Integrated Medical Monitoring Platform for Home-care assistance

Student: Cristian – Ioan Blaga
Group: 30443

1. Requirements Analysis

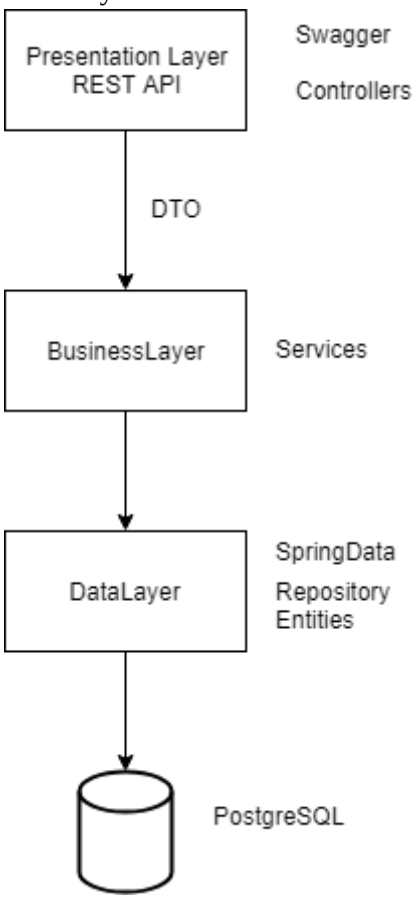
This assignment represents the first module of the distributed software system "Integrated Medical Monitoring Platform for Home-care assistance" that represents the final project for the Distributed Systems course.
The module consists of an online platform designed to manage patients, caregivers and medication. The system can be accessed by three types of users after a login process: doctor, patient and caregiver. The doctor can perform CRUD operations on patient (defined by ID, name, birth date, gender, address, medical record) and caregiver (defined by ID, name, birth date, gender, address, list of patients taken care of) accounts and on the list of medication (defined by ID, name, list of side effects, dosage) available in the system. Furthermore, the doctor can create a medication plan for a patient, consisting of a list of medication and intake intervals needed to be taken daily, and the period of the treatment. The patients can view their accounts and their medication plans. The caregivers can view their associated patients and the corresponding medication plans.

2. System Design

2.1 Architectural Description Server side

The Architectural Pattern I have used is the Layered Architecture Design Pattern for the server side. Each layer represents a certain step of the operation flow. The layers are:

- The Presentation Layer – This layer consists of RestControllers. The access to these controllers



is restricted with help from Spring Security.

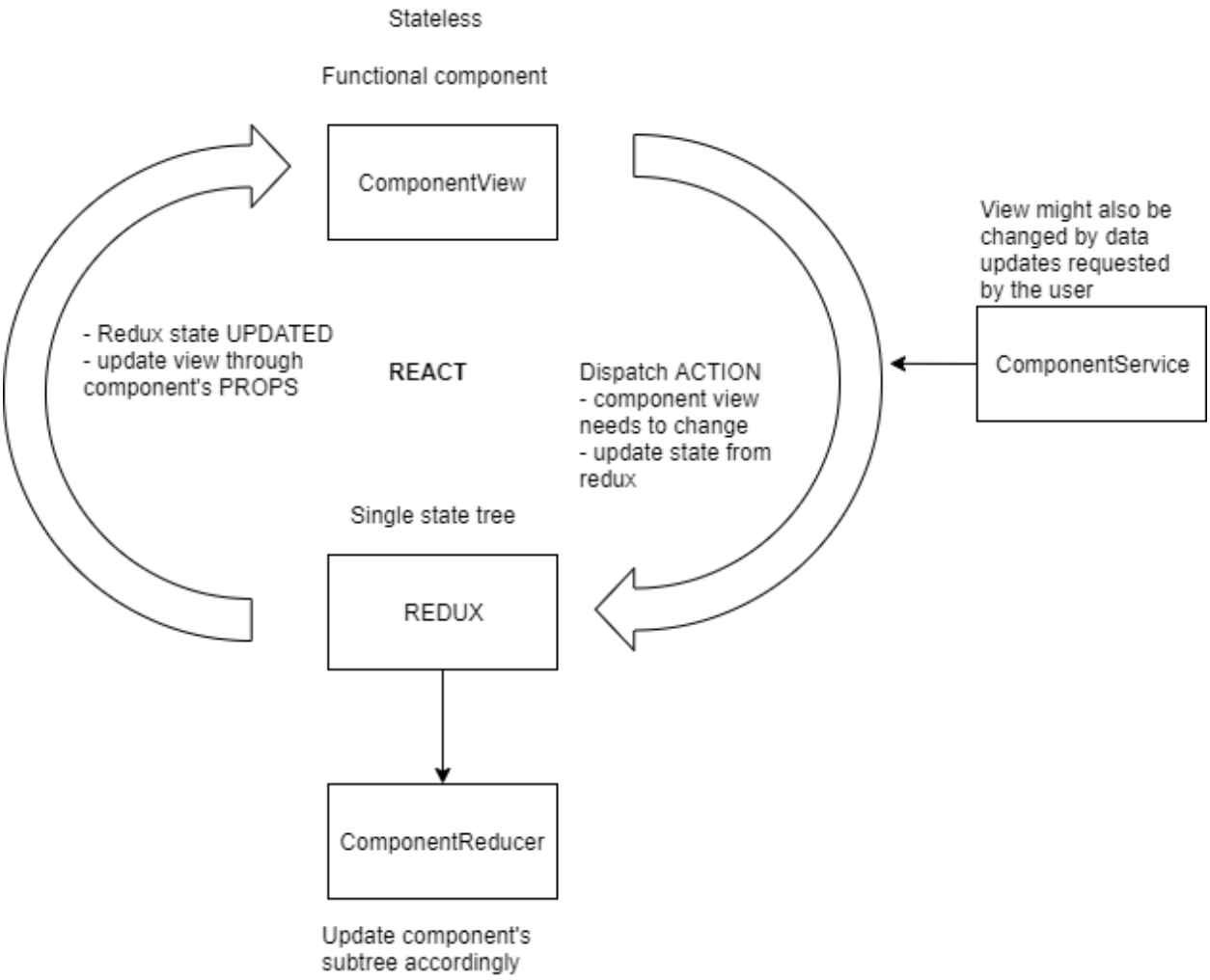
- The Business Layer consists of Service classes. The communication between the BL and the PL

is done through DTO (data transfer objects).

- The Data Layer is represented by the model classes and the repository classes. The model classes map the tables from the database into objects. The repositories define methods and queries to interact with the database. The implementation has been done with Spring Data.

Each layer is closed and the operation flow goes successively, layer to layer, top to bottom, from PL to DL. No layer can be skipped in the operation flow.

2.2 Architectural Description Client side



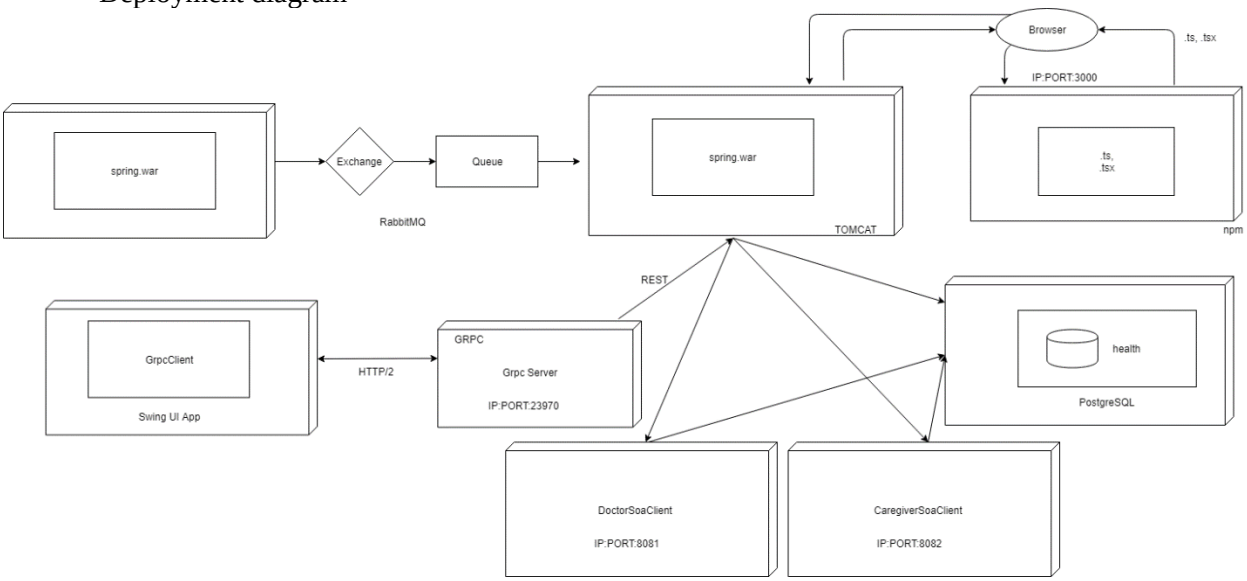
The architecture of the Client side revolves around the fact we are using stateless components together with Redux as a single state tree. Because of this, we have as components the following:

- ComponentView – the data display the user sees and interacts with

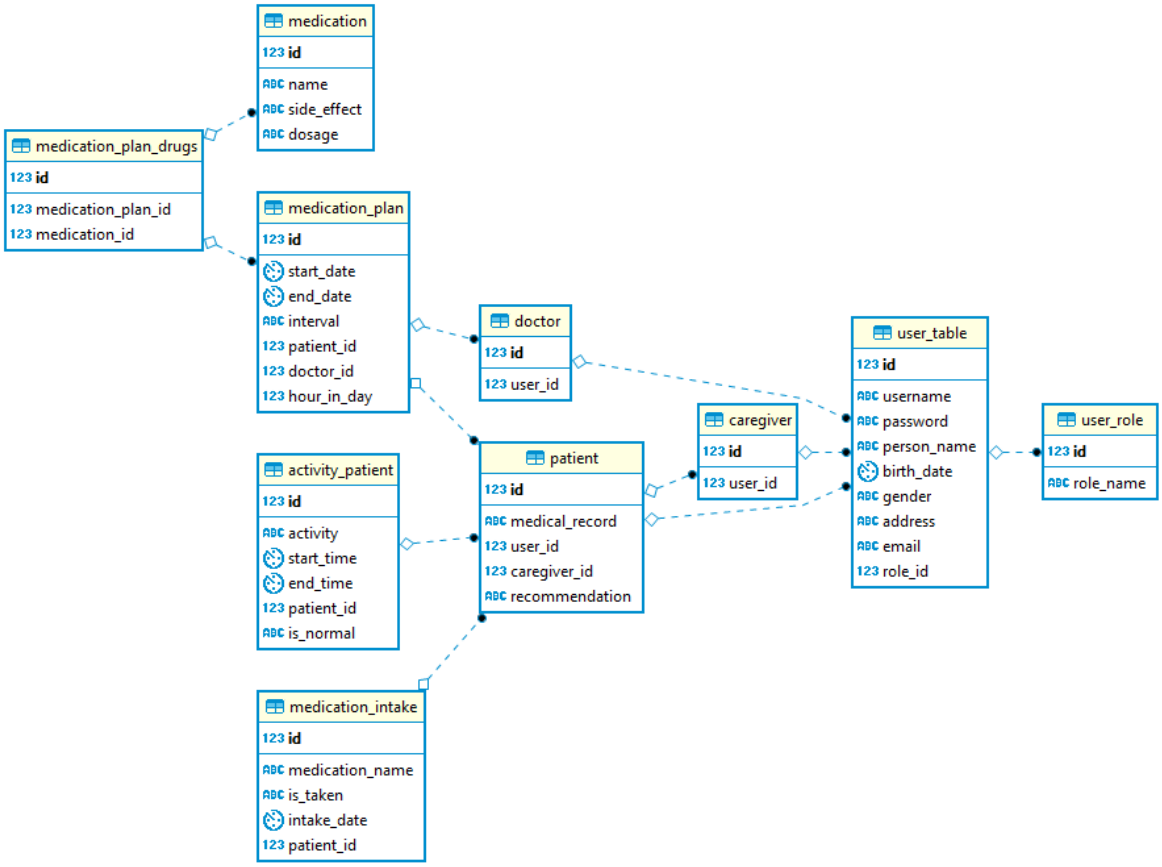
- ComponentService – the service responsible for handling data updates and communication with the server
- Redux – state tree manager
- ComponentReducer – based on action type, updates the state tree accordingly

2.3 Diagrams

Deployment diagram



DB diagram


medication
id
name
side_effect
dosage
medication_plan_drugs
id
medication_plan_id
medication_id
medication_plan
id
start_date
end_date
interval
patient_id
doctor_id
hour_in_day
doctor
id
user_id
user_table
id
username
password
person_name
birth_date
gender
address
email
role_id
caregiver
id
user_id
user_role
id
role_name
activity_patient
id
activity
start_time
end_time
patient_id
is_normal
patient
id
medical_record
user_id
caregiver_id
recommendation
medication_intake
id
medication_name
is_taken
intake_date
patient_id