

Санкт-Петербургский государственный университет

Кафедра Системного Программирования

Группа 21.Б15-мм

# Реализация уведомлений и расширение поиска с форумом в Medical Web App

*Гагин Артур Сергеевич*

Отчёт по учебной практике  
в форме «Решение»

Научный руководитель:  
доцент кафедры СП, к.т.н. Ю. В. Литвинов

Санкт-Петербург  
2023

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Постановка задачи</b>	<b>4</b>
<b>2. Обзор</b>	<b>5</b>
2.1. Существующие решения . . . . .	5
2.2. Используемые технологии . . . . .	5
<b>3. Реализация</b>	<b>7</b>
3.1. Серверная часть приложения . . . . .	7
3.2. Клиентская часть приложения . . . . .	8
3.3. Тестирование . . . . .	9
<b>Заключение</b>	<b>14</b>
<b>Список литературы</b>	<b>15</b>

# Введение

В современном мире людям достаточно часто приходится обращаться к врачам. При этом выбрать хорошего специалиста очень сложно. Человек вынужден посетить множество сайтов, чтобы узнать, насколько выбранный им врач хорош. Но даже если выбрать хорошего специалиста для посещения, нет никакой гарантии, что он поставит точный диагноз, поскольку врачебные ошибки неизбежны.

Для того, чтобы избежать плохих последствий от неправильно поставленного диагноза, существует второе мнение — это практика получения дополнительной консультации по уже проведенным медицинским исследованиям. Она необходима для того, чтобы уточнить поставленный человеку диагноз, наметить более корректный план лечения. Более того, данная практика обычно осуществляется высококвалифицированными врачами, что дает возможность провести очень качественную диагностику и лечение пациента.

Студенты кафедры системного программирования СПбГУ уже на протяжении нескольких лет разрабатывают проект Medical Web App [2], с помощью которого можно получить второе мнение.

В прошлом семестре нами был расширен личный кабинет врача, была добавлена страница редактирования профиля, однако поиск в веб-приложении все еще требует доработки, поскольку найти специалиста — первый и самый важный шаг на пути к получению дополнительной консультации. Более того, важно дать возможность и врачам искать пациентов, а для этого необходимо расширить форум, позволить пациентам обращаться за помощью с помощью постов и присылать уведомления соответствующим специалистам.

# 1. Постановка задачи

Целью работы является расширение поиска и форума в веб-приложении, а также реализация уведомлений. Для ее достижения были поставлены следующие задачи.

1. Провести обзор существующих реализаций поиска врачей на разных сайтах.
2. Дополнить страницу поиска недостающими параметрами, выявленными посредством проведенного обзора.
3. Реализовать серверную часть уведомлений в Medical Web App.
4. Добавить возможность пациентам создавать посты с поиском врачей и отправлять последним уведомления.
5. Протестировать код, написанный в ходе работы.

## 2. Обзор

Данный раздел содержит обзор существующих аналогов, помогающий выявить недостающие параметры поиска в веб-приложении, а также используемых технологий.

### 2.1. Существующие решения

Обзор существующих решений производился с целью определения недостающих параметров поиска. На всех сайтах, представленных ниже, есть возможность искать врача по специальности. Помимо этого, на всех сайтах существует возможность сортировки врачей по рейтингу.

- «ПроДокторов» [10] — сайт отзывов о врачах. На странице поиска можно выбрать специальность, стаж (любой, от 5 лет, от 15 лет), сортировку.
- «СберЗдоровье» [11] — данный сайт содержит в себе много всего, в том числе возможность получить второе мнение от врача. В поиске есть возможность отсортировать врачей, выбрать специальность. Выбор стажа отсутствует.
- «НаПоправку» [9] — медицинский портал, который тоже заслуживает внимания. В поиске присутствует выбор специальности и сортировки врачей. Можно искать специалистов с рейтингом выше 4 (от 4 до 5).
- «Докту» [8] — еще один сайт для поиска медицинских специалистов. Есть выбор специалиста и сортировка. Нет возможности выбора стажа от определенного количества лет.

### 2.2. Используемые технологии

Так как веб-приложение Medical Web App было написано с помощью следующих технологий, нами было решено продолжать разработку на них.

- Язык Java и фреймворк Spring Boot [4] для серверной части.
- Язык CSS, Язык JavaScript и библиотека React [7] для клиентской части.
- Библиотека Axios [6], с помощью которой обеспечивается взаимодействие клиентской и серверной частей.
- База данных PostgreSQL [5].

## 3. Реализация

В этом разделе описывается реализация: что было сделано в рамках данной практической работы.

### 3.1. Серверная часть приложения

Серверная часть приложения требовала реализации уведомлений, а также добавления врачу нового поля, в котором должны были содержаться диагнозы, на которых он специализируется. Последнее делалось достаточно просто, однако на первом стоит остановиться поподробнее.

В таблицу уведомления было решено добавить следующие поля.

- Поле “data”, в котором описывается сообщение, которое пользователь должен получить.
- Поле “creationTime” — время получения уведомления.
- Поле “timeZone”, необходимое для корректного отображения времени с учетом часового пояса пользователя.
- Поле “notificationType”, которое говорит нам о том, какого типа уведомления. Пока что тип один — новый пост на форуме. Однако с помощью этого поля, в будущем, можно создавать другие уведомления, что делает уведомления легко расширяемыми.
- Поле “notificationLink”, содержащее ссылку, если уведомление позволяет пользователю перенаправиться на какую-нибудь страницу при необходимости.
- Поле “numberOfOwners”, которое содержит количество пользователей, получивших данное уведомление.

Дело в том, что нет смысла хранить у пользователя сами уведомления, особенно если мы хотим посылать большому количеству разных пользователей одинаковые уведомления, ведь тогда мы, по

сути, будем «плодить» много одинаковых объектов. Гораздо проще хранить у пользователей id's уведомлений. При удалении пользователем уведомления убирать один id из списка идентификаторов, а у самого уведомления уменьшать значение numberOfOwners на единицу. А когда это значение достигнет нуля (это значит, что все пользователи удалили данное уведомление), можно просто удалять notification из нашей базы данных.

При этом стоит отметить, что при отправке одинаковых уведомлений не делается много Axios-запросов, ведь был добавлен метод, позволяющий по списку идентификаторов пользователей (которым должно прийти уведомление) создать одно новое уведомление и добавить всем пользователям id этого уведомления.

Таким образом, алгоритм получился достаточно эффективным по сравнению с простейшей идеей, которая первой может прийти в голову — добавление списка уведомлений каждому пользователю.

## 3.2. Клиентская часть приложения

Попутно работая над серверной частью, нами также расширялись страницы поиска и форума.

После проведенного обзора было решено, что при поиске специалистов очень важно уметь их сортировать. Было решено добавить следующие пункты.

- Сортировка по алфавиту.
- Сортировка по рейтингу.
- Сортировка по количеству отзывов.

Также мы позволили пациентам выбирать специальности врачей, диагнозы, на которых последние специализируются (все это было реализовано с помощью React-Select [3] — гибкого элемента управления Select Input для ReactJS с множественным выбором, автозаполнением и



возможностью создания [этот элемент управления мы решили использовать также на странице регистрации и редактирования профиля]). Более того, было решено добавить возможность выбирать цену (в пределах от и до) и стаж (в пределах от), что сделало поиск еще более гибким (рис. 1-2).

Далее, когда работа над поиском была закончена, создание постов на форуме было разделено на «Обсуждение» и «Поиск специалиста». Для последнего была добавлена возможность указать все те же данные: специальность(-и), диагноз(-ы) (с помощью React-Select) и цена (в пределах до), которую пациент готов заплатить (рис. 3-6).

Наконец, мы сделали так, чтобы после создания поста соответствующим врачам приходили уведомления с ссылкой на пост на форуме. Переходя по этой ссылке, врач может взглянуть на пост и откликнуться, обратившись к пациенту в личные сообщения (рис. 7).

### **3.3. Тестирование**

Для клиентской части производилось ручное тестирование. Для серверной части было произведено модульное тестирование с помощью фреймворка JUnit [12]. Также использовалась библиотека Java Code Coverage Library [1] для нахождения покрытий тестами (рис. 8).

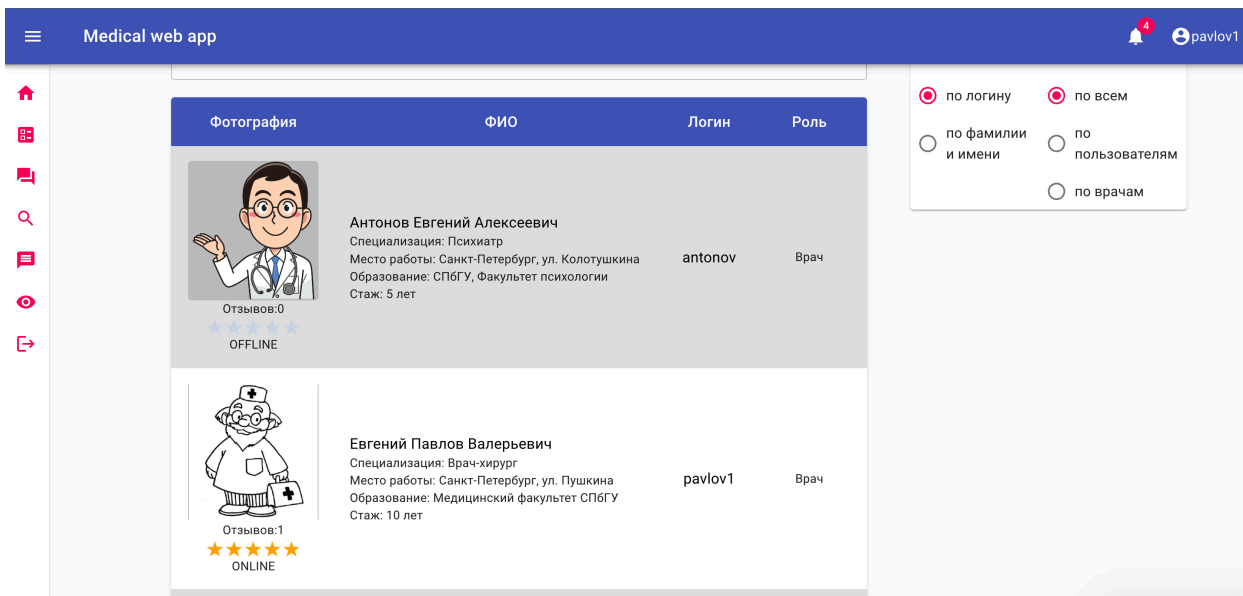


Рис. 1: Страница поиска (до изменений)

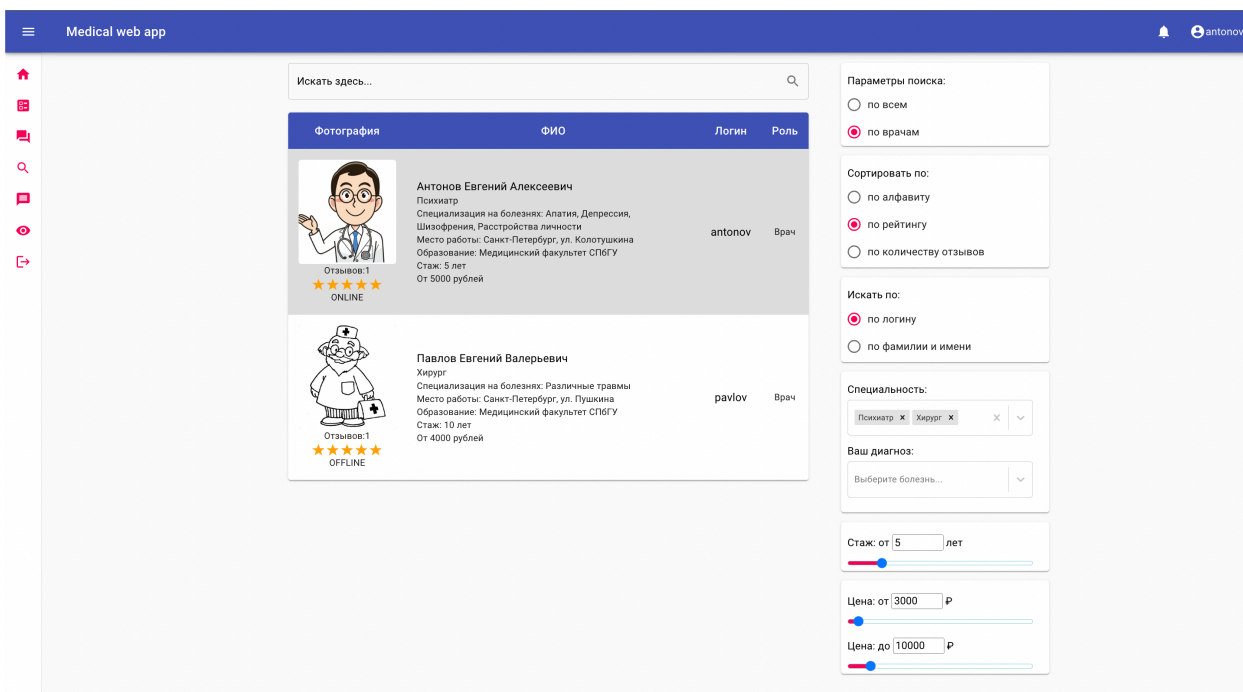


Рис. 2: Страница поиска (после изменений)

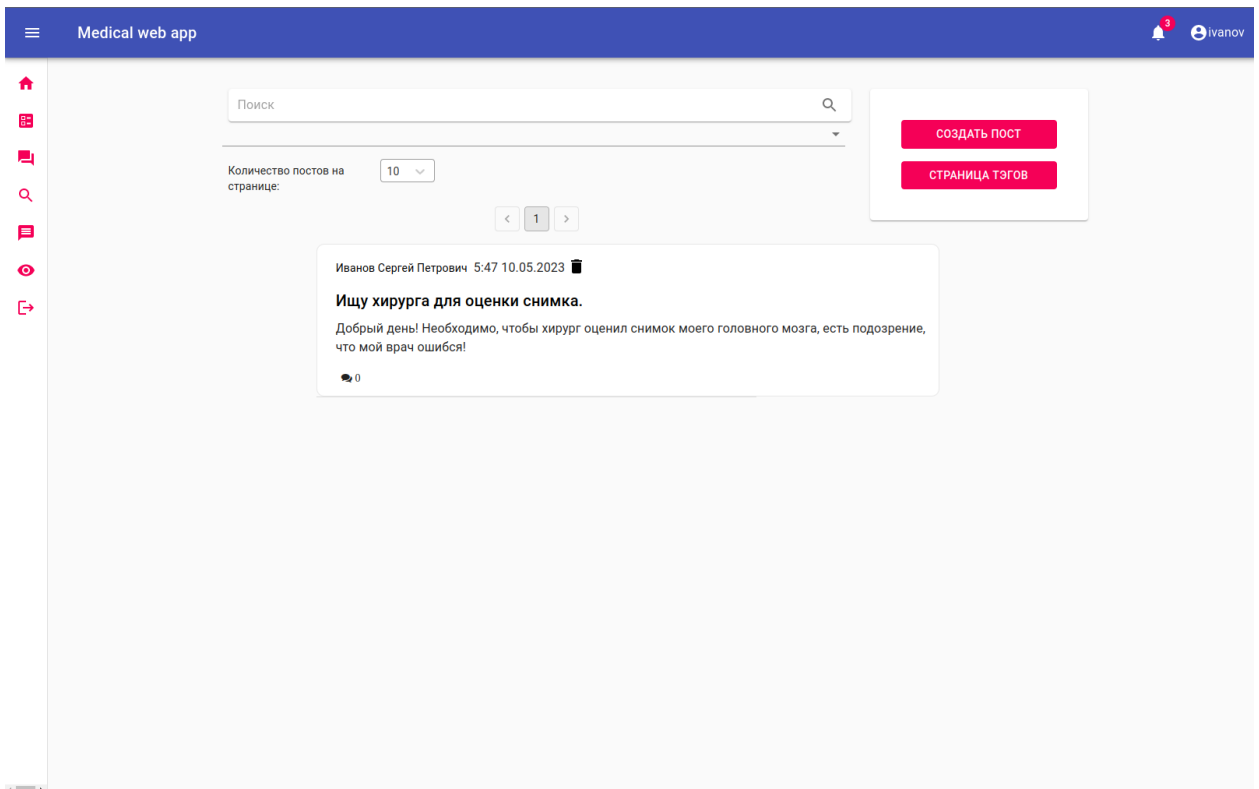


Рис. 3: Страница форума (до изменений)

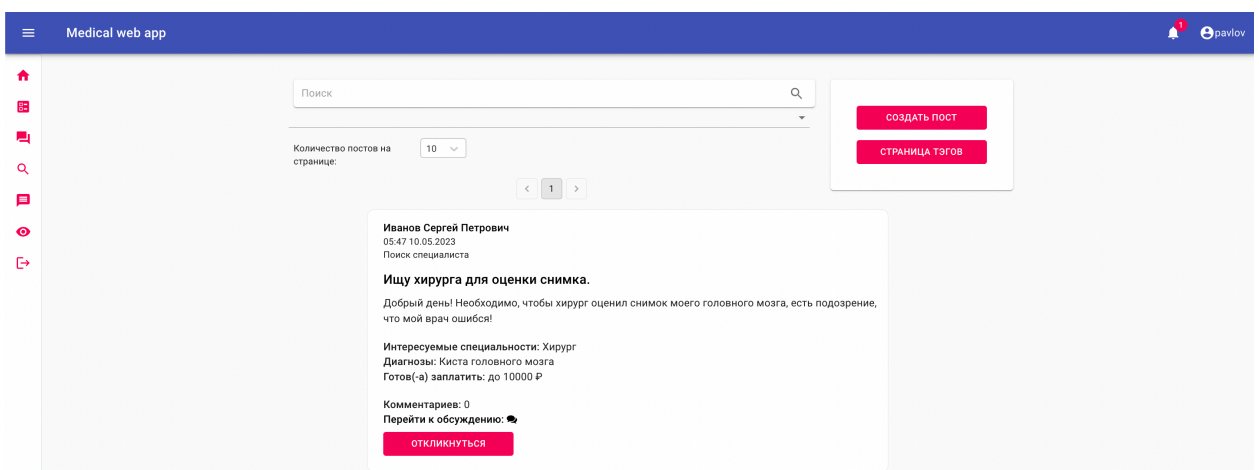


Рис. 4: Страница форума (после изменений)

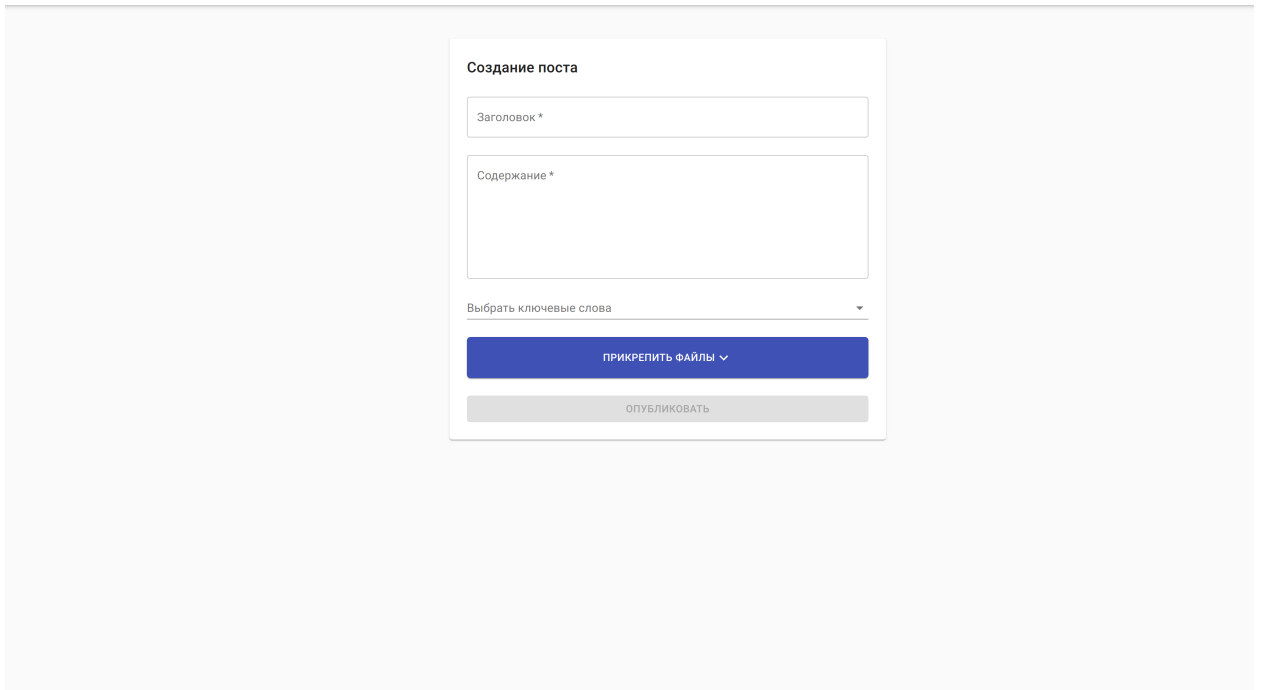


Рис. 5: Страница создания поста (до изменений)

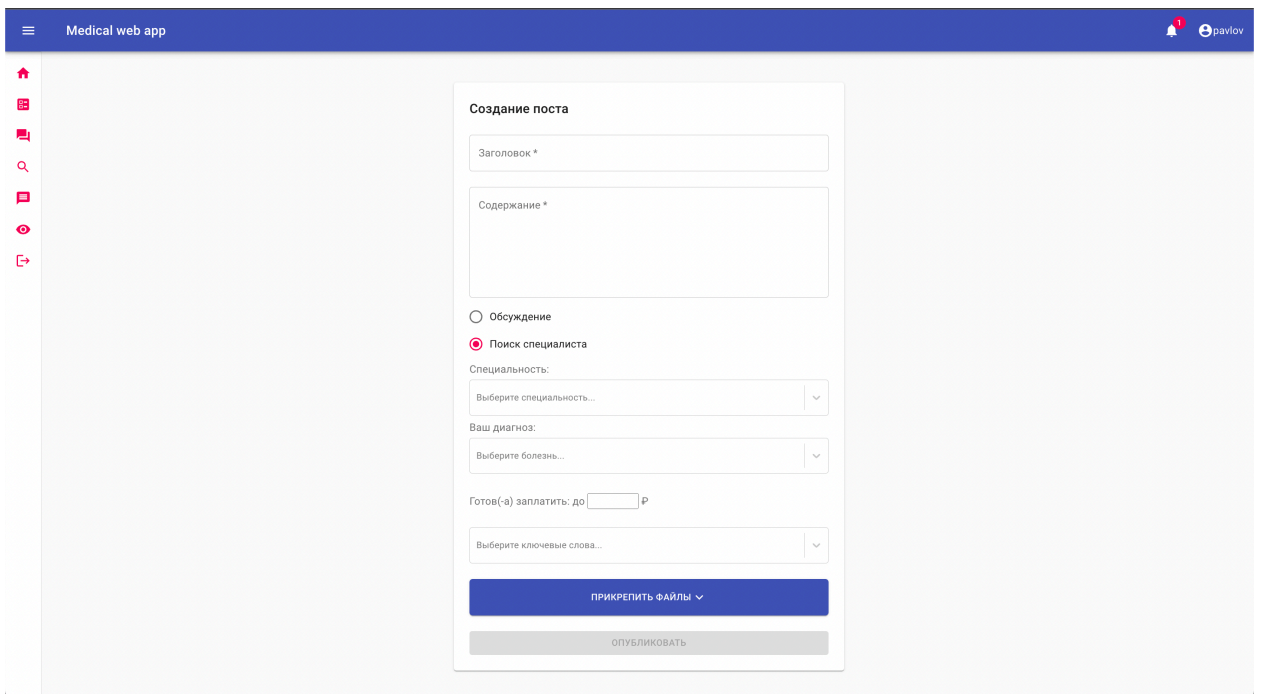


Рис. 6: Страница создания поста (после изменений)

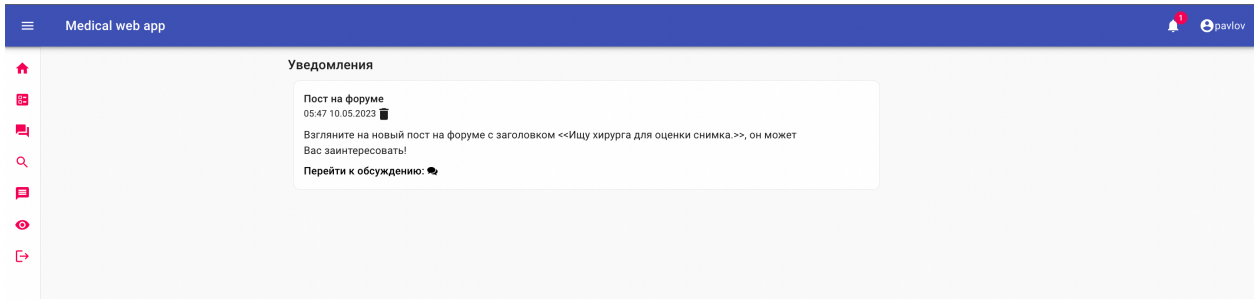


Рис. 7: Страница уведомлений

### NotificationService

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Ctxty	Missed	Lines	Missed	Methods
NotificationService(UserRepository, NotificationRepository)		100 %		n/a	0	1	0	4	0	1
saveNotification(Notification, Long)		100 %		n/a	0	1	0	4	0	1
editNotificationContent(String, Long)		100 %		n/a	0	1	0	4	0	1
editNotificationsContent(String, List)		100 %		100 %	0	2	0	4	0	1
getAllNotifications(Long)		100 %		100 %	0	3	0	8	0	1
saveNotification(Notification, List)		100 %		100 %	0	3	0	12	0	1
deleteNotification(Long, Long)		88 %		87 %	1	5	2	19	0	1
<b>Total</b>	10 of 247	95 %	1 of 18	94 %	1	16	2	55	0	7

Created with JaCoCo 0.8.2.201808211720

Рис. 8: Результаты покрытий тестами

# Заключение

В результате практической работы за весенний семестр второго курса были выполнены следующие задачи.

1. Проведен обзор существующих реализаций поиска врачей на разных сайтах.
2. Страница поиска была дополнена недостающими параметрами, выявленными посредством проведенного обзора.
3. Реализована серверная часть уведомлений в Medical Web App.
4. Добавлена возможность пациентам создавать посты с поиском врачей и отправлять последним уведомления.
5. Протестирован код, который был написан в ходе работы.

Весь код доступен в репозитории проекта<sup>1</sup> на GitHub. Имя аккаунта: lastdesire.

---

<sup>1</sup><https://github.com/MathAndMedLab/Medical-Web-App/tree/develop-new>

## Список литературы

- [1] JaCoCo Java Code Coverage Library. — URL: <https://www.jacoco.org/jacoco/> (дата обращения: 2022-12-19).
- [2] Medical Web App. — URL: <https://github.com/MathAndMedLab/Medical-Web-App> (дата обращения: 2022-10-01).
- [3] React-Select. — URL: <https://react-select.com/home> (дата обращения: 2023-03-18).
- [4] Spring Boot. — URL: <https://spring.io/projects/spring-boot> (дата обращения: 2022-10-19).
- [5] База данных PostgreSQL. — URL: <https://www.postgresql.org/> (дата обращения: 2022-11-01).
- [6] Библиотека Axios. — URL: <https://axios-http.com/> (дата обращения: 2022-10-08).
- [7] Библиотека React. — URL: <https://reactjs.org/> (дата обращения: 2022-10-04).
- [8] Докту. — URL: <https://doctu.ru/> (дата обращения: 2022-10-18).
- [9] НаПоправку. — URL: <https://napopravku.ru/> (дата обращения: 2022-10-18).
- [10] ПроДокторов. — URL: <https://prodoctorov.ru/> (дата обращения: 2022-10-18).
- [11] СберЗдоровье. — URL: <https://sberhealth.ru/> (дата обращения: 2022-10-18).
- [12] Фреймворк JUnit. — URL: <https://junit.org> (дата обращения: 2022-11-28).