistrat în aplicație

PatientService – în acest serviciu este implementată logica necesară pentru un pacient, cuprinzând metode pentru crearea și căutarea unui pacient după nume

✚ *Doctor* – în acest model sunt stocate proprietățile pentru un medic. Între această entitate și *Hospital*, care va fi prezentată în continuare, există o relație *one-to-many*, motiv pentru care în modelul *Doctor* am pus o proprietate de tip *Hospital*.

În serviciul *DoctorService* am implementat metodele necesare pentru funcționalitățile ce țin de medici, și anume crearea acestora, listarea lor și căutarea unui medic după nume.

- ✚ *Hospital* – în acest model se află proprietățile necesare pentru spitale, iar în serviciul asociat, *Hospital Service*, se regăsesc metode pentru a căuta un spital după nume și după id.
- ✚ *Diagnostic* – în acest model se află informațiile despre diagnosticul unui pacient. Între aceste două entități există o relație *one-to-many*, marcată prin prezența unui field de tip *Patient* în entitatea *Diagnostic*. În serviciul asociat, *DiagnosticService* sunt implementate metode pentru crearea unui diagnostic și pentru căutarea unuia după id.
- ✚ *Treatment* – acest model conține câmpuri necesare pentru stocarea informațiilor despre tratamentul asociat unui diagnostic, fiind o relație *one-to-many* între aceste două entități. În *TreatmentService* există implementată logica pentru crearea unui tratament.
- ✚ *Appointment* – în această entitate sunt stocate informații despre programări. Ea se află în relație de tip *one-to-many* cu alte două modele, *Doctor* și *Patient*. În *AppointmentService* sunt implementate metodele pentru crearea, modificarea și anularea unei programări.

3. Cerințele de business

Acestea sunt:

- ✚ crearea/modificarea/ștergerea unei programări
- ✚ adăugarea unui diagnostic
- ✚ adăugarea unui tratament pentru un diagnostic
- ✚ adăugarea de pacienți/medici
- ✚ prezentarea medicilor

4. Alte detalii

Documentația endpoint-urilor în Swagger: <http://localhost:8080/swagger-ui/>

Diagrama bazei de date:

