

Aplicația JobGetter - Documentație

Proiect

Introducere

JobGetter este o aplicație web proiectată pentru a simplifica procesul de aplicare și recrutare, conectând cei care caută un loc de muncă, angajatori și recrutori într-o platformă cuprinzătoare pentru căutarea de locuri de muncă și achiziția de talente.

Tech Stack

Frontend

- React.js
- HTML
- CSS
-

Backend

- Spring Boot (Java)
- Arhitectură N-layer
- API RESTful

Bază de Date

- MySQL
- Docker pentru virtualizarea bazei de date

Tehnologii Adiționale

- Spring Data JPA (pentru interacțiuni cu baza de date)

Cerințe Funcționale

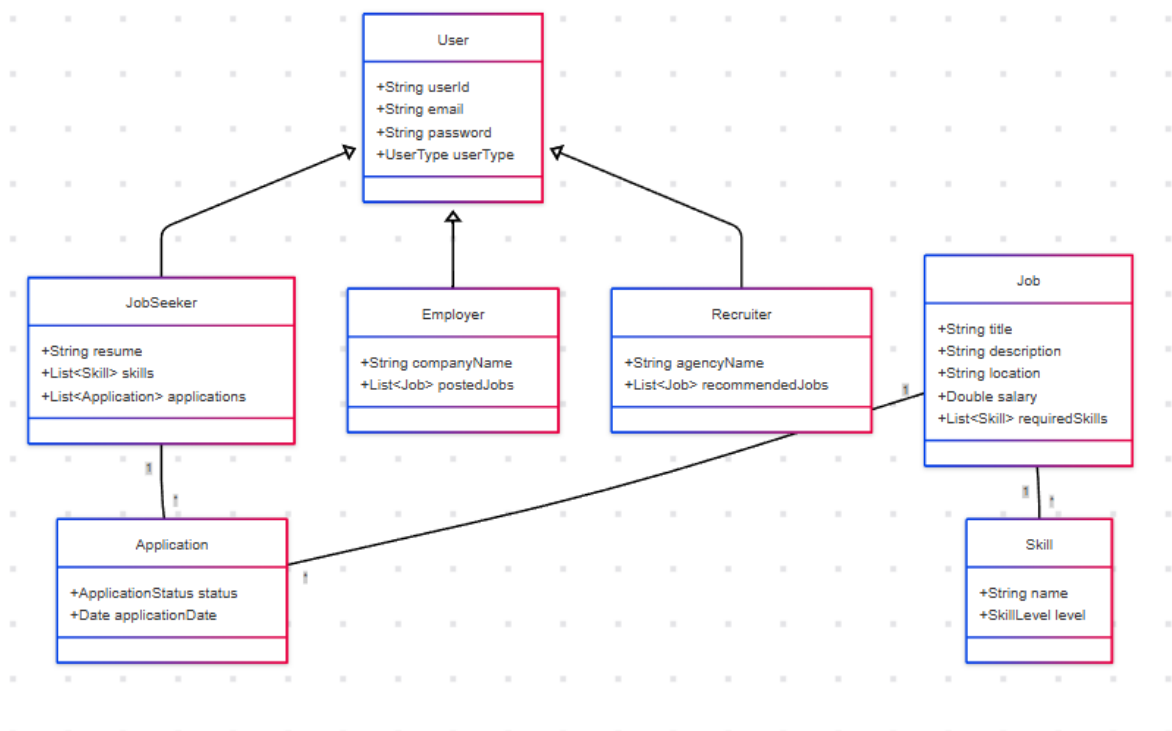
1. Înregistrare și Autentificare Utilizator

2. Căutare și Filtrare Locuri de Muncă
3. Gestionare Aplicații la Locuri de Muncă
4. Creare și Gestionare Profil
5. Postare Locuri de Muncă pentru Angajatori
6. Urmărire și Recomandare Candidați

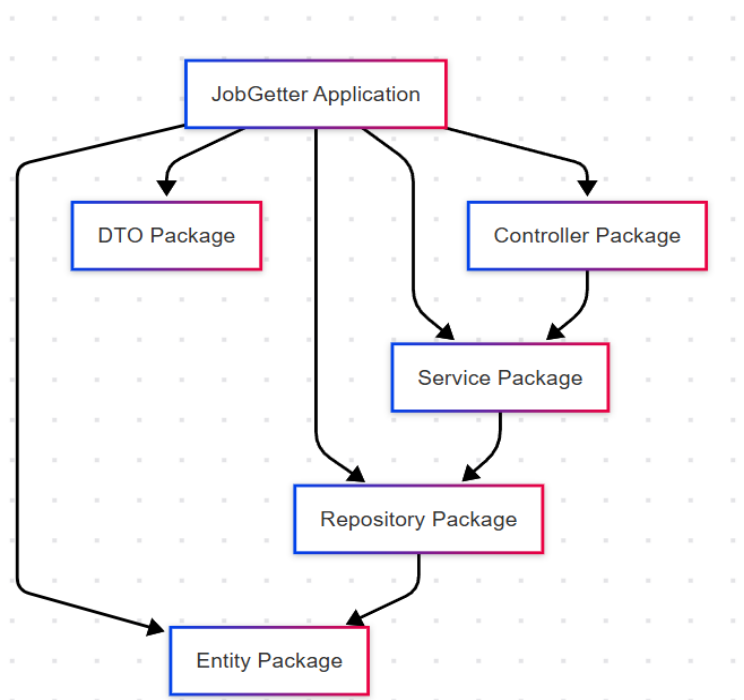
Cerințe Non-Funcționale

1. Securitate
 - a. Autentificare utilizator securizată
 - b. Protecție confidențialitate date
2. Performanță
 - a. Procesare rapidă a căutărilor și aplicațiilor
 - b. Arhitectură de sistem scalabilă
3. Utilitate
 - a. Design responsiv pentru desktop și mobil
 - b. Interfață utilizator intuitivă
4. Capacitate
 - a. Stocare a cel puțin 1000 de locuri de muncă
 - b. Suport pentru interacțiuni utilizator concurente

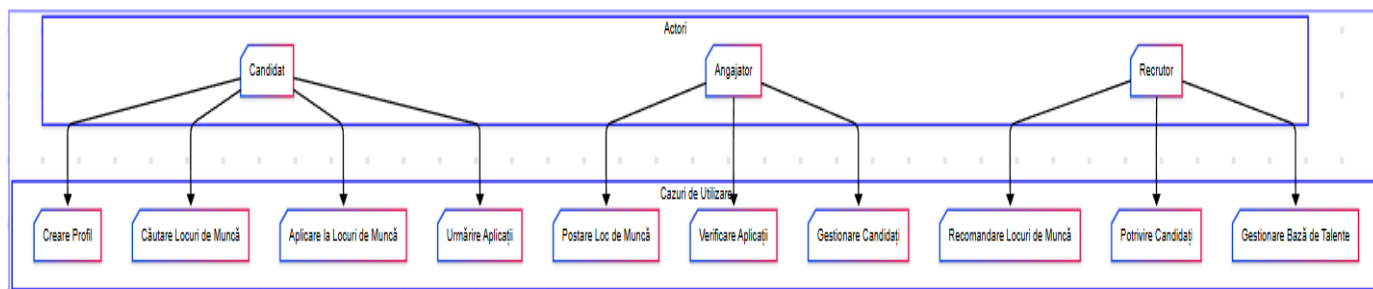
Class Diagram



Package Diagram



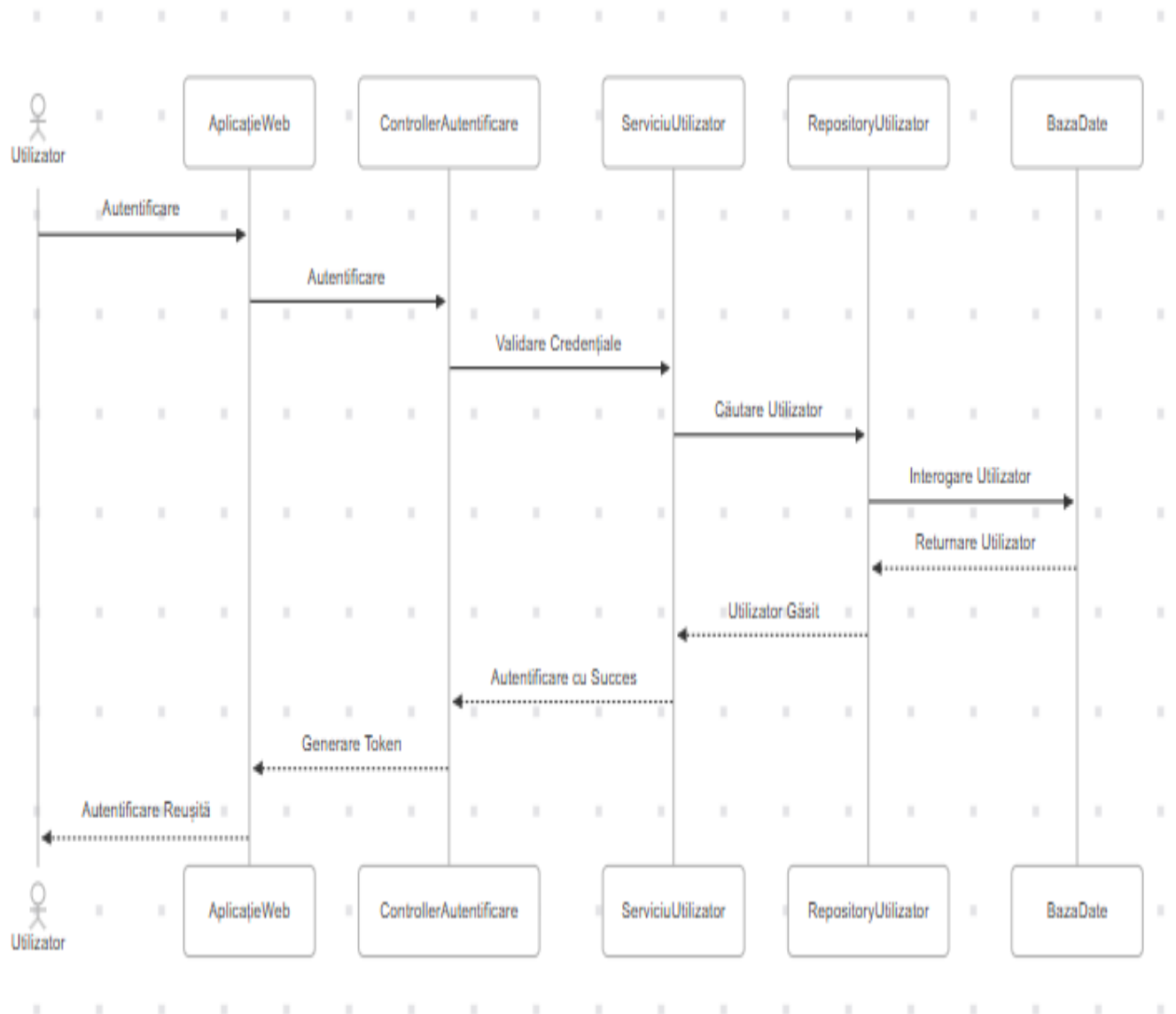
Use Case Diagram



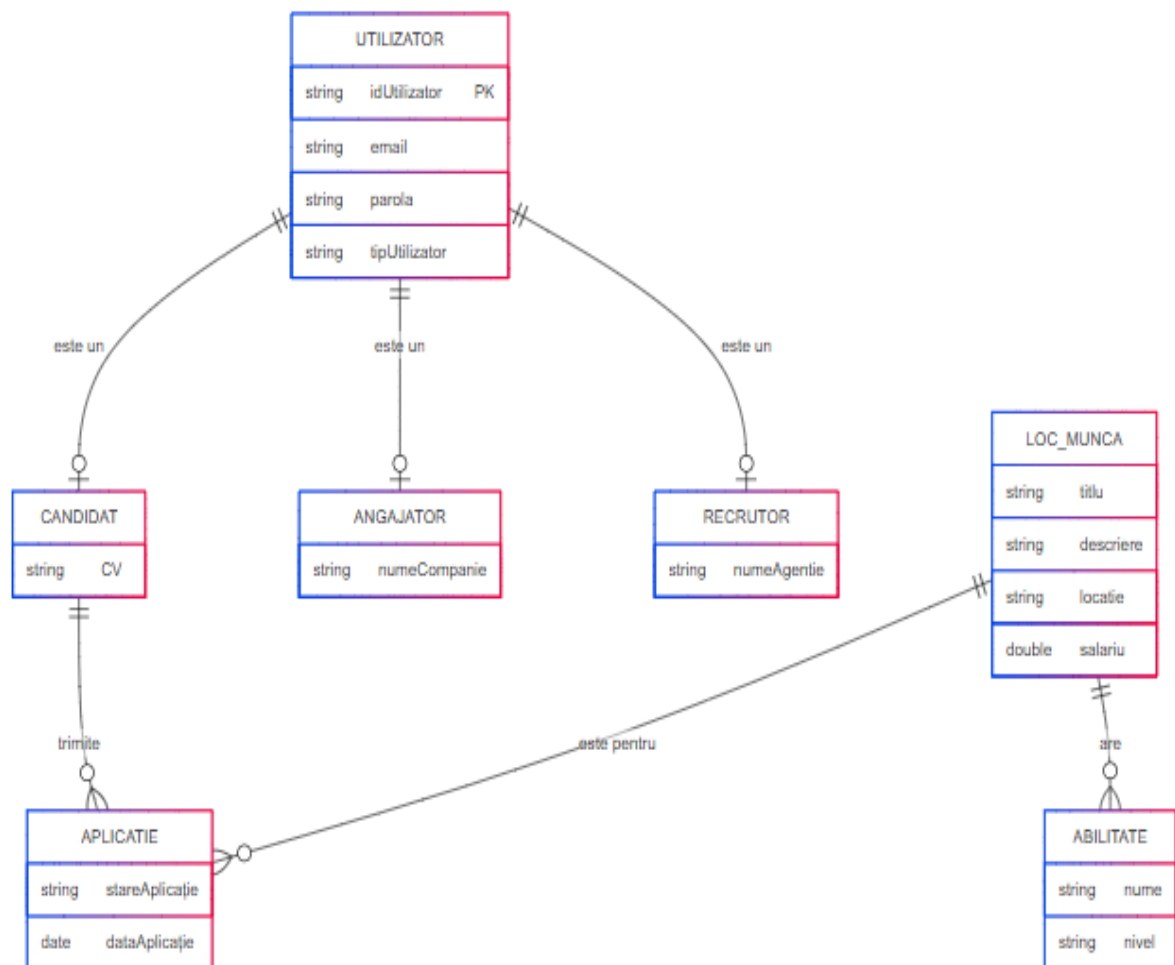
- Utilizator intră pe platformă
- Selectează rol (Candidat/Angajator/Recrutor)
- Autentificare/Înregistrare
- Candidat caută locuri de muncă
- Aplică la locuri de muncă
- Angajator postează job-uri
- Sistemul procesează și notifică aplicațiile
- Utilizatorii pot comunica

- Finalizare proces aplicare/recrutare

Sequence Diagram



DB Diagram

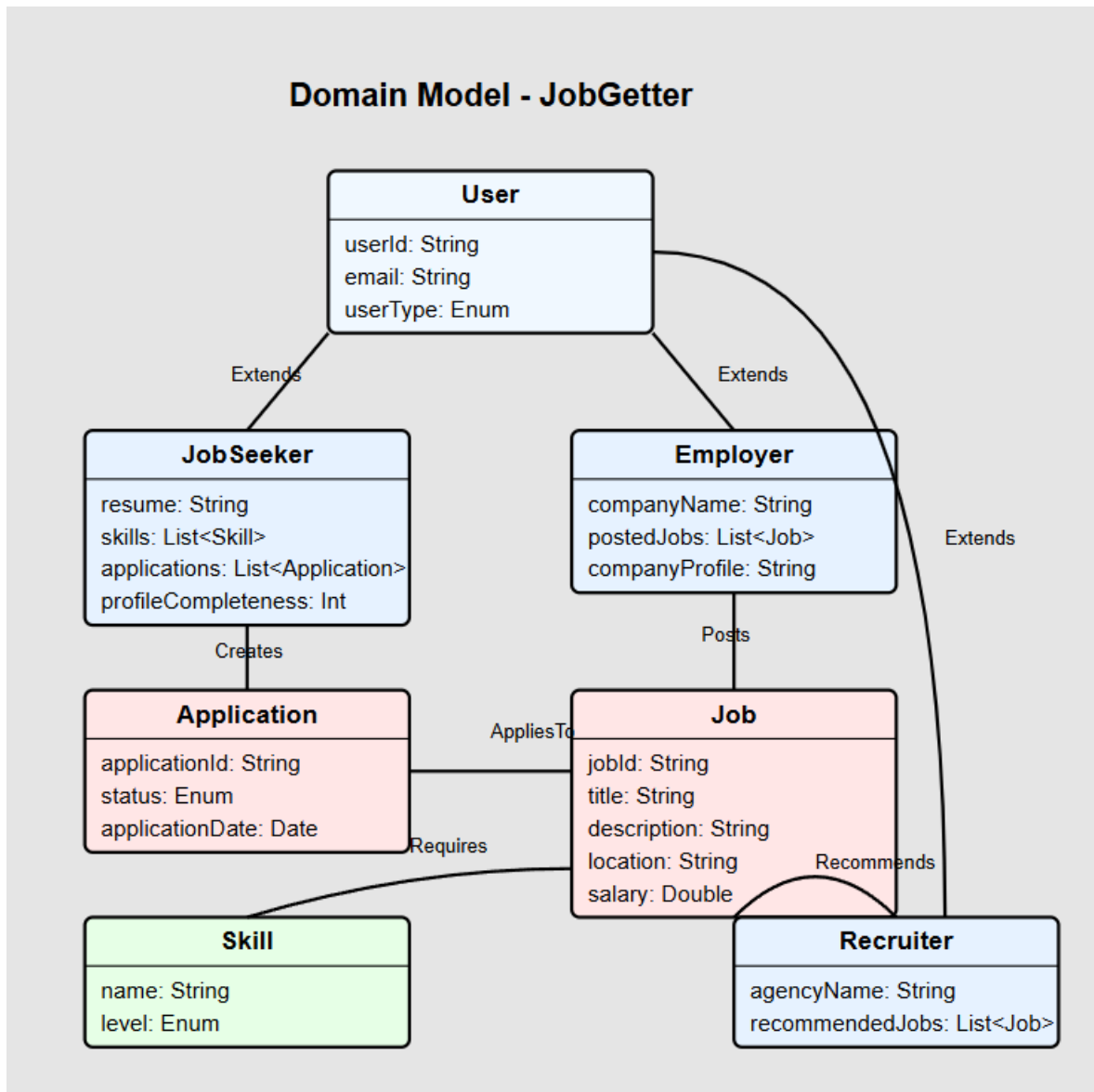


PROJECT DELIEVERABLE - PART 2

Domain Model

- prezintă entitățile principale din sistem și relațiile dintre ele:

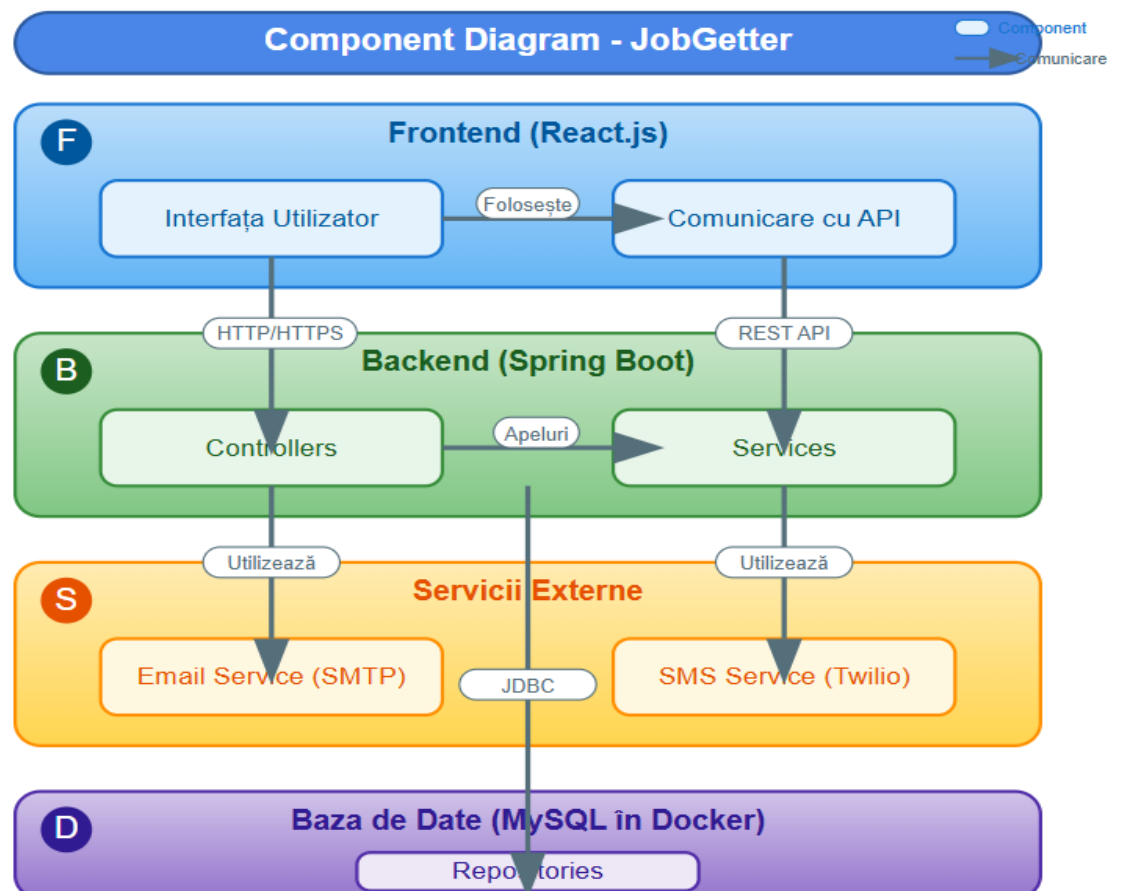
- Reflectă structura conceptuală a aplicației JobGetter
- Include entitățile: User, JobSeeker, Employer, Recruiter, Job, Application și Skill
- Ilustrează relațiile între aceste entități (extends, creates, applies to, etc.)
- Prezintă atributele principale ale fiecărei entități



Component Diagram

- arată componentele software ale sistemului și relațiile dintre ele:

- Organizat pe 4 straturi principale cu design modern și codificare prin culori:
 - Frontend (albastru): interfața utilizator și comunicarea cu API
 - Backend (verde): controllers și services care procesează logica aplicației
 - Servicii Externe (galben/portocaliu): Email Service (SMTP) și SMS Service (Twilio)
 - Baza de Date (violet): repositories pentru gestionarea accesului la date
- Ilustrează fluxurile de comunicare între componente:
 - Frontend comunică cu Backend prin HTTP/HTTPS și REST API
 - Controllers fac apeluri către Services pentru procesarea datelor
 - Services utilizează serviciile externe pentru notificări
 - Services se conectează la baza de date prin JDBC
- Evidențiază separarea clară a responsabilităților între componentele aplicației
- Folosește iconițe și etichete vizuale pentru a facilita înțelegerea structurii



Deployment Diagram

- Ilustrează mediul de execuție fizic al sistemului:
- Client environment (browser web și aplicația React)
- Application server cu Spring Boot
- Notification services (Email cu SMTP/Gmail și SMS cu Twilio)
- Database server cu MySQL în container Docker
- Conexiunile între componente (HTTP/HTTPS, REST API, JDBC)

