

Университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники

ОТЧЕТ ПО ПРОЕКТИРОВАНИЮ И РАЗРАБОТКЕ ДИЗАЙН ПРОЕКТА

Курса «Рефакторинг баз данных и приложений»

Санкт-Петербург 2022

В данном спринте было проведено следующее:

Создана новая роль Patient, теперь пациент может зайти в систему и просматривать свои назначения (своего рода личный кабинет

Изменения в коде.

Backend:

Создана новая роль Patient и добавлена в конфигурацию Spring Security

```
public enum Position implements GrantedAuthority {  
    DOCTOR,  
    NURSE,  
    PATIENT;  
  
    @Override  
    public String getAuthority() {  
        return name();  
    }  
}
```

Создан запрос для получения данных о назначениях конкретного пациента:

```
@PostMapping("patient")  
public ResponseEntity getData(@RequestBody String patientName) {  
    List<AppointmentDto> appointmentDtos = appointmentsService.getAppointmentsForPatient(patientName);  
    if(appointmentDtos.isEmpty()) {  
        return ResponseEntity.status(404).body(Collections.emptyList());  
    }  
    return ResponseEntity.status(200).body(appointmentsService.getAppointmentsForPatient(patientName));  
}
```

```

public List<AppointmentDto> getAppointmentsForPatient(String patientName) {
    List<Appointment> appointments = appointmentsDao.findAllForPatient(patientName);
    List<AppointmentDto> appointmentDtos = new ArrayList<>();
    AppointmentDto currentAppointmentDto = new AppointmentDto();
    int iteration = 0;
    for (Appointment appointment: appointments) {
        currentAppointmentDto.setId(iteration);
        currentAppointmentDto.setPatientName(appointment.getPatient().getName());
        currentAppointmentDto.setTreatmentName(appointment.getTreatment().getName());
        currentAppointmentDto.setTreatmentType(appointment.getTreatment().getType());
        currentAppointmentDto.setTimePattern(appointment.getTimePattern());
        currentAppointmentDto.setStartDate(appointment.getStartDate());
        currentAppointmentDto.setEndDate(appointment.getEndDate());
        currentAppointmentDto.setDose(String.valueOf(appointment.getDose()));
        appointmentDtos.add(currentAppointmentDto);
        currentAppointmentDto = new AppointmentDto();
        iteration++;
    }
    return appointmentDtos;
}

return Collections.emptyList();
}

public List<Appointment> findAllForPatient(String patientName) {
    Query query = entityManager.createQuery("From Appointment Where patient = " +
        patientsDao.findPatientByName(patientName));
    return query.getResultList();
}

```

Frontend:

Было необходимо создать еще один ProtectedRoute:

```

<Route element={<ProtectedRoutes />}>
    <Route path="/doctor_main_page" element={<DoctorMainPage />} />
    <Route path="/nurse_main_page" element={<NurseMainPage />} />
    <Route path="/patient_main_page" element={<PatientMainPage />} />
</Route>

```

Была создана страница, содержащая необходимую информацию о пациенте:

Patient	Name of treatment	Type	Time pattern	Period	Dose
Марина Краюхина	массаж	procedure	Twice a day	15/11/2022 - 19/11/2022	-

Запрос на бэкенд:

```
getAppointments(patientName) {  
  const dto = {  
    patientName: patientName  
  }  
  return axios.post(APPOINTMENTS_FOR_PATIENT_URL, dto)  
}
```

В результате получаем новую роль пользователей – пациент и он получает возможность просматривать только свои назначения.