|  |  |
| --- | --- |
| **­­** | **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 \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ДОМАШНЕМУ ЗАДАНИЮ***

***НА ТЕМУ:***

***Определение индекса тяжести панкреатита по\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рэнсону \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_ИУ5-56Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Турсков \_\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.И. Канев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2025 г.*

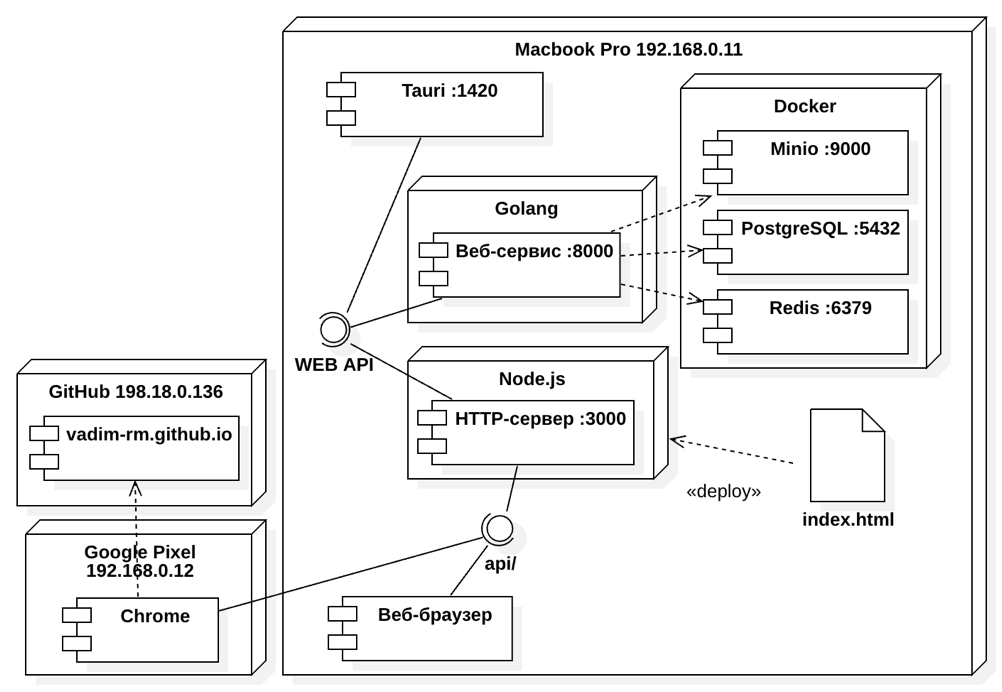
# АННОТАЦИЯ

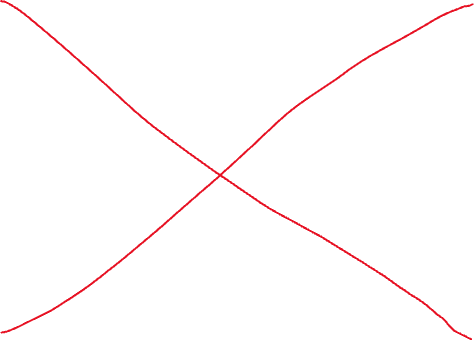
# СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

# ПРЕДМЕТНАЯ ОБЛАСТЬ

# АРХИТЕКТУРА

Архитектура системы отображена на диаграмме развертывания (рисунок 4). Рисунок 4 – Диаграмма развертывания



Веб-сервис, реализованный на языке Golang [3] связан с серверами Minio [6], Redis [5] и СУБД PostgreSQL [7]. В Redis хранятся отозванные JWT [4] пациентов, добавление которых происходит при выходе пациента из системы.

Язык программирования Golang был выбран благодаря его высокой производительности и простоте синтаксиса. Использование стандартной библиотеки Go и встроенных инструментов для работы с сетевыми запросами позволяет создавать надежные и эффективные веб-сервисы. Веб-сервис на Golang является общим для веб-сервера и десктопного приложения Tauri [8].

Данные хранятся в СУБД PostgreSQL. Она была выбрана, является стандартом современной индустрии разработки.

Структура данных отражена на ER диаграмме (рисунок 5). Модель договоров представляет собой набор полей, отражающих параметры договора. Данные о договоре хранятся в таблице contracts. Для хранения в одной заявке на счёт нескольких договоров используется промежуточная таблица связи М-М account\_contracts. Таблица accounts представляет собой список заявок на открытие счетов. Данные о пациентах и врачах системы хранятся в таблице users.

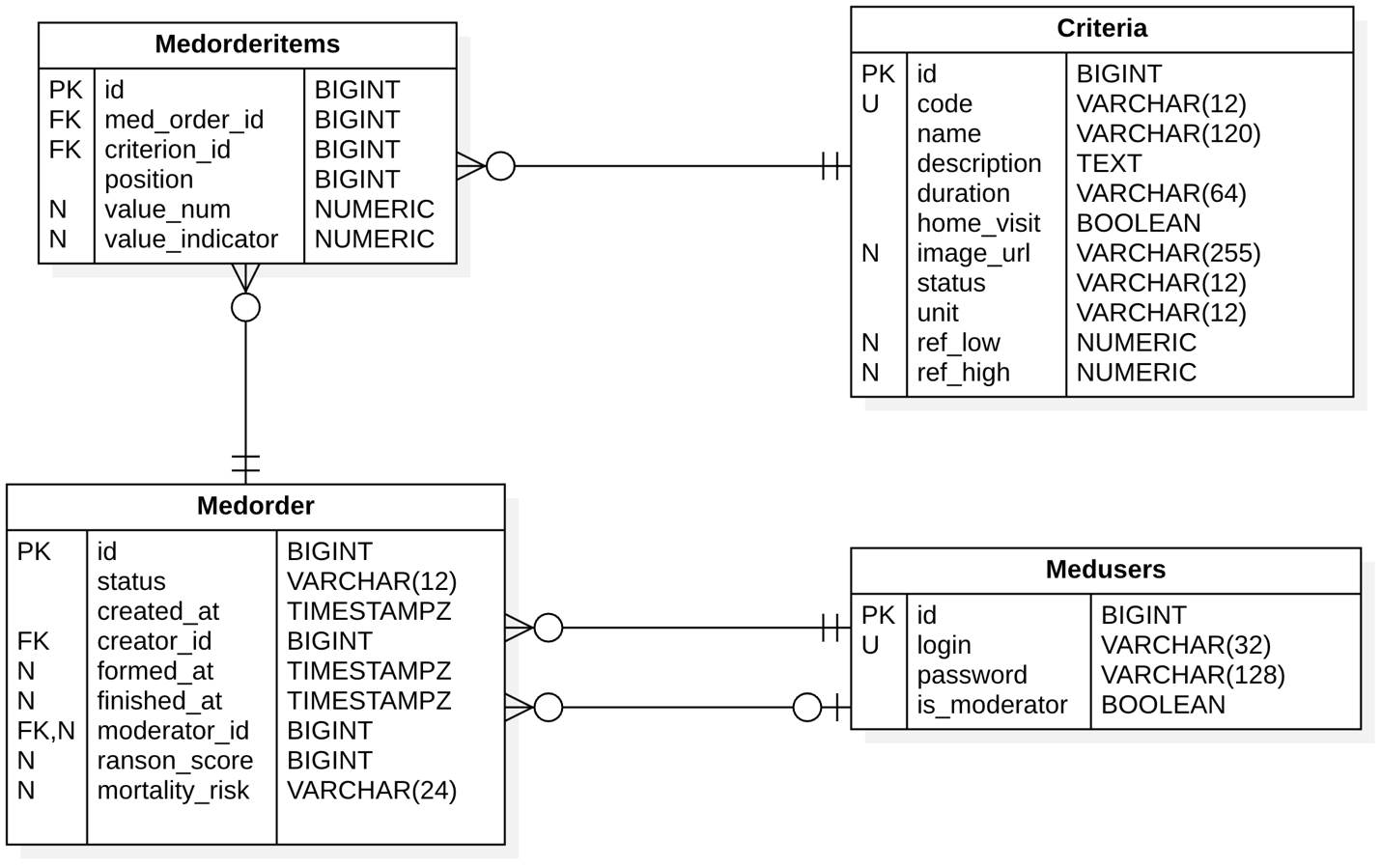


Рисунок 5 – ER диаграмма.

Веб-сервер реализован с использованием языка TypeScript и фреймворка React [7]. Устройство бэкенда приложения разработанной системы приведено на диаграмме классов бэкенда (рисунок 6). Пациенты и врачи взаимодействуют с доменами. Домены связаны с моделями. Модели имеют связи с таблицами в базе данных.

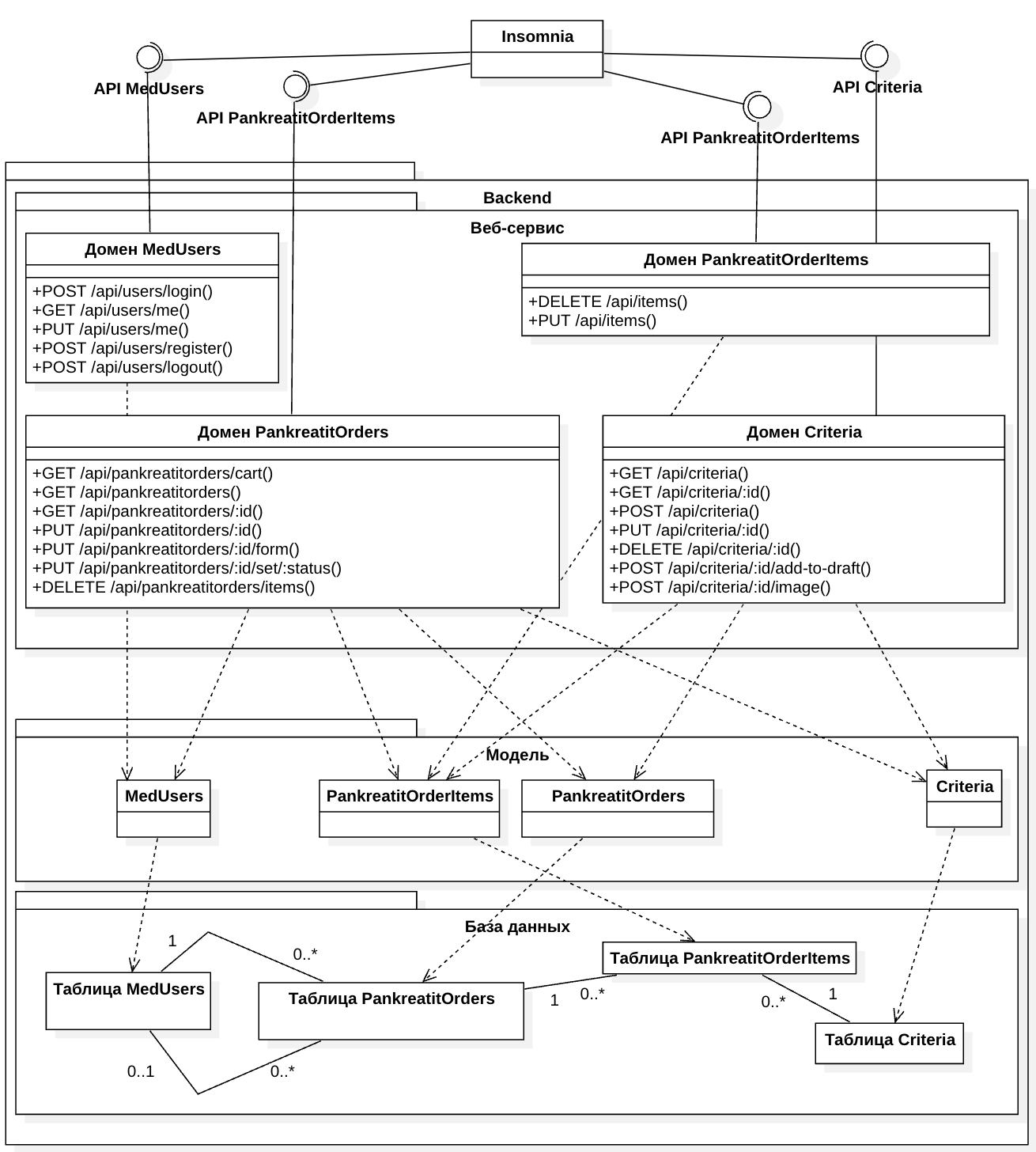
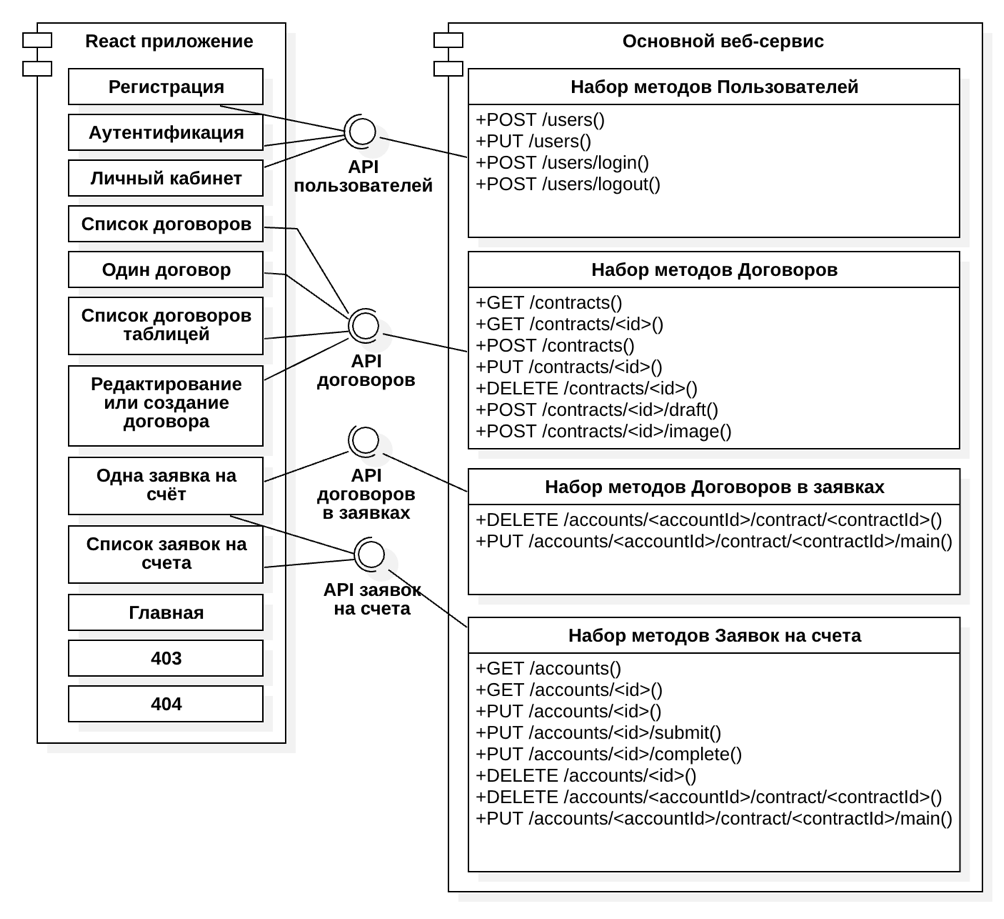


Рисунок 6 – Диаграмма классов бэкенда

Связь фронтенда и бэкенда отражена на диаграмме классов фронтенда (рисунок 7). Каждая страница связана с API, которое используется для взаимодействия с данными на соответствующей странице.



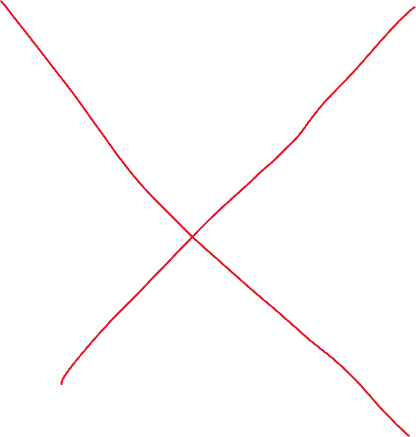


Рисунок 7 – Диаграмма классов фронтенда

# АЛГОРИТМЫ

Алгоритм работы разработанной системы отображен на диаграмме последовательности (рисунок 8). В основе системы лежит веб-сервис, реализующий внутри себя всю бизнес-логику. Он предоставляет доступ к методам из следующих доменов: пользователи (пациенты и врачи), договоры, заявки на счета, договоры в заявках на счета. Методы следуют правилам REST API.

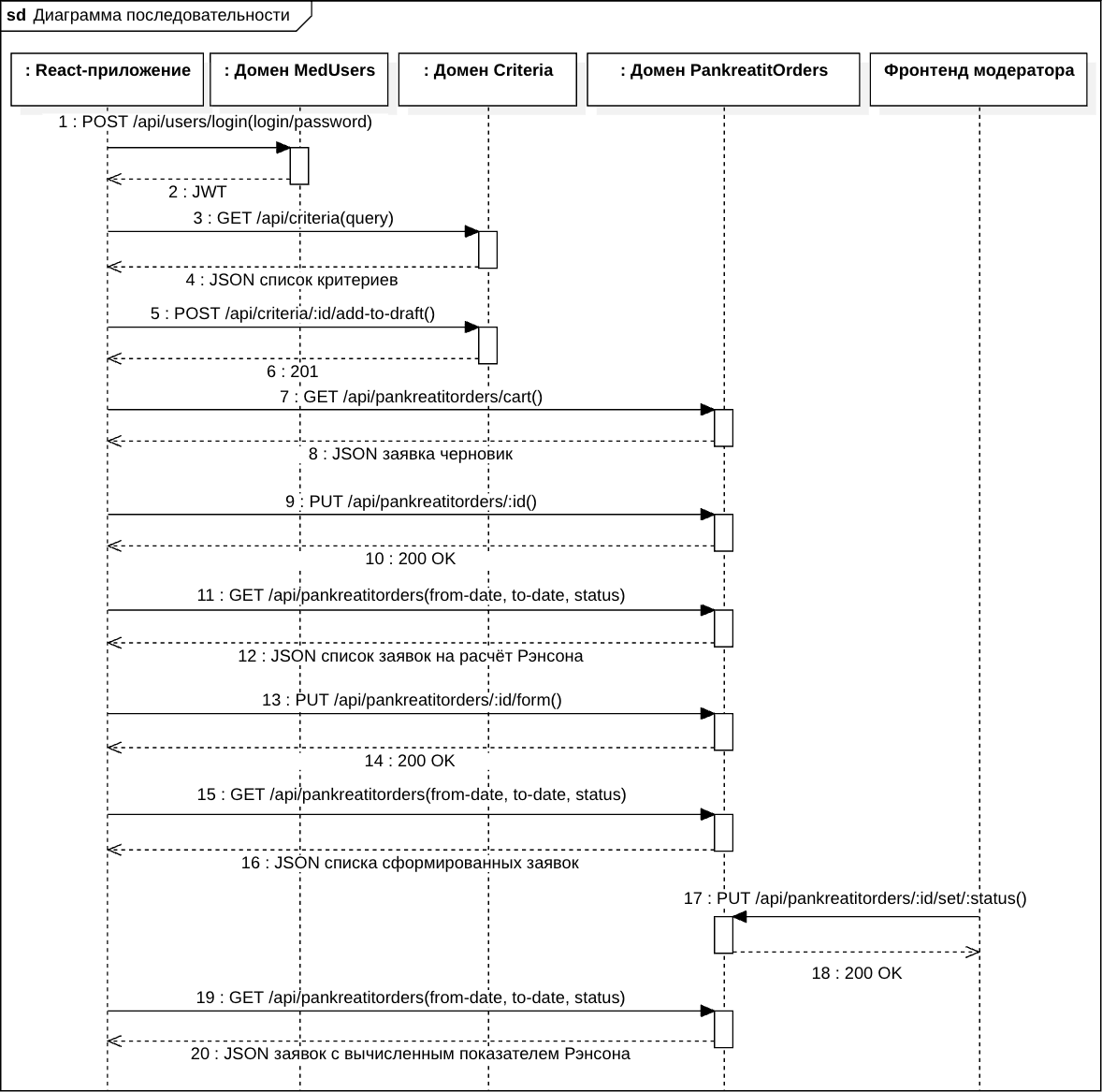


Рисунок 8 – Диаграмма последовательности

# ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2025 г. |  | \_\_\_\_\_\_\_\_\_\_А.И. Канев  "\_\_"\_\_\_\_\_\_\_\_\_\_2025 г. |

**Определение индекса тяжести панкреатита по Рэнсону**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

20

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| Турсков Евгений |  |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2025 г. |  |

Москва - 2024

## Введение

Необходимо разработать систему, включающую в себя веб-сервис, веб-приложение и десктопное приложение на Tauri, которая позволит медицинскому учреждению (врачам) отслеживать состояние пациентов по заявкам, а пациентам – оставлять их.

## Назначение разработки

Основное назначение разработки — создание единой платформы, объединяющей пациента и врача, где осуществляется полный цикл работы с заявками: от формирования черновика до подтверждения расчёта и анализа итоговых данных.

Пациенты могут создавать и редактировать заявки, вводить показатели и получать итоговые значения индекса тяжести. Врачи имеют возможность просматривать все созданные заявки, проверять корректность введённых данных, утверждать расчёты либо отклонять их при наличии ошибок.

## Стадии и этапы разработки

### Создание MVP и базового дизайна на основе [invitro.ru](https://www.invitro.ru/);

### Создание базы данных для хранения информации об этапах производства;

### Создание веб-сервиса в бэкенде на Go;

### Реализация авторизации и хранения сессий в Redis;

### Разработка базового SPA на React для гостя;

### Внедрение адаптивности, разработка Tauri приложения;

### Создание интерфейса пациента в React с менеджером состояний Redux Toolkit;

### Реализация интерфейса врача в React;

### Разработка нативного приложения;

### Развертка приложения при помощи GitHub Pages;

### Подготовка набора документации, включающий РПЗ, ТЗ и набор диаграмм;

### Оформление git-репозитория на сервисе GitHub, содержащего исходный код проекта;

## Требования к функциональным характеристикам

### Методы HTTP

#### POST — регистрация нового пациента;

#### POST — вход в систему (аутентификация пациента по логину и паролю);

#### POST — выход из аккаунта с добавлением токена в blacklist;

#### GET — получение полей пациента (текущего из JWT);

#### PUT — изменение данных пациента (например, имени, пароля);

#### GET — получение списка доступных медицинских критериев;

#### GET — получение информации об одном критерии (например, «Глюкоза», «Кальций»);

#### POST — добавление нового критерия (услуги) в систему (для врача);

#### PUT — изменение информации о критерии (например, референсных значений);

#### DELETE — удаление критерия из системы (для врача);

#### POST — загрузка или обновление изображения критерия (например, иконки лабораторного показателя);

#### POST — добавление критерия в заявку (через «Добавить в черновик»);

#### GET — получение черновой заявки (для кнопки корзины);

#### GET — получение списка всех заявок (расчётов критериев Рэнсона);

#### GET — получение одной заявки с детальной информацией о расчёте;

#### PUT — изменение данных заявки (редактирование показателей);

#### PUT — формирование заявки (перевод из черновика в статус «сформирована»);

#### PUT — изменение статуса заявки (завершение или отклонение врачом);

#### DELETE — удаление заявки (в статусе «черновик»);

#### DELETE — удаление критерия из конкретной заявки;

#### PUT — изменение показателей критерия (обновление диапазонов, единиц измерения).

### Регистрация

#### Доступно только неавторизованным пациентам (гостям);

#### Отображает форму регистрации с полями:

##### Логин (уникальное имя пациента);

##### Пароль (минимум 6 символов);

#### Действия:

##### Регистрация пациента (вызывает метод регистрации);

##### Переход к странице аутентификации.

### Аутентификация

#### Доступно только гостям;

#### Отображает форму входа в систему с полями:

##### Логин;

##### Пароль;

#### Действия:

##### Вход — выполняет авторизацию пациента;

##### Регистрация — перенаправляет на форму регистрации.

### Личный кабинет

#### Доступно авторизованным пациентам и врачам;

#### Действия:

##### Изменить данные пациента — выполняет запрос на обновление;

##### Выйти из аккаунта — выполняет запрос на выход.

### Главная страница

#### Доступно всем;

#### Содержит статическую информацию:

##### Назначение сервиса (описание системы и принципа расчёта Рэнсона);

##### Контакты и справочную информацию.

### Список медицинских показателей

#### Доступно всем;

#### Отображается список доступных критериев (услуг) с полями:

##### Название (например, «Возраст», «Глюкоза»);

##### Единицы измерения;

##### Референсные значения (норма);

##### Изображение;

#### Действия:

##### Поиск по названию — фильтрация списка;

##### Просмотр подробностей — переход к разделу критерия;

##### Добавить в заявку — добавление показателя в черновик;

##### Перейти к заявке — переход к черновику.

### Один медицинский показатель

#### Доступно всем;

#### Отображается подробная информация по выбранному критерию.

### Черновик заявки (расчёт Рэнсона)

#### Доступно авторизованным пациентам;

#### Отображает черновик заявки пациента;

#### Содержит:

##### Список выбранных показателей;

##### Поля для ввода значений (глюкоза, ЛДГ, возраст и др.);

#### Действия (доступны при статусе «черновик»):

##### Удалить показатель;

##### Сохранить изменения;

##### Отправить на расчёт (формирование заявки);

##### Удалить заявку.

### Список заявок

#### Доступно авторизованным пациентам и врачам;

#### Отображается список всех заявок с фильтрацией по дате и статусу;

#### Действия:

##### Фильтрация — поиск по дате, статусу, пациенту;

##### Завершить — отметить заявку как выполненную;

##### Отклонить — отметить заявку как отклонённую;

##### Просмотреть — открыть подробную информацию.

### Список медицинских критериев

#### Доступно врачам;

#### Отображается таблица всех доступных критериев;

#### Действия:

##### Удалить критерий;

##### Изменить/создать — переход на страницу редактирования.

### Редактирование/создание критерия

#### Доступно врачам;

#### Отображается форма редактирования:

##### Название;

##### Описание;

##### Единица измерения;

##### Нормативные значения;

##### Изображение;

#### Действия:

##### Сохранить — добавление или обновление критерия;

##### Загрузить изображение.

### Страницы ошибок

#### 404 — отображается при отсутствии ресурса (например, при попытке открыть несуществующий критерий или заявку);

#### 403 — отображается при отсутствии прав доступа (например, если пациент пытается просмотреть заявку другого пациентам);

## Требования к составу и параметрам технических средств

### Серверная часть

#### Процессор минимум 6-ядерный с частотой от 3.3 ГГц

#### Оперативная память от 1 Гб

#### Место на жестком диске от 2 Гб

### Клиентская часть

#### Процессор с частотой от 23.3 ГГц

#### Оперативная память от 8 Гб

#### Место на жестком диске от 2 Гб

## Требования к информационной и программной совместимости

### Серверная часть

#### ОС Linux (6.4.12 и выше)

#### Minio (RELEASE.2023-12-14T18-51-57Z и новее)

#### Redis (7.2 и выше)

#### Golang (1.23 и выше)

#### СУБД PostgreSQL (16 и выше)

#### Доступ к интернету

### Клиентская часть

#### Браузер (Safari 16.5.2, Firefox 121.0, Chrome 119.0.6045, Yandex 24.6.3.729, Opera 105.0.4970.16)

# 

# ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ

Таблица 1 – HTTP методы разрабатываемого веб-сервиса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Тип** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| 4.1.1 | POST | /api/users/auth/register | Регистрация. Доступно всем. | {  login: string,  password: string  } | {  access\_token: string, token\_type: 'Bearer', expires\_in: int  } |
| 4.1.2 | POST | /api/users/auth/login | Вход в систему. Доступно всем. | {  login: string,  password: string  } | {  access\_token: string, token\_type: 'Bearer', expires\_in: int  } |
| 4.1.3 | POST | /api/users/auth/logout/{token} | Выход из аккаунта: добавление JWT в blacklist. Доступно врачу. | 200 Ok | {  status: 'ok', message: 'logout success'  } |
| 4.1.4 | GET | /api/users/me | Профиль текущего пациента. Доступно авторизованным. | Authorization: Bearer <jwt> | {  id: uint, login: string, is\_moderator: bool  } |
| 4.1.5 | PUT | /api/users/me | Изменение данных пациента (логин/пароль). Доступно авторизованным. | Authorization: Bearer <jwt>, Body:  {  login: string,  password: string  } | 201 Created |
| 4.1.6 | GET | /api/criteria | Список услуг (критериев) с фильтрацией по подстроке. Доступно всем. | query: string | { items: [ { id: uint, code: string, name: string, description: string, duration: string, home\_visit: bool, image\_url?: string, status: string, unit: string, ref\_low?: number,ref\_high?: number } ] } |
| 4.1.7 | GET | /api/criteria/{id} | Получить одну услугу по ID. Доступно всем. | path: { id: uint } | { id: uint, code: string, name: string, description: string, duration: string, home\_visit: bool, image\_url?: string, status: string, unit: string, ref\_low?: number, ref\_high?: number } |
| 4.1.8 | POST | /api/criteria | Создать услугу (без изображения). Доступно врачу. | Authorization: Bearer <jwt>, Body: {  code: string, name: string, description: string, duration: string, home\_visit: bool, status: 'active'|'deleted', unit:string, ref\_low: number, ref\_high: number  } | 201 Created |
| 4.1.9 | PUT | /api/criteria/{id} | Изменить поля услуги. Доступно врачу. | Authorization: Bearer <jwt>, path: { id: uint }, Body: {  code: string, name: string, description: string, duration: string, image\_url: string, home\_visit: bool, unit: string, ref\_low: number, ref\_high?: number  } | 200 OK |
| 4.1.10 | DELETE | /api/criteria/{id} | Удалить услугу (встроенное удаление изображения). Доступно врачу. | Authorization: Bearer <jwt>, path: { id: uint } | 200 OK |
| 4.1.11 | POST | /api/criteria/{id}/image | Загрузить/заменить изображение услуги (MinIO). Доступно врачу. | Authorization: Bearer <jwt>, path: { id: uint }, multipart/form-data: image: file(jpg/png/webp) | { status: 'ok', url: string } |
| 4.1.12 | POST | /api/criteria/{id}/add-to-draft | Добавить услугу в заявку-черновик пациента. Доступно авторизованным. | Authorization: Bearer <jwt>, path: { id: uint } | 200 OK |
| 4.1.13 | GET | /api/pankreatitorders/cart | Иконка корзины: ID черновика и количество услуг. Доступно авторизованным. | Authorization: Bearer <jwt> | { pankreatit\_order\_id: uint, criteria\_amount: uint } |
| 4.1.14 | GET | /api/pankreatitorders | Список заявок с фильтрацией по статусу и диапазону даты формирования. Доступно авторизованным. | Authorization: Bearer <jwt>, query: { status: 'formed'|'completed'|'rejected'|'draft'|'deleted', from\_date: RFC3339, to\_date: RFC3339 } | [ { id: uint, status: string, formed\_at?: string, finished\_at?: string, ranson\_score?: int, mortality\_risk?: string } ] |
| 4.1.15 | GET | /api/pankreatitorders/{id} | Получить одну заявку и её услуги. Доступно авторизованным. | Authorization: Bearer <jwt>, path: { id: uint } | { id: uint, status: string, creator\_id: uint, formed\_at?: string, finished\_at?: string, moderator\_id?: uint, ranson\_score?: int, mortality\_risk?: string, criteria: [ { id: uint, criterion\_id: uint, criterion: {...}, position: int, value\_num?: number, value\_indicator: bool } ] } |
| 4.1.16 | PUT | /api/pankreatitorders/{id} | Изменить поля заявки (по теме). Доступно врачу. | Authorization: Bearer <jwt>, path: { id: uint }, Body: { status: string, ranson\_score: int, mortality\_risk: string } | 200 OK |
| 4.1.17 | PUT | /api/pankreatitorders/{id}/form | Сформировать заявку (создателем). Проверка обязательных полей. | Authorization: Bearer <jwt>, path: { id: uint } | 200 OK | 409 Conflict |
| 4.1.18 | PUT | /api/pankreatitorders/{id}/set/{status} | Завершить/отклонить заявку (врачом). | Authorization: Bearer <jwt>, path: { id: uint, status: 'completed'|'rejected' } | 200 OK | 409 Conflict |
| 4.1.19 | DELETE | /api/pankreatitorders/{id} | Удалить черновик (soft delete по ТЗ). Доступно авторизованным. | Authorization: Bearer <jwt>, path: { id: uint } | 200 OK | 409 Conflict |
| 4.1.20 | DELETE | /api/pankreatitorders/items | Удалить услугу из заявки (без PK м-м). Доступно авторизованным. | Authorization: Bearer <jwt>, query: { pankreatit\_order\_id: uint, criterion\_id: uint } | 200 OK | 404 Not Found |
| 4.1.21 | PUT | /api/pankreatitorders/items | Изменить м-м (кол-во/порядок/значение) без PK м-м. Доступно врачу. | Authorization: Bearer <jwt>, query: { pankreatit\_order\_id: uint, critrion\_id: uint }, Body: { value\_num: number } | 200 OK | 404 Not Found |