Technical Solution Description

Online Healthcare Service

Author: Alexey Yefremov

T-Systems JavaSchool, 2020

Contents

1. Introduction

Healthcare Service application was designed for the employees of rehabilitation medical centers. Its main goal is to prevent unnecessary paperwork and reduce bureaucracy in daily work of medical workers.

2. Tech Stack

- Spring 5.2.5
- Spring Security 5.3.0
- MySQL 8
- Hibernate 5.4.12
- C3P0 0.9.5
- JSP, JSTL, CSS
- JavaScript, JQuery
- Log4j 2.13.1
- JUnit 5.6.2
- Mockito 3.3.3
- ActiveMQ 5.15.12
- Lombok 1.18.12
- ModelMapper 2.3.6
- Jackson 2.10.3
- Maven
- Docker
- Git

3. Functionality

Doctors can create, add and discharge patients. Make appointments, control dose and time pattern of receipt.

Nurses can complete or cancel events.

All changes and current events are sent to second application, which displays this info on special events board.

4. Database

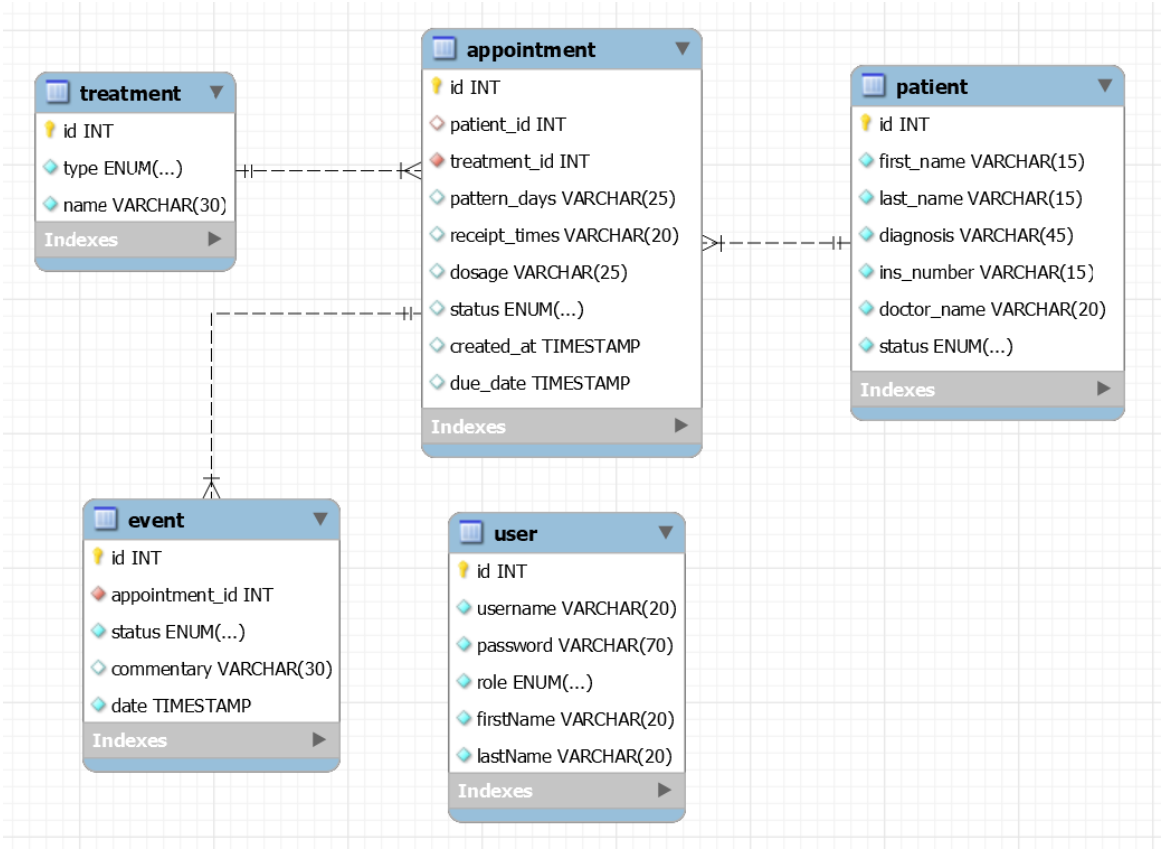


Table relationships:

appointment ManyToOne ***patient***

treatment OneToOne ***appointment***

event ManyToOne ***appointment***

5. Project Structure

Main application packages:

tsystems.rehab.config – contains hibernate and web configuration

- ➔ *tsystems.rehab.config.security* – security configuration, custom implementation of user principal and success handler

tsystems.rehab.controller – admin, patient, treatment, appointment and event controllers

tsystems.rehab.dao – user, appointment, event, treatment and patient DAO implementations

tsystems.rehab.dto – contains DTO objects

tsystems.rehab.entity – entity objects

tsystems.rehab.messaging – JMS configuration, implementation of Message Producer and Message Listener

tsystems.rehab.mapper – custom implementations of mapper objects with help of ModelMapper library

tsystems.rehab.service – services

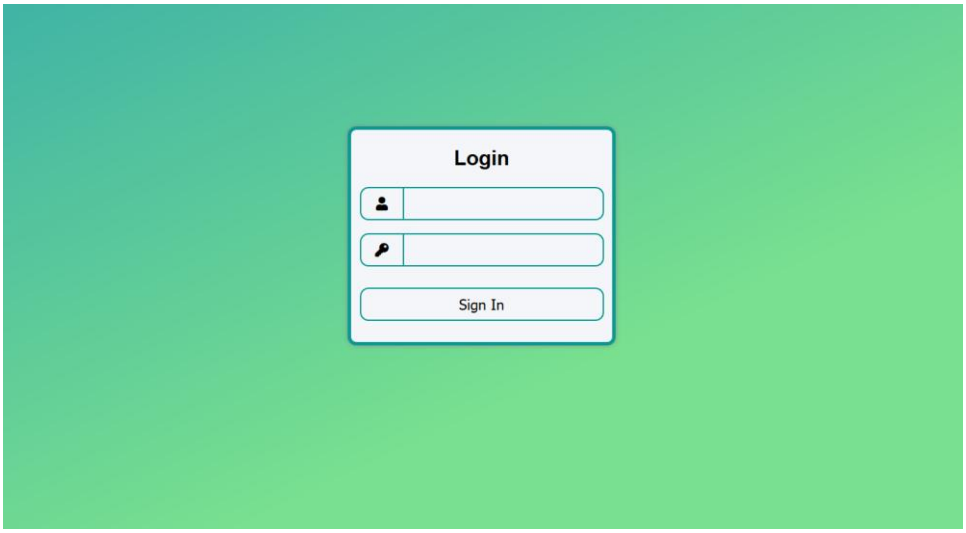
6. Technical Solutions

Main application was built on Spring MVC framework. Also Security, JMS, ORM and Test modules were used.

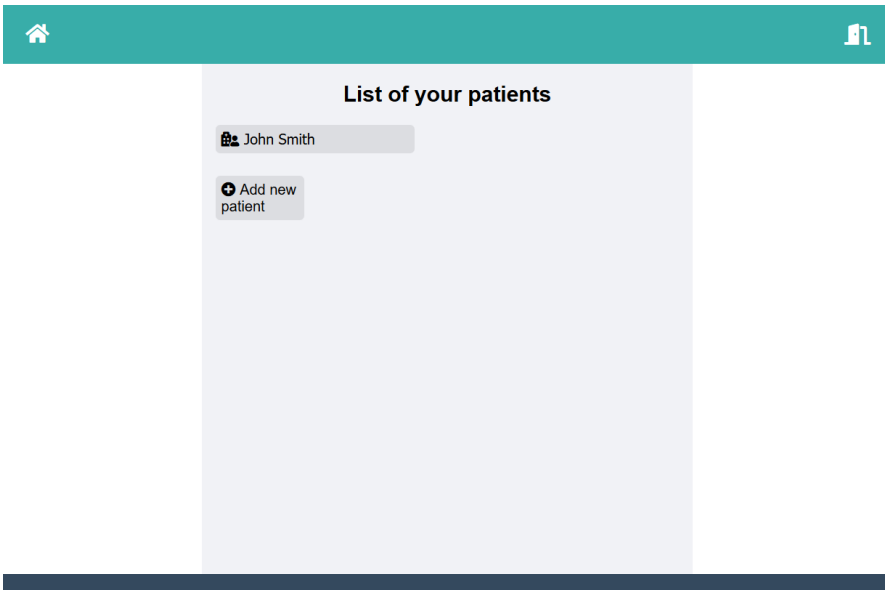
Second application was built on JSF framework. Its main functionality was achieved by using Stomp JS library, which helps us to easily connect to ActiveMQ broker via WebSockets protocol.

7. User Interface

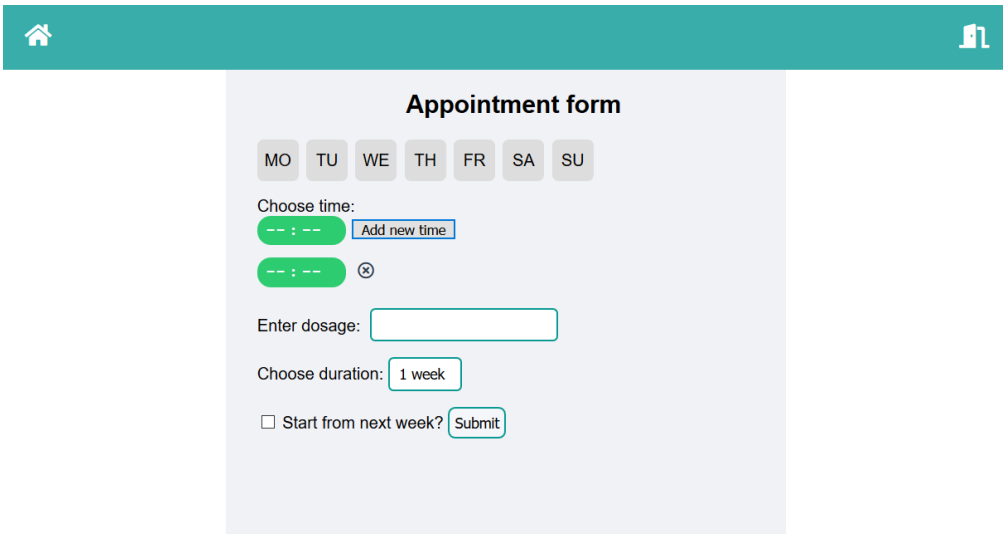
Login page:



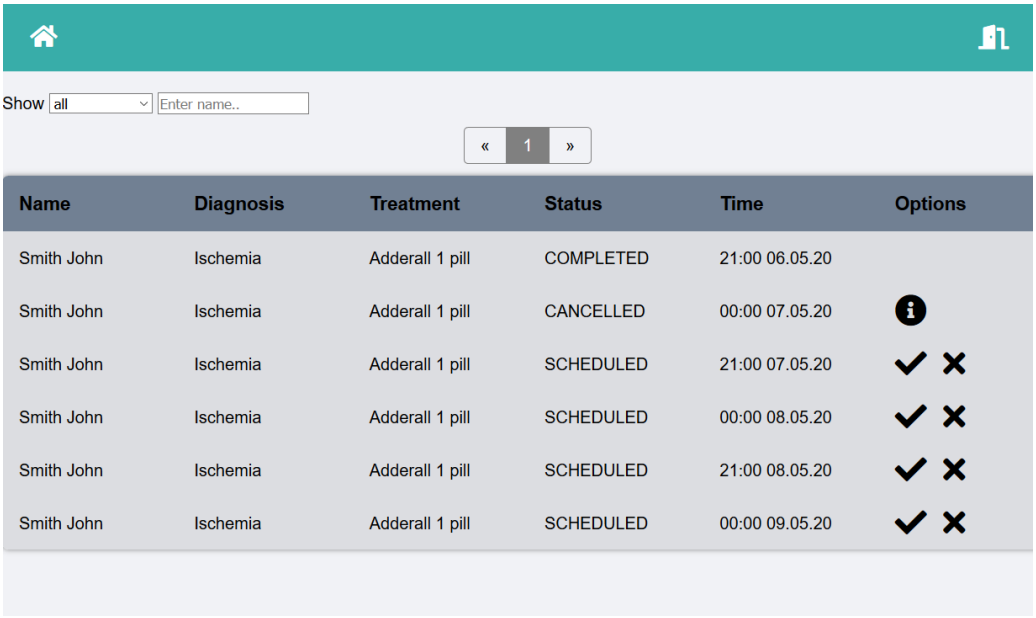
Doctor Main Page:



Appointment Creation Form:



Nurse Main Page (pagination.js library was used for pagination implementation):



8. Testing

Service and DAO layers were fully covered by tests with JUnit 5. H2 Database was used for integration testing.

9. Deployment

Application was built with Maven and later deployed on docker containers with help of docker-compose.