



**UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA**

FACULTATEA: Automatică și Calculatoare

SPECIALIZAREA: Calculatoare Romana

DISCIPLINA: Sisteme Distribuite

ASSIGNMENT 2:

**Asynchronous Communication Sensor Monitoring
System and Real-Time Notifications**

Îndrumător de laborator:

Claudia-Daniela Pop

Student:

Filip-Dud Mihai-Bogdan

1. Specificația proiectului

Aplicația are a scop implementarea unei platforme web ce va administra activitățile realizate de doctori, asistenți medicali și pacienți. Această aplicație va putea fi accesată de toate cele 3 tipuri de utilizatori, oferind pentru fiecare dintre aceștia o experiență diferită în urma unui proces de login.

Doctorul poate efectua operații de tip CRUD pe rolurile de pacienți și asistenți, CRUD pe lista de medicamente, să a dauge rețete pacienților și să a atribuie un a sistent către un pacient.

Asistentul își poate vedea detaliile contului și, de asemenea, lista de pacienți pe care îi are în îngrijire.

Pacienții pot vedea pe pagina a ferentă lor detalii personale, cât și lista de prescripții medicale pe care le au primite de la doctor.

Securitatea aplicației prevede ca niciun rol să nu poată vedea pagina a ltui rol fără a fi a utentificat în preakabil cu un cont a ferent rolului respectiv.

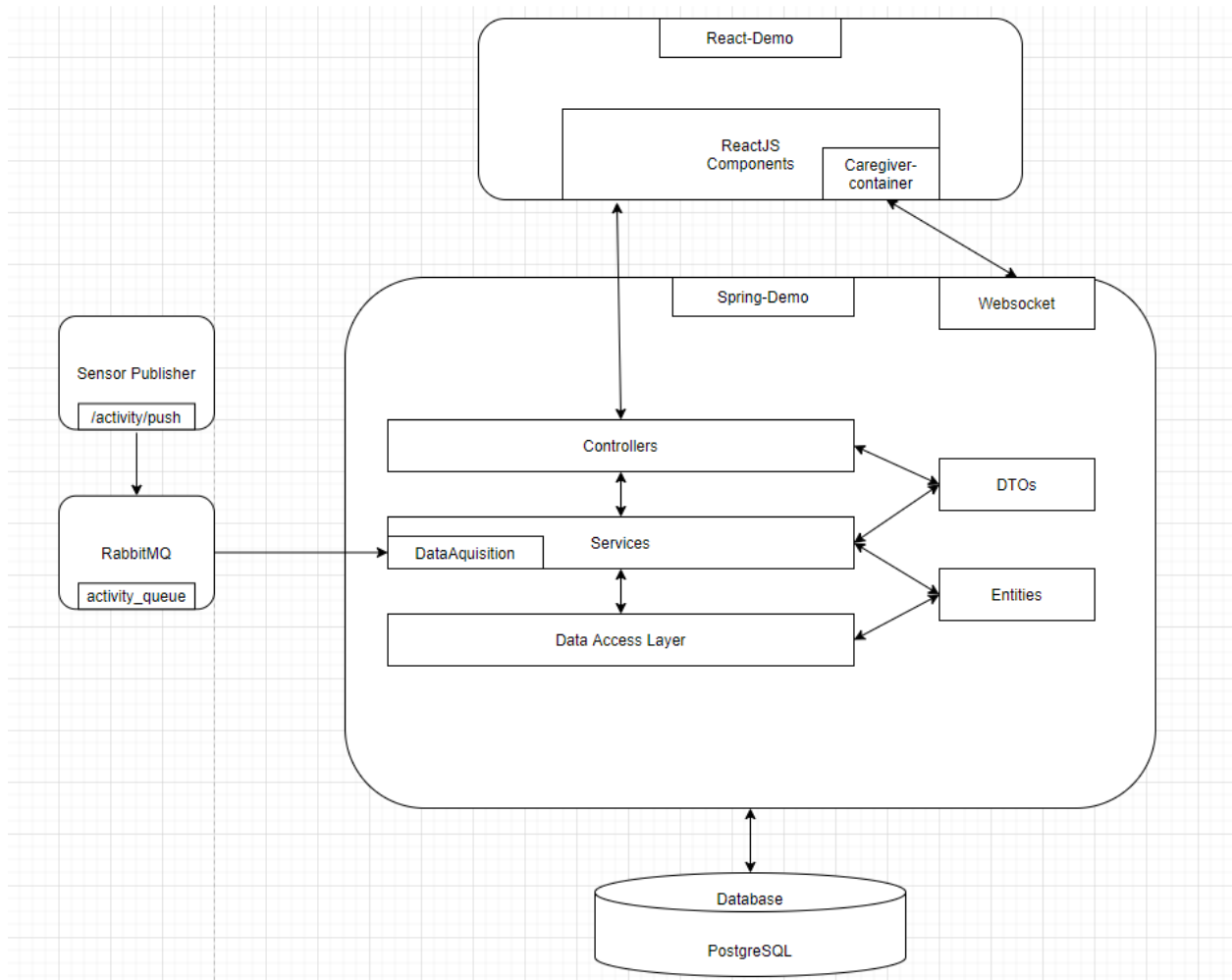
Pentru a implementa un sistem de monitorizare a pacienților, se va alcătui o listă a activităților desfășurate de fiecare pacient în timpul zilei care vor fi trimise printr-o coadă către aplicația principală pentru a fi prelucrate, iar în caz de o activitate suspectă, aplicația va avertiza caregiverul a ferent pacientului.

2. Arhitectura conceptuală a sistemului distribuit

Aplicația este structurată pe 3 niveluri principale:

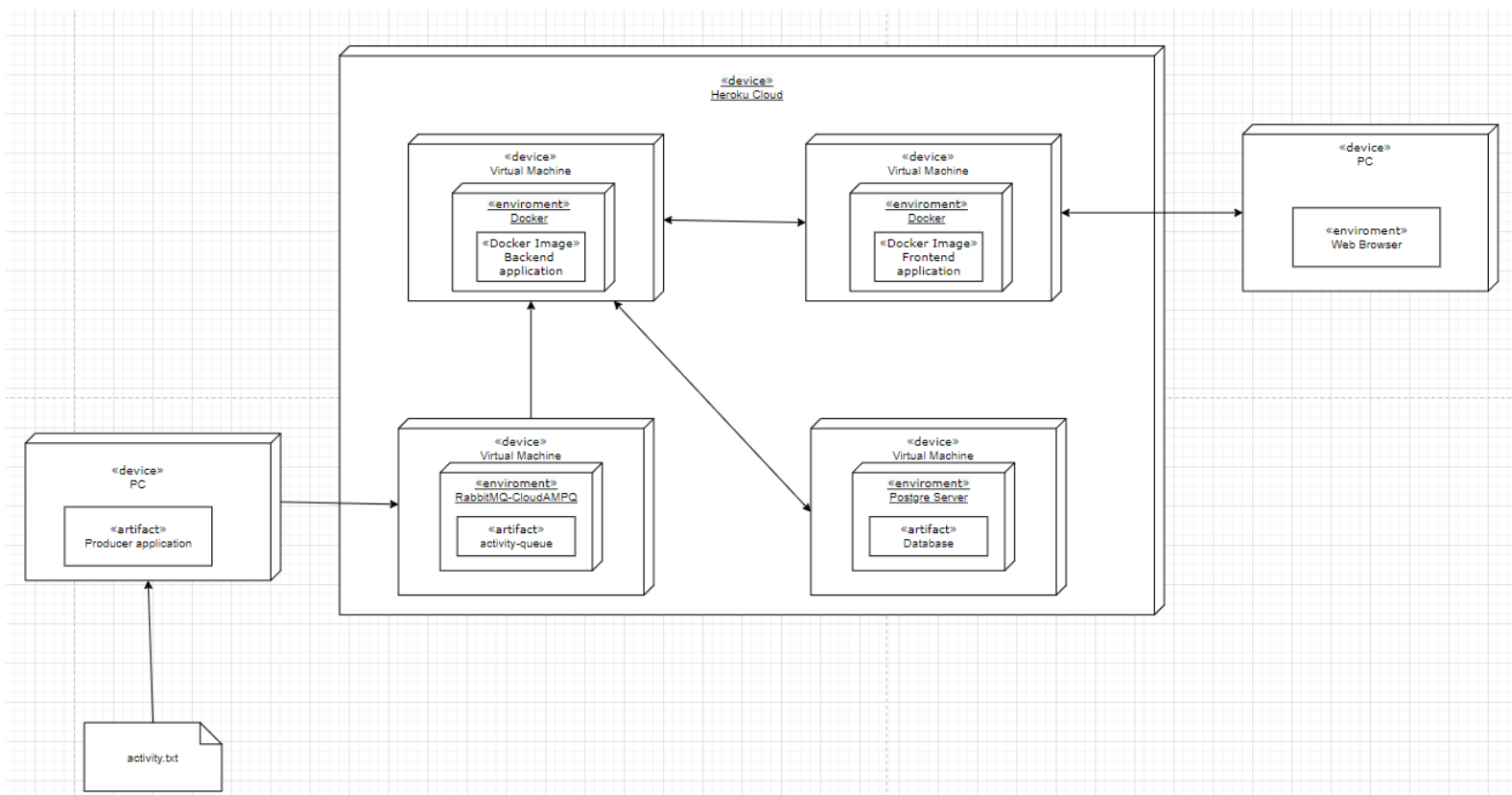
- Front-end-ul: reprezintă unealta de interacțiune cu utilizatorul a aplicației. Această componentă a fost implementată cu ajutorul framework-ului ReactJS ce pune la dispoziție o multitudine de unelte pentru crearea de pagini web.
- Back-end-ul: această componentă se ocupă cu primirea datelor de la front-end prin intermediul REST-API și procesarea lor pentru a putea fi stocate mai apoi într-o bază de date cu ajutorul JPA Repository. Pentru implementarea sa, am utilizat framework-ul Spring-Boot ce rulează aplicația pe un server Tomcat integrat.
- Baza de date: aceasta stochează toate datele necesare sistemului primite de la back-end și le pune la dispoziție acestuia atunci când are nevoie de ele. Am utilizat PostgreSQL ca și platformă de gestiune a bazei de date relațională.

La acestea, în această temă se a daugă o aplicație terță pentru a simula recepționarea datelor de la senzori și trimiterea acestora prin intermediul unor mesaje manageriate de o coadă de activități implementată cu RabbitMQ. Această coadă va fi descărcată în aplicația de backend unde pentru fiecare activitate se verifică îndeplinirea condițiilor și inserarea acesteia în baza de date. În cazul observării unor neregularități, backendul va notifica caregiverul corespunzător pacientului în aplicația de frontend prin intermediul mesajelor a sincrone transmise prin WebSockets.



4. Diagrama de deployment

Pentru a prezenta structura fizică a aplicației, am realizat următoarea diagramă UML de deployment:



5. Considerații generale de utilizare

Aplicația a fost amplasată pe un pipeline de Continuous Integration and Deployment ce are ca rezultat expunerea acesteia în cloud, pe platforma Heroku. Pentru a fi accesat:

- Back-end-ul: <https://mihaifilipdud-backend-sd.herokuapp.com/>
- Front-end-ul: <https://mihaifilipdud-frontend-sd.herokuapp.com/>

Logarea în aplicație ca și doctor se poate face cu username-ul: mihai.f și parola: mihai.f.

Operații pentru rolul de doctor:

- CRUD-uri pe tabele se realizează prin apăsarea butoanelor aferente
- Asignarea unui asistent la un pacient se realizează prin selectarea radio-buttonurilor aferente celor 2 și mai apoi apăsarea pe butonul de Assign Caregiver
- Pentru a crea un medication plan pentru un pacient, trebuie să se selecteze un pacient apăsând pe radio-button, să se selecteze medicationurile dorite cu ajutorul checkbox-urilor și apoi se apasă pe butonul de Add Medication Plan pentru a completa detaliile necesare.

Pentru a trimite datele despre activități înspre coada, se pornește aplicația de producer și se accesează endpointul: `http://localhost:9292/activity/push`.