

# Technical Solution Description

## Online Healthcare Service

Author: Alexey Yefremov  
T-Systems JavaSchool, 2020

## Contents

1. Introduction
2. Tech Stack
3. Functionality
4. Database
5. Project Structure
6. Technical Solutions
7. User Interface
8. Testing
9. Deployment

## 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

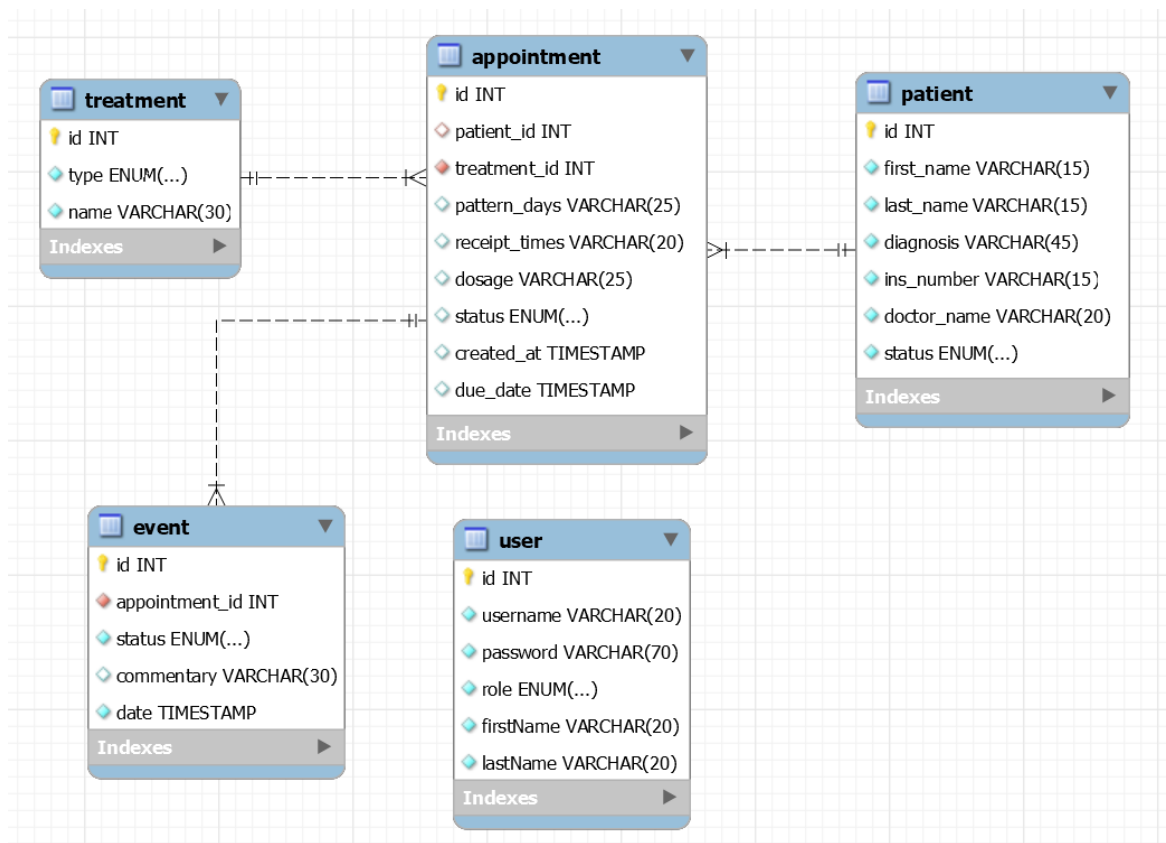


Table relationships:

***appointment*** ManyToOne ***patient***

***treatment*** OneToOne ***appointment***

***event*** ManyToOne ***appointment***

## 5. Project Structure

Main application packages:

*tssystem.rehab.config* – contains hibernate and web configuration

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

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

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

*tssystem.rehab.dto* – contains DTO objects

*tssystem.rehab.entity* – entity objects

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

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

*tssystem.rehab.service* – services

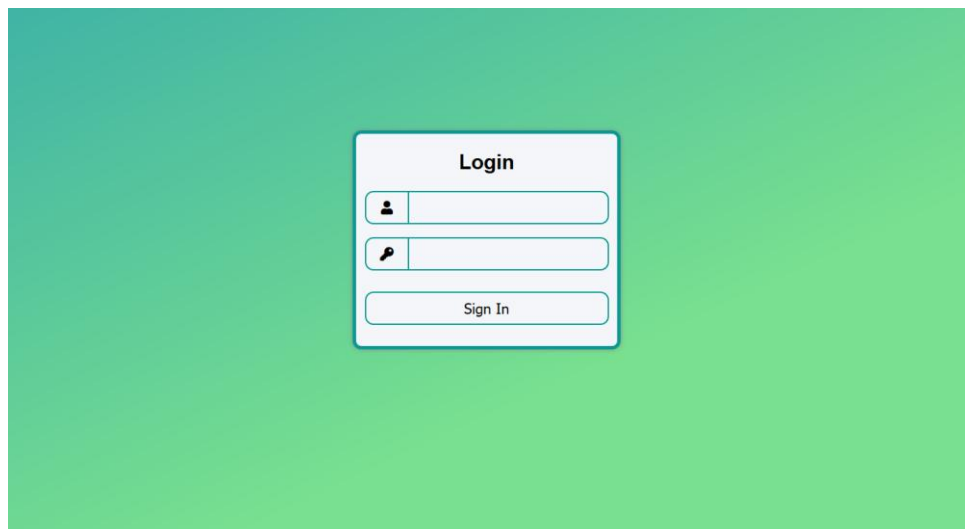
## 6. Technical Solutions

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

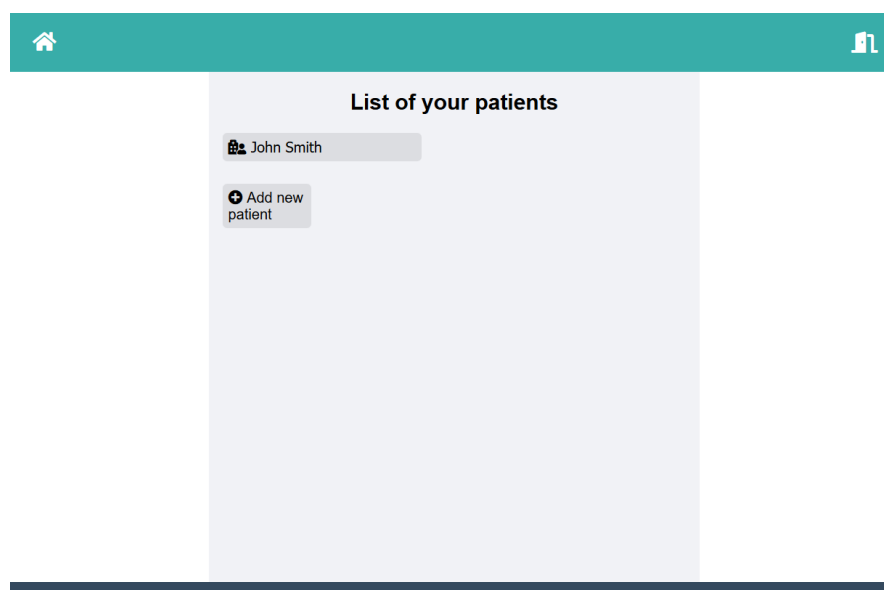
Second application was built on JSF framework. Its main functionality was achieved by using Stomp JS library, which helps us to easily connect to ActiveMQ broker via WebSockets protocol.

## 7. User Interface

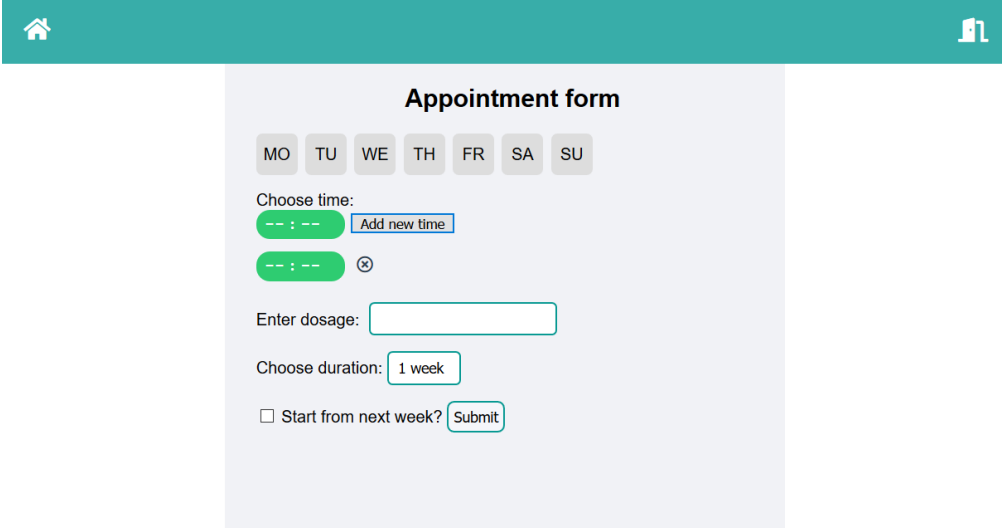
Login page:



Doctor Main Page:

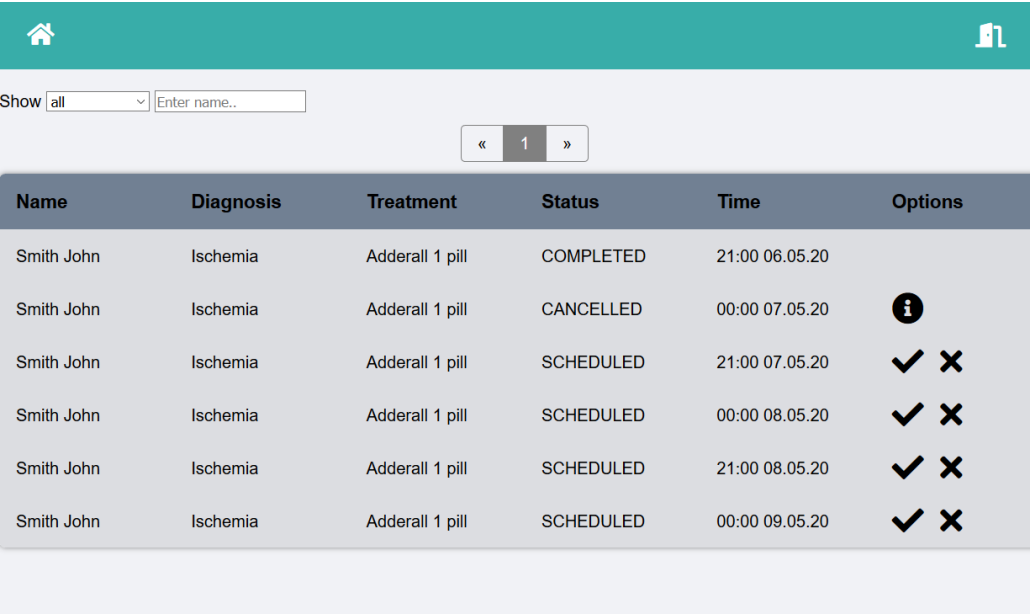


## Appointment Creation Form:



The screenshot shows a web application interface for creating an appointment. At the top is a teal header bar with a home icon on the left and a user profile icon on the right. Below the header is a light gray box titled "Appointment form". Inside this box, there are several input fields and buttons: a row of day-of-the-week buttons (MO, TU, WE, TH, FR, SA, SU); a "Choose time:" section with two green time pickers and an "Add new time" button; an "Enter dosage:" text input field; a "Choose duration:" section with a "1 week" button; and a checkbox labeled "Start from next week?" next to a "Submit" button.

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



The screenshot shows a web application interface for a nurse's main page. It features a teal header bar with a home icon on the left and a user profile icon on the right. Below the header is a light gray box containing a search bar with a "Show" dropdown menu set to "all" and a text input field labeled "Enter name..". Below the search bar is a pagination control showing "« 1 »". Below the search bar is a table with the following data:

Name	Diagnosis	Treatment	Status	Time	Options
Smith John	Ischemia	Adderall 1 pill	COMPLETED	21:00 06.05.20	
Smith John	Ischemia	Adderall 1 pill	CANCELLED	00:00 07.05.20	
Smith John	Ischemia	Adderall 1 pill	SCHEDULED	21:00 07.05.20	
Smith John	Ischemia	Adderall 1 pill	SCHEDULED	00:00 08.05.20	
Smith John	Ischemia	Adderall 1 pill	SCHEDULED	21:00 08.05.20	
Smith John	Ischemia	Adderall 1 pill	SCHEDULED	00:00 09.05.20	

## 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.