



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**Факультет «Радиотехнический»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Дисциплина «Разработка интернет-приложений»**

**Расчетно-пояснительная записка**

**Тема: «Составление занятий»**

**Студент: Толкачев Р.М.**

**Группа РТ5-51Б**

**Преподаватель: Канев А.И.**

**2024г.**

## ВВЕДЕНИЕ

В современном образовательном контексте, характеризующемся стремительными изменениями, преподаватели сталкиваются с увеличением объема задач и ответственности. Согласно последнему отчету Образовательной ассоциации, опубликованному в сентябре 2023 года, нагрузка на преподавателей увеличилась на 15% за последние два года. Этот тренд подчеркивает необходимость в эффективных инструментах для управления временем и задачами. Интеграция онлайн-обучения также продолжает свой взлет. Согласно статистике Обзора образовательных трендов 2023 года, за последние три года объем онлайн-курсов увеличился на 35%. Этот рост подчеркивает актуальность создания гибких инструментов, которые могут адаптироваться к различным форматам обучения. Отчет "Образование в цифрах: Тенденции и перспективы" предостерегает о том, что более 40% преподавателей выражают беспокойство по поводу недостатка эффективных средств для составления и управления списками задач для проведения занятий.

Цель данного проекта заключается в создании системы для составления списков задач для занятий, предоставляющей преподавателям простой и интуитивно понятный интерфейс для создания и управления этими списками, включающей в себя веб-сервис, веб-приложение и десктопное приложение, обеспечивая возможность удобного отслеживания и редактирования списков задач.

Платформа создана для удобства преподавателей и сотрудников деканата, специализирующихся на формировании занятий и управлении списками заданий. Система обеспечивает эффективное планирование задач, позволяя преподавателям выбирать необходимые для выполнения задания. Для создания списка заданий преподаватель формирует занятие, в которой указывает желаемые задания. Платформа предоставляет автоматизированный

способ создания, учета и отслеживания занятий. Также она предоставляет сотрудникам деканата функционал принятия или отклонения занятий. Сотрудник имеет возможность редактировать существующие и создавать новые задания.

Нефункциональные требования к разрабатываемой системе:

1. Должна поддерживаться кроссплатформенность.
2. Интерфейс системы и текст ошибок должны быть русифицируемы.

В ходе работы необходимо выполнить следующие задачи:

1. Разработать дизайн приложения.
2. Создать базу данных в PostgreSQL.
3. Создать веб-сервис на языке GoLang 1.20.
4. Реализовать интерфейс гостя на технологии React.
5. Развернуть веб-приложение React на Github Pages.
6. Добавить авторизацию и аутентификацию в веб-сервис.
7. Реализовать интерфейс преподавателя в React.
8. Реализовать интерфейс сотрудника деканата в React.
9. Создать десктопное приложение на Tauri.
10. Создать асинхронный сервис формирования занятия на языке Python
11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.

## 1. БИЗНЕС-ПРОЦЕСС

В настоящее время существует много различных трудностей при создании преподавателем списка заданий для студентов, который им необходимо выполнить. Для этого необходимо искать эти задания по разным ресурсам и объединять их в список заданий [1]. Для того, чтобы вручную не искать какие-то задания, а потом в каком-то другом месте сохранять список этих заданий, создана система, которая позволяет объединить решения этих 2 проблем. Такой же функционал предоставляет образовательная платформа "GeekBrains", где участники могут прикреплять образовательные материалы, относящиеся к определенным курсам (например, "Лекции по программированию", "Проекты по искусственному интеллекту" и т.д.). При добавлении нового материала, прямо связанного с тем или иным курсом, требуется указать соответствующую информацию в материале. В случае отсутствия курса в системе, необходимо заполнить информацию о нем. Таким образом, студенты и преподаватели могут ознакомиться со всеми материалами, связанными с образовательными курсами: базой данных курсов и связанными с ними образовательными материалами. При разработке проекта необходимо ориентироваться на функциональность платформы "GeekBrains" и подобных образовательных сервисов, перенимая положительный пользовательский опыт.

Когда преподаватель определится со списком заданий, который он выбрал для студентов, он формирует занятие, и оно отправляется на проверку. После этого его уже нельзя редактировать. Можно также посмотреть историю своих занятий. Сотрудник деканата может принимать занятия или же отклонять их.

Задания могут быть совершенно по разными предметам. Их наполнение может сильно отличаться друг от друга, это может быть общее задание по написанию сочинения по литературе на свободную тему, так и узконаправленное задание на решение конкретного типа задач по математике

или физике [2]. Особенности задания видны по её текстовому описанию. Некоторые задания рано или поздно могут устареть, поэтому деканат может отредактировать существующие задания, исправить какие-либо ошибки, обновить описание, название, предмет или же поменять изображение задания. Также можно создать совершенно новое задание или удалять старые, которые потеряли свою актуальность. Функции пользователей с разными ролями описаны на диаграммах прецедентов (рис. 1).

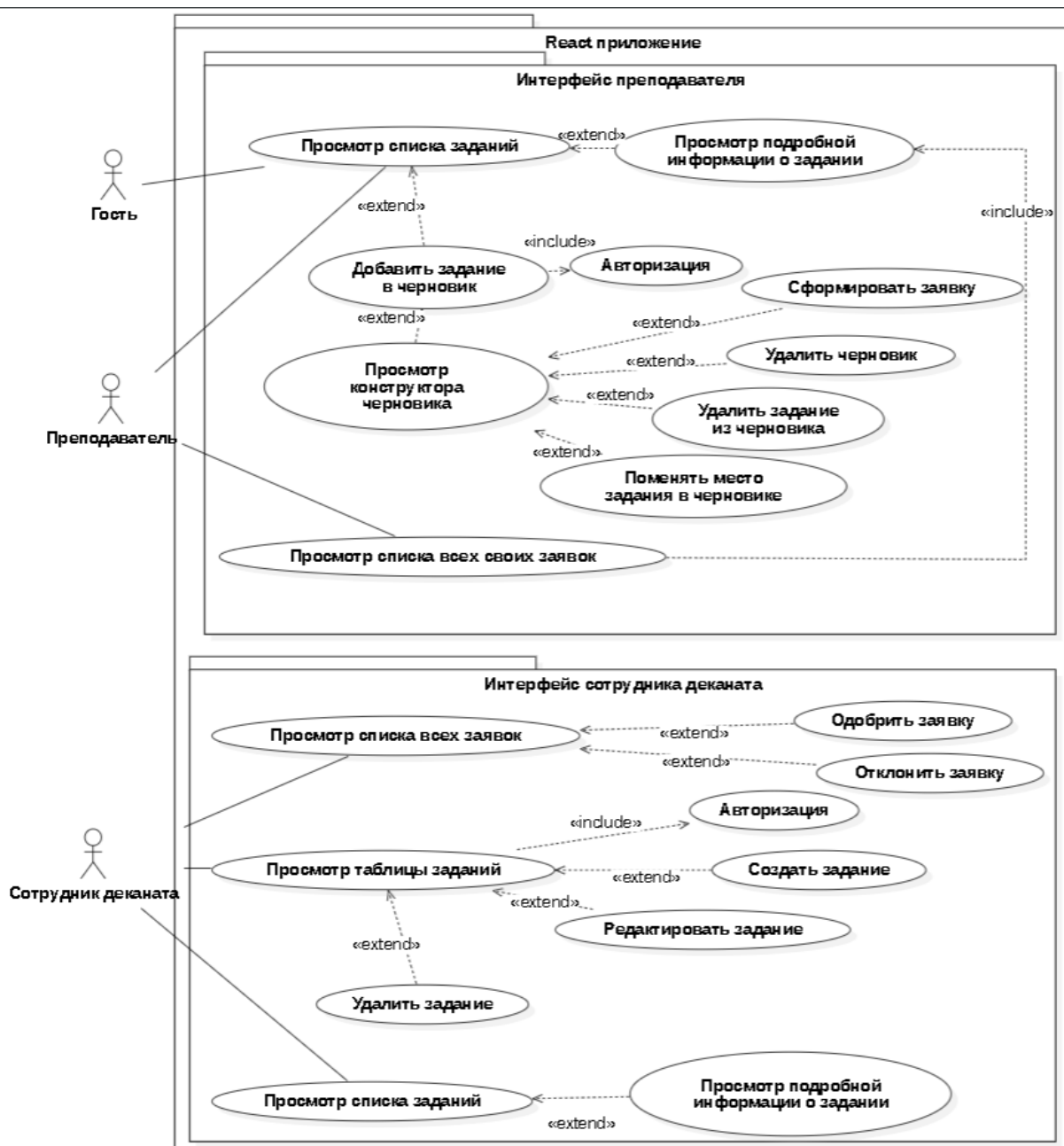


Рисунок 1 - Диаграмма прецедентов

Гостям доступен просмотр заданий. Зарегистрированные гости – преподаватели. Они могут добавлять задания в занятие, просматривать список своих занятий и формировать текущую занятие, добавлять задания в занятие, удалять их оттуда, либо полностью очистить черновую занятие. Отправленные занятия обрабатываются сотрудниками деканата. В результате обработки занятия её либо одобряют, либо отклоняют. Также, сотрудники могут просматривать список заданий в табличном виде, редактировать отдельные задания, либо же создавать новые [3]. Процесс оформления занятия отражен на диаграмме деятельности (рис. 2).

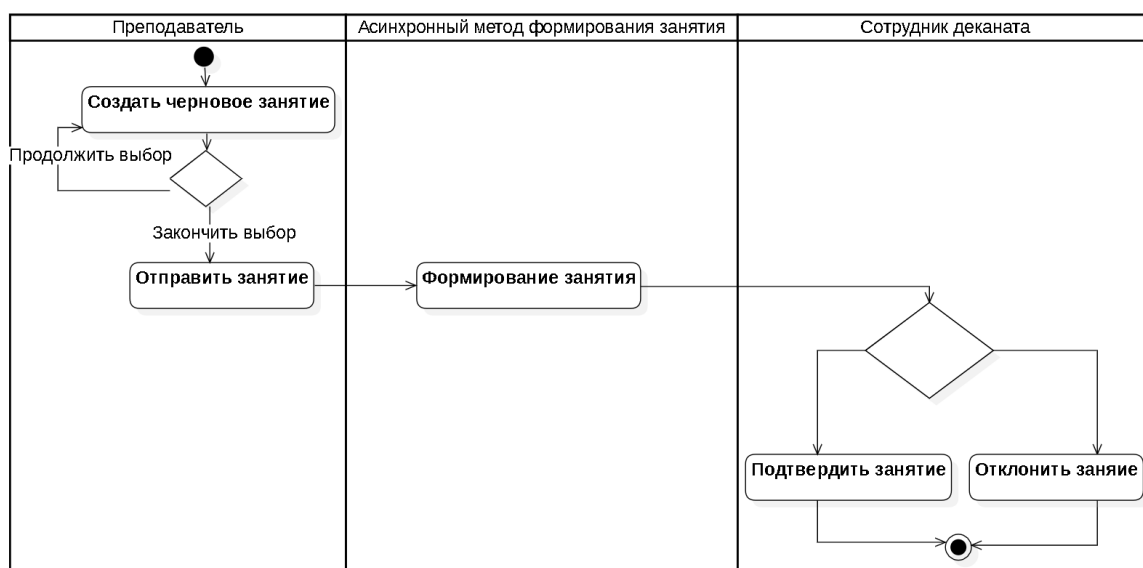


Рисунок 2 - Диаграмма деятельности

Преподаватель выбирает задания, затем формирует на основе выбранных заданий занятие. Это занятие затем обрабатывает асинхронный сервис, а затем и сотрудник деканата. Возможные состояния занятий отражены на диаграмме состояний (рис. 3).

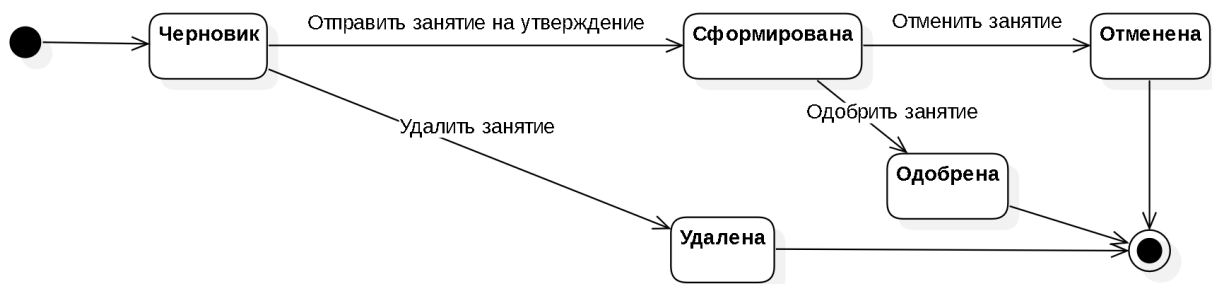


Рисунок 3 - Диаграмма состояний занятий

При выборе первого задания формируется черновик. Последующие выбранные задания добавляются в этот черновик. Преподаватель затем формирует занятие, удаляет её или выходит из приложения. Сформированное занятие обрабатывает сотрудники деканата. Они могут одобрить или отклонить её.

## 2. АРХИТЕКТУРА

Архитектура системы отображена на диаграмме разворачивания (рис. 4).

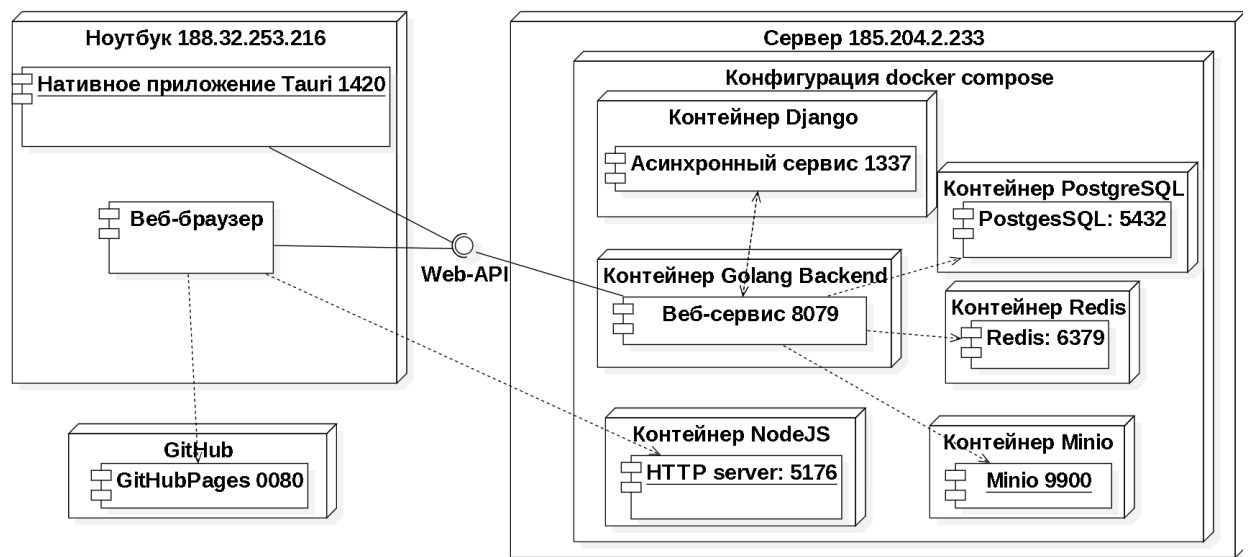


Рисунок 4 - Диаграмма разворачивания

Фронтенд был реализован с помощью JS-библиотек – React [5,6] и Redux более удобного управления состояниями приложения, Redux отделяет логику обновлений состояний от основной и упрощает отслеживание изменения данных [8]. Бекенд разворачивается в кластере докер контейнеров. Такое решение было принято в связи с тем, что технология докер контейнеров позволяет быстро и удобно разворачивать целые системы на любых устройствах, поддерживающих докер. Контейнеры включают в себя все зависимости и библиотеки, необходимые для выполнения приложения. Это гарантирует, что приложение будет работать одинаково на разных средах, избегая проблем с зависимостями, а также контейнеры обеспечивают высокий уровень изоляции, что позволяет запускать приложения в изолированных средах без влияния на другие приложения. Еще одним неоспоримым преимуществом докер контейнеров является то, что контейнеры легко масштабируются в зависимости от изменяющихся требований. Конфигурация контейнеров описана в `docker-compose.yml` файле [4]. Асинхронный сервис



формирования занятий разворачивается отдельно и необязательно на том же компьютере, что и кластер основных контейнеров.

Данные хранятся в СУБД PostgreSQL [7], их структура отражена на ER диаграмме (рис. 5). СУБД PostgreSQL является одним из стандартов индустрии, поэтому было решено использовать её. Часть данных хранится в MinIO – высокопроизводительном объектном хранилище, в частности, там лежат изображения заданий. Redis используется для хранения недействительных токенов. Интерфейс гостя развернут на GitHub pages.

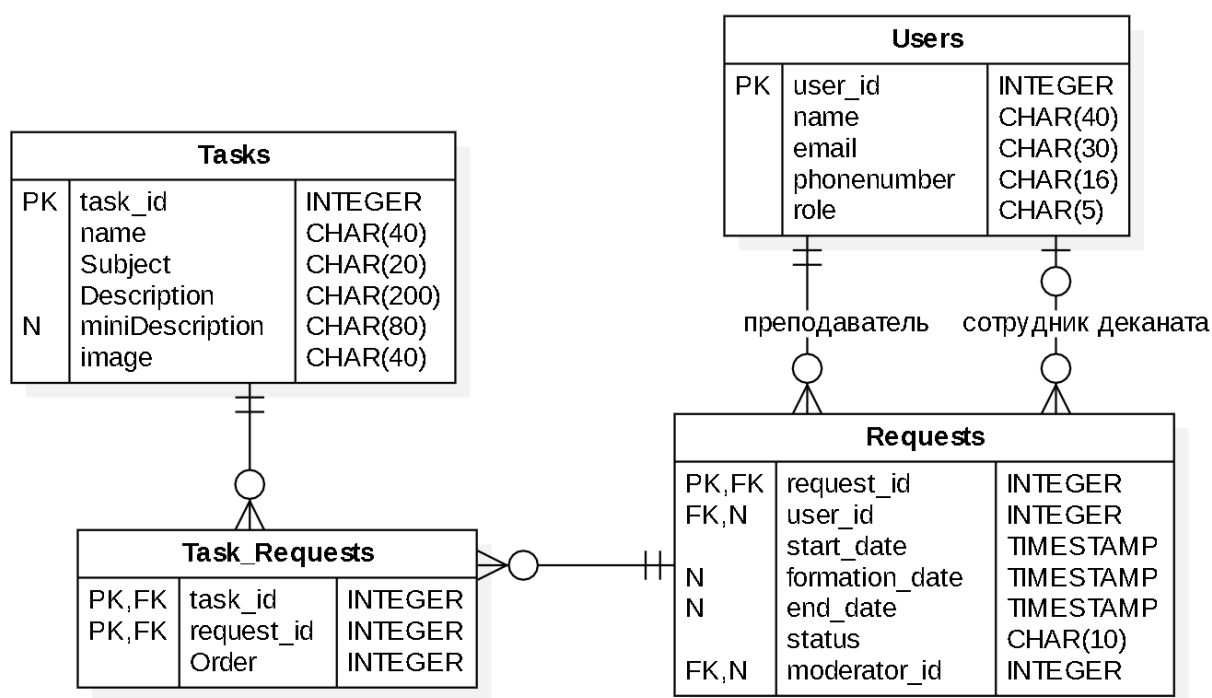


Рисунок 5 - ER диаграмма

Структура данных довольно проста. Помимо базовых полей, в занятии также есть поле преподавателя, который создал занятие и поле сотрудника деканата, который будет подтверждать или отклонять эту занятие. Модель заданий представляет собой набор полей, необходимых исключительно для бизнес-логики. Модель преподавателя содержит контактные данные и поля, необходимые для авторизации и аутентификации. Для хранения в одном занятии нескольких заданий используется промежуточная таблица task-request, реализующая связь М-М.

Устройство бекенда приложения изображено на диаграмме классов бекенда (рис 6.). Модели имеют связи с таблицами в базе данных. Также некоторые модели имеют связи с внешними сервисами. В частности, задания имеют связь с сервером статических файлов, т.к. в заданиях хранится ссылка на их изображение, которое лежит на сервере статических файлов [9].

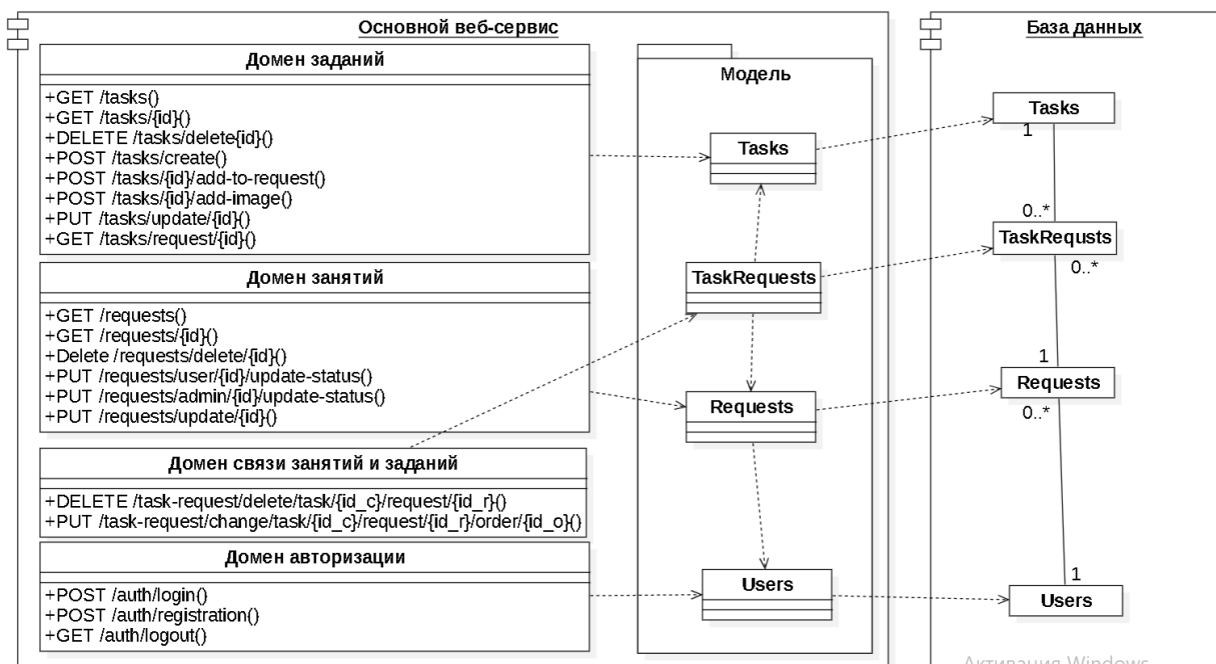


Рисунок 6 - Диаграмма классов бекенда

Связь фронтенда и бекенда отражена на диаграмме классов фронтенда (рис. 7). Ключевые страницы имеют связь с API аутентификации, т.к. доступ к ним осуществляется только для авторизированных пользователей с определенными правами (ролями).

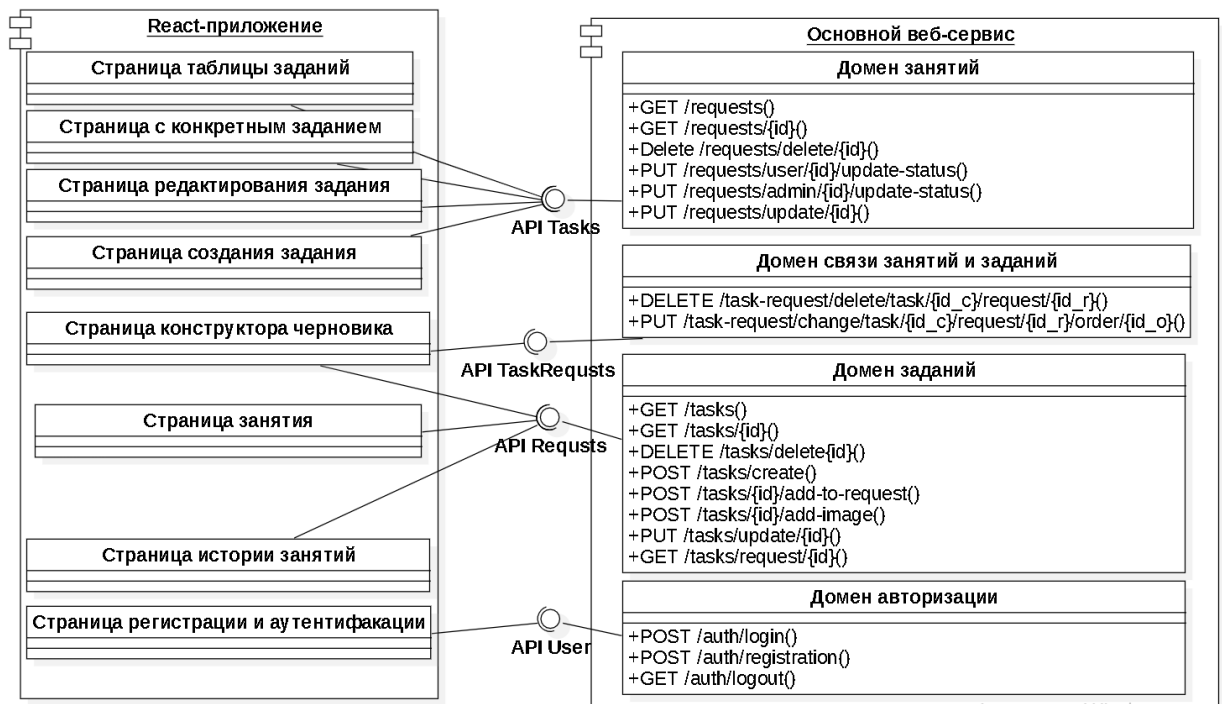


Рисунок 7 - Диаграмма классов фронтенда

### 3. АЛГОРИТМЫ

Алгоритм работы системы отображен на диаграмме последовательности (рис. 8). В основе системы лежит веб-сервис, реализующий внутри себя всю бизнес-логику. Он предоставляет доступ к методам из следующих доменов: пользователя, задания, занятия. Методы следуют правилам REST API.

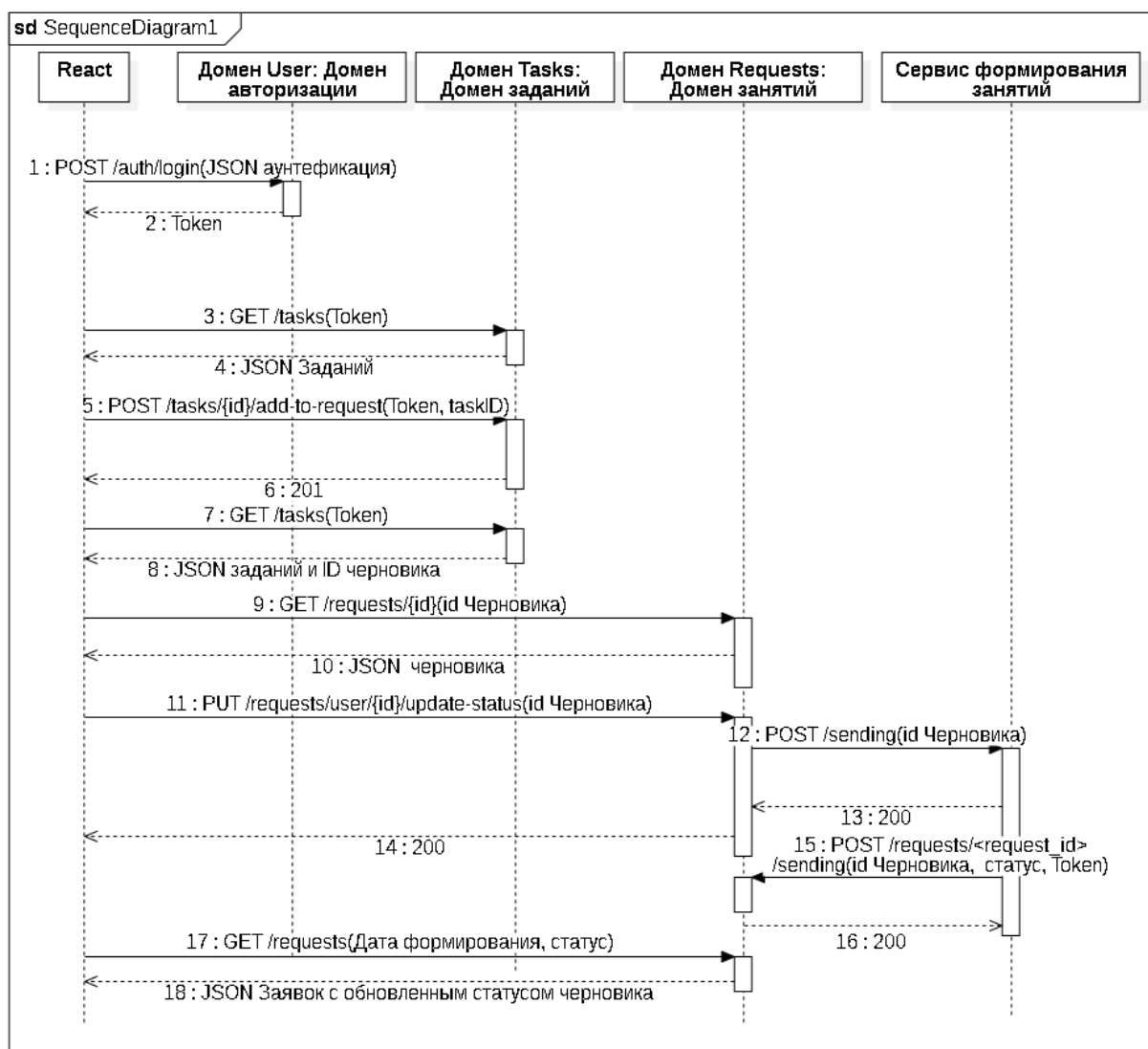


Рисунок 8 - Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация преподавателя. Для этого он отправляет через графический интерфейс запрос, передавая в нем логин и пароль. Если аккаунт с такими данными существует, то пользователь получает токен в ответном запросе. Если же такого аккаунта не существует, или пароль введен неверно, преподаватель получит ошибку. В таком случае

ему надо либо пройти регистрацию, либо ввести свои данные для входа верно. Затем графический интерфейс преподавателя запрашивает у веб-сервиса список заданий, которые возвращаются в JSON формате. Преподаватель выбирает задание, которую хочет добавить, и, нажимая на кнопку «Добавить» в графическом интерфейсе, отправляет запрос на добавление задания в свое черновое занятие. Этот процесс может продолжаться несколько раз.

Когда преподаватель определится с выбором, он переходит в корзину и нажимает на кнопку «отправить» в графическом интерфейсе. После этого приложение запрашивает id чернового занятия преподавателя и затем отправляет запрос на формирование этого занятия. В этот момент основной веб-сервис выполняет асинхронный запрос к сервису формирования занятия на то, чтобы получить разрешение на формирование занятия. Когда занятие будет одобрено, преподаватель сможет увидеть через некоторое время, что его занятие получило новый статус.

Процесс рассмотрения занятий происходит также через графический интерфейс сотрудника деканата. Сотрудники могут просматривать списки всех занятий и, нажимая на соответствующие кнопки, отправлять запросы на одобрение или отклонение занятия в основной веб-сервис. В эти запросы также можно включить фильтры по диапазону дат и статусу занятия, в котором должны быть занятия. Кроме обработки занятий, сотрудники деканата также могут через графический интерфейс управлять непосредственно заданиями. Им доступны такие функции, как создание и редактирование заданий, просмотр списка заданий в табличном формате и их удаление. Для каждой из этих функций присутствует свой метод, отправляемый на основной веб-сервис.

## 4. ОПИСАНИЕ ИНТЕРФЕЙСА

Главное меню приложения включает пункты, которые доступны в зависимости от роли пользователя (рис. 9, 10).

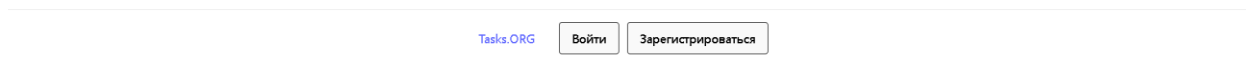


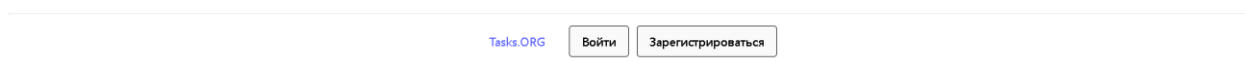
Рисунок 9 - Главное меню (неавторизированный пользователь)



Рисунок 10 - Главное меню (авторизованный пользователь - преподаватель)

Меню видно для всех преподавателей и гостей. В зависимости от типа пользователя его содержимое меняется. Для гостей, например, там отображаются только кнопки «войти», «зарегистрироваться» и «главная», а для преподавателя — «главная», «занятия» и «выйти».

На странице с формой авторизации (рис. 11) отображается форма, через которую гость входит в свой аккаунт. При успешном вводе логина и пароля гость получает JWT, который сохраняется в localStorage и используется при отправлении запросов.



### Авторизация

Почта  
rock@example.com

Пароль  
\*\*\*\*

Войти

[Еще нет аккаунта? Зарегистрироваться](#)

Рисунок 11 - Страница авторизации

На странице с формой регистрации (рис. 12) гости могут завести аккаунт. Для этого нужно указать имя, логин, email, телефон пароль, и подтверждение пароля. Если введенный логин, телефон, email уже заняты, то система попросит пользователя сменить данные.

Tasks.ORG   Войти   Зарегистрироваться

### Регистрация

Имя

Почта

Телефон

Пароль

Повторите пароль

[Есть аккаунт? Войдите](#)

Рисунок 12 - Страница регистрации

На странице со списком заданий (рис. 13) они отображаются в виде списка карточек. На каждую карточку можно нажать и тогда пользователь переносится на страницу с подробной информацией о задании. Также на карточках есть кнопка «Добавить», которая добавляет задание в черновое занятие. Сверху находится поле поиска заданий, там можно ввести искомый предмет задания. Под этим поле поиска есть кнопка корзины, которая становится активной при наличии хотя бы одного задания в занятии. При нажатии на нее пользователь переходит на страницу корзины (рис. 14), где он может посмотреть все свои добавленные задания в занятие

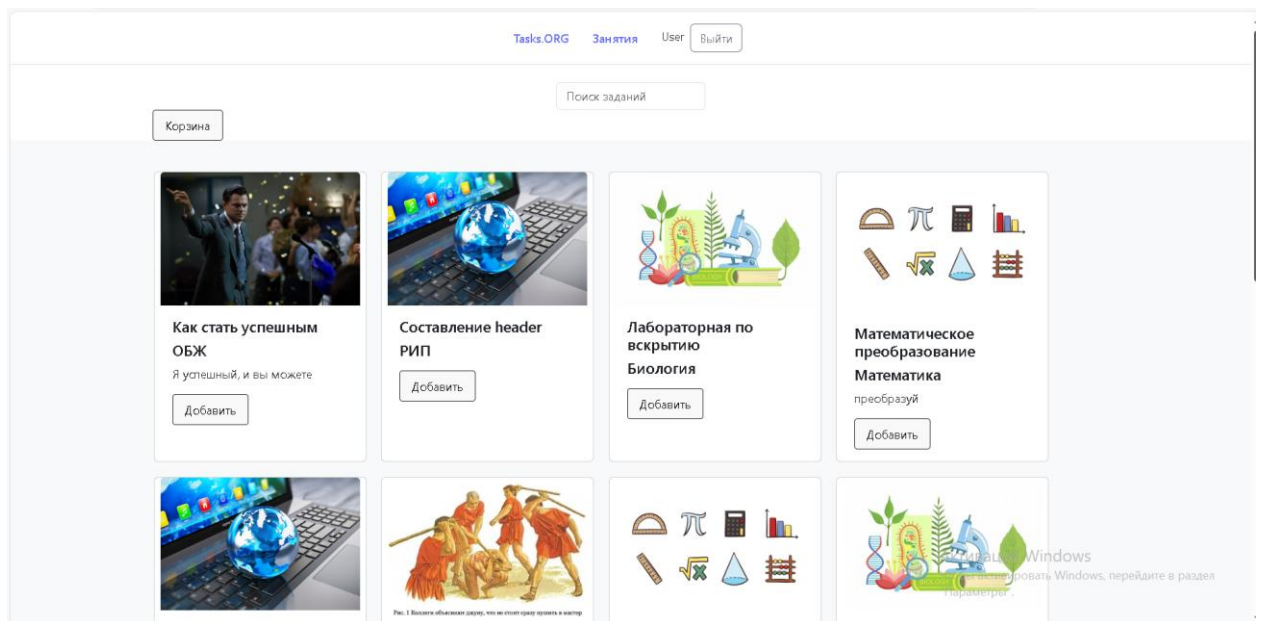


Рисунок 13 - Страница со списком заданий

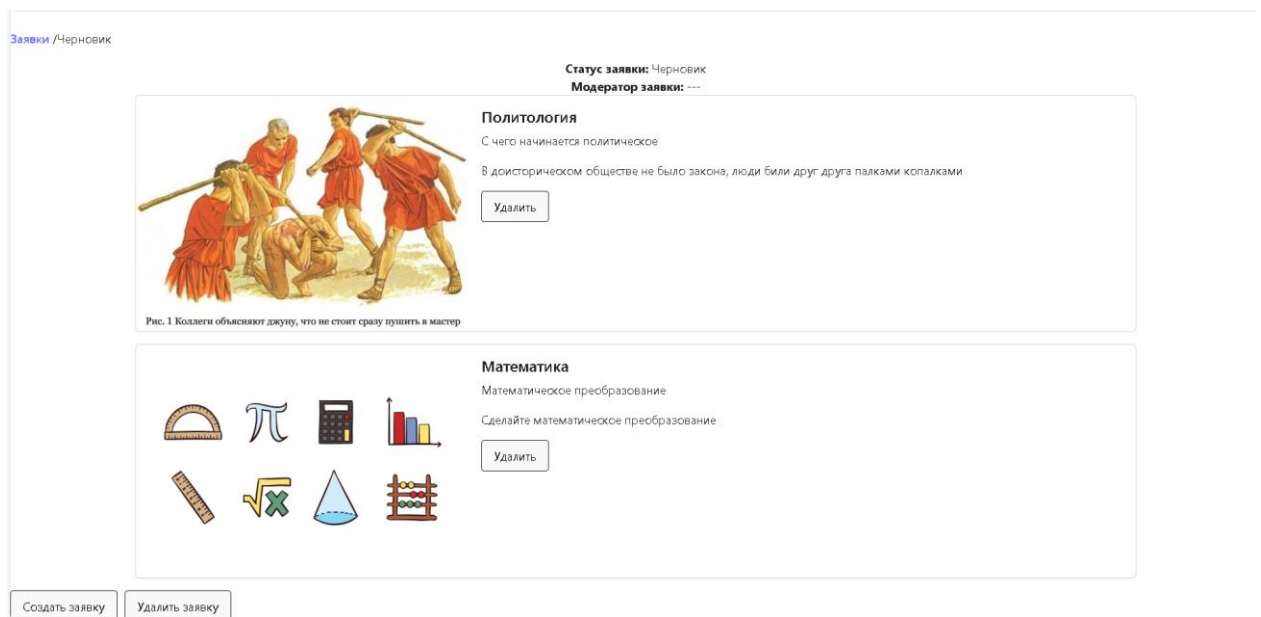


Рисунок 14 - Страница корзины

Также в корзине можно поменять порядок заданий перетаскивая задание на нужное место зажатием мышки



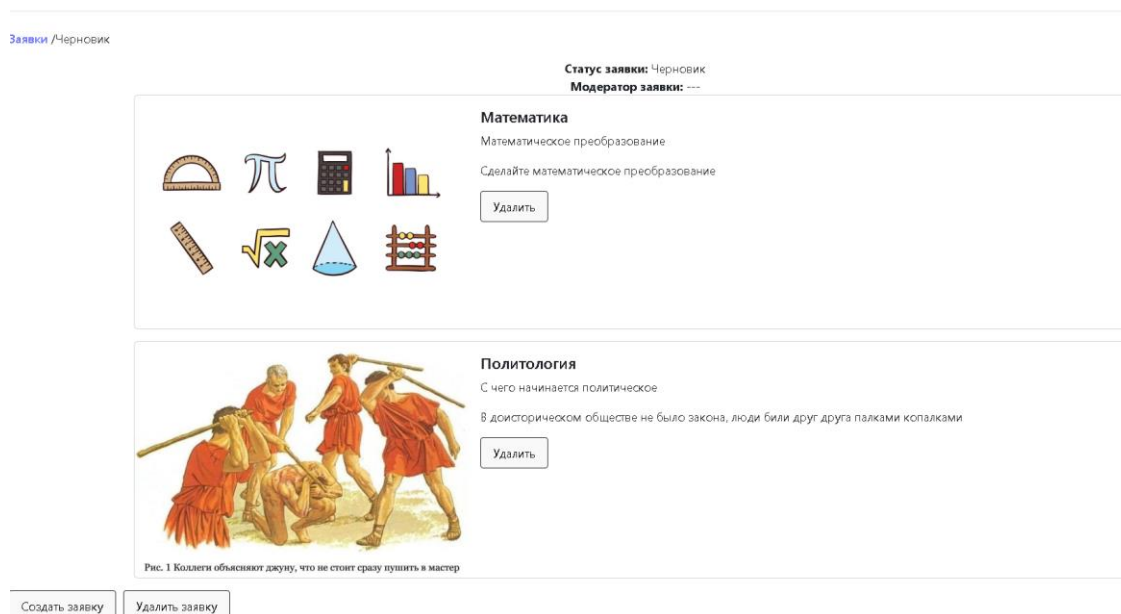


Рисунок 15 - Страница корзины (измененной)

На странице с подробным описанием задания (рис. 16) отображается подробная информация о задании: название, предмет, полное описание.

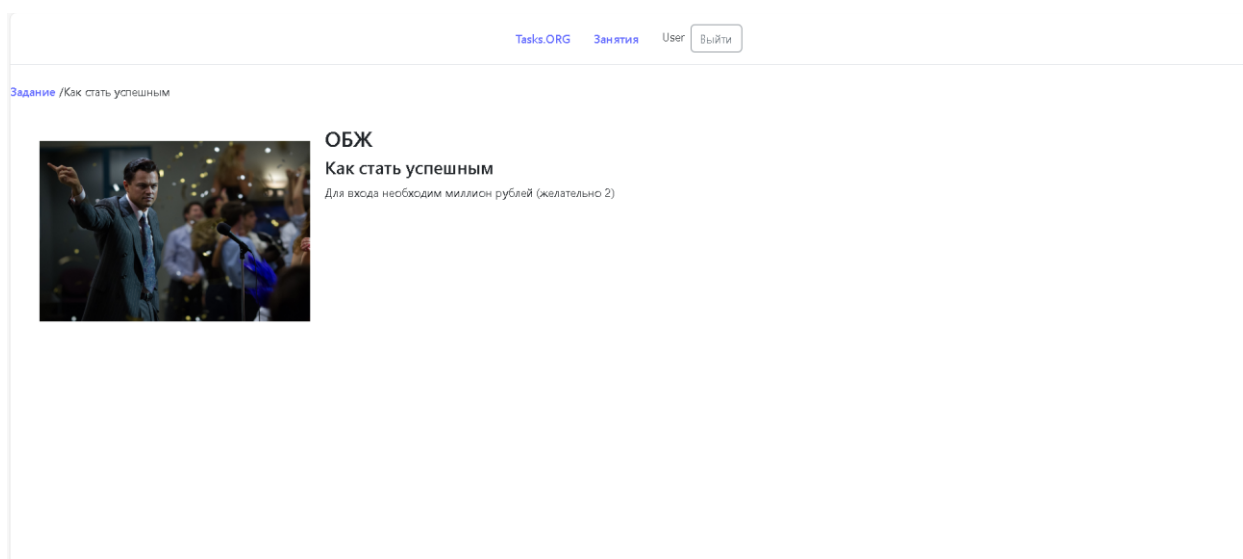


Рисунок 16 - Страница с подробным описанием задания

На странице со списком занятий (рис. 17, 18) отображается список занятий. В зависимости от типа пользователя этот список будет функционально отличаться. Так, для преподавателей отображается список созданных ими занятий: статус, дата создания, дата формирования, дата

завершения, сотрудник деканата, если он уже есть, который подтвердил или отклонил занятие.

Tasks.ORG Заявки User Выйти

Фильтрация

ДД.ММ.ГГГГ ДД.ММ.ГГГГ All

Заявки

Дата создания	Дата формирования	Дата завершения	Статус	Модератор
12-01-2024 02:57:03	12-01-2024 12:40:18	---	На проверке	---
10-01-2024 22:20:41	12-01-2024 01:57:41	---	На проверке	---
07-01-2024 04:00:21	10-01-2024 22:20:04	---	На проверке	---
06-01-2024 22:59:46	06-01-2024 23:00:43	---	На проверке	---
25-12-2023 14:49:59	25-12-2023 14:50:10	---	На проверке	---
25-12-2023 08:22:45	25-12-2023 08:24:47	---	На проверке	---
25-12-2023 08:22:01	25-12-2023 08:22:06	---	На проверке	---
25-12-2023 03:43:30	25-12-2023 08:18:14	---	На проверке	---
01-05-2023 03:00:00	05-05-2023 03:00:00	06-12-2023 12:12:02	Завершена	Charlie Brown
01-04-2023 03:00:00	05-04-2023 03:00:00	10-04-2023 03:00:00	Завершена	Charlie Brown

Активация Windows

Рисунок 17 - Страница со списком занятий (преподаватель)

Для сотрудника деканата функционал этой страницы шире. Для него отображается список всех занятий всех преподавателей. А также сотрудник может отфильтровать занятия по: начальной дате формирования, конечной дате формирования и статусу занятия.

Tasks.ORG Занятия Задания Rock Выйти

Фильтрация

ДД.ММ.ГГГГ ДД.ММ.ГГГГ Все Поиск по пользователю

Занятия

Дата создания	Дата формирования	Дата завершения	Статус	Пользователь	Модератор	Подтвердить	Отклонить
17-01-2024 21:10:52	21-01-2024 17:44:02	---	На проверке	User	---	Подтвердить	Отклонить
17-01-2024 21:10:51	17-01-2024 21:11:39	20-01-2024 14:12:53	Завершена	User	Rock	---	---
17-01-2024 21:10:51	20-01-2024 14:12:20	---	На проверке	User	---	Подтвердить	Отклонить
17-01-2024 19:54:04	17-01-2024 19:55:30	17-01-2024 19:55:45	Завершена	User	Rock	---	---
17-01-2024 16:57:53	17-01-2024 16:59:07	17-01-2024 16:59:42	Завершена	User	Rock	---	---
17-01-2024 16:47:38	17-01-2024 16:47:51	17-01-2024 16:48:05	Завершена	Rock	Rock	---	---
17-01-2024 16:10:23	17-01-2024 16:10:33	17-01-2024 16:11:00	Завершена	User	Rock	---	---
17-01-2024 13:04:41	17-01-2024 16:47:08	17-01-2024 16:47:27	Завершена	Rock	Rock	---	---
17-01-2024 12:59:00	17-01-2024 12:59:29	17-01-2024 13:00:11	Завершена	User	Rock	---	---
17-01-2024 12:16:38	17-01-2024 12:16:47	17-01-2024 17:23:45	Отклонена	User	Rock	---	---
17-01-2024 12:16:19	17-01-2024 12:16:27	17-01-2024 13:00:17	Отклонена	User	Rock	---	---

Рисунок 18 - Страница со списком занятий (сотрудник деканата)

На странице с таблицей заданий (рис. 19) сотрудник деканата может в компактном и удобном формате просмотреть список всех заданий, существующих в системе. Отображаются следующие поля: название, предмет, миниописание и описание. Также можно перейти к окну создания нового задания (рис. 20) или редактирования старого (рис. 21).

Tasks.ORG Занятия Задания Rock Выйти

Задания

Создать задание

Название	Предмет	Миниописание	Описание	Редактирование	Удаление
Как стать успешным	ОБЖ	Я успешный, и вы можете	Для входа необходим миллион рублей (желательно 2)	Редактировать	Удалить
Составление header	РИП		Научиться составлять headers и его поля в запросе	Редактировать	Удалить
Лабораторная по вскрытию	Биология		Принести и вскрыть лягушку	Редактировать	Удалить
Математическое преобразование	Математика	преобразуй	Сделайте математическое преобразование	Редактировать	Удалить
Разработка интернет приложения	РИП		Разработать интернет приложение с возможностью добавить услуги в заявки и заявки для заказчика Канева А.И.	Редактировать	Удалить
С чего начинается политическое	Политология	Люди стадо	В доисторическом обществе не было закона, люди били друг друга палками копалками	Редактировать	Удалить
Решение диф. уравнений	Математика		Решить диф. уравнение: $y' = -2y$	Редактировать	Удалить
Эксперимент по генетике	Биология	Эксперимент по генетике	Проведите эксперимент по изучению законов наследования генетических признаков	Редактировать	Удалить
Анализ экономических показателей	Экономика	Анализ экономики	Проанализируйте ключевые экономические показатели страны за последний год	Редактировать	Удалить
Исследование языковых особенностей	Лингвистика	Исследование языка	Проведите исследование особенностей выбранного языка: фонетика, морфология, синтаксис	Редактировать	Удалить

Рисунок 19 - Страница с таблицей заданий

Tasks.ORG Занятия Задания Rock Выйти

Создание задачи

Название

Введите название

Предмет

Введите предмет

Миниописание (необязательно)

Введите миниописание

Описание

Введите описание

Сохранить

Рисунок 20 - Страница создания задания

## Редактирование задачи

Название

Как стать успешным

Предмет

ОБЖ

Миниописание

Я успешный, и вы можете

Описание

Для входа необходим миллион рублей (желательно 2)

Изображение

Выбер файл

Не выбран ни один файл

Сохранить

Рисунок 21 - Страница редактирования задания

## ЗАКЛЮЧЕНИЕ

В ходе работы были достигнуты следующие результаты:

1. Был разработан дизайн приложения с помощью набора стилей CSS и HTML тегов.
2. База данных была создана и расположена в docker контейнере.
3. Был создан веб-сервис.
4. Разработан интерфейс гостя с использованием технологии React Framework и подключен к веб-сервису.
5. Приложение интерфейса было развернуто на сервисе Github Pages по ссылке [https://dedeimos.github.io/tasks\\_front/](https://dedeimos.github.io/tasks_front/)
6. В веб-сервис добавлена авторизация с использованием JWT токенов, а методы задокументированы через Swagger.
7. Был реализован интерфейс преподавателя. Доступ к нему имеют только авторизированные пользователи.
8. Выделенный сервис был разработан и развернут в отдельном docker кластере.
9. Реализован интерфейс сотрудника деканата для подтверждения занятий и создания новых заданий
10. Было реализовано десктопное приложение на Tauri, повторяющее интерфейс веб-приложения на React.
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор UML диаграмм.
12. Исходный код проекта доступен в GitHub.  
[https://github.com/DeDeimos/tasks\\_back](https://github.com/DeDeimos/tasks_back)

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Проблема выбора образовательного веб-ресурса в современном вузе: Непрерывное образование: XXI век [Электронный ресурс] // Petsru. URL: <https://lll21.petsru.ru/journal/article.php?id=3407> (дата обращения 13.10.2023)
2. Социально-педагогическая работа в образовательных организациях [Электронный ресурс] // Zabgu. URL: [https://zabgu.ru/files/html\\_document/pdf\\_files/fixed/MP\\_Social%27no\\_pedagogicheskaya\\_rabota\\_v\\_obrazovatel%27ny%27x\\_organizatsiyah/Sovremennye\\_problemy\\_nauki\\_i\\_obrazovaniya.pdf](https://zabgu.ru/files/html_document/pdf_files/fixed/MP_Social%27no_pedagogicheskaya_rabota_v_obrazovatel%27ny%27x_organizatsiyah/Sovremennye_problemy_nauki_i_obrazovaniya.pdf) (дата обращения 15.10.2023).
3. JavaScript полное руководство [Книга] // Д.Флэнаган. (дата обращения 11.10.2023).
4. Полное практическое руководство по Docker [Электронный ресурс] // Habr. URL: <https://habr.com/ru/articles/310460/> (дата обращения: 15.11.2023).
5. Руководство по React [Электронный ресурс] // Habr. URL: <https://habr.com/ru/companies/ruvds/articles/428077/> (дата обращения: 11.10.2023).
6. Quick Start – React [Электронный ресурс] // React. URL: <https://react.dev/learn> (дата обращения: 12.11.2023).
7. Документация PostgreSQL [Электронный ресурс] // PostgreSQL. URL: <https://postgrespro.ru/docs/postgresql> (дата обращения: 20.10.2023).
8. Документация Redux [Электронный ресурс] // Redux URL: <https://reactdev.ru/libs/redux/> (дата обращение 10.12.2023)
9. Документация MinIO [Электронный ресурс] // MinIO URL: <https://min.io/docs/minio/linux/index.html> (дата обращения: 30.10.2023)

## **Приложение. Техническое задание**



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

Техническое задание

Тема: «Составление занятий»

Студент: Толкачев Р.М.

Группа РТ5-51Б

Преподаватель: Канев А.И.

2023г.



## **1. Цель работы**

Реализовать систему для составления преподавателем списка занятий со списком заданий для студентов (учеников), включающая в себя веб-сервис, веб-приложение, десктопное приложение и асинхронный сервис формирования занятий.

## **2. Назначение**

Основное назначение разработанной системы заключается в автоматизации процесса создания комплекса занятий. Система помогает создать комплекс занятий, чтобы упростить для преподавателя составление списка заданий, необходимых для выполнения студентами. Система предоставляет автоматизированный способ создания, учета и ведения занятий. Также она позволяет сотрудникам деканата принимать или отклонять занятия. Сотрудники имеют возможность редактировать существующие и создавать новые задания.

## **3. Задачи:**

1. Создание дизайна
2. Создать базу данных
3. Создать веб-сервис
4. Реализовать интерфейс гостя
5. Подключить интерфейс к веб-сервису
6. Задеплоить на Github Pages
7. Написать документацию методов API
8. Добавить авторизацию и аутентификацию
9. Реализовать интерфейс преподавателя
10. Реализовать интерфейс сотрудника деканата
11. Создать нативное приложение

#### 4. Методы веб-сервиса:

№	Метод	Описание	URL	Входные данные	Выходные данные
а. Методы заданий					
4.1.1	GET	Возвращает список заданий, удовлетворяющих переданным критериям  Будут возвращены id чернового задания для текущего пользователя и задания со статусом: active  Доступно неавторизованным пользователям	api/tasks/	{ "title": char(100), }	{ "Request_id": int [ { "id": int, "name": char(100) "status": char(8), "subject": char(20), "minidsescription": char(8), "image": char(40), "description": char(8), }, ... ] }
4.1.2	GET	Возвращает задание  Доступно неавторизо	api/tasks/<task_id >/	{ "task_id": int }	{ "id": int, "name": char(100) "status": char(8), "subject": char(20),

		ванным пользовате лям			“minidsescription”: char(8), “image”: char(40), “description”: char(8), }
4.1.4	DELETE	Удаляет конкретное задание  Доступно только сотруднику деканата	api/tasks/delete/<task_id>/	{ “task_id”: int }	
4.1.5	POST	Создает новое задание Доступно только сотруднику деканата	api/tasks/create	{ “name”: char(100) “status”: char(8), “subject”: char(20), “minidsescription”: char(8), “image”: char(40), “description”: char(8), },	
4.1.6	PUT	Добавляет картинку занятия в хранилище Minio  Доступно только сотруднику деканата	api/tasks/<task_id>/add-image	{ “image”: file.jpg },	

4.1.7	POST	Добавляет задание в черновое занятие  Доступно только авторизованным пользователям	api/tasks/:<task_id>/add-to-request	{ "task_id": int }	
4.1.8	PUT	Изменяет задание  Доступно только сотруднику деканата	api/tasks/update/<task_id>/	{ "name": char(100) "status": char(8), "subject": char(20), "minidsdescription": char(8), "image": char(40), "description": char(8), },	
b. Методы занятий					
4.2.1	GET	Возвращает список занятий	api/requests	{ "startDate": string, "endDate": string, "status": string }	[ { "Request_id": int, "Status": string, "start_date": time.Time, "formation_date": time.Time, "end_date": time.Time, "UserID": int,

					<pre>       "ModeratorID": int,       "User": {         "Name": string,       },       "Moderator": {         "Name": string,       },       "Tasks": null,     },   ] </pre>
4.2.2	GET	<p>Возвращает конкретное занятие</p> <p>Доступно только авторизованным пользователям</p>	api/requests/<request_id>	<pre> {   "request_id": int } </pre>	<pre> [   {     "Request_id": int,     "Status": string,     "start_date": time.Time,     "formation_date": time.Time,     "end_date": time.Time,     "UserID": int,     "ModeratorID": int,     "User": {       "Name": string,     },     "Moderator": {       "Name": string,     },     "Tasks": [{       "id": int,       "name": char(100) </pre>

					"status": char(8), "subject": char(20), "minidsdescription": char(8), "image": char(40), "description": char(8), }, {...}], },]
4.2.3	DELETE	Удаляет конкретное занятие	api/requests/delete/<request_id>	{ "request_id": int }	
4.2.4	PUT	Изменяет статус на завершен или отклонен Только сформированные занятия Доступно только сотрудникам деканата	api/ /requests/admin/<request_id>/update-status	{ "request_id": int, "status": string, }	
4.2.5	PUT	Сформируется черновое занятие Доступно только авторизованным	api/ /requests/user/<request_id>/update-status	{ "request_id": int, }	

		пользовате лям			
4.2.6	POST	Устанавлив ает статус формирова ния занятия. Вызываетс я внешним сервисом.	/api/requests/<req uest_id>/sending	{ id: string, token: string, status: string }	{ request_id: string, status: string }
с. Методы связи занятия и задания					
4.3.1	DELETE	Удаляет связь между занятием и заданием Доступно только авторизова нным пользовате лям	api/task- request/delete/tas k/<task_id>/reque st/<request_id>/	{ “request_id”: int, “task_id”: int, }	
4.3.2	POST	Изменения порядка задания в занятии	api/task- request/delete/tas k/<task_id>/reque st/<request_id>/or der/<order>	{ “request_id”: int, “task_id”: int, “order”: int }	
4.4.Методы авторизации и аутентификации					
4.4.1	POST	Регистриру ет нового пользовате ля в базе данных	api/auth /register	{ “email”: char(100), “name”: char(100),	{ “access_token”: char(100) }

				<pre> “password”: char(100) } </pre>	
4.4.2	POST	Выполняет аутентификацию пользователя	api/auth /login	<pre> {   “email”: char(100),   “password”: char(100) } </pre>	<pre> {   “access_token”: char(100) } </pre>
4.4.3	POST	Выполняет выход из аккаунта пользователя и сброс его токена	api/auth /logout	<pre> {   “access_token”: char(100) } </pre>	
4.5.Методы внешнего сервиса					
4.5.1	POST	Отслеживает формирование занятий. Вызывается основным сервисом	/api/sending	<pre> {   request_id: string } </pre>	

## 5. Функциональные требования:

5.1. Главное меню. Здесь находятся кнопки для регистрации и авторизации или информация о текущем пользователе, и меню со списком доступных пунктов.

5.1.1. Доступна всем пользователям

5.1.2. Действия

5.1.2.1. Войти – переход на страницу 5.9. Только для гостей.

5.1.2.2. Зарегистрироваться – переход на страницу 5.10. Только для гостей.



- 5.1.2.3. Открыть список заданий – переход на страницу 5.2.  
Для всех пользователей.
- 5.1.2.4. Открыть список занятий – переход на страницу 5.5.  
Только для авторизованных пользователей.
- 5.1.2.5. Открыть список заданий – переход на страницу 5.8.  
Только для модераторов.
- 5.2. Страница со списком заданий. На этой странице располагается список всех заданий и поле ввода для поиска задания по предмету.
  - 5.2.1. Доступна всем пользователям
  - 5.2.2. Выводится информация о заданиях в виде карточек (метод 4.1.1.).
    - 5.2.2.1. Изображение предмета (поясняющее).
    - 5.2.2.2. Название задания
    - 5.2.2.3. Предмет задания
    - 5.2.2.4. Краткое описание задания
  - 5.2.3. Действия
    - 5.2.3.1. Отфильтровать задания по введенному предмету (метод 4.1.1).
    - 5.2.3.2. Подробнее - перенаправляет на страницу 5.3.
- 5.3. Страница с подробным описанием задания
  - 5.3.1. Доступна всем пользователям.
  - 5.3.2. Выводится информация о задании
    - 5.3.2.1. Изображение предмета (поясняющее).
    - 5.3.2.2. Название задания
    - 5.3.2.3. Предмет задания
    - 5.3.2.4. Подробное описание задания
  - 5.3.3. Действия
    - 5.3.3.1. Добавить задание в черновое занятие (метод 4.1.7.). Только для авторизованных пользователей.
    - 5.3.3.2. Задания – перенаправляет на страницу 5.2.

#### 5.4. Страница со списком занятий

5.4.1. Доступна только авторизированным пользователям

5.4.2. Выводится информация о занятиях в табличной форме (метод 4.2.1).

5.4.2.1. Статус занятия.

5.4.2.2. Дата создания.

5.4.2.3. Дата формирования.

5.4.2.4. Дата завершения.

5.4.2.5. Сотрудник деканата

5.4.2.6. Список выбранных заданий

5.4.3. Действия

5.4.3.1. Открыть занятия – перенаправляет на страницу

5.4.

5.4.3.2. Установить занятию статус «отклонено» или «завершено» (метод 4.2.4.). Только для сотрудников деканата.

5.4.3.3. Отфильтровать список занятий по диапазону дат и/или статусу занятий. Доступно всем пользователям

5.5. Страница с подробным описанием занятия. Пользователь может удалить задания в занятии.

5.5.1. Доступна только для авторизованных пользователей.

5.5.2. Выводится информация о занятии в карточной форме. Список заданий, выбранных в занятии, в виде карточек (метод 4.2.2.).

5.5.2.1. Изображение предмета (поясняющее).

5.5.2.2. Название задания

5.5.2.3. Предмет задания

5.5.2.4. Краткое описание задания

5.5.3. Действия

5.5.3.1. Удалить задание из занятия (метод 4.3.1.). Если занятие находится в статусе черновика. Только для владельца занятия

5.5.3.2. Сформировать занятие (метод 4.2.5.). Если занятие находится в статусе черновика. Только для владельца занятия

5.5.3.3. Удалить занятие (метод 4.2.5.). Если занятие находится в статусе черновика. Только для владельца занятия

5.6. Страница редактирования и создания задания. Сотрудник деканата может изменить существующее или создать новое задание.

5.6.1. Доступна только сотрудникам деканата.

5.6.2. Выводится информация о задании в построчном формате.

5.6.2.1. Изображение предмета (поясняющее).

5.6.2.2. Название задания

5.6.2.3. Предмет задания

5.6.2.4. Краткое описание задания

5.6.2.5. Подробное описание задания

5.6.3. Действия

5.6.3.1. Изменение полей задания. Можно изменять все вышеперечисленные поля задания и изображение прикрепленное к заданию.

5.6.3.2. Сохранить изменения (методы 4.1.5, 4.1.8).

5.7. Страница с таблицей заданий. Предоставляет сотруднику деканата удобный способ отображения всех заданий.

5.7.1. Доступна только сотрудникам деканата.

5.7.2. Выводит информацию о всех заданиях в табличном виде (метод 4.1.1.).

5.7.2.1. Изображение предмета (поясняющее).

5.7.2.2. Название задания

5.7.2.3. Предмет задания

5.7.2.4. Краткое описание задания

5.7.2.5. Подробное описание задания

5.7.3. Действия

- 5.7.3.1. Открыть задание - перенаправляет на страницу 5.3.
- 5.7.3.2. Добавить задание – перенаправляет на страницу 5.6.
- 5.8. Страница с формой авторизации. На этой странице находится форма авторизации.
  - 5.8.1. Доступна гостям
  - 5.8.2. Действия
    - 5.8.2.1. Войти – производит запрос (метод 4.4.2.).
    - 5.8.2.2. Зарегистрироваться – перенаправляет на страницу 5.9.
- 5.9. Страница с формой регистрации
  - 5.9.1. Доступна гостям
  - 5.9.2. Действия
    - 5.9.2.1. Зарегистрироваться – производит запрос (метод 4.4.1).
    - 5.9.2.2. Войти – перенаправляет на страницу 5.8.

## **6. Требования к программному обеспечению:**

- 6.1. Серверная часть
  - 6.1.1. ОС: Linux (6.4 и выше)/Windows (7 и выше)
  - 6.1.2. Docker 24.0.8
  - 6.1.3. Докер образы
  - 6.1.4. Redis 6.2-alpine
  - 6.1.5. Minio RELEASE.2022-10-15T19-57-03Z (go1.21.3 linux/amd64)
  - 6.1.6. Golang (1.20 и выше)
  - 6.1.7. PostgreSQL (12.0 и выше)
- 6.1. Клиентская часть
  - 6.1.1. ОС: Windows (7 и выше)/MacOS (12.0 и выше)/Linux (6.4 и выше)
  - 6.1.2. Веб-браузер: Safari 11.1+/Chrome 40+/Opera 27+/Firefox 44+

## **7. Требования к аппаратному обеспечению:**

- 7.1. Серверная часть

7.1.1. Процессор минимум 4-ядерный с частотой от 2 ГГц

7.1.2. Оперативная память от 8 Гб

7.1.3. Место на жестком диске от 1 Гб

7.2. Клиентская часть

7.2.1. Процессор с частотой от 2ГГц

7.2.2. Оперативная память от 1 Гб