

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 20.Б11-мм

Шеремет Ирина Дмитриевна

Medical web app: реализация тем в постах и интеграция Dicom Viewer

Отчёт по учебной практике
в форме «Решение»

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

Санкт-Петербург
2022

Оглавление

1. Введение	3
2. Постановка задачи	4
3. Обзор	5
3.1. Обзор существующих решений	5
3.2. Обзор существующей реализации	6
4. Описание реализации	9
4.1. Топики	9
4.2. Пользовательский интерфейс	9
4.3. Интеграция DICOM Viewer	12
4.4. Тестирование	13
5. Заключение	14
Список литературы	15

1. Введение

В настоящее время важно заботиться о своём здоровье. Однако поставленный лечащим врачом диагноз вполне может быть ошибочным. Так, по данным Министерства здравоохранения о работе патологоанатомической службы, каждому двадцатому умершему в 2018 году был поставлен неправильный диагноз и назначено ошибочное лечение [1]. В таком случае было бы полезно его перепроверить, то есть узнать второе мнение по интересующей проблеме уже из другого источника.

Таким образом, второе мнение (англ. *Second Opinion*) — это получение повторной консультации у специалиста по уже имеющимся результатам медицинских исследований.

Medical web app — веб-система диагностики патологий по медицинским снимкам. Планируется несколько способов предоставления второго мнения:

- автоматический анализ с использованием библиотеки *MIRF* [5];
- обсуждение диагноза на форуме или поиск в постах схожих симптомов;
- нахождение врача для личного общения с ним.

Для удобной навигации по форуму необходимы ключевые слова (топики), по которым врачу будет легче найти вопросы по своей специальности или пациенту – по интересующей теме. Также обычный пользователь будет использовать только тот сайт, в котором наиболее приятный и понятный ему интерфейс.

Для передачи, хранения и визуализации медицинских изображений наиболее широко используется стандарт DICOM (Digital Imaging and Communications in Medicine). Medical web app является медицинской платформой, таким образом, необходима программа, позволяющая производить полноценную работу с DICOM-файлами. Данная программа в дальнейшем будет называться DICOM Viewer.

2. Постановка задачи

Целью работы является создание и прикрепление топиков к постам в веб-приложении; улучшение пользовательского интерфейса; интеграция DICOM Viewer с медплатформой.

Для её выполнения были поставлены следующие задачи:

1. Изучение существующей реализации.
2. Обзор других сервисов, предоставляющих второе мнение.
3. Реализация клиентской и сервисной части топиков.
4. Улучшение UI.
5. Интеграция DICOM Viewer.
6. Написание тестов.

3. Обзор

3.1. Обзор существующих решений

Обзор проводится для выявления критериев в реализации топиков и представления желаемого интерфейса сайта. Изначально под обзор попадали самые популярные сайты, предоставляющие услугу второго мнения, которые можно было найти по запросу в Google.

1. Radiology [10] — сайт, предоставляющий услуги телерадиомедицины. Из плюсов можно отметить выделение важной информации, достаточно удобную навигацию по сайту. Однако загрузить данные можно только в архивированном виде, присутствует информация не о всех врачах.
2. ПЭТ Технолоджи [18] - веб-платформа центра оказания медицинской помощи. Нужно отметить наличие инструкции по получению второго мнения, возможность написать сообщение операторам, но периодическое появление напоминания об этом мешает чтению.
3. Скандинавия [16] - сайт медицинской клиники. Из особенного могу выделить версию сайта на английском языке, отсутствие навигации по странице второго мнения.

Во всех уже описанных сайтах отсутствует автоматическая обработка медицинских данных, чье наличие является преимуществом Medical web app, и форум, необходимый для добавления топиков и поиска по нему, поэтому были рассмотрены некоторые другие сервисы.

1. Русский PubMed [17] - медицинский форум. Все вопросы изначально объединены по заболеваниям, навигация лишь в виде структуры подтем, наличие ошибок в информации о сайте.
2. Reddit [14] - популярная социальная сеть или веб-форум. Топики (flairs в Reddit) расположены рядом с названием поста, по нажатию выполняется поиск.

3.2. Обзор существующей реализации

В работах ранее уже была написана черновая версия сайта [20].

3.2.1. Серверная часть

Серверная часть Medical web app написана на языке Java с использованием библиотеки SpringBoot [15] для облегчения создания веб-приложения.

За обработку HTTP-запросов отвечают контроллеры. Уже были реализованы:

- AuthController — регистрация и аутентификация.
- FileObjectController — загрузка и скачивание файлов.
- PipelineController — получение всех pipeline-ов (конвейеров).
- PipelineJobController — создание нового и получение именно своих pipeline-ов.
- RecordController — создание и просмотр постов, поиск по названию.

Сервисы содержат в себе некоторую логику контроллеров для лучшей структуризации и читаемости.

Репозитории и модели помогают взаимодействовать с базой данных, используя возможности SpringBoot.

3.2.2. Клиентская часть

Клиентская часть сайта написана на языке JavaScript с помощью фреймворка React [11]. Также используется Axios (библиотека для выполнения HTTP-запросов) [2].

Весь код со стороны клиента разделен на компоненты, сервисы и стили. В компонентах представлен основной внешний вид, в сервисах с помощью Axios происходит HTTP-запрос, стили - это просто css описание внешнего вида используемых в компонентах элементов.

Интерфейс некоторых страниц представлен на рисунках 1-3.

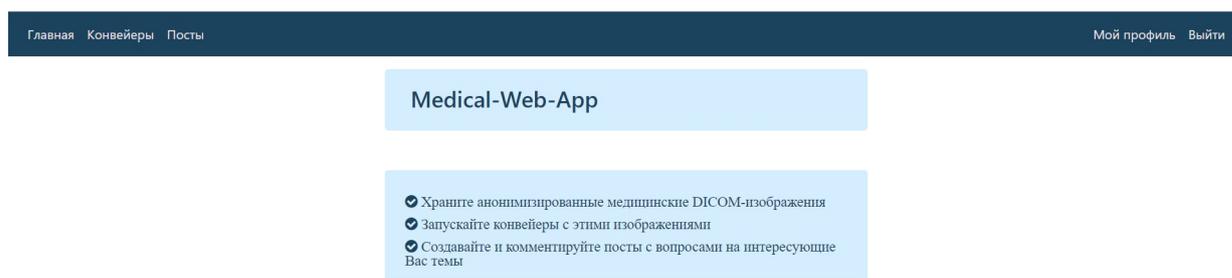


Рис.1: Главная страница

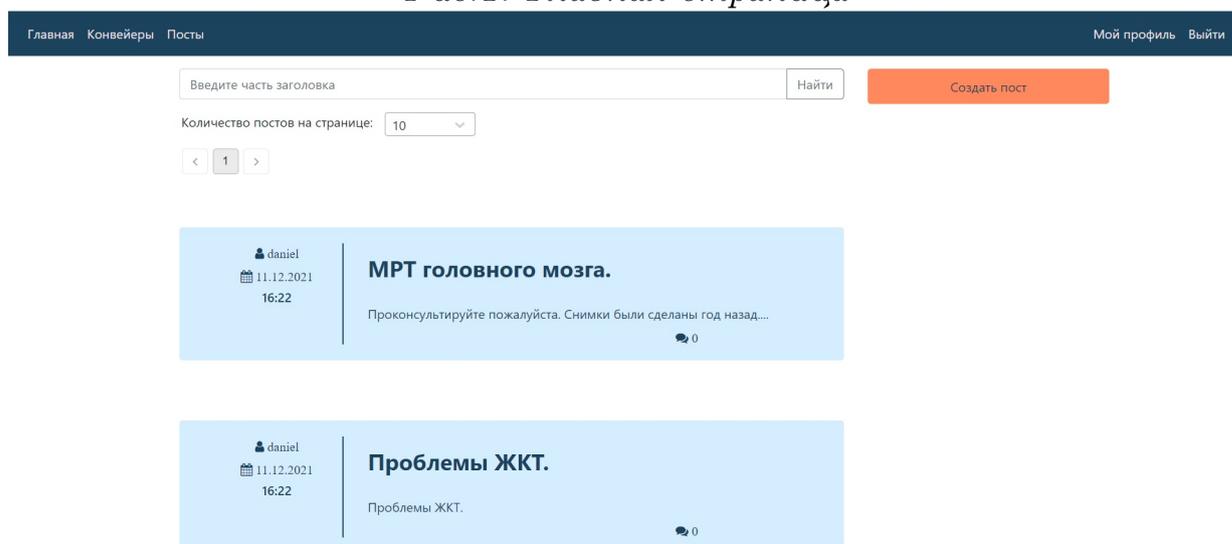


Рис.2: Просмотр всех опубликованных постов

Логин

Пароль

Рис.3: Регистрация пользователя

4. Описание реализации

4.1. Топики

Так как весь код написан на Java с использованием библиотеки SpringBoot, было принято решение и в дальнейшем писать с помощью данных технологий.

Изначально была создана модель топика, включающего в себя id, который генерируется автоматически, название, создателя и время создания. Также для взаимодействия с базой данных написан TopicRepository.

Для обработки HTTP-запросов необходим TopicController. В нём реализованы возможности получения всех существующих топиков и создания нового (часть логики для этого представлена в TopicService).

Ранее уже был реализован поиск поста по заголовку, так что необходимо было добавить только опциональный выбор нужного топика, что и было сделано. Запрос на поиск принимает RecordController, уже он вызывает функцию из класса-сервиса: обычный поиск в случае отсутствия выбранного топика или поиск и по заголовку, и по топикам. Далее с помощью репозитория в базе данных ищутся необходимые посты.

4.2. Пользовательский интерфейс

Ранее представленный интерфейс создавался лишь с целью представления работоспособности разработанного решения автоматизации обработки медицинских данных, поэтому необходимо было полностью переписать UI. React как JavaScript-библиотека имеет ряд преимуществ – например, легкость в изучении и виртуальная DOM (document object model) [13], так что мы продолжили использовать React. Аналогично Axios в качестве библиотеки для выполнения HTTP-запросов нас полностью устраивала, поэтому осталась без изменений.

Также нужно было выбрать библиотеку компонентов для React. Их достаточно много, с помощью различных статей (таких как [19]) из библиотек были отобраны самые популярные — React Bootstrap [12] и Material UI [6]. В дальнейшем главным образом благодаря множеству

компонентов был выбран Material UI.

После обзора существующих решений было получено представление о желаемом виде интерфейса, которое я реализовывала.

Необходимо было написать клиентскую часть топиков. Для этого был создан сервис, который передает запрос на серверную часть, компонент топика, представляющий собой новую страницу, где представлены все ранее созданные топики, и компонент с визуальным отображением информации о каждом элементе на этой странице. При создании поста была добавлена возможность выбрать один или несколько топиков, в сам пост — их отображение и поиск по клику, на страницу форума — переход на страницу топиков, а к фильтру по заголовку добавлено раскрывающееся поле с выбором для топиков.

Всей команде, работающей над Medical web app, не нравились используемые главные цвета интерфейса, так что одним из первых этапов стал подбор сочетающихся цветов, подходящих для медицинского сайта. Совместно они были определены (рисунки 4-7).

После этого были разделены по команде и переписаны все страницы нашего сайта.

Главная страница (находится в разработке, ожидается необходимый текст от руководителей проекта) — рисунок 4.

Навигация по сайту. Незарегистрированные пользователи могут просматривать только главную страницу, страницы регистрации и входа. Находится на всех рисунках 4-7.

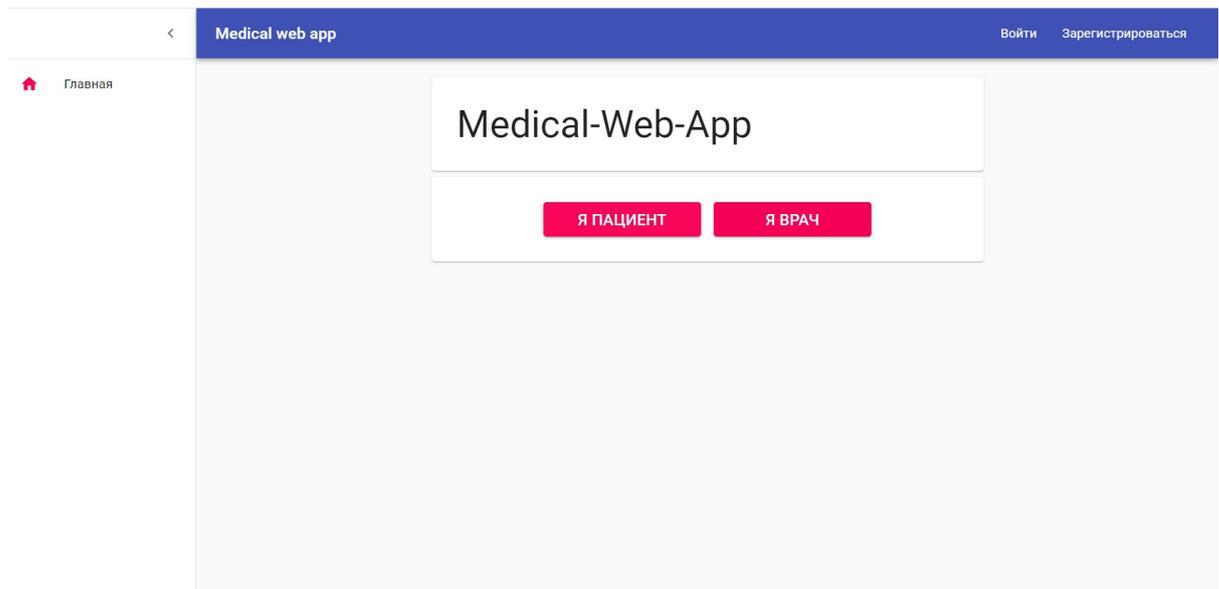


Рис.4: Главная страница и навигация для незарегистрированного пользователя

Страница форума. Общая концепция осталась прежней, совершен переход на компоненты Material UI и выбранную цветовую гамму (рисунок 5).

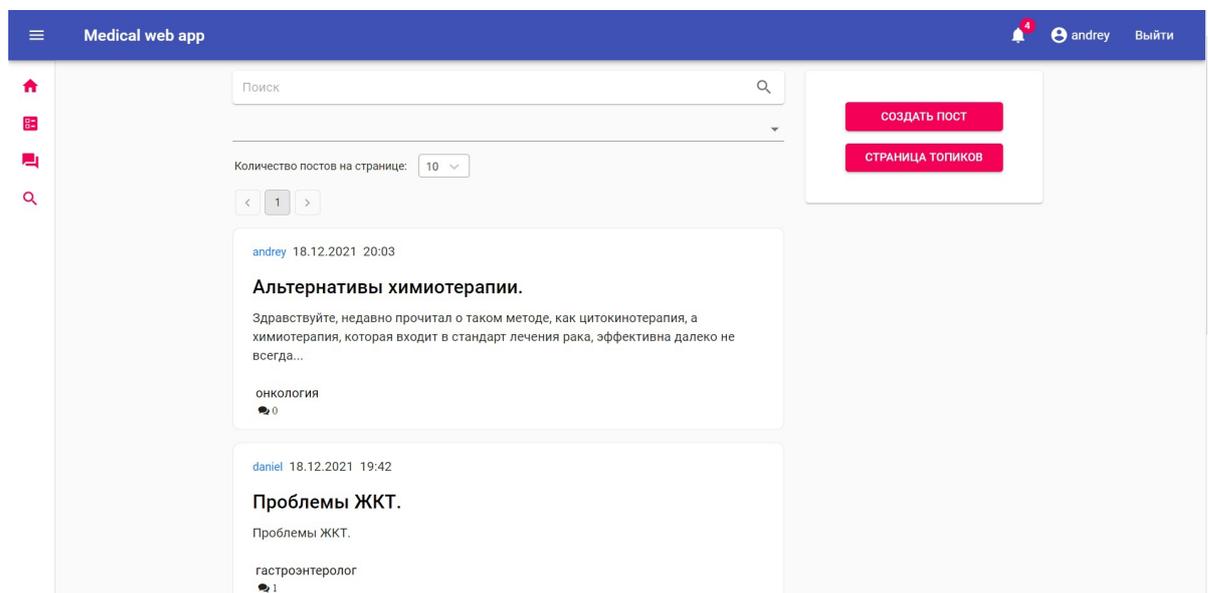


Рис.5: Интерфейс форума, закрытый Drawer

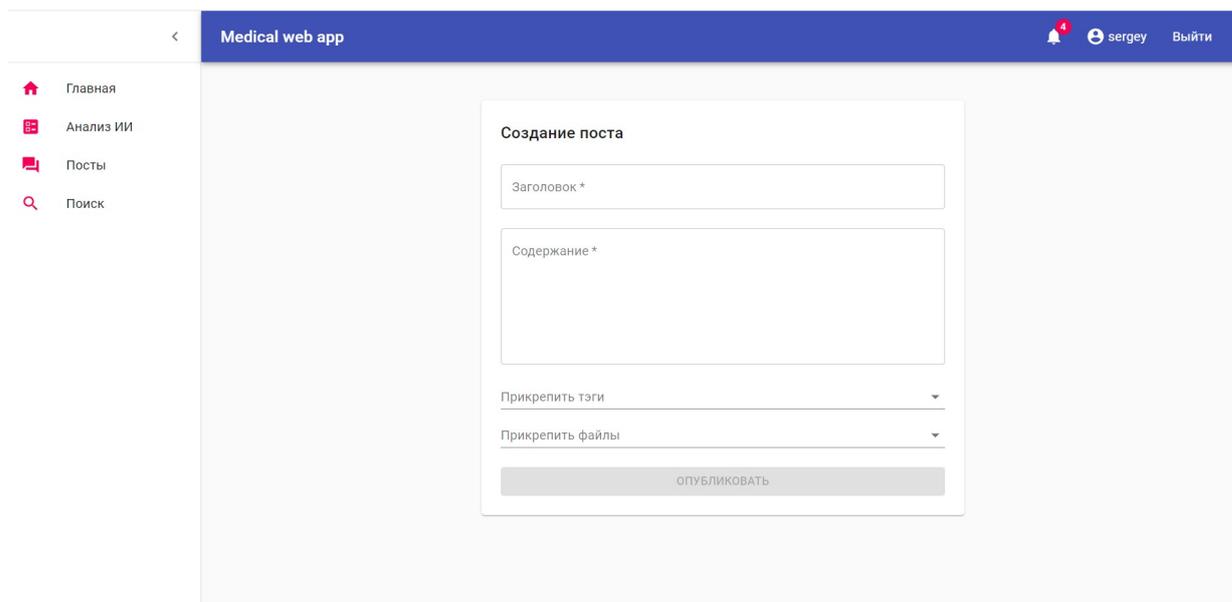


Рис.6: Интерфейс создания поста

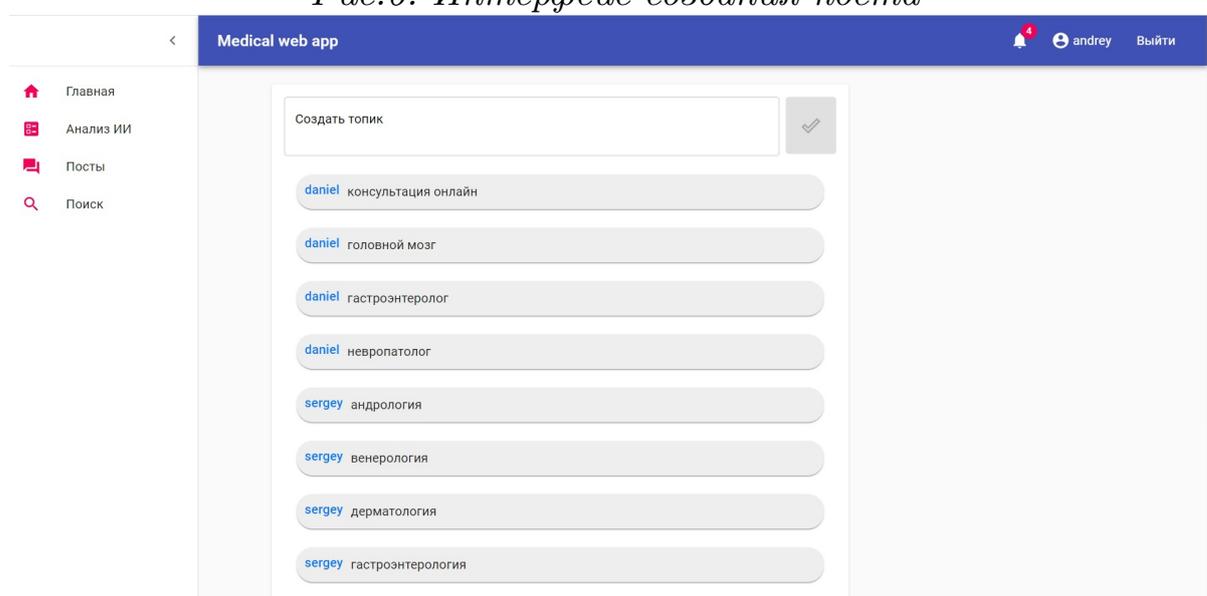


Рис.7: Интерфейс страницы топиков

4.3. Интеграция DICOM Viewer

Для интеграции с веб-платформой в рамках курсовой работы уже был написан DICOM Viewer [4], за основу которого был взят ONIF [8].

Все DICOM-файлы, которые пользователь загрузил на медплатформу, анонимизированно хранятся на сервере Orthanc [9]. Однако он не поддерживает механизм CORS [3]. Для устранения этой проблемы был использован почтовый прокси nginx [7].

На данный момент работа с DICOM Viewer не полностью заверше-

на, так как с любого профиля можно увидеть абсолютно все файлы, загруженные в Orthanc. Также доступ к единичным файлам всё ещё блокируется политикой CORS.

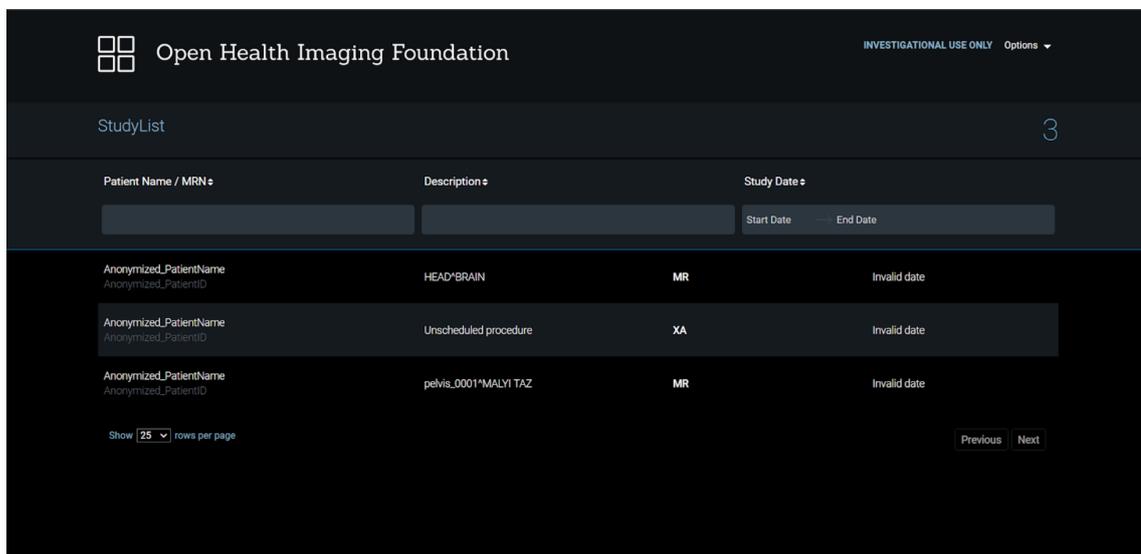


Рис.8: Интерфейс DICOM Viewer

4.4. Тестирование

Для проверки того, что программа ведет себя так, как ожидается, были написаны некоторые тесты. Проверяется правильность создания топика и поста через сервисы и дальнейшее их нахождение непосредственно в самой базе данных, для чего был создан специальный тестовый контейнер.

5. Заключение

В ходе выполнения данной работы были получены следующие результаты.

- Изучена ранее существующая реализация.
- Проведен обзор существующих аналогов.
- Реализованы топики.
- Изменен UI.
- Частично интегрирован Dicom Viewer.
- Произведено тестирование серверной части топиков и постов.

Код доступен в репозитории¹ GitHub.

¹<https://github.com/MathAndMedLab/Medical-Web-App> (дата обращения: 2022-05-21).

Список литературы

- [1] <https://newizv.ru/news/society/22-05-2019/minzdrav-nazval-chislo-umershih-ot-vrachebnyh-oshibok>. — [Дата обращения 2022-05-21].
- [2] Axios. — <https://axios-http.com/docs/intro>. — [Дата обращения 2022-05-21].
- [3] Cross-Origin Resource Sharing (CORS). — <https://developer.mozilla.org/ru/docs/Glossary/CORS>. — [Дата обращения 2022-05-21].
- [4] DICOM Viewer. — <https://github.com/MathAndMedLab/DicomViewer>. — [Дата обращения 2022-05-21].
- [5] MIRF. — <https://github.com/MathAndMedLab/MIRF2>. — [Дата обращения 2022-05-21].
- [6] Material UI. — <https://mui.com/>. — [Дата обращения 2022-05-21].
- [7] Nginx. — <https://nginx.org/ru/https://nginx.org/ru/>. — [Дата обращения 2022-05-21].
- [8] The OHIF Viewer. — <https://github.com/OHIF/Viewers>. — [Дата обращения 2022-05-21].
- [9] Orthanc. — <https://www.orthanc-server.com/>. — [Дата обращения 2022-05-21].
- [10] Radiology-online. — <https://radiology-online.ru/usluga-vtoroe-mnenie>. — [Дата обращения 2022-05-21].
- [11] React. — <https://ru.reactjs.org/>. — [Дата обращения 2022-05-21].
- [12] React Bootstrap. — <https://react-bootstrap.github.io/>. — [Дата обращения 2022-05-21].

- [13] React VS Angular VS Vue. — <https://mrehead.com/ru/blog/angular-vs-react-vs-vue-2021/>. — [Дата обращения 2022-05-21].
- [14] Reddit. — <https://www.reddit.com/>. — [Дата обращения 2022-05-21].
- [15] Spring Boot. — <https://spring.io/projects/spring-boot>. — [Дата обращения 2022-05-21].
- [16] Клиника внимательной медицины Скандинавия. — <https://www.avaclinic.ru/vtoroe-mnenie/>. — [Дата обращения 2022-05-21].
- [17] Медицинский форум. — <http://ruspubmed.ru/>. — [Дата обращения 2022-05-21].
- [18] ПЭТ Технолоджи. — <https://www.pet-net.ru/pacientu/vtoroe-mnenie.php>. — [Дата обращения 2022-05-21].
- [19] Топ-13 библиотек компонентов React на 2020 год. — <https://techrocks.ru/2020/01/18/13-top-react-component-libraries/>. — [Дата обращения 2022-05-21].
- [20] Швыркова Александра Алексеевна. Платформа автоматизации обработки медицинских данных [2021]. — <https://dspace.spbu.ru/handle/11701/32463>. — [Дата обращения 2022-05-21].