|  |  |
| --- | --- |
|  | **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»

Расчетно-пояснительная записка

Тема: «Заявка на проведение тендера»

Студент: Комаров Д.С.

Группа ИУ5-53Б

Преподаватель: Канев А.И.

2023г.

# **ВВЕДЕНИЕ**

Организация тендеров всегда была прибыльным и ключевым сектором. Существует значительное количество компаний-участников по всему миру. Учитывая потребность заказчиков в эффективном поиске компаний для участия в тендере, было принято решение разработать систему автоматического формирования тендера. Эта система обеспечит заказчика более простым и удобным способом поиск и выбор компаний.

Целью работы является реализация системы для формирования тендера с компаниями, включающую в себя веб-сервис, веб-приложение, десктопное приложение и выделенный сервис для рассмотрения тендера.

Система предназначена для заказчиков, и модераторов сайта. В сайте предусмотрен ограниченный доступ к компаниям. Для получения доступа пользователю необходимо создать заявку на получение компаний к выбранному тендеру. Система предоставляет автоматизированный способ создания, учета и ведения заявок. Также она позволяет модератору завершать или отклонять заявки, возможность редактировать существующие и добавлять новые компании.

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 пользователя в React.
  8. Реализовать интерфейс модератора React.
  9. Создать мобильное приложение на React-Native.
  10. Создать асинхронный сервис для рассмотрения тендера на Python.
  11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.

# **БИЗНЕС-ПРОЦЕСС**

Существует множество различных компаний, которые хотят участвовать в тендере[1]. Для того, чтобы вручную не искать компании готовые участвовать в тендере, заказчики могут оставлять заявки на формирования тендера и выбирать наиболее подходящие компании. Каждая заявка вначале создается как черновик, в который можно добавлять и из которого можно удалять компании. В одной заявке можно указать сразу несколько компаний, и тогда все эти компании будут участвовать в тендере.

Каждой компании можно выставить ценник. Когда заказчик определится с компаниями, которые будут участвовать в заявке, он формирует заявку и она отправляется на модерирование. После этого её уже нельзя редактировать. Можно также посмотреть историю своих заявок. Для того, чтобы компании, оплата компаний совпадали c ожиданиями модератор отклоняет и принимает заявки таким образом, чтобы все происходили честно. На случай если заказчик передумает формировать тендер присутствует возможность отменить заявку.

Тендеры разнообразны по этапам, участникам и типам закупок. Этапы бывают одно- или двухэтапные, где второй этап позволяет уточнить условия и выбрать победителя. Участники бывают в закрытых, специализированных торгах. Типы закупок включают аукционы, конкурсы и запрос котировок. Тендеры могут быть открытыми или закрытыми, проводиться в форме торгов или аукционов на один или несколько этапов[2]. За рассмотрение заявки на тендер отвечает выделенный сервис, который проверяет информацию о компании и её возможности участвовать в данном тендере[3]. Функции пользователей с разными ролями описаны на диаграммах прецедентов (рис. 1).

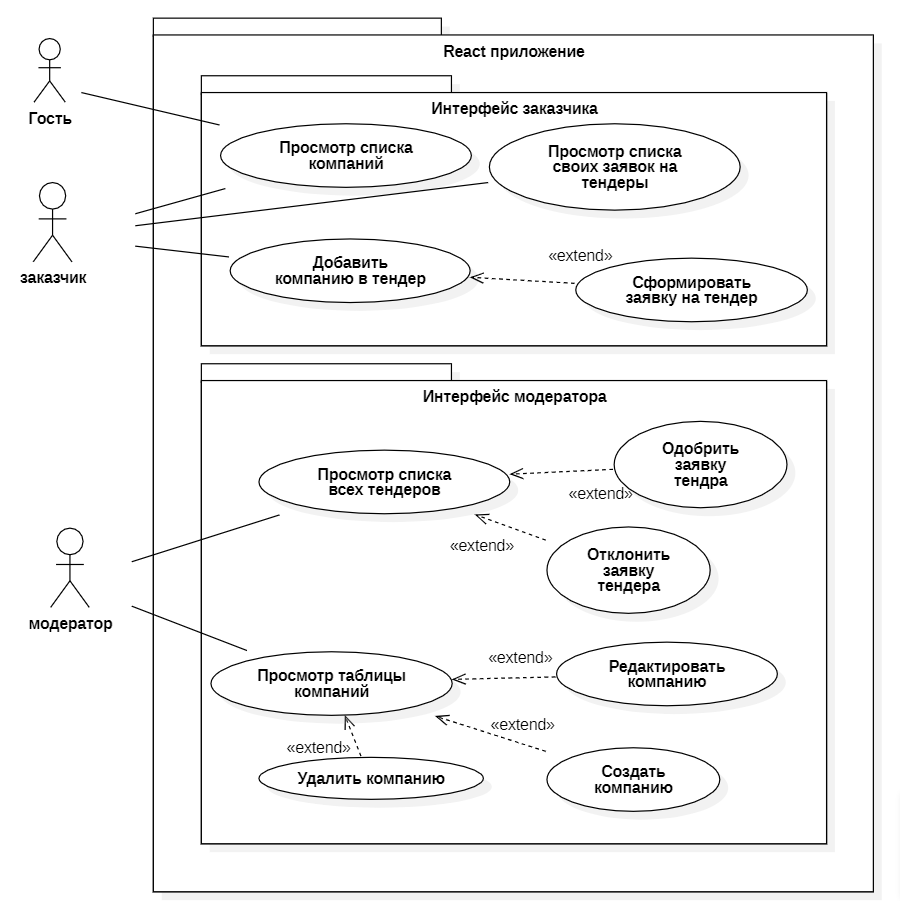


Рисунок 1 - Диаграмма прецедентов

Гостям доступен просмотр компаний. Зарегистрированные гости – заказчики. Они могут добавлять компании в заявку, просматривать список своих заявок и сформировывать текущую заявку. Заявки обрабатываются советом директоров. В результате обработки заявки её либо одобряют, либо отклоняют. Помимо возможностей модератору также доступны уникальные функции для работы с компаниями, а именно: просмотр всех компаний, редактирование, создание и удаление компаний, а также просмотр списка всех тендеров в табличном виде. Процесс оформления заявки отражен на диаграмме деятельности (рис. 2).

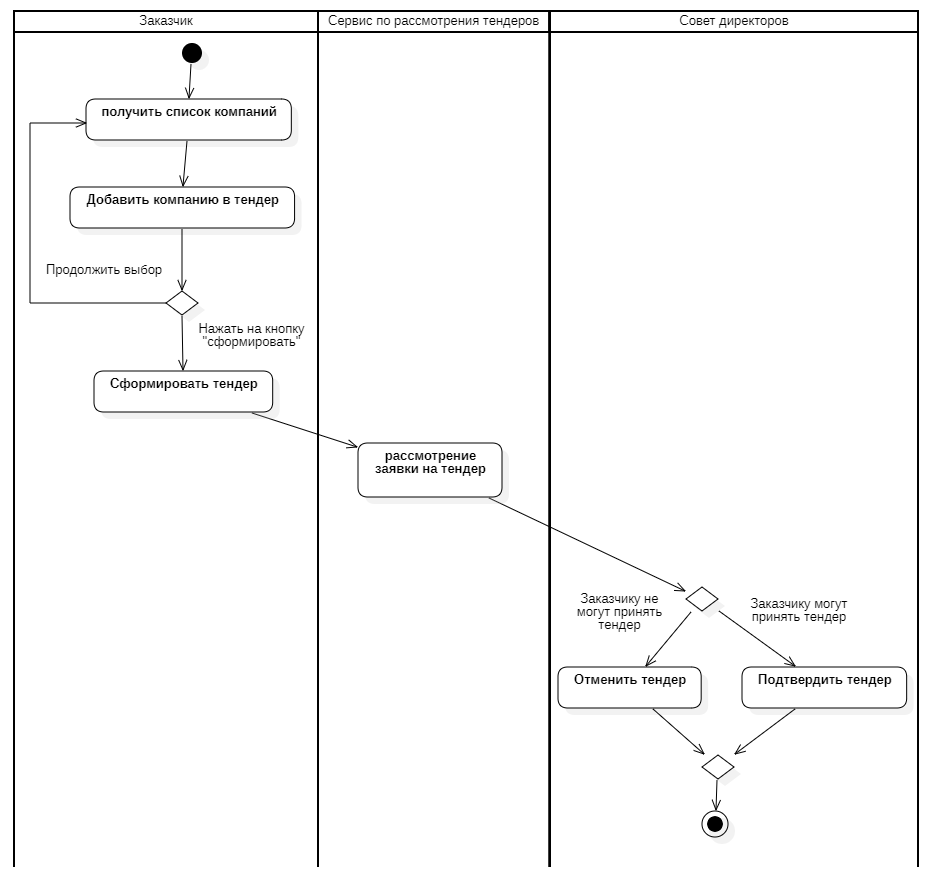


Рисунок 2 - Диаграмма деятельности

Заказчик выбирает компании, затем формирует на основе выбранных компаний заявку. Эту заявку обрабатывает совет директоров. Возможные состояния заявки отражены на диаграмме состояний (рис. 3).

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 3 - Диаграмма состояний уведомления

При выборе первой компании формируется черновик. Последующие выбранные компании добавляются в этот черновик. Пользователь затем формирует заявку, удаляет её или выходит из приложения. Сформированную заявку обрабатывает совет директоров. Они могут одобрить или отклонить её.

# **АРХИТЕКТУРА**

Архитектура системы отображена на диаграмме развертывания (рис. 4). Бэкенд и база данных разворачивается на самом устройстве, а s3 хранилище изображений, Redis в докер контейнерах [6]. Такое решение было принято в связи с тем, что технология докер контейнеров позволяет быстро и удобно разворачивать целые системы на любых устройствах, поддерживающих докер. Еще одним неоспоримым преимуществом докер контейнеров является то, что их контейнеры представляют собой виртуальные машины, а значит их работа не зависит от платформы устройствах. Сервис рассмотрение тендера [3] разворачиваются отдельно и необязательно на том же компьютере, что и основной сервис.

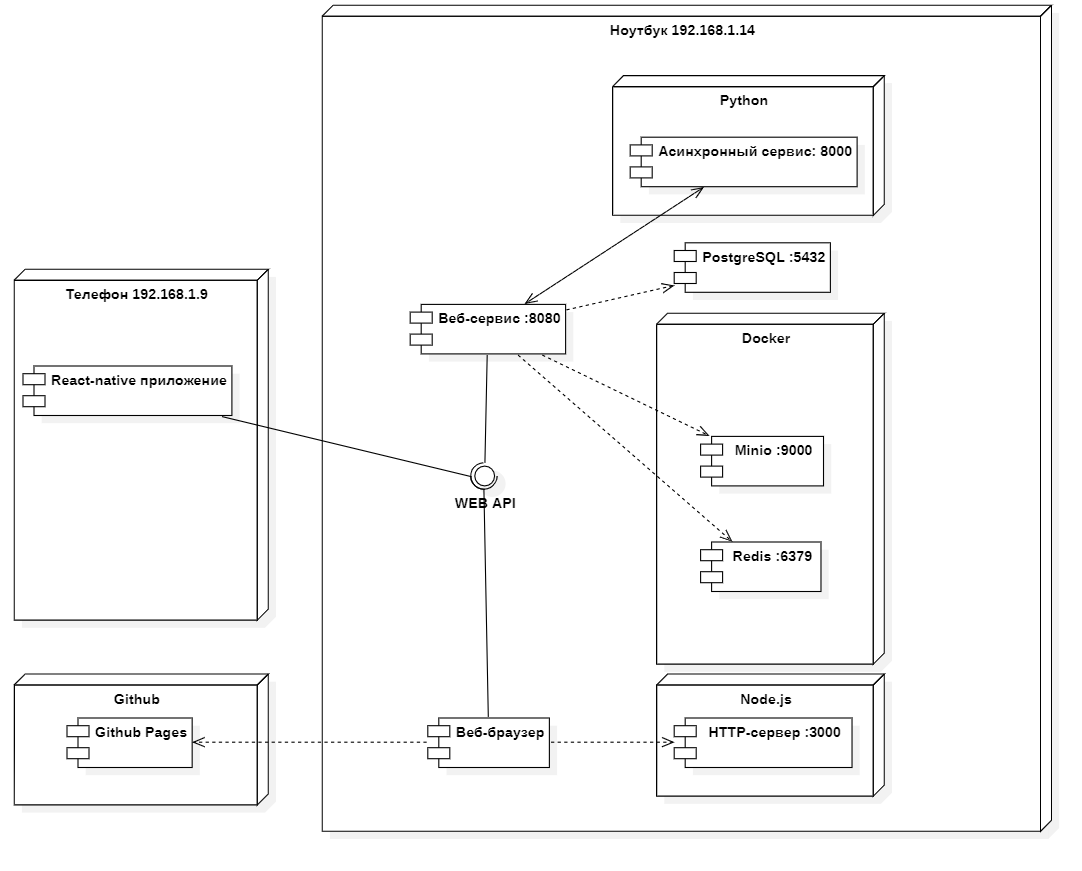


Рис. 4 - Диаграмма развертывания

Мобильное [10] и браузерное приложения [7, 8, 9] обращаются к веб-сервису на базе языке Golang 1.20 [3, 4, 5] через REST Web-API. Использование Golang обосновано его проверенной эффективностью, признанием в индустрии и современностью языка.

Данные хранятся в СУБД PostgreSQL [11], их структура отражена на ER диаграмме (рис. 5). СУБД PostgreSQL является одним из стандартов индустрии, поэтому было решено использовать её. Структура данных довольна проста. Помимо базовых полей, тендера также имеет поле status\_check, отражающее рассмотрения тендера, полученный от асинхронного сервиса. Для хранения в одной заявке нескольких карт используется промежуточная таблица Tender\_Company, реализующая связь М-М. Устройство бекенда приложения изображено на диаграмме классов бекенда.

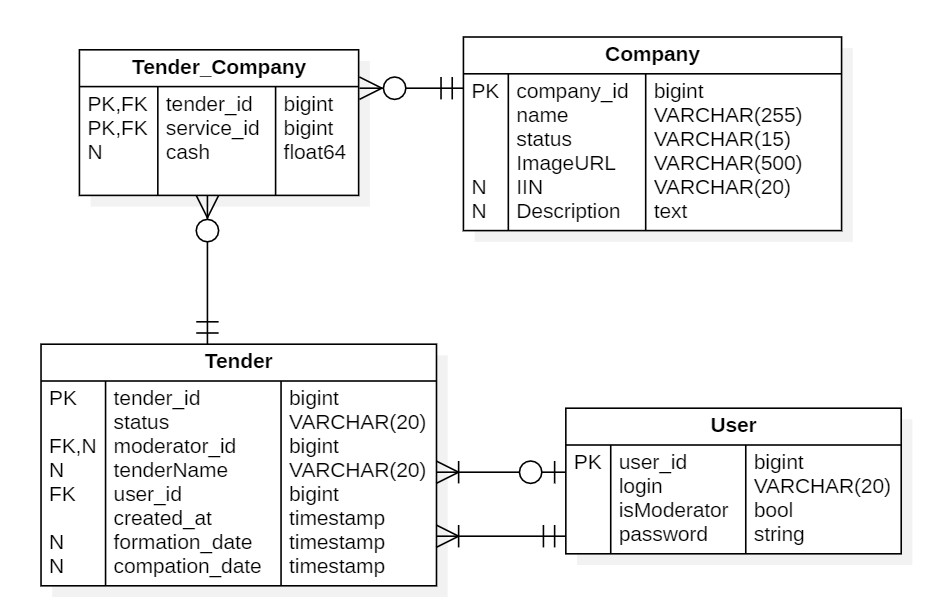


Рисунок 5 - ER диаграмма

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, компании имеют связь с сервером статических файлов, т.к. в данных получателя хранится ссылка на их изображение, хранимое на сервере статических файлов.

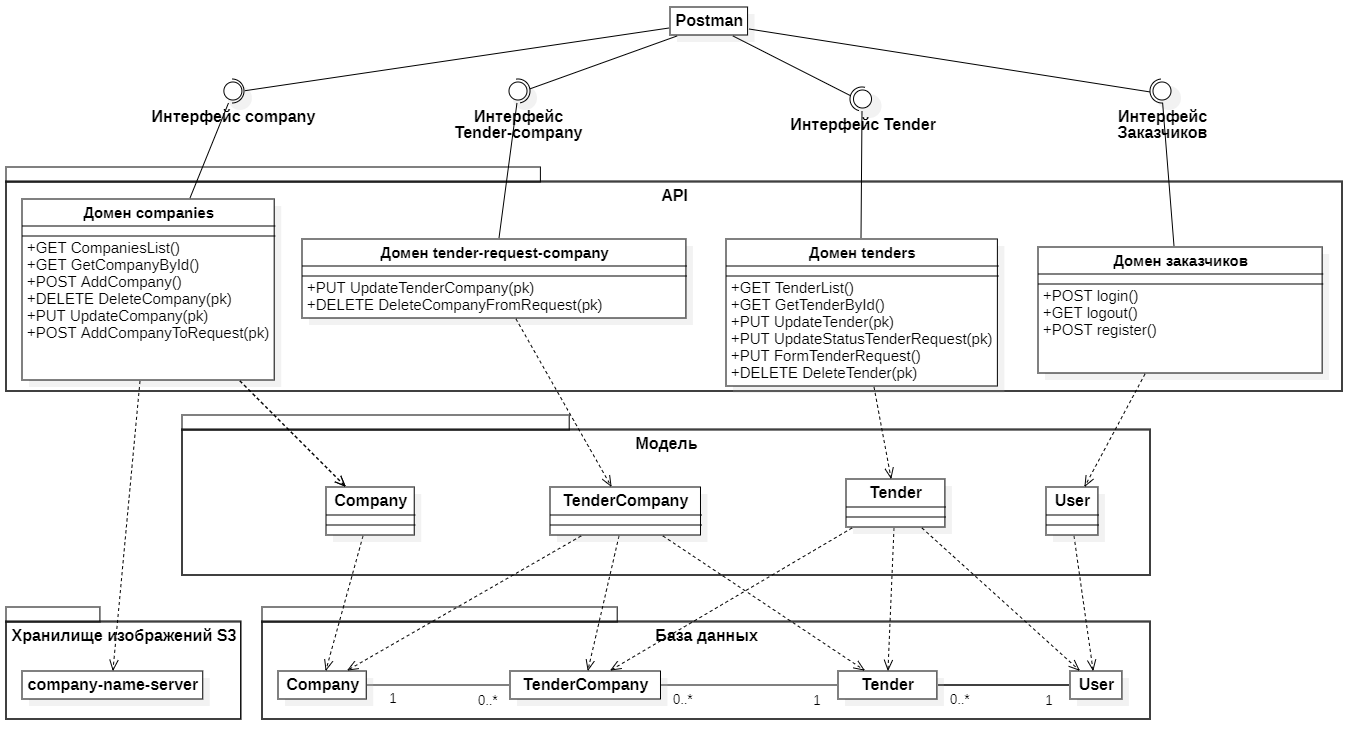


Рисунок 6 *-* Диаграмма классов бекенда

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).

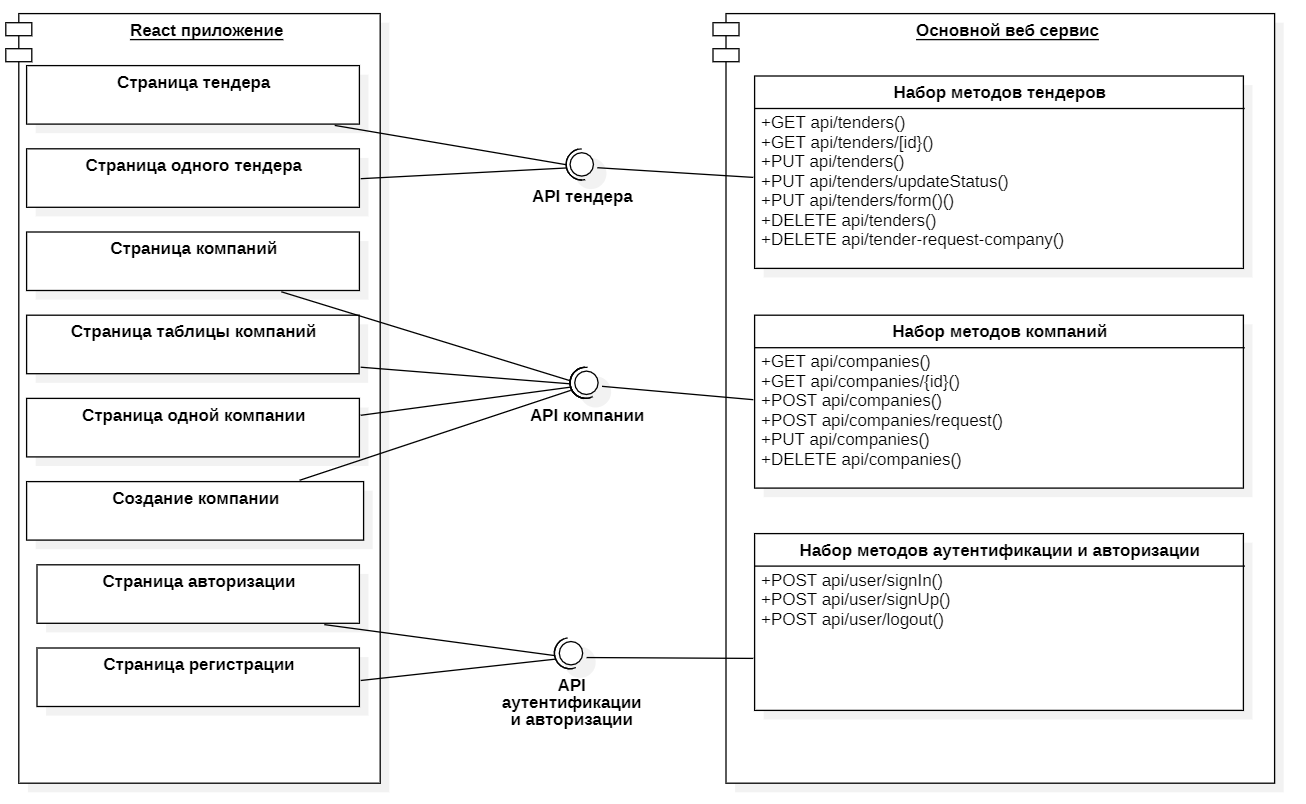


Рисунок 7 *-* Диаграмма классов фронтенда

# **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: тендера, компании, и аутентификация. Методы следуют правилам REST API.

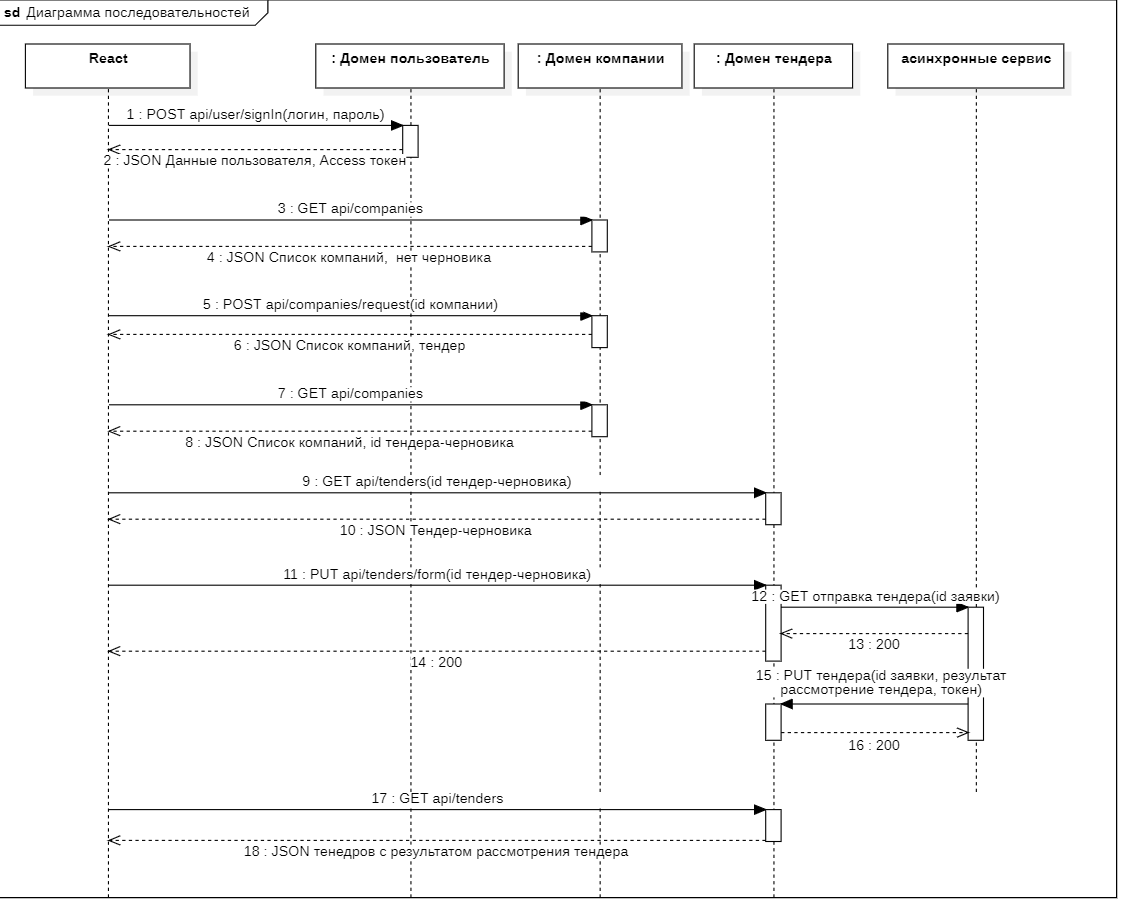


Рисунок 8 - Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. Для этого он отправляет через графический интерфейс запрос, передавая в нем логин и пароль. Если аккаунт с такими данными существует, то пользователь получает JWT в ответном запросе. Если же такого аккаунта не существует, или пароль введен неверно, пользователь получит ошибку. В таком случае ему надо либо пройти регистрацию, либо ввести пароль верно. Затем графический интерфейс пользователя запрашивает у веб-сервиса список тендеров, которые возвращаются в JSON формате. Заказчик выбирает компанию, которую хочет добавить в тендер, и, нажимая на кнопку «плюс» в графическом интерфейсе, отправляет запрос на добавление компании в свою черновую заявку. Этот процесс может продолжаться несколько раз.

Когда закзачик определится с выбором, он нажимает на кнопку «сформировать» в графическом интерфейсе. После этого приложение запрашивает id черновой заявки тендера и затем отправляет запрос на формирование этой заявки. В этот момент основной веб-сервис выполняет асинхронный запрос к другому сервису, чтобы он рассмотрел заявку на проведение тендера и вернул статус с решением в основной сервис. Через некоторое количество времени уведомитель может увидеть, что статус его решение по тендеру изменился.

Процесс рассмотрения тендера происходит также через графический интерфейс. Модераторы могут просматривать списки всех тендеров и, нажимая на соответствующие кнопки, отправлять запросы на подтверждение или отклонение тендеров в основной веб-сервис. В эти запросы также можно включить фильтры по имени создателя тендера, одному из статусов тендера и диапазону дат, в которых эти тендеры были сформированы. Также модераторы могут управлять тендерами через графический интерфейс. Им доступны такие функции, как создание и редактирование тендеров, просмотр списка тендеров в виде таблицы и удаление их. Для каждой из этих функций присутствует свой метод, отправляемый на основной веб-сервис.

# **ОПИСАНИЕ ИНТЕРФЕЙСА**

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

Изображение выглядит как снимок экрана, текст, логотип, программное обеспечение

Автоматически созданное описание

Рисунок 9 - Главное меню (неавторизированный пользователь)

Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

Рисунок 10 - Главное меню (модератор)

Первоначальная страница для всех пользователей и гостей. В зависимости от типа пользователя её содержимое меняется. Гости, например, могут посмотреть список компаний и подробную информацию о них, а также нажать на кнопку «Войти», либо «Регистрация». А у пользователя уже есть кнопки, «Корзина» и «Выход».

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 cookie и используется при отправлении запросов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 11 - Страница авторизации

На странице с формой регистрации (рис. 12) гости могут завести аккаунт. Для этого нужно указать логин, и пароль. Если введенный логин уже занят система попросит пользователя сменить его.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 12 - Страница регистрации

На странице со списком компаний (рис. 13) отображается список компаний в виде карточек. У каждой компании есть хендлер нажатия для перехода в режим «Подробнее», переносящая пользователя на страницу с подробной информацией о компании.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 13 - Страница со списком компаний

На странице с подробным описанием компании (рис. 14) отображается подробная информация о компании: Название, ИИН, статус, описание компании.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 14 - Страница с подробным описанием компании

На странице со списком уведомлений (рис. 15, 16) отображается список тендеров. В зависимости от типа пользователя этот список будет функционально отличаться. Так, для заказчика отображается список созданных ими заявок: название тендера, статус рассмотрения, дата создания, дата начала процесса, дата принятия, автор, статус. Также можно отфильтровать тендера по дате формирования тендера, выбрав либо одну дату, либо промежуток.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 15 - Страница со списком тендеров (пользователь)

Для модератора функция этой страницы шире. Для них отображается список всех заявок всех пользователей с более подробной информацией: название тендера, статус рассмотрения, дата создания, дата начала процесса, дата принятия, автор, модератор, статус, и две кнопки «Завершить» и «Отказать» для подтверждения или отмены тендера. Также присутствует фильтр по создателю тендера.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 16 - Страница со списком тендеров (модератор)

На странице с подробным описанием транзакции (рис. 17) отображается подробная информация об транзакции. Список выбранных компаний в виде таблицы, а также все поля транзакции.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 17 - Страница с подробным описанием тендера

На странице с таблицей компаний (рис. 18) модератор может в компактном и удобном формате просмотреть список всех компаний, существующих в системе. Отображаются следующие поля: логотип компании, название компании, статус, описание, кнопки с удалением и возможностью редактировать. Также можно перейти на страницу создания новой компании(рис. 19).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 18 - Страница с таблицей компаний

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 19 - Страница создания компании

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 20 - Страница редактирование компании

На странице редактирования/создания компании (рис. 19-20) модератор может отредактировать существующую, или создать новую компанию. Для редактирования доступны все поля: название, ИИН, статус, описание.

# **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 на Golang, с использованием веб-фреймворка Gin.
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://qaleka.github.io/R\_I\_P\_labs\_front/recipients.
6. В веб-сервис добавлена авторизация через JWT, а методы задокументированы через Swagger.
7. Реализован интерфейс пользователя. Доступ к нему имеют только авторизированные пользователи.
8. Выделенный сервис был разработан и развернут в отдельном виртуальном окружении Python.
9. Реализован интерфейс модератора для подтверждения новых тендеров и редактирования/добавления получателей.
10. Было реализовано мобильное приложение на React Native, 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/DmitriyKomarovCoder/RIP.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Tender in Finance Definition: How It Works, With Example [Электронный ресурс]. URL: https://www.investopedia.com/terms/t/tender.asp (дата обращения 04.01.2024).
  2. Какие бывают тендеры и как повысить шансы на победу [Электронный ресурс]. URL: https://journal.tinkoff.ru/love-me-tender/ (дата обращения 06.01.2024).
  3. Руководство по Go [Электронный ресурс] // GoLang. URL: https://metanit.com/go/tutorial/ (дата обращения: 11.09.2023)
  4. Руководство по Gin Web Framework [Электронный ресурс] // GoLang. URL: https://github.com/gin-gonic/gin (дата обращения: 12.09.2023).
  5. Руководство по Gorm [Электронный ресурс] // Gorm. URL: https://gorm.io/docs/index.html (дата обращения: 14.10.2023)
  6. Полное практическое руководство по Docker [Электронный ресурс] // Habr. URL: https://habr.com/ru/articles/310460/ (дата обращения: 10.10.2023).
  7. Руководство по React [Электронный ресурс] // Metanit. URL: https://metanit.com/web/react/ (дата обращения: 11.11.2023).
  8. Quick Start – React [Электронный ресурс] // React. URL: https://react.dev/learn (дата обращения: 11.11.2023).
  9. Руководство по React Bootstrap [Электронный ресурс] // React Bootstrap. URL: <https://react-bootstrap.netlify.app/docs/getting-started/introduction>
  10. Руководство по React Native [Электронный ресурс] // React Nativ. URL: <https://reactnative.dev/docs/getting-started>
  11. Руководство по PostgreSQL [Электронный ресурс] // Metanit. URL: https://metanit.com/sql/postgresql/ (дата обращения: 05.10.2023).

**Приложение. Техническое задание**

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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: «Заявка на проведение тендера»

Студент: Комаров Д.С.

Группа ИУ5-53Б

Преподаватель: Канев А.И.

Москва, 2023

# **Цель**

Реализовать систему для заявок на проведение тендера, включающую в себя веб-сервис, веб-приложение, десктопное приложение и выделенный сервис для рассмотрения результата для тендера.

# **Назначение**

Система предназначена для пользователей, модераторов. В сервисе предусмотрен ограниченный доступ к тендерам. Для получения доступа пользователю необходимо создать заявку на получение доступа к выбранному тендеру. Система предоставляет автоматизированный способ создания, учета и ведения заявок. Также она позволяет модераторам принимать или отклонять заявки. Модераторы имеют возможность редактировать существующие и создавать новые компании.

# **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 пользователя в React.
  8. Реализовать интерфейс модератора в React.
  9. Создать мобильное приложение на React Native.
  10. Создать асинхронный сервис для отслеживания отправки уведомления на Python.
  11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.

# **Методы веб-сервиса**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Метод | URL | Описание | Входные данные | Выходные данные |
| Методы авторизации и аутентификации | | | | | |
| 4.1.1 | POST | api/users/sign\_up | Регистрация | login: string  password: string  user\_name: string |  |
| 4.1.2 | POST | api/users/login | Вход в аккаунт | login: string  password: string | {  access\_token: string,  expires\_type: string  } |
| 4.1.3 | GET | api/logout/ | Выход из аккаунта  Доступно только авторизированным пользователям | auth cookies | status\_code: int |
| Методы компаний | | | | | |
| 4.2.1 | GET | api/comapanies/upload-image | Обновляет фотографию города. Доступно только модератору | auth header  file: File  company\_id: Int | {  image\_url: string,  status: string  } |
| 4.2.2 | GET | api/companies | Возвращает список городов, удовлетворяющих переданным критериям | auth header  {  company: string,  search: string  } | {  comapanies: {  "company\_id": int,  name: string,  status\_:string,  description: string  image\_url: string  }[]  } |
| 4.2.3 | PUT | api/companies/request | Добавляет в по-ход пользовате-ля компании {company\_id}, возращает id тендера, в которую был добавлен компания  Доступно только авторизированным пользова-телям | auth header  {  compnay\_id: int,  draft\_id: int  } | {  id: int,  status: string  } |
| 4.2.4 | GET | api/companies/{id} | Возвращает компании | id: int | {  "id": int,  name: string,  status: string,  description: string  image\_url: string  } |
| 4.2.5 | POST | api/companies | Создает компании  Доступно только модераторам | auth header  {  name: string,  description: string,  inn  image\_url: file?  } | {  company\_id: string,  status: string  } |
| 4.2.6 | PUT | api/сompanies | Обновляет информацию о компании  Доступно только модераторам | auth header  {  name: string?  description: string?  id: int?  status: string?  inn: string?  } | {  name: string,  description: string,  id: int,  status: string,  inn: string  } |
| 4.2.7 | DELETE | api/companies | Помечает город как удаленный  Доступно только модераторам | auth header  id: int | {  deleted\_id: int  } |
| Методы тендера | | | | | |
| 4.3.1 | GET | api/tenders | Возвращает список заявок пользователя  Доступно только авторизированным пользователям | auth header | tenders: {  creation\_date: string,  completion\_date: string,  formation\_date: string,  description: string,  destination\_tenders: {  {  company: {  name: string,  description: string,  id: 0,  image\_url: string,  status: string  },  company\_id: int,  tender\_id: int,  id: int,  cash: int  }[]  }[] |
| 4.3.2 | GET | api/tenders/{id} | Возвращает по-дробную ин-формацию о за-явке  Доступно только авторизированным пользова-телям | auth header  id: int | {  tenders: {  creation\_date: string,  completion\_date: string,  formation\_date: string,  company\_tenders: Company [],  }[],  status: string  } |
| 4.3.3 | GET | api/tenders/current | Возвращает те-кущую черновую заявку пользова-теля  Доступно только авторизированным пользова-телям | auth header | tenders: {  creation\_date: string,  completion\_date: string,  formation\_date: string,  company\_tenders: Company []  } |
| 4.3.4 | DELETE | api/tender-request-company | Удаляет из тендера пользователя компании {company\_id}  Доступно только авторизированным пользова-телям | auth header  {  id: int  } | {  deleted\_company\_tender: int  status": string"  } |
| 4.3.5 | PUT | api/tenders/form | Устанавливает заявке статус «сформирована»  Только если её текущий статус «черновик»  Доступно только авторизированным пользователям | auth header  status: string? | Status\_code:int |
| 4.3.6 | DELETE | api/tenders | Устанавливает заявке статус «удалена»  Только если её текущий статус «черновик»  Доступно только авторизированным пользова-телям | auth header  start\_date: datetime?  end\_date: datetime?  Status: string | {  Tenders:Tenders[],  Status:string  } |
| 4.3.7 | GET | api/tenders?status&start\_date&end\_date | Возвращает отфильтрованный по датам созда-ния и статусу тендера.  Доступно только модераторам | auth cookies  start\_date: datetime?  end\_date: datetime?  Status: string | {  Tenders:Tenders[],  Status:string  } |
| 4.3.8 | PUT | api/tenders/updateStatus | Устанавливает заявке статус «отклонена» или «заверше-на»  Только если её текущий статус «сформирована»  Доступно только модераторам | {  tender\_id: int,  Status: string  } | Status code |
| .Методы внешнего сервиса board\_of\_Directors\_check | | | | | |
| 4.4.1 | POST | api/tenders/user-form-start | Совет директоров принимает решение о рассмотрении тендера. | tender\_id: int |  |

1. **Функциональные требования:**
   1. Главное меню. Здесь находятся кнопки для регистрации и авторизации или информация о текущем пользователе, и меню со списком доступных пунктов.
      1. Доступна всем пользователям.
      2. Действия
         1. Войти – переход на страницу 5.2. Только для гостей.
         2. Зарегистрироваться – переход на страницу 5.3. Только для гостей.
         3. Открыть список компаний – переход на страницу 5.4. Для всех пользователей.
         4. Открыть список выбранных компаний – переход на страницу 5.6. Только для авторизированных пользователей.
         5. Список заявок – переход на страницу 5.6. Только для пользователей.
   2. Страница с формой авторизации. На этой странице находится форма авторизации.
      1. Доступна гостям
      2. Действия
         1. Войти – производит запрос (метод 4.1.2).
         2. Зарегистрироваться – перенаправляет на страницу 5.3.
   3. Страница с формой регистрации. На этой странице находится форма авторизации.
      1. Доступна гостям
      2. Действия
         1. Зарегистрироваться – производит запрос (метод 4.1.1.), в котором передаются введенные данные формы.
         2. Войти - перенаправляет пользователя на страницу 5.2.
   4. Страница со списком компании. На этой странице располагается список всех получателей и панель фильтрации.
      1. Доступна всем пользователям.
      2. Выводится информация о компаниях в виде карточек (метод 4.2.2.).
         1. Логотип компании.
         2. Название компании.
      3. Действия
         1. Отфильтровать компанию по названию (метод 4.2.2.).
         2. Добавить компанию в поход (метод 4.2.3.). Только для авторизированных пользователей.
         3. Нажатие на карточку – перенаправляет на страницу 5.5. Только для авторизированных пользователей.
   5. Страница с подробным описанием компании
      1. Доступна всем пользователям.
      2. Выводится информация о компании.
         1. Логотип компании.
         2. Название компании.
         3. Статус компании.
         4. Описание компании.
         5. ИИН владельца.
      3. Действия
         1. Удалить компанию (метод 4.2.7.). Только для модератора.
         2. Назад – перенаправляет на предыдущую страницу.
   6. Страница со списком тендеров
      1. Доступна только авторизированным пользователям.
      2. Выводится информация об тендерах в табличном формате (метод 4.3.1).
         1. Название тендера.
         2. Дата создания.
         3. Дата формирования.
         4. Дата окончания.
         5. Создатель
         6. Кем обработана. Только для модераторов.
      3. Действия
         1. Открыть подробную информацию об тендере – перенаправляет на страницу 5.7.
         2. Установить уведомлению статус «отклонено» или «завершено» (метод 4.3.8.). Только для модераторов.
         3. Отфильтровать список тендеров по диапазону дат формирования и/или статусу (метод 4.2.1) и/или создателю (только для модератора).
   7. Страница с подробным описанием тендера. Заказчик может изменить набор получателей компаний. Модератор может просматривать тендеры всех пользователей.
      1. Доступна только авторизированным пользователям.
      2. Выводится информация о заявке. Список компаний, выбранных в заявке, в виде таблицы (метод 4.3.2.).
         1. Логотип компаний.
         2. Имя создателя тендера.
         3. Название тендера.
         4. Статус тендера
         5. Дата формирования
      3. Действия
         1. Удалить компанию из тендера (метод 4.3.4.). Если тендер находится в статусе черновика. Только для создателя тендера.
         2. Сформировать тендер (метод 4.3.5.). Если тендер находится в статусе черновика. Только для создателя тендера.
         3. Удалить тендер (метод 4.3.6.). Если тендер находится в статусе черновика. Только для создателя уведомления.
   8. Страница создания компании.
      1. Доступна только модераторам.
         1. Название компании.
         2. Описание компании.
         3. Логотип компании.
         4. ИИН.
      2. Действия
         1. Создать компании (метод 4.2.5)
   9. Страница с таблицей городов. Предоставляет модератору удобный способ отображения всех городов в табличной форме с возможностью редактирования Доступна только модераторам.
      1. Доступна только модераторам.
         1. Название компании.
         2. Статус компании.
         3. Описание компании.
         4. Логотип компании.
      2. Действия
         1. Изменение полей компании. Можно изменять все вышеперечисленные поля компании.
         2. Сохранить изменения (методы 4.2.5. и 4.2.6.).
         3. Удалить компанию (метод 4.2.7)
         4. Добавить компанию. Открывает страницу 5.8.
2. **Требования к программному обеспечению:**
   1. Серверная часть
      1. ОС: Linux 6.5 / Windows 10
      2. Docker 24.0.6
      3. Python 3.11.5
      4. Golang 1.21.1
      5. redis 6.2-alpine
      6. nginx 1.19.2-alpine
      7. minio RELEASE.2022-10-15T19-57-03Z
      8. postgres 16.1
   2. Клиентская часть
      1. ОС: Windows/MacOS/Linux
      2. Веб-браузер: Safari 11.1+/Chrome 40+/Opera 27+/Firefox 44+/Yandex 20+
3. **Требования к аппаратному обеспечению:**
   1. Серверная часть
      1. Процессор минимум 2-ядерный с частотой от 2 ГГц.
      2. Оперативная память от 4 Гб.
      3. Место на жестком диске от 2 Гб.
   2. Клиентская часть
      1. Процессор с частотой от 1ГГц.
      2. Оперативная память от 512 Мб.