|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практическим работам №9-10**

по дисциплине «Системная и программная инженерия»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-24-21 | Оганнисян Г.А.  Шаралапов Д.А.  Кудинов А.В. |
| Принял преподаватель | Запорожских А.И. |

Москва 2024 г.

**СОДЕРЖАНИЕ**

[Практическая работа №9 3](#_Toc164765740)

[Практическая работа №10 5](#_Toc164765741)

[Вывод 7](#_Toc164765742)

# **Практическая работа №9**

В качестве методологии управления процессом разработки выбран каскадный (Waterfall) подход, поскольку он обеспечивает чёткое разделение всех этапов проекта, позволяет заранее детализировать требования и спланировать ресурсное обеспечение. За счёт последовательного прохождения фаз – сбор и анализ требований, проектирование, реализация, тестирование и внедрение – упрощается контроль за выполнением задач и оформление сопутствующей документации, а также снижается риск «перехода» функционала между различными командами.

В качестве платформы для хранения и совместной работы с исходным кодом проекта выбран GitHub. Созданный репозиторий «Lamadjo‑Task‑Board» доступен по адресу https://github.com/Anabol1ks/Lamadjo-Task-Board.

На рисунке 1 приведён скриншот главной страницы репозитория.

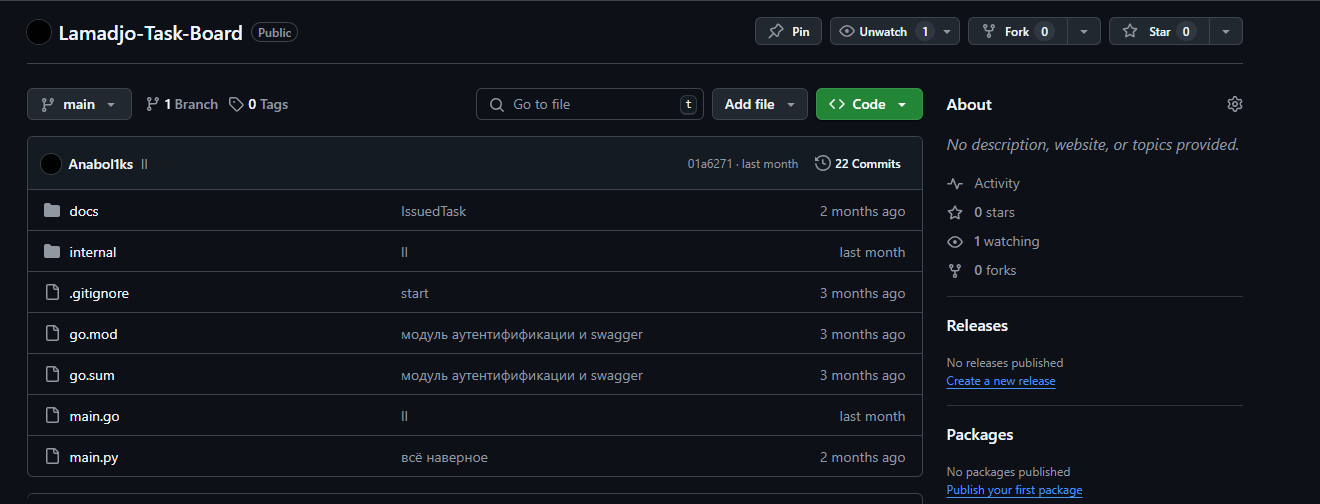


Рисунок 1 – Удаленный репозиторий

Ниже описаны используемые в рамках проекта средства разработки:

1. Go – Компилируемый язык с высокой производительностью и строгой статической типизацией. На Go реализован REST‑API‑сервер проекта благодаря его эффективности и простоте развертывания в любых средах.
2. Gin – лёгкий HTTP‑фреймворк для Go, обеспечивающий удобную маршрутизацию, поддержку middleware и минимальные оверхеды. Используется для обработки запросов, управления группами роутов и быстрой интеграции Swagger.
3. GORM – ORM‑библиотека для Go, позволяющая описывать модели базы данных в виде структур и автоматически выполнять миграции. GORM упрощает работу с SQL‑запросами, транзакциями и связями между сущностями.
4. PostgreSQL – надёжная реляционная СУБД, используемая для хранения данных пользователей, команд, задач и встреч. Обеспечивает транзакционную целостность и масштабируемость при увеличении объёма данных.
5. Swaggo (gin‑swagger) – пакет для автогенерации интерактивной документации API на основе аннотаций в коде. Позволяет любому участнику команды оперативно просматривать список эндпоинтов и тестировать их через Swagger UI.
6. Godotenv – утилита для загрузки переменных окружения из файла  .env. Обеспечивает гибкую настройку параметров подключения к базе данных и других секретных значений без хардкода в исходниках.
7. Python – язык выбран для вспомогательных скриптов, выполняющих тестовые запросы к API и демонстрирующих работу сервиса. Позволяет быстро автоматизировать повторяемые проверки.

# **Практическая работа №10**

Для технической документации был использован генератор документации Swagger. На рисунках 2–3 представлена сгенерированная документация.

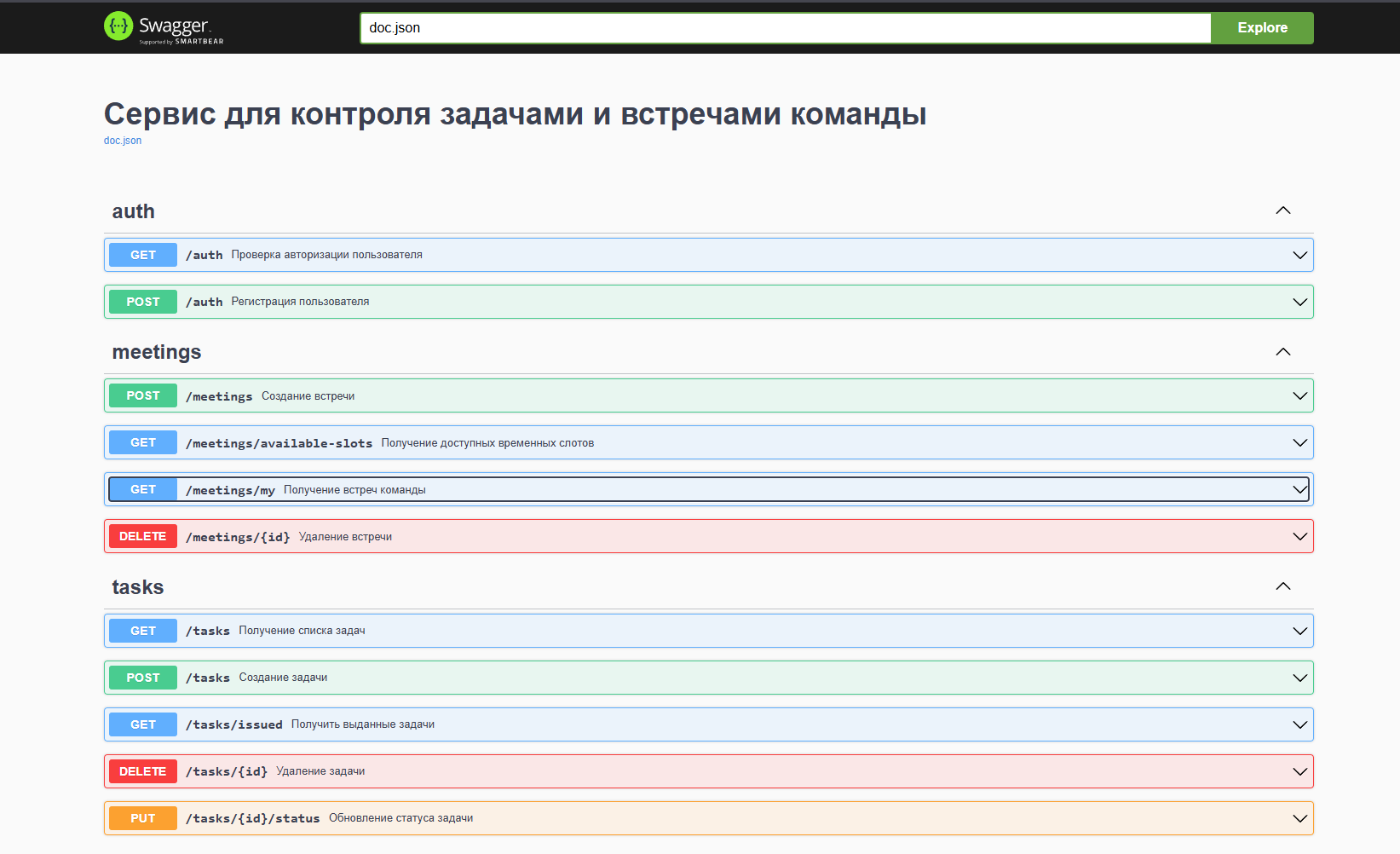


Рисунок 2 – Документация

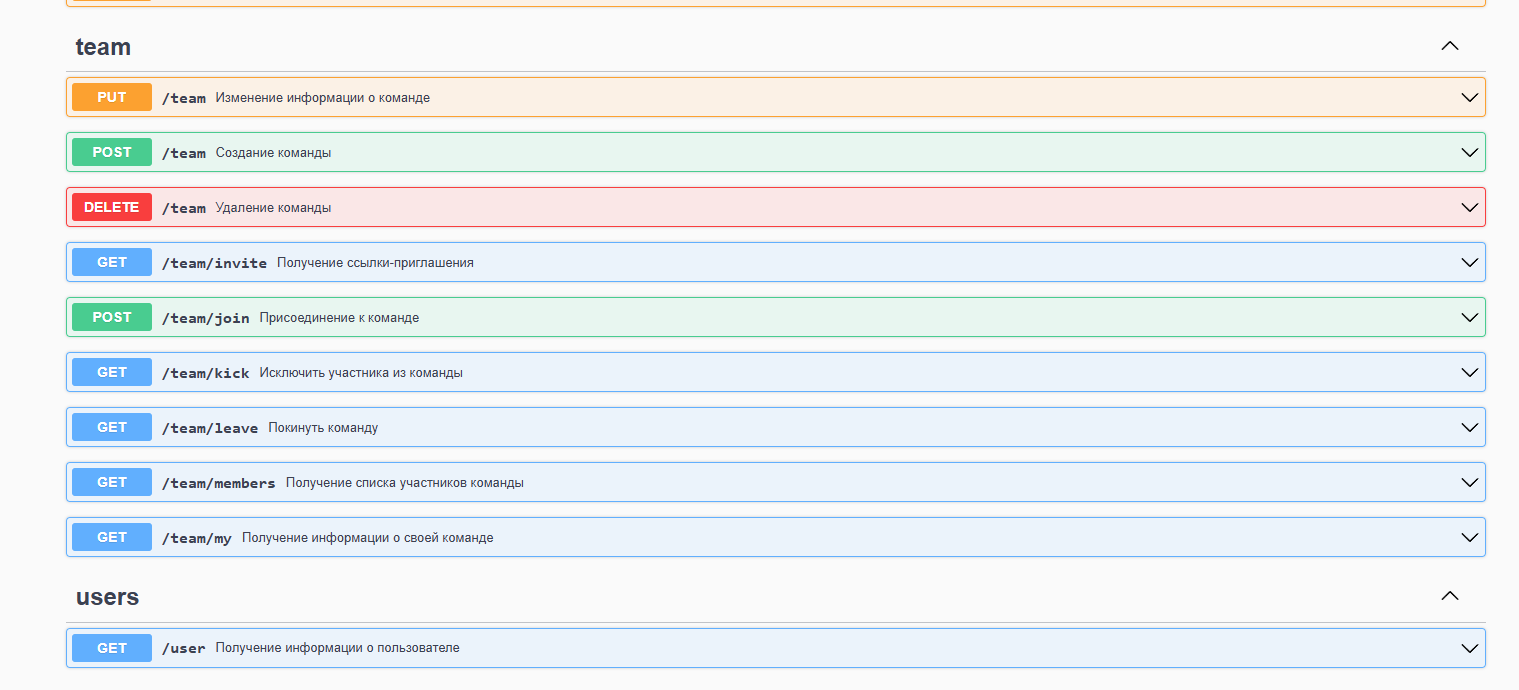


Рисунок 3 – Документация

Пользовательская документация представлена в виде README файла, который хранится в репозитории проекта.

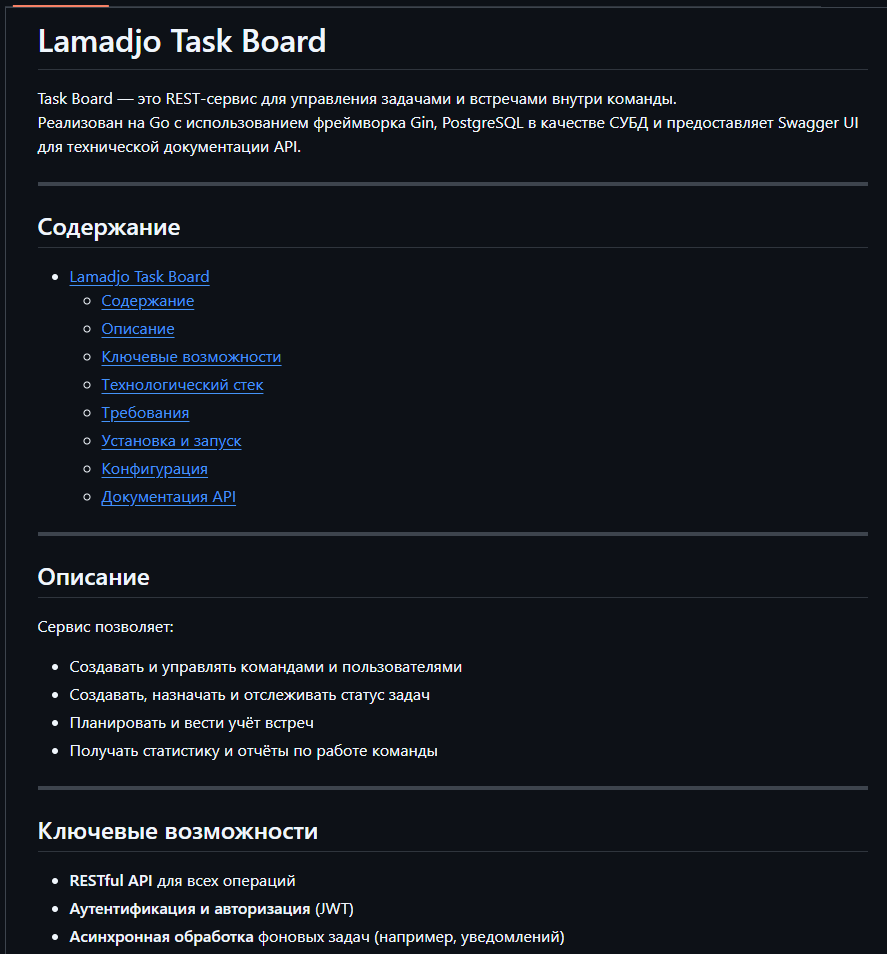


Рисунок 4 – Содержимое README.md

# **Вывод**

В результате выполнения практических работ были получены навыки по организации процесса командной разработки с применением различных методологий управления процессом разработки, а также технологий командной разработки. Была выбрана методология управления процессом разработки проекта исходя из потребностей команды, создан удаленных git репозиторий на сервисе Github, а также описаны выбранные инструменты разработки программного обеспечения. Были получены навыки по формированию документаций разработчика и пользователя.