DISTRIBUTED SYSTEMS 2020-2021

Assignment 3 Remote Procedure Call

**Medication Dispenser**

Pop Cristian

Grupa: 30244

# Requirements

Se presupune ca fiecare pacient are un dispenser inteligent de pastille care poate fi programat in functie de planul medicational prescris de catre doctor. Dispenser-ul de pastile va notifica pacientul cand trebuie luat un medicament si cand un pacient nu a luat medicamentul in intervalul de timp corespunzator. Acesta poate comunica cu serverul folosind Java remote method invocation. Sa se creeze o aplicatie client care reprezinta dozatorul de pastile si o aplicatie server pentru a obtine/procesa planul medicational dupa urmatoarele reguli:

1. Dozatorul de pastile poate descarca planul medicational de pe server.
2. Daca butonul este apasat in intervalul de timp dat, medicamentul este marcat ca si luat si mesajul corespunzator este trimis la server, iar in cazul in care butonul nu este apasat in intervalul dat, medicamentul este marcat ca nefiind luat si mesajul corespunzator este trimis, de asemenea, la server pentru a-l informa.

# Functional requirements

* + Aplicatia clientului pentru dozatorul de pastile este in timp real.
  + Aplicatia client pentru dozatorul de pastile afiseaza o lista de medicamente care sunt ale id-ul trimis de server (ora curenta este in intervalul de administrare),
  + Am introdus in aplicatia Desktop, un timer in timp real.
  + Aplicatia poate descarca medicationPlanul la o ora stabilita

# Conceptual architecture of the distributed system

Aplicatia este formata din doua parti, cea de Backend si partea de Frontend. Pentru partea de Backend s-a folosit aceeasi aplicatie de Backend ca pana acum, cu mentiunea ca s-au adus imbunatatiri introducand partea de remote method invocation, iar pentru partea de Frotend s-a creat o aplicatia noua, aceasta fiind una desktop cu ajutorul acesteia putand fi verificate functionalitatile adaugate in maniera de “Remote Procedure Call”. Cele doua aplicatii sunt ambele aplicatii in Java, partea de Backend fiind o aplicatie Web API, iar cea de Frontend fiind una Java Swing.

Legatura dintre cele doua aplicatii se face cu ajutorul framework-ului Hessian care permite un standard bazat pe HTTP. Cu ajutorul acestui framework se realizeaza comunicarea dintre cele doua aplicatii. Relatia dintre cele doua este una de client-server. Astfel, in momentul in care aplicatia desktop doreste sa primeasca planul medicational de la un anumit pacient, aceasta ii trimite serverului, adica aplicatiei de Backend o cerere pentru care se doreste returnarea planului medicational. In momentul in care aceasta trimite id-ul catre server, acesta il receptioneaza si ii trimite clientului, adica aplicatiei desktop, planul medicational care exista in baza de date. De asemenea se poate „hardcoda” o ora la care sa fie descarcat adica afisat medicationPlanu.Adica daca se doreste descarcarea medicationPlanului peste 5 min de la ora curenta si in acest timp se adauga un nou MedicationPlan atunci aplicatia Desktop va fi actualizata,medicationPlanul creat in acest interval se va putea vedea.

UML Deployment Diagram

