|  |  |
| --- | --- |
| **О­­** | **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:***

***\_\_\_\_\_Консалтинг по ИТ безопасности\_\_\_\_\_\_\_\_\_\_\_\_\_***

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

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

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

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

Студент \_\_ИУ5-55Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_Гонов М. И.\_\_\_**

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

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

(Подпись, дата) (И.О.Фамилия)

*2025 г.*

**АННОТАЦИЯ**

Расчётно-пояснительная содержит 31 страницы. С приложениями объем составляет 46 страницы. Работа включает в себя 8 диаграмм и 33 изображений системы. В процессе выполнения было использовано 8 источников.

Объектом разработки является веб-сервис «Consulting» для оказания консультационных услуг в сфере информационной безопасности. Данный программный сервис позволяет автоматизировать процесс взаимодействия между консультантами и клиентами, управлять заявками, анализировать статистику.

Цель работы заключается в создании набора программных компонентов и сопутствующих веб-сервисов для управления заявками, планирования консультаций.

В ходе работы была разработана архитектура веб-сервиса «Consulting», рассчитанного на автоматизацию оказания консультационных услуг в сфере информационной безопасности. Создан интерфейс для взаимодействия с данным сервисом, а также реализованы и развернуты веб-сервер, нативное приложение и прогрессивное веб-приложение, способные работать с данным веб-сервисом.

Пояснительная записка содержит 2 приложения.

**СОДЕРЖАНИЕ**

[АННОТАЦИЯ 1](#_Toc185419960)

[СОДЕРЖАНИЕ 2](#_Toc185419961)

[ВВЕДЕНИЕ 3](#_Toc185419962)

[1 ПРЕДМЕТНАЯ ОБЛАСТЬ 5](#_Toc185419963)

[2 АРХИТЕКТУРА 10](#_Toc185419964)

[3 АЛГОРИТМЫ 14](#_Toc185419965)

[4 ОПИСАНИЕ ИНТЕРФЕЙСА 16](#_Toc185419966)

[ЗАКЛЮЧЕНИЕ 20](#_Toc185419967)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#_Toc185419968)

[ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ 22](#_Toc185419969)

[1 Введение 23](#_Toc185419970)

[2 Назначение разработки 23](#_Toc185419971)

[3 Стадии и этапы разработки 23](#_Toc185419972)

[4 Требования к функциональным характеристикам 24](#_Toc185419973)

[5 Требования к составу и параметрам технических средств 29](#_Toc185419974)

[6 Требования к составу и параметрам технических средств 29](#_Toc185419975)

[ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ 0](#_Toc185419976)

**ВВЕДЕНИЕ**

В современной цифровой эпохе информационная безопасность приобретает критическое значение для организаций различных отраслей. С ростом числа кибератак и усложнением методов взлома, необходимость в квалифицированных консультационных услугах по ИТ-безопасности становится все более актуальной. В условиях высокой конкуренции на рынке ИТ-услуг, внедрение автоматизированной системы консалтинга по ИТ-безопасности становится важным шагом для оптимизации процессов взаимодействия между консультантами и клиентами, управления заявками и обеспечения высокого уровня защиты корпоративных данных.

Целью данной работы является разработка системы «Consulting», предоставляющей интерфейс для оказания консультационных услуг в сфере информационной безопасности. Система включает в себя веб-сервис, веб-приложение и нативное приложение, обеспечивающие полный цикл оказания консультационных услуг, управление заявками.

Эта система предназначена для клиентов, желающих получить экспертные консультации по защите корпоративных сетей и данных, а также для консультантов, предоставляющих эти услуги. Клиенты могут формировать заявки на консультации, выбирать необходимые услуги из каталога и получать рекомендации по улучшению ИТ-безопасности. Консультанты, в свою очередь, могут оперативно обрабатывать заявки.

Система автоматизирует процесс оказания консультационных услуг, обеспечивая удобство для клиентов и консультантов, а также повышая эффективность управления заявками и контролем исполнения задач. Реализована возможность получения подробной информации о заявках, что позволяет оперативно принимать решения и вести учет.

Нефункциональные требования к разрабатываемой системе:

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

В ходе работы необходимо выполнить следующие задачи:

* 1. Создать MVP и базового дизайн на основе in4security.com
  2. Создать базу данных для хранения информации о услугах и заявках
  3. Создать веб-сервис на языке Python с использование Django Rest Framework.
  4. Реализовать авторизацию и хранение сессий в Redis .
  5. Разработать базовый SPA на React для гостя
  6. Внедрить адаптивность, менеджер состояний Redux Toolkit, PWА, разработать Tauri приложение
  7. Завершить разработку интерфейса клиента в React, использовать для обращений к методам веб-сервиса Axios
  8. Реализовать интерфейс модератора в React
  9. Разработать десктопное приложение Tauri
  10. Развернуть приложение при помощи GitHub Pages
  11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм
  12. Реализовать мобильное приложение на React Native.
  13. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта.

**1 ПРЕДМЕТНАЯ ОБЛАСТЬ**

В условиях стремительного развития информационных технологий и увеличения числа кибератак, обеспечение информационной безопасности становится приоритетной задачей для организаций различных отраслей. Консультирование в области ИТ-безопасности играет ключевую роль в защите корпоративных данных, систем и инфраструктуры от потенциальных угроз. Основной задачей системы является создание удобного и эффективного механизма для взаимодействия между клиентами и консультантами, управления заявками на консультации, а также поддержание регулятивных норм и стандартов безопасности, что обеспечивается за счет участия модераторов [1].

Клиенты представляют собой организации или частных лиц, заинтересованных в получении экспертных рекомендаций по защите своих информационных систем. Они могут формировать заявки на консультации, выбирая необходимые услуги из каталога, и получать индивидуальные решения для своих задач. При создании новой заявки клиент указывает ключевые параметры: номер телефона для связи, уровень приоритета и услуги, с дополнительными комментариями. Заявка изначально создается в статусе черновика, что позволяет клиенту на этапе подготовки вносить необходимые изменения, добавлять дополнительные данные и редактировать информацию. Это значительно упрощает планирование.

После того как заявка полностью оформлена, сотрудник отправляет её на рассмотрение модератору. На этом этапе редактировать заявку становится невозможно. Модератор получает доступ к заявке и проверяет её на соответствие требованиям безопасности и реальностью выполнения заявки [1]. Модераторы обладают ключевыми функциями для работы с заявками. Они могут просматривать все поступившие заявки, фильтровать их по дате формирования, статусу и создателю заявки. Это позволяет модераторам быстро находить нужную информацию и сосредотачиваться на заявках, требующих внимания.

После проверки модератор может принять одно из следующих решений: одобрить заявку, отклонить её. В случае одобрения заявка получает статус “завершена”, а клиент может считать процесс завершенным. Если заявка не соответствует требованиям, она отклоняется.

Кроме обработки заявок, модераторы имеют возможность управлять данными об услугах [2]. В их распоряжении функционал для добавления новых услуг, редактирования информации о существующих услугах и удаления устаревших данных. Таким образом, модераторы поддерживают актуальность данных, что позволяет клиентам выбирать только актуальные услуги.

Система «Consulting» по ИТ-безопасности представляет собой комплексное решение, которое помогает оптимизировать процессы, связанные с организацией процесса оказания консультационных услуг. Возможность добавлять несколько услуг в рамках одной заявки позволяет минимизировать временные и ресурсные затраты, упрощая планирование и ускоряя подготовку к перелетам.

Одной из ключевых задач системы является обеспечение высокого уровня безопасности и соблюдение регулятивных норм. Проверка заявок модераторами на всех этапах помогает выявлять и устранять потенциальные проблемы до завершения заявок. Модераторы строго контролируют соответствие заявок требованиям безопасности [1], что делает процесс прозрачным и надежным.

Таким образом, система «Consulting» по ИТ-безопасности представляет собой комплексное решение, направленное на оптимизацию процесса оказания консультационных услуг, повышение уровня информационной безопасности организаций и обеспечение эффективного взаимодействия между клиентами и консультантами. Модераторы, в свою очередь, получают все необходимые инструменты для контроля, проверки и корректировки заявок, а также для управления данными о заявках. Благодаря этому достигается максимальная эффективность и прозрачность работы, поддерживается высокий уровень безопасности, а все заказы выполняются в строгом соответствии с регулятивными требованиями.

Функциональные возможности клиентов и модераторов представлены на диаграмме прецедентов (рисунок 1).

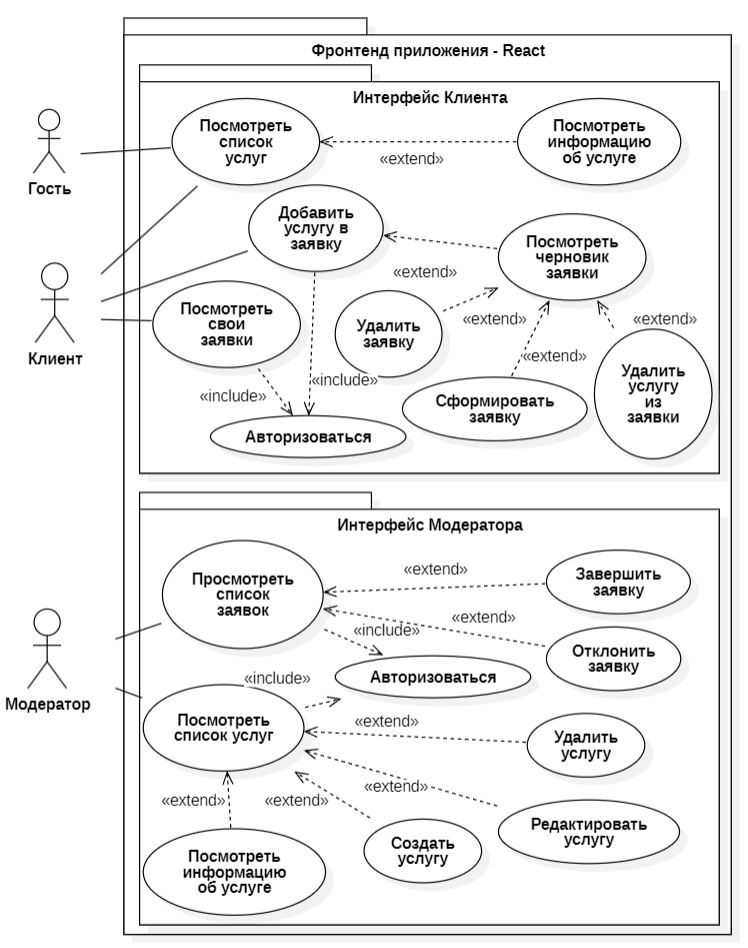


Рисунок 1 – Диаграмма прецедентов

Гостям доступен просмотр списка доступных услуг. После регистрации гости становятся клиентами. Они могут выбирать услуги для заявки, формировать заявку и просматривать список своих заявок.

После этого модератор может просмотреть заявку и принять решение о ее согласовании или отклонении. Возможные статусы заявки отображены на диаграмме состояний(рисунок 2).

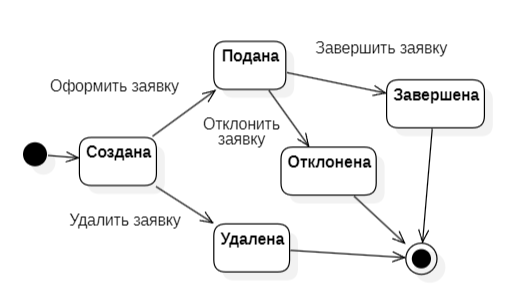


Рисунок 2 – Диаграмма состояний заявок

Заявки обрабатываются модераторами. В результате обработки заявки ее либо одобряют, либо отклоняют. Модератору также доступны операции для работы с услугами: просмотр всех доступных услуг, редактирование информации о них, создание новых и удаление существующих услуг, а также просмотр списка всех услуг в табличном виде.

Процесс оформления заявки отображен на диаграмме деятельности (рисунок 3).

В начале взаимодействия с системой оформления заявок на перелеты клиент запрашивает список доступных для добавления в заявку услуг. При необходимости клиент может добавить услуги в текущую черновую заявку. После чего клиент может продолжить выбор услуг из каталога или перейти к оформлению заявки. В случае, если клиент выбрал оформление заявки, ему необходимо выбрать услуги. Затем указать контактный номер телефона, уровень приоритета и дополнительные комментарии. По нажатии кнопки «Сформировать заявку» заявка формируется и становится доступна в интерфейсе модератора.

Модератор имеет возможность просматривать заявки клиентов, а также принимать решение об одобрении или отклонении заявки.

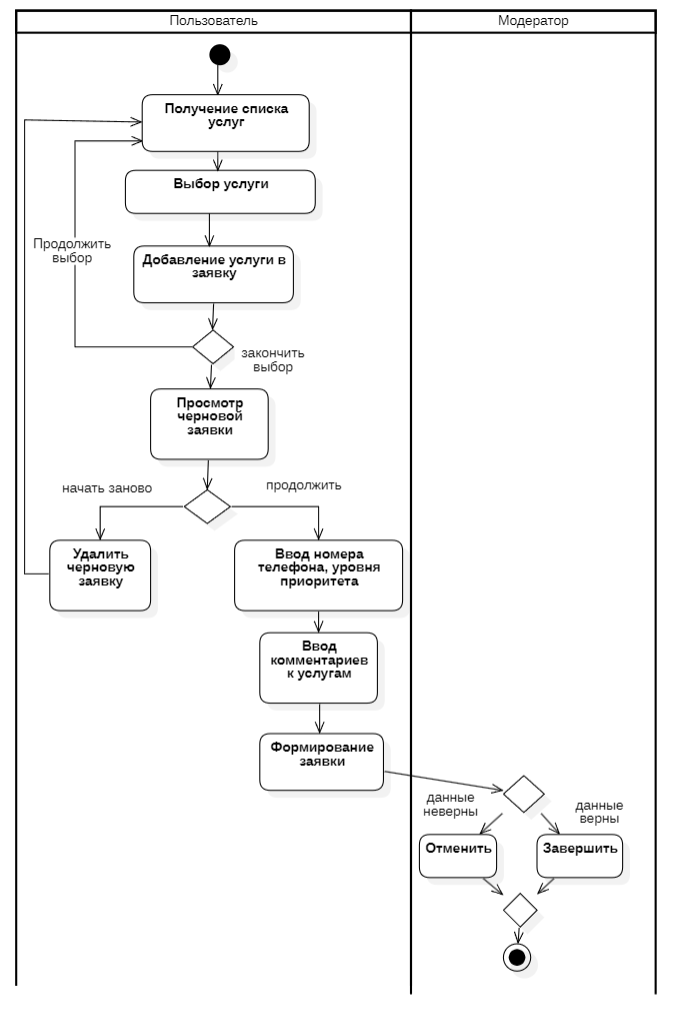


Рисунок 3 – Диаграмма деятельности

1. **АРХИТЕКТУРА**

Архитектура системы отображена на диаграмме развертывания (рисунок 4).

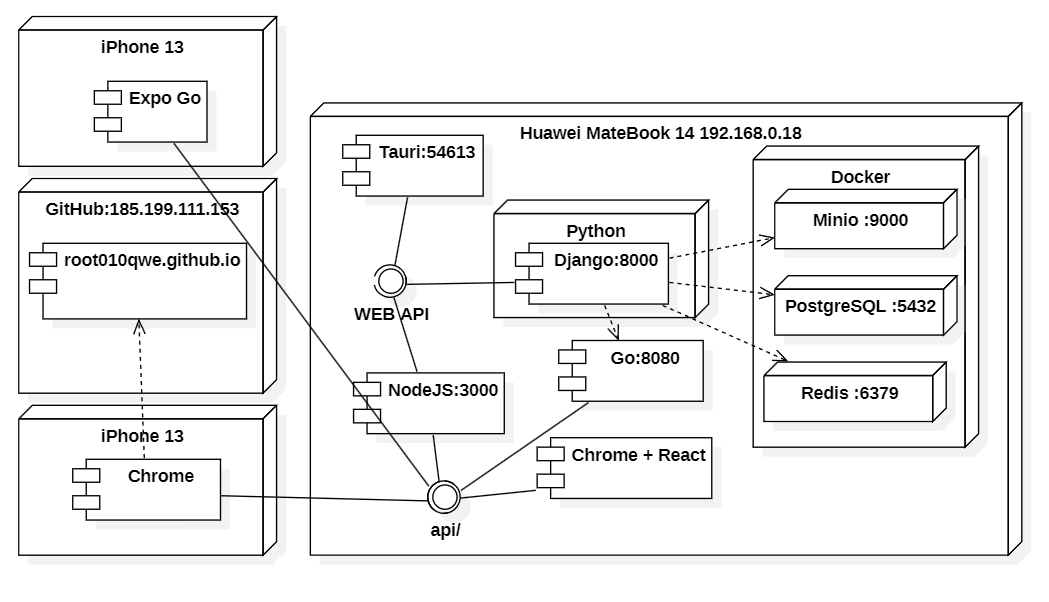


Рисунок 4 – Диаграмма развертывания

Веб-сервис, реализованный на фреймворке Django с использованием DRF [3] связан с серверами Minio [5], Redis [4] и СУБД PostgreSQL [6]. В Redis [4] хранятся активные сессии пользователей, при выходе пользователя из системы – удаляются.

Язык программирования Python был выбран в силу его простоты, а также популярности. Использование Django обусловлено тем, что этот фреймворк имеет достаточно широкие возможности и является стандартом индустрии. Веб-сервис Django является общим для веб-сервера и десктопного приложения Tauri [7].

Данные хранятся в СУБД PostgreSQL [6]. Она была выбрана, поскольку, на ровне с Python и Django она является стандартом современной индустрии разработки. Структура данных отражена на ER диаграмме (рисунок 5). Модель услуг представляет собой набор полей, отражающих параметры услуги. Данные об услугах хранятся в таблице consulting\_services. Для хранения в одной заявке используется промежуточная таблица связи М-М service\_requests. Таблица consulting\_requests представляет собой список заявок. Данные о пользователях системы хранятся в таблице сonsulting\_customuser.

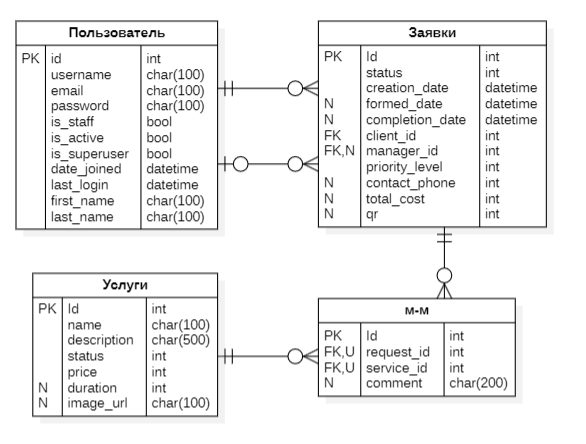


Рисунок 5 – ER диаграмма.

Веб-сервер реализован с использованием языка TypeScript и фреймворка React [8]. Устройство бэкенда приложения разработанной системы приведено на диаграмме классов бэкенда (рисунок 6). Пользователи взаимодействуют с доменами. Домены связаны с моделями. Модели имеют связи с таблицами в базе данных.

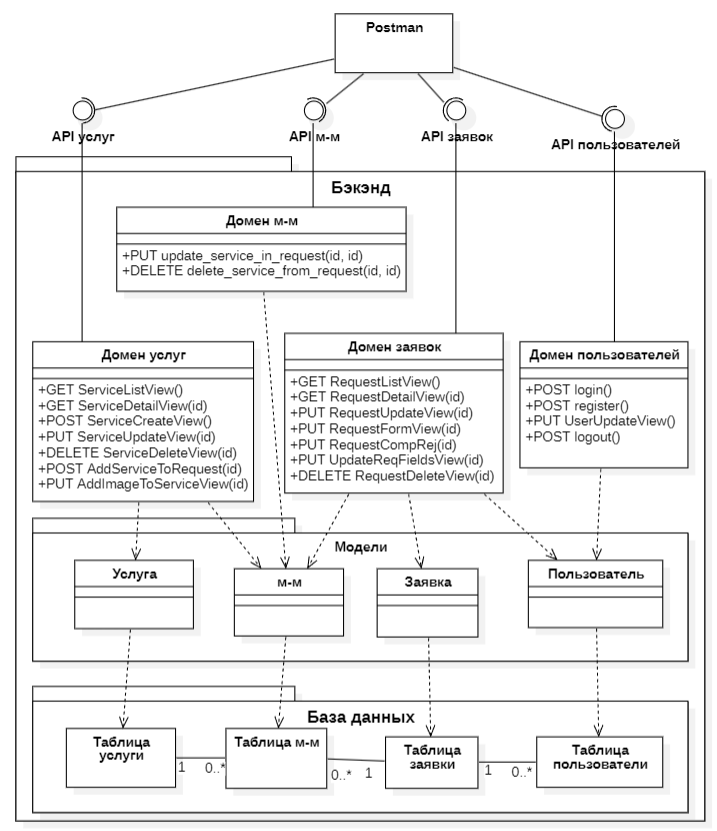


Рисунок 6 – Диаграмма классов бэкенда

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.

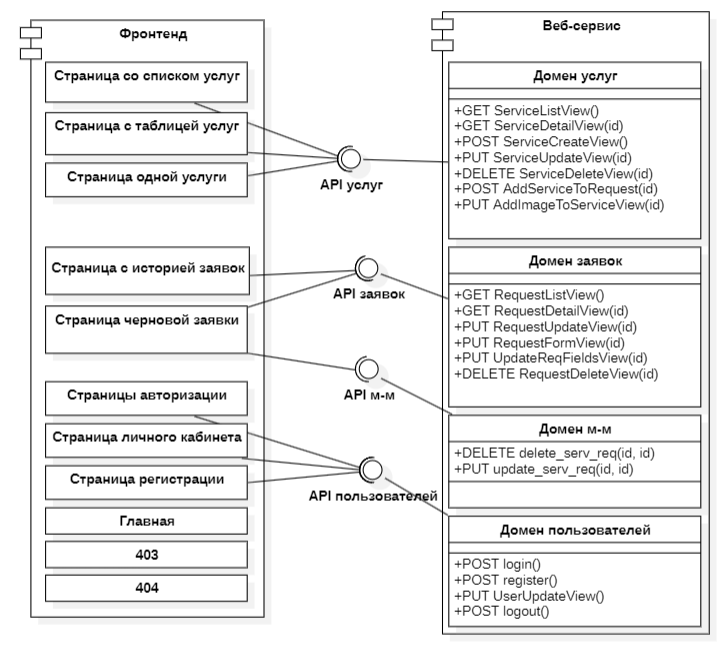


Рисунок 7 – Диаграмма классов фронтенда

1. **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, договоры, заявки на счета, договоры в заявках на счета. Методы следуют правилам REST API.

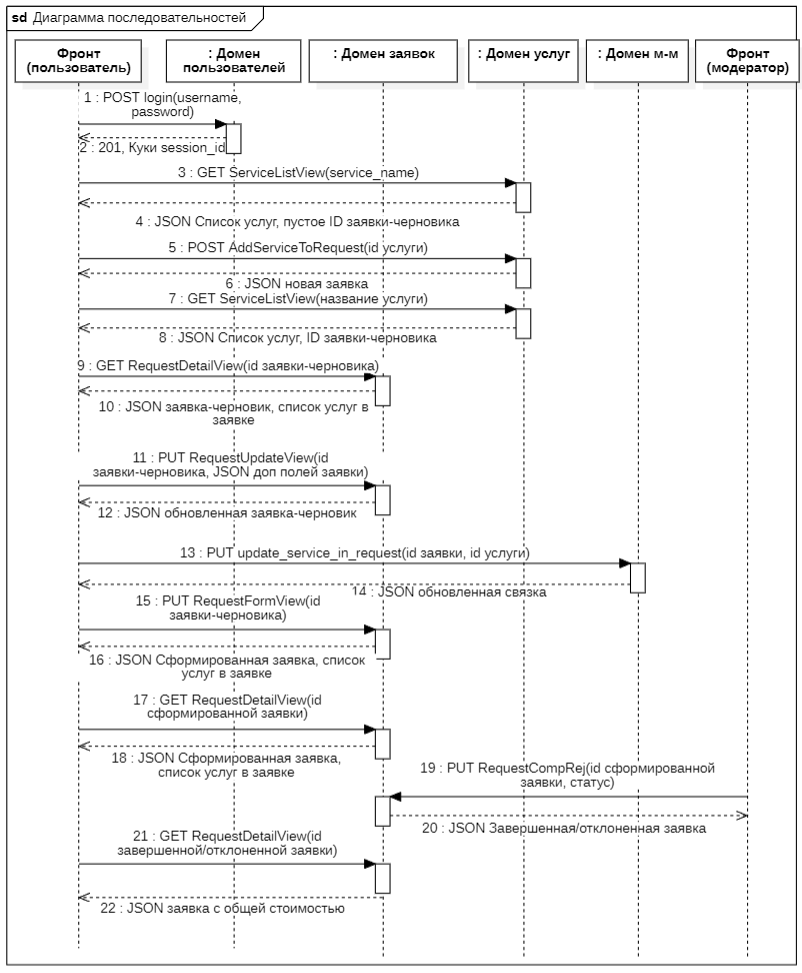


Рисунок 8 — Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. При помощи графического интерфейса клиент отправляет запрос, передавая в нем логин и пароль. Если аккаунт с указанными данными существует в базе, на клиент возвращается информация о пользователе и устанавливаются куки с полем session\_id (идентификатор текущей сессии) в ответном запросе. Если аккаунта с введённым логином не существует или пароль введён неверно, на клиент возвращается ошибка. В таком случае пользователю нужно пройти регистрацию или ввести корректные логин и пароль. На этом же этапе происходит проверка: является ли пользователь клиентом или модератором.

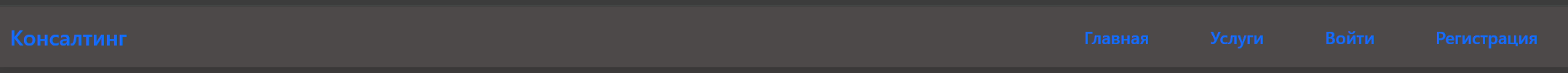
Графический интерфейс пользователя запрашивает у веб-сервиса список доступных услуг, которые возвращаются в формате JSON. Клиент выбирает услугу, которая будет добавлена в заявку, и, нажимая на кнопку «Добавить», отправляет запрос на добавление выбранной услуги в свою черновую заявку. Это действие может быть выполнено несколько раз.

Когда клиент определился с выбором услуг и укажет все необходимые параметры заказа (контактный номер телефона, уровень приоритета и дополнительные комментарии), он нажимает кнопку «Сформировать заявку» в графическом интерфейсе. После этого приложение отправляет запрос на веб-сервис для формирования заявки. Клиент может отслеживать статус сформированных заявок в соответствующем разделе графического интерфейса.

Процесс отслеживания заявок модераторами также происходит при помощи графического интерфейса. Модераторы могут просматривать список всех оформленных клиентами заявок; завершать или отклонять их при помощи соответствующих кнопок. Доступна фильтрация по логинам пользователей, а также по статусам и датам оформления заявок. Модераторы FAA также могут редактировать список доступных услуг, изменять данные о каждой услуге и добавлять новые услуги в систему. Для каждой из этих операций существует соответствующий метод, отправляемый на веб-сервис.

1. **ОПИСАНИЕ ИНТЕРФЕЙСА**

Главное меню приложения включает разделы, доступность которых зависит от роли пользователя. Рисунок 9 иллюстрирует разделы доступные гостю.

 Рисунок 9 — Меню приложения для гостя

На рисунке 10 изображено меню, которое видит клиент.

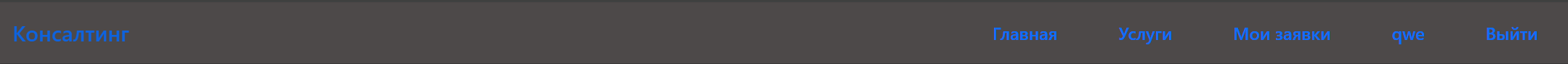


Рисунок 10 — Меню приложения для клиента

На рисунке 11 изображено меню, которое видит модератор.



Рисунок 11 — Меню приложения для модератора

На странице с формой регистрации (рисунок 12) отображается форма, при помощи которой гость может создать новый аккаунт. После успешной регистрации открывается форма аутентификации.

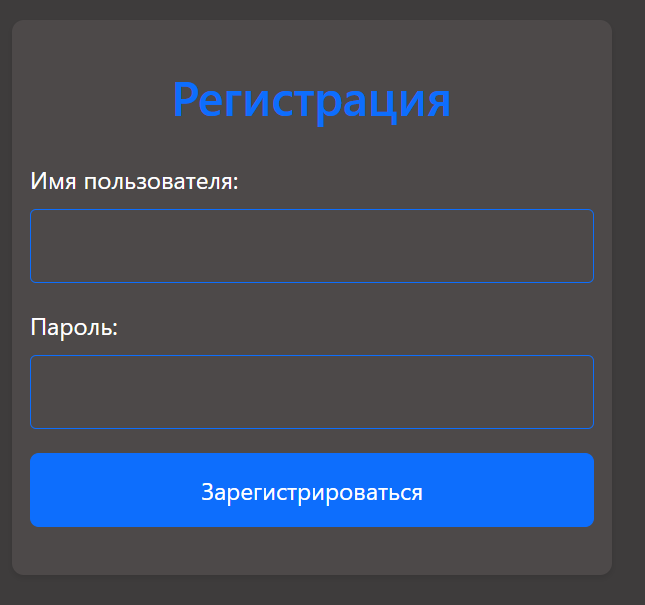


Рисунок 12 — Форма регистрации

На странице с формой входа (рисунок 13) отображается форма, через которую пользователь может войти в свой аккаунт. При успешном вводе данных аккаунта на клиент приходят куки с идентификатором текущей сессии.

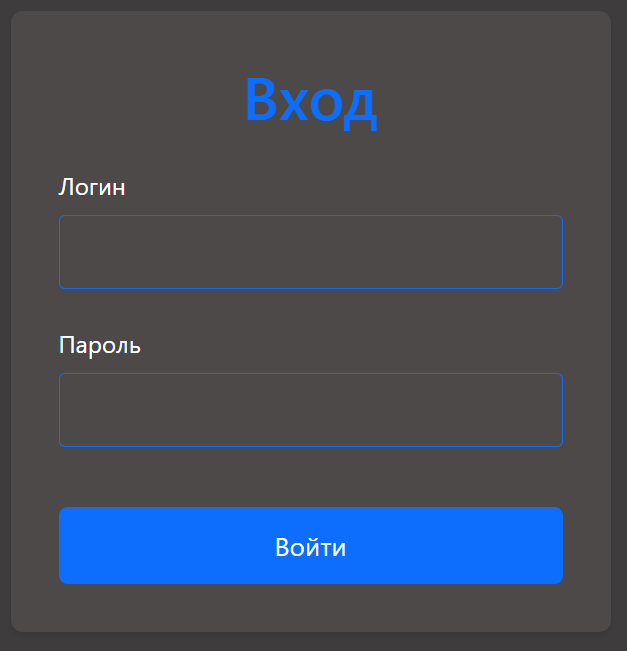


Рисунок 13 — Форма входа

После входа в аккаунт пользователь попадает на главную страницу (рисунок 14), на которой располагается описание разработанной системы.

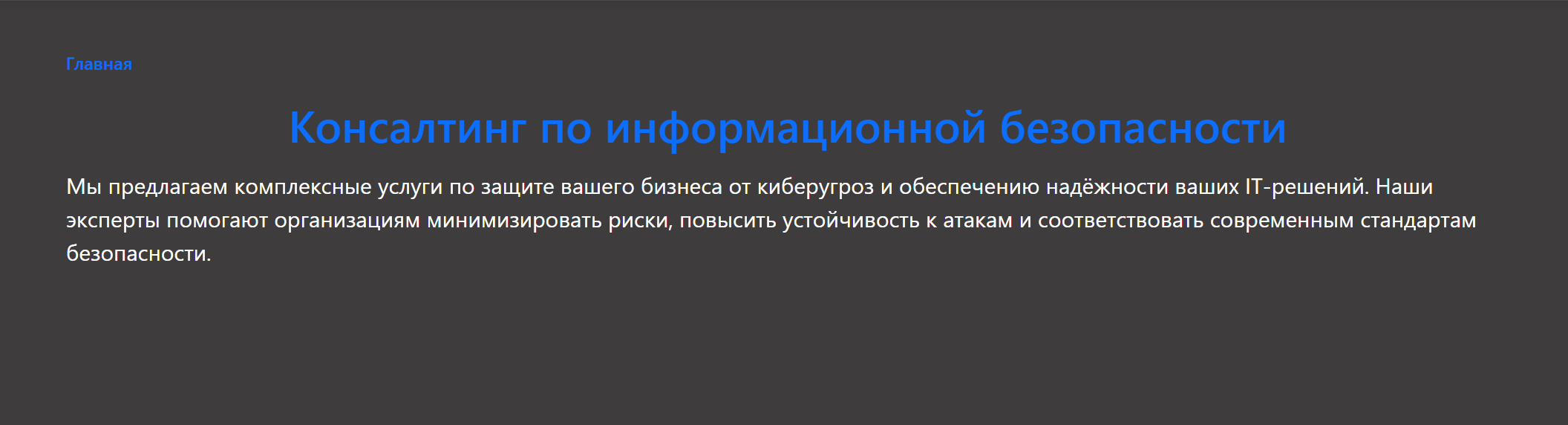


Рисунок 14 — Главная страница

Страница списка услуг (рисунок 15) содержит список услуг в виде карточек доступных к добавлению в перелет. При нажатии на кнопку «Подробнее» на карточке, открывается страница с подробным описанием выбранной услуги.

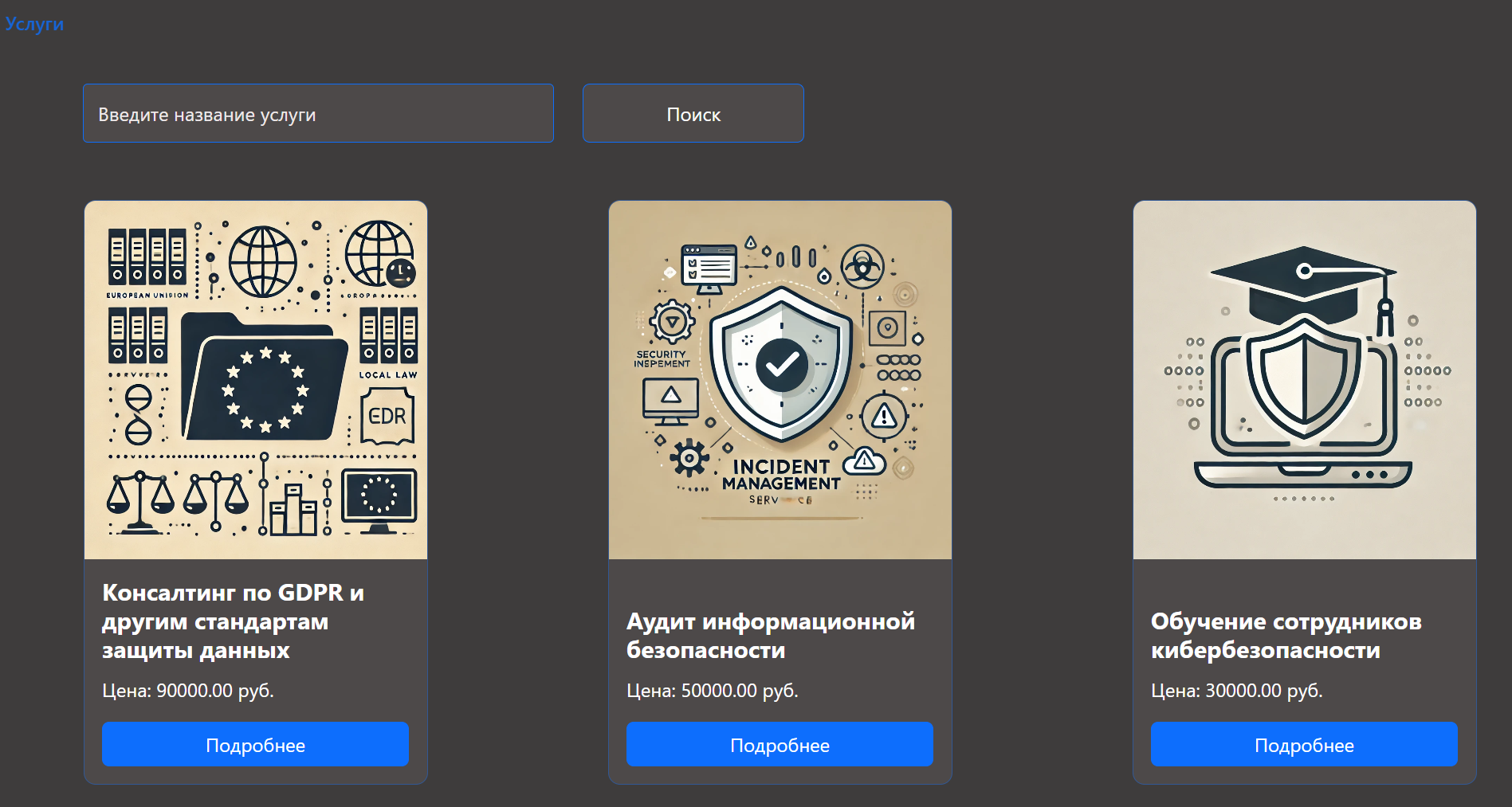


Рисунок 15 — Страница списка услуг для гостя

На странице услуги представлена информация о наименовании, стоимости и продолжительности выполнения, а также ее подробное описание и картинка (рисунок 16).

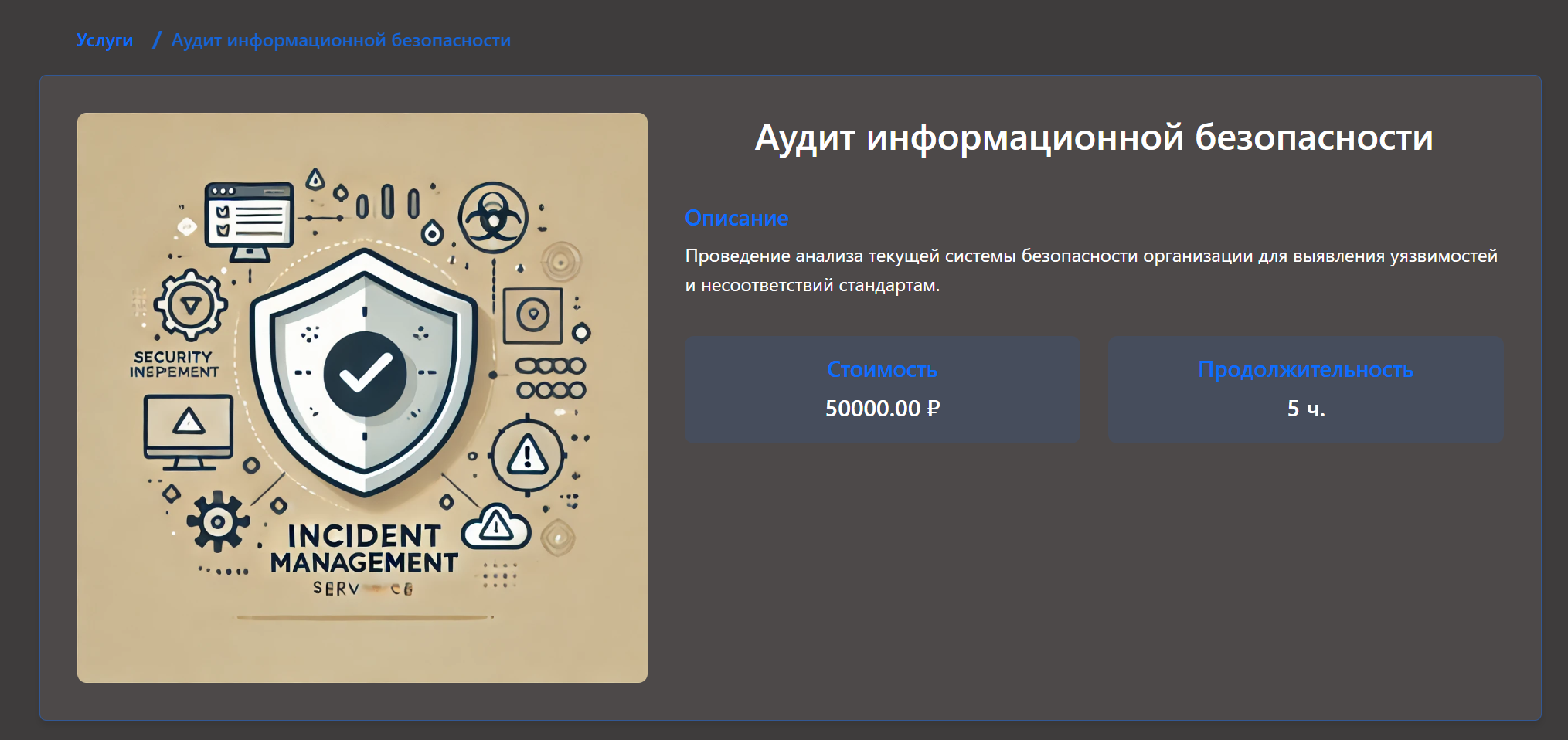


Рисунок 16 — Страница с описанием услуги

В отличие от гостя клиенту в карточке каждой услуги доступна кнопка «Добавить» (рисунок 16), с помощью которой он может добавить услугу в черновик заявки. Просмотреть черновик заявки, при его наличии, автор может, нажав на кнопку с иконкой «Корзины» в правом верхнем углу страницы. Вверху страницы также располагается поисковая строка, позволяющая найти интересующую услугу по названию. Поиск не чувствителен к регистру.

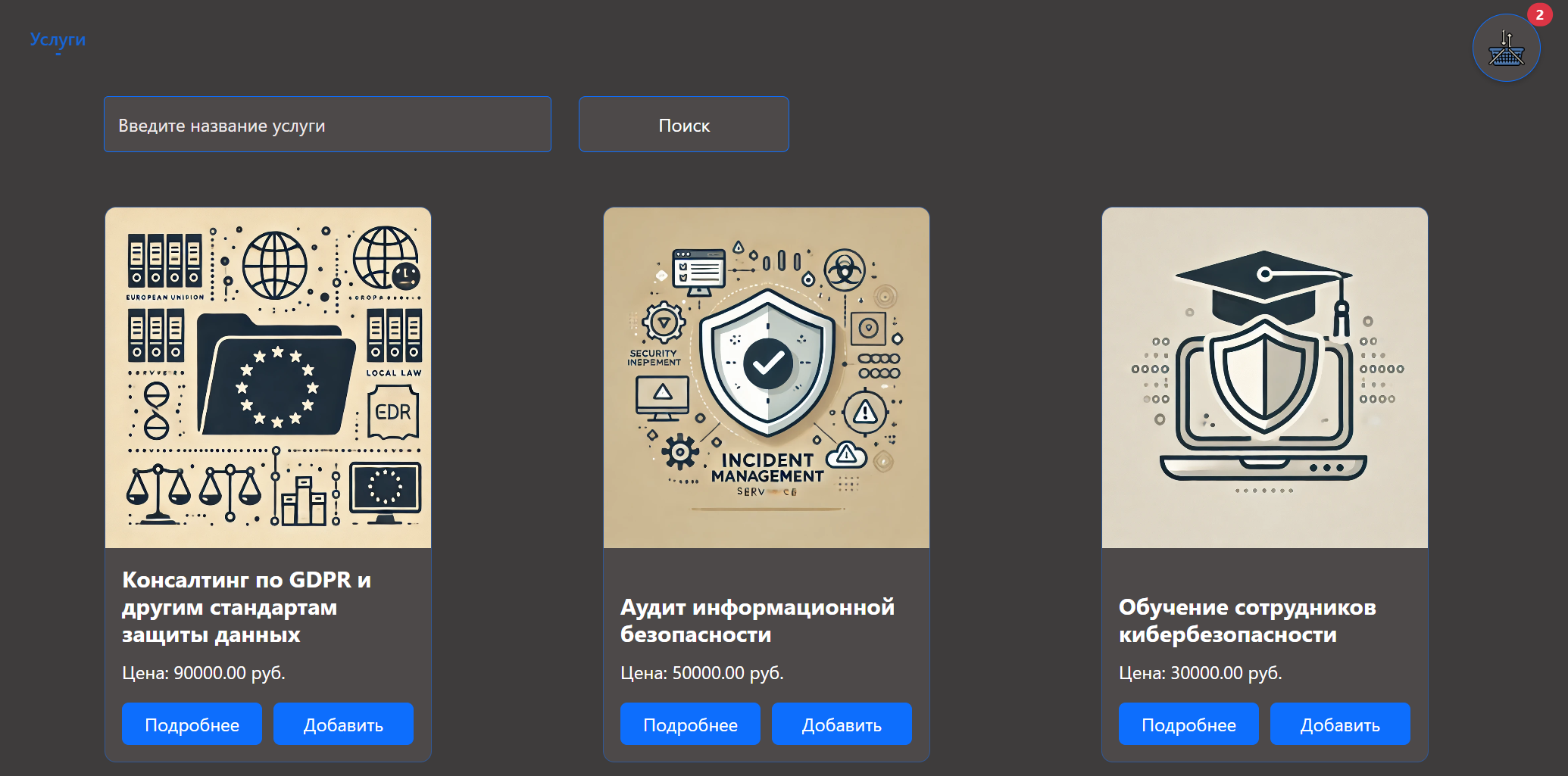


Рисунок 16 — Страница списка услуг для клиента

На странице одной заявки (рисунок 17) отображается текущая заявка на перелет, есть возможность удалить услугу из заявки, указать дополнительный комментарий для услуги, указать контактный номер телефона, уровень приоритета, посмотреть все заявки, а также отправить заявку на проверку или удалить ее. Редактирование оформленных заявок запрещено.

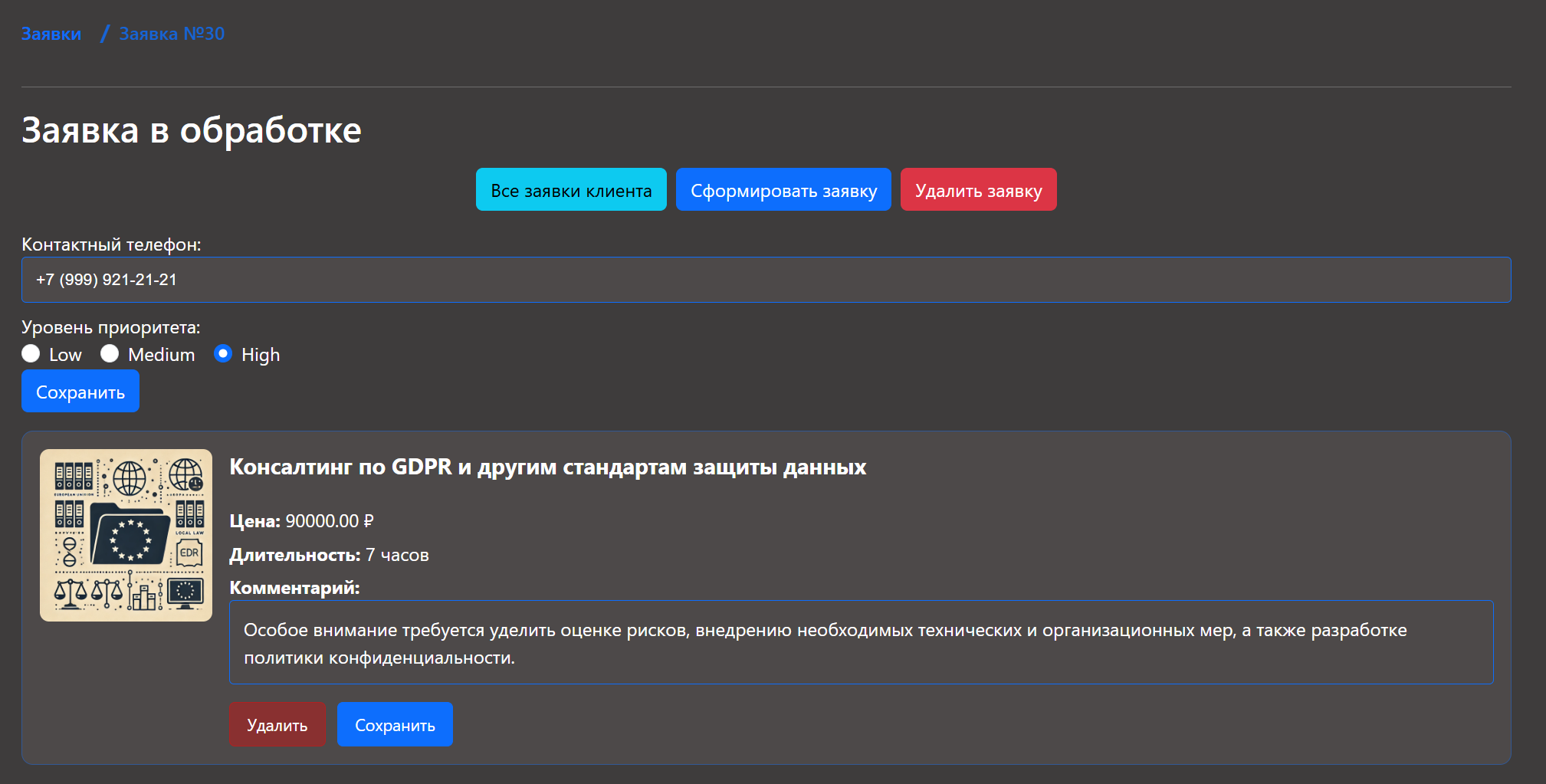


Рисунок 17 – Страница заявки

На странице списка заявок (рисунок 18) клиенты могут просматривать созданные ими заявки. На этой странице можно посмотреть подробную информацию о заявке, нажав на id перелета

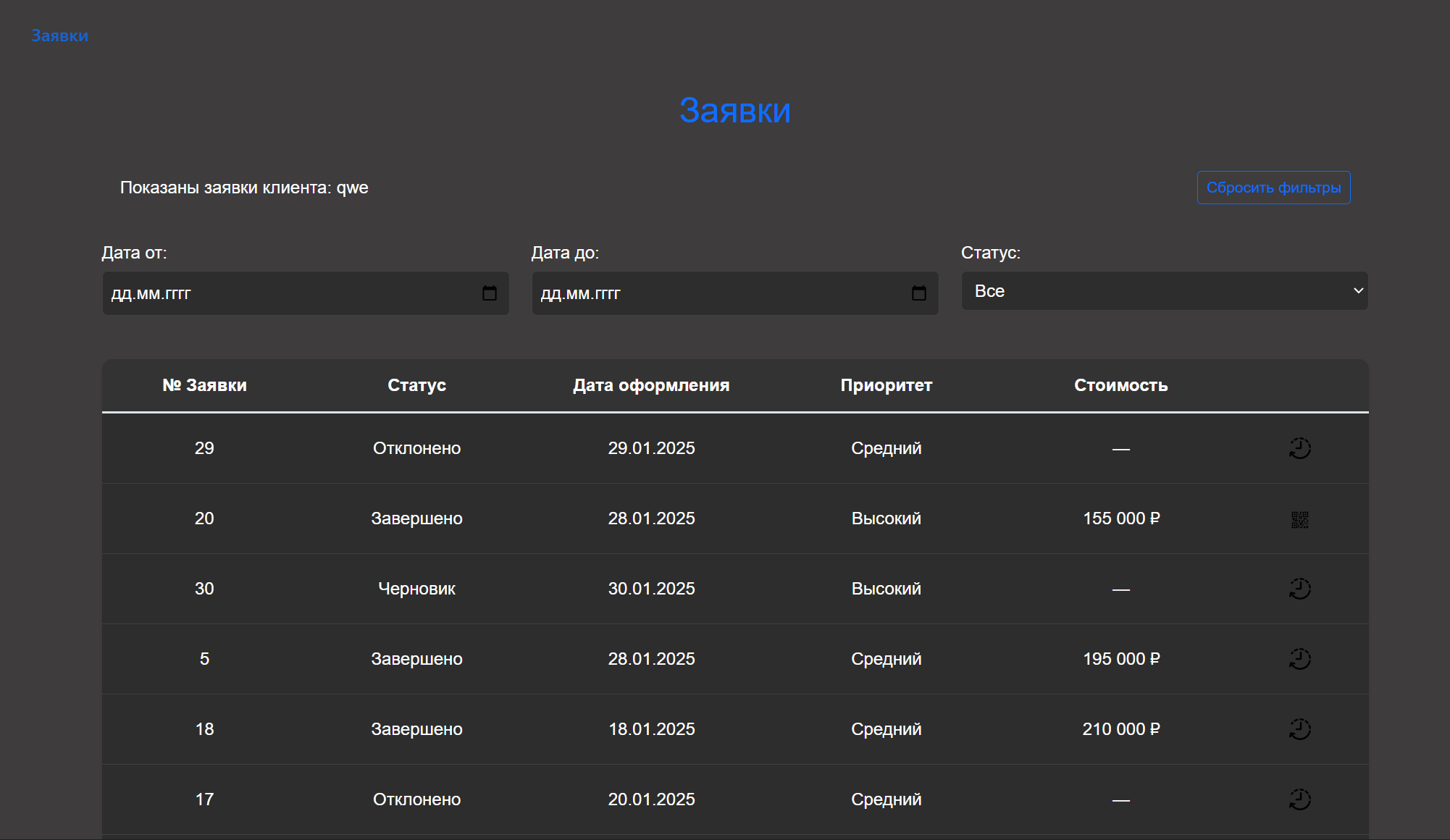


Рисунок 18 – Страница списка заявок для клиента

Пользователь может изменить свой пароль и личные данные на странице редактирования данных пользователя (рисунок 19). Она открывается при нажатии на логин пользователя в меню.

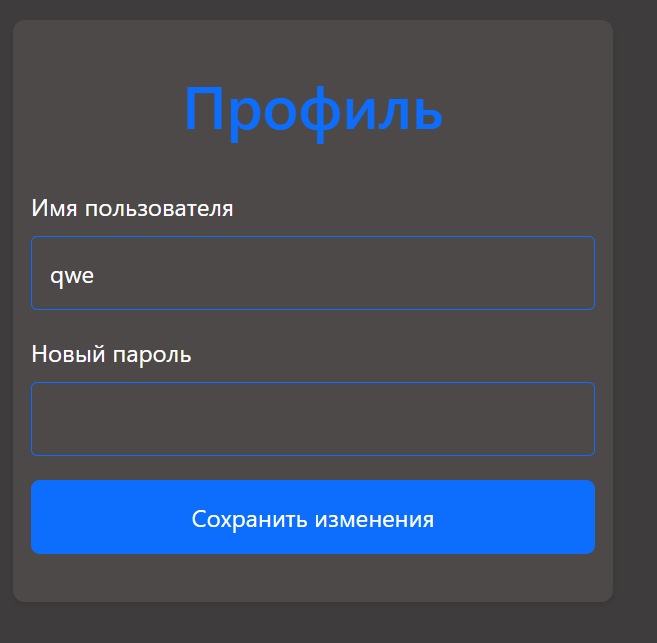


Рисунок 19 – Страница редактирования информации о пользователе

Модератор может просматривать информацию о всех услугах, находящихся в системе на странице списка услуг (рисунок 20). У него есть возможность добавить новая услуга, отредактировать или удалить существующий.

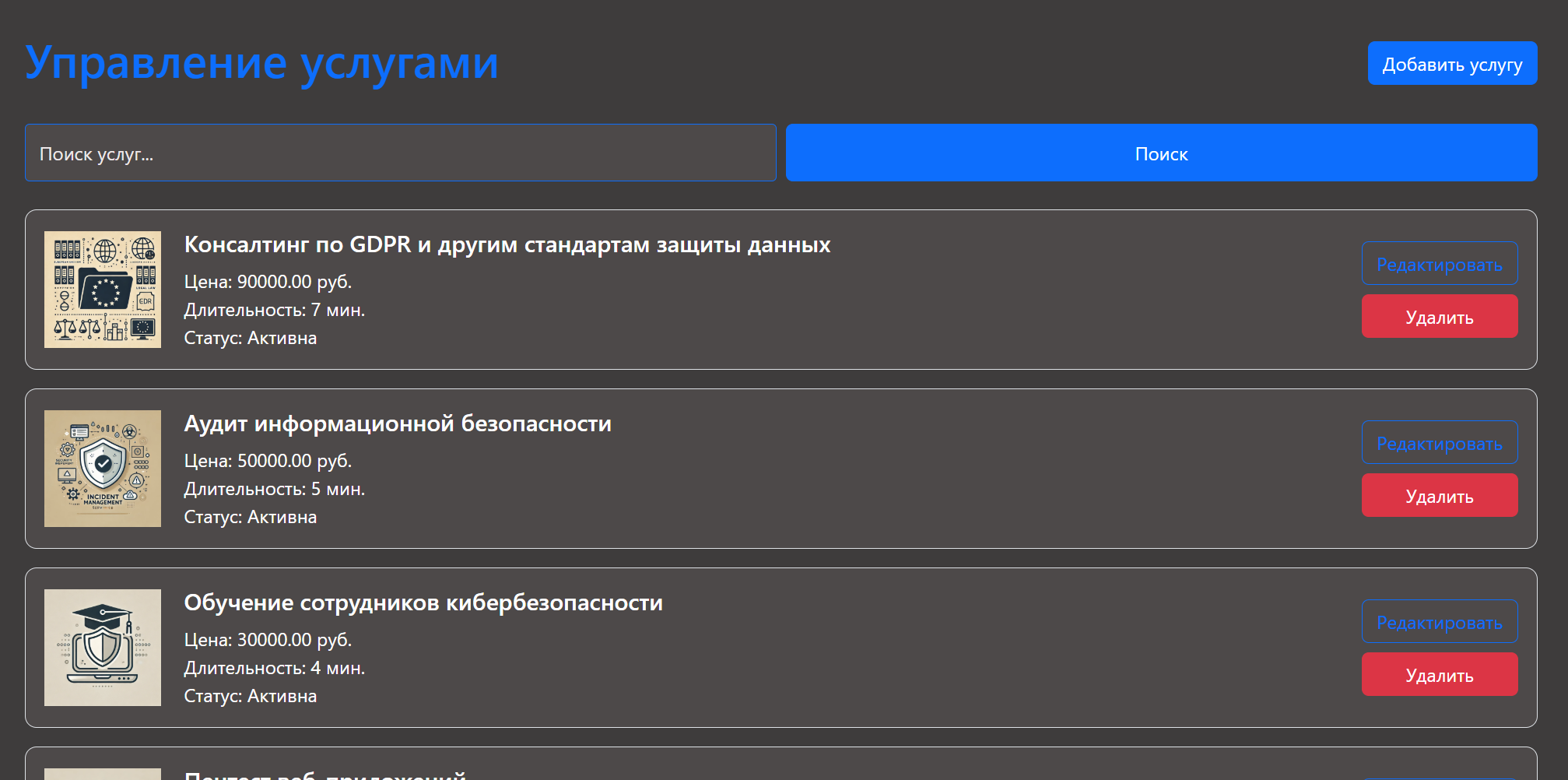


Рисунок 20 – Страница списка услуг у модератора

На странице списка всех заявок пользователей (рисунок 21), модератор может выполнить или отклонить заявки пользователей путём нажатия соответствующих кнопок в интерфейсе. Модератор может просматривать подробную информацию о заявке с помощью, нажав на неё. Заявки можно фильтровать по логину пользователя, статусу и датам формирования и исполнения.

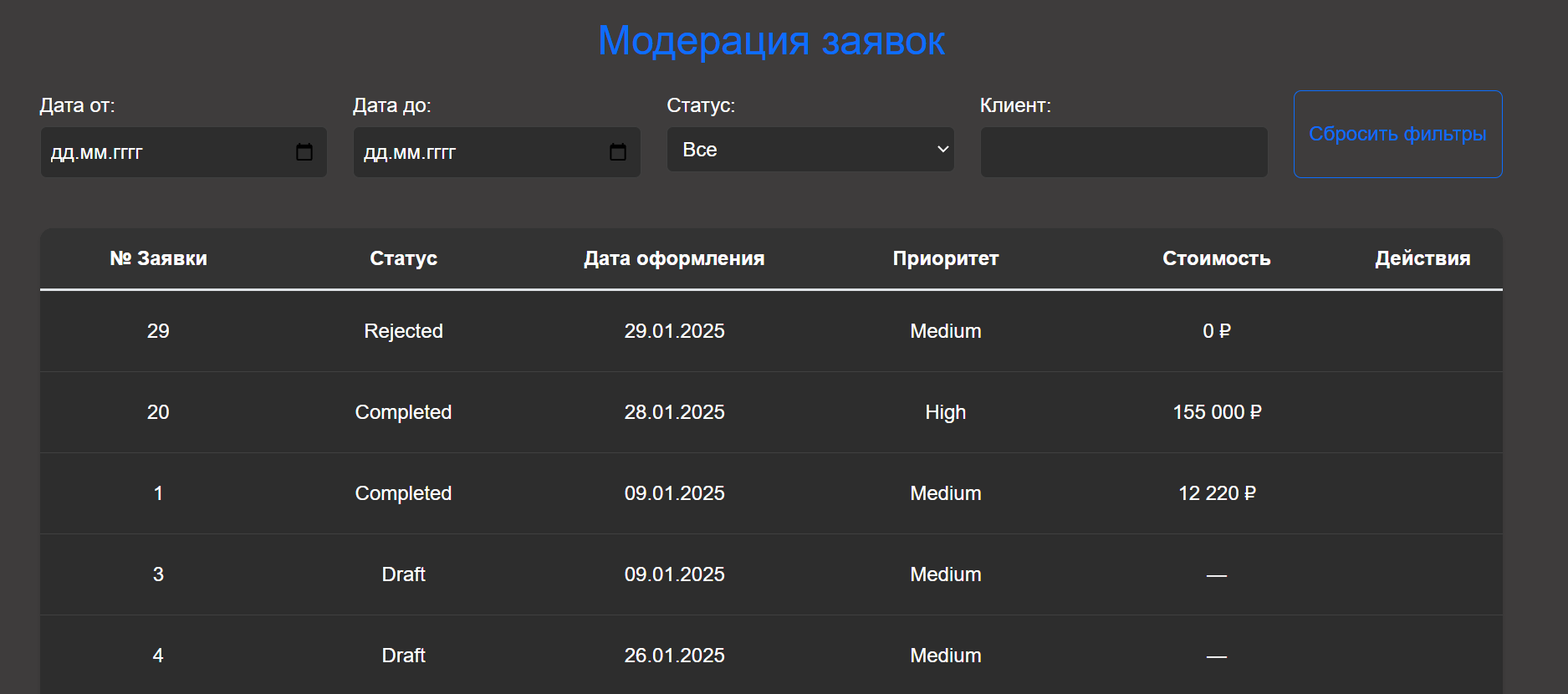




Рисунок 21 – Страница списка заявок модератора

С помощью страницы редактирования и создания услуги(рисунок 22) модератор FAA может добавлять новые услуги или изменять существующие.

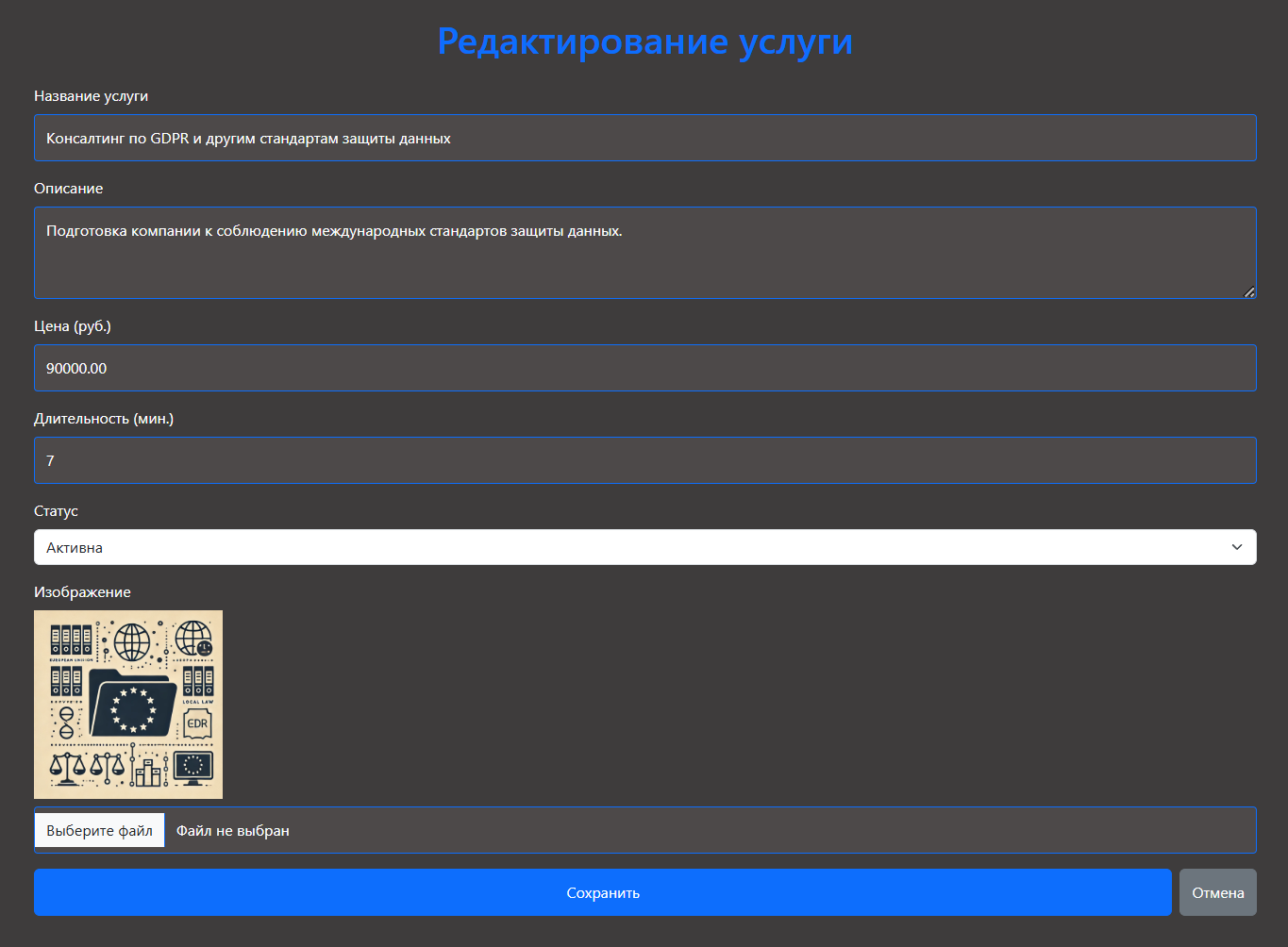
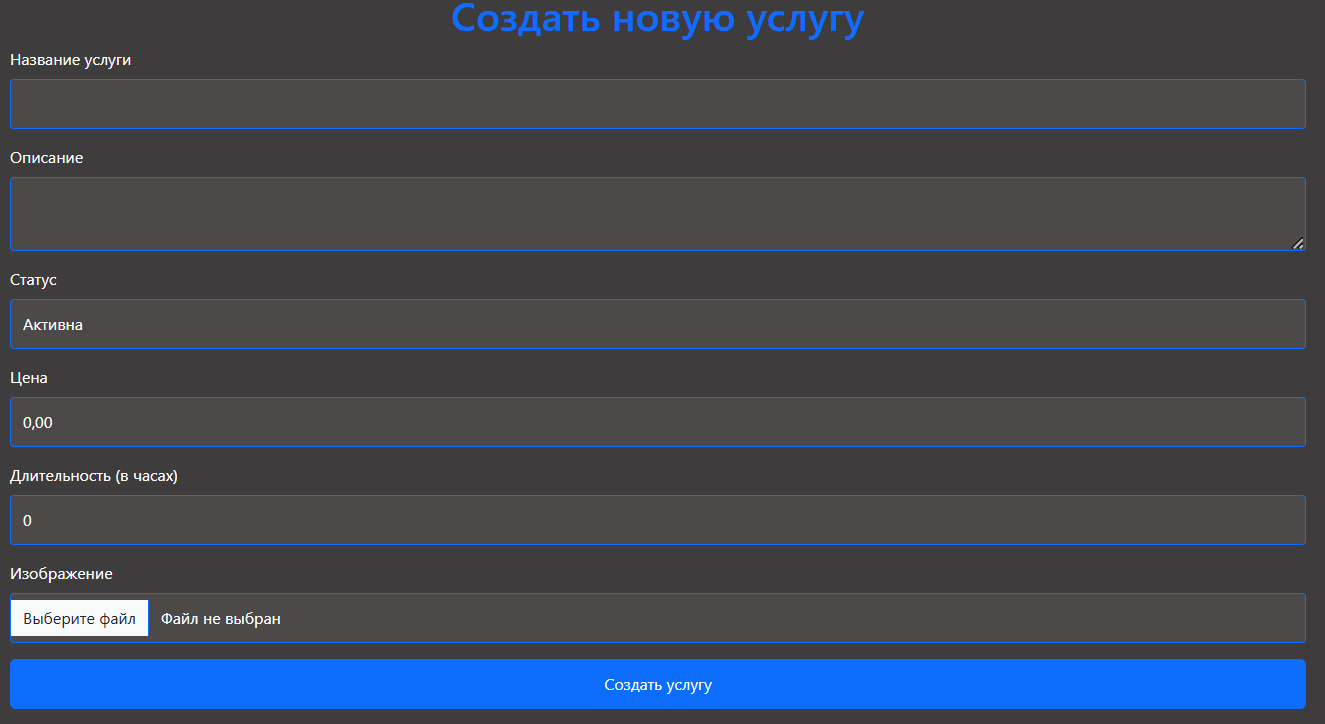
 

Рисунок 22, 23 – Страница редактирования и создания услуги

В случае, если пользователь запрашивает несуществующую станицу, он перенаправляется на страницу ошибки 404 (рисунок 24).

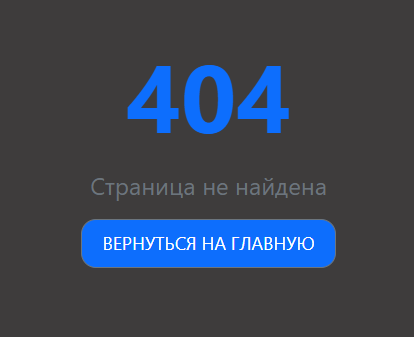


Рисунок 24 – Страница 404

В случае, если пользователь пытается получить доступ к страницам, для просмотра которых ему не хватает прав, он перенаправляется на страницу ошибки 403 (рисунок 25).

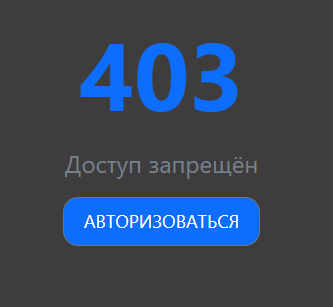
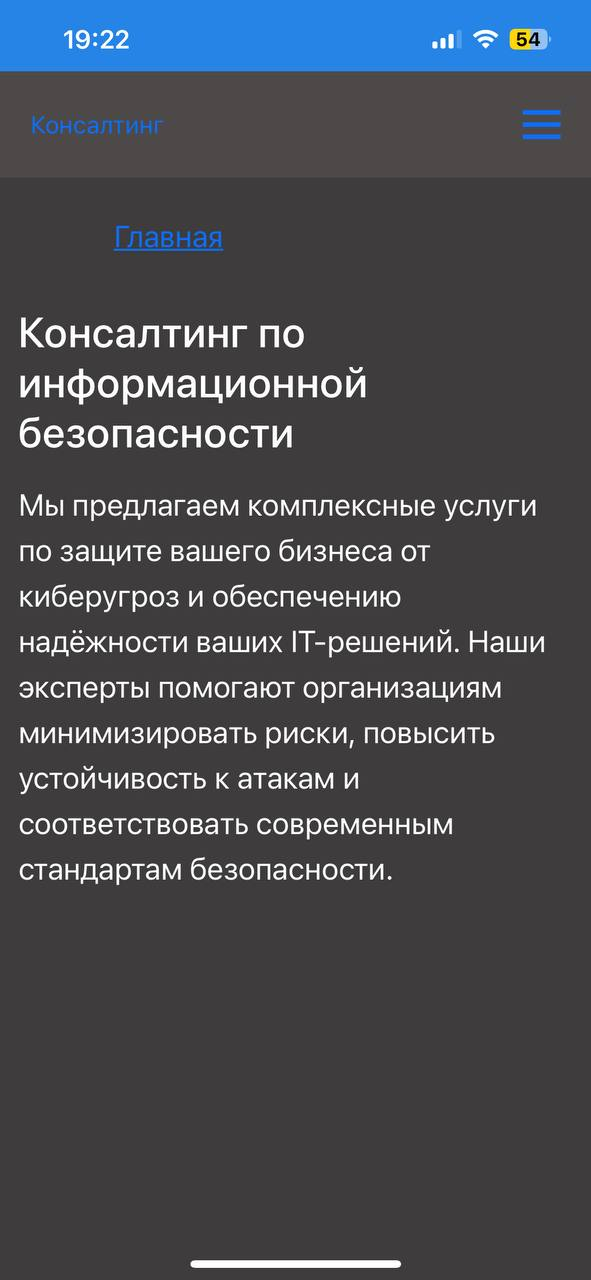


Рисунок 25 – Страница 403

Приложение поддерживает возможность работы как прогрессивное веб-приложение (PWA), что позволяет пользователю устанавливать его на устройство как нативное приложение. Интерфейс адаптирован для мобильных устройств, сохраняя функциональность основной версии, и состоит из трех страниц: главная страница PWA (рисунок 26), страница списка услуг (рисунок 27) и страница с подробным описанием услуги (рисунок 28).

  
Рисунок 26 – Главная страница PWA

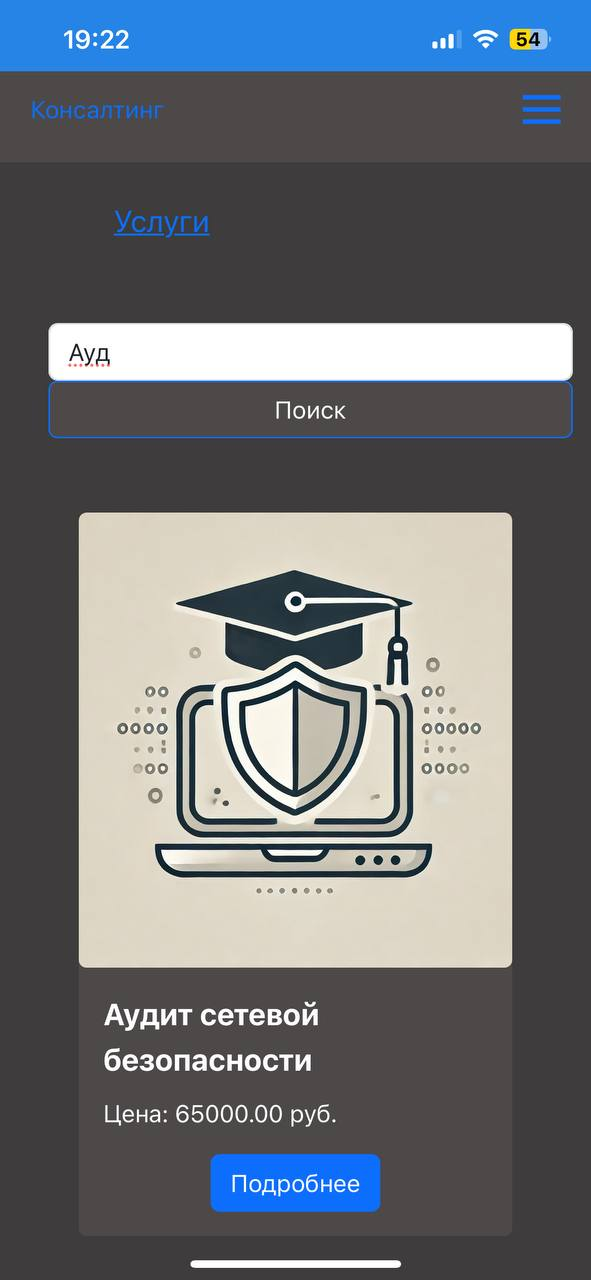


Рисунок 27 – Страница списка услуг PWA



Рисунок 28 –Страница с подробным описанием услуги PWA

Была разработана десктопная версия приложения на основе Tauri. Интерфейс состоит из трех страниц: главная страница (рисунок 29), страница списка услуг (рисунок 30) и страница с подробным описанием услуг (рисунок 31).

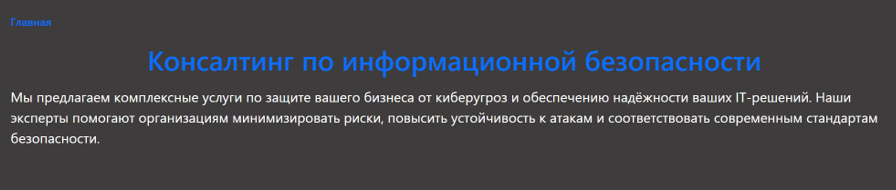


Рисунок 28 – Главная страница Tauri



Рисунок 29 – Страница списка услуг Tauri

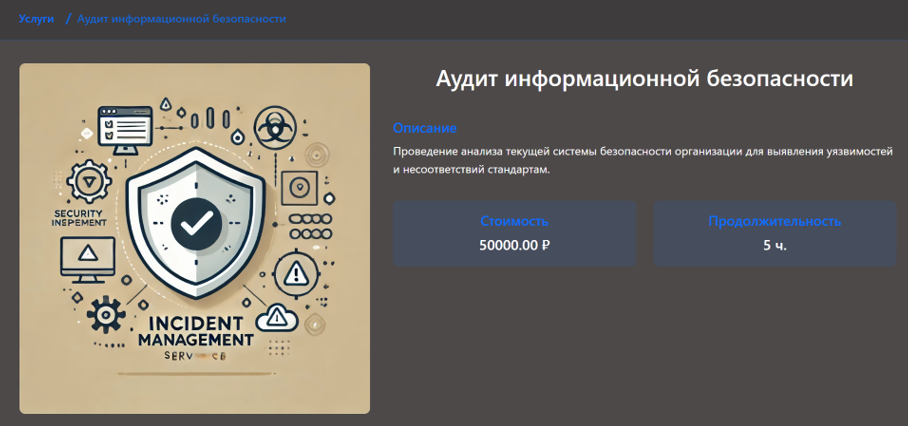


Рисунок 30 – Страница с подробным описанием услуг Tauri

Была разработана мобильная версия приложения на основе React Native. Интерфейс включает три основных экрана: экран списка услуг (рисунок 32) и экран с подробным описанием услуг (рисунок 33).



Рисунок 32 – Экран списка услуг React Native



Рисунок 33 – Экран с подробным описанием услуг React Native

**ЗАКЛЮЧЕНИЕ**

В ходе работы были достигнуты следующие результаты:

1. Создан MVP и базового дизайн на основе in4security.com
2. Создана база данных для хранения информации о услугах и заявках.
3. Создан веб-сервис на языке Python с использование Django Rest Framework.
4. Реализована авторизация и хранение сессий в Redis.
5. Разработан базовый SPA на React для гостя
6. Внедрена адаптивность, менеджер состояний Redux Toolkit, PWА, разработано Tauri приложение
7. Завершена разработка интерфейса клиента в React
8. Реализован интерфейс модератора FAA в React
9. Реализовано десктопное приложение Tauri
10. Приложение развернуто при помощи GitHub Pages и доступно по ссылке: https://root010qwe.github.io/Consalting\_Frontend/
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Реализовать мобильное приложение на React Native.
13. Оформлен git-репозиторий на сервисе GitHub, содержащий исходный код проекта: https://github.com/Root010qwe/Consalting\_Frontend

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Регулирование космических полетов и лицензирование FAA [Электронный ресурс] // FAA. URL: https://www.faa.gov/space (дата обращения: 11.09.2024);
2. Starship Overview – Описание космического корабля [Электронный ресурс] // SpaceX. URL: https://www.spacex.com/vehicles/starship/ (дата обращения: 11.09.2024);
3. Документация по DRF [Электронный ресурс] // Django-rest-framework. URL: https://www.django-rest-framework.org/topics/documenting-your-api/ (дата обращения: 20.10.2024).
4. Документация по Redis [Электронный ресурс] // Netlify. URL: https://master--redis-doc.netlify.app/docs/ (дата обращения: 10.10.2024);
5. Документация Minio [Электронный ресурс] // Min. URL: https://min.io/docs/minio/kubernetes/upstream/index.html (дата обращения: 09.09.2024);
6. Документация PostgreSQL [Электронный ресурс] // Postgresql. URL: https://www.postgresql.org/docs/ (дата обращения: 15.09.2024);
7. Документация по Tauri [Электронный ресурс] // Tauri. URL: https://v2.tauri.app/develop/ (дата обращения: 20.11.2024);
8. Документация по React [Электронный ресурс] // React. URL: https://react.dev/learn (дата обращения: 01.11.2024);

**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. |

**Консалтинг по ИТ безопасности**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

8

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гонов Марат |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2025 г. |  |

Москва - 2025

1. Введение

Система «Consulting» представляет собой веб-сервис, включающий в себя веб-приложение и нативное приложение для организации процесса оказания консультационных услуг в сфере информационной безопасности, управления заявками и взаимодействия между консультантами и клиентами.

1. Назначение разработки

Данная система предназначена для автоматизации полного цикла оказания консультационных услуг в сфере информационной безопасности. С её помощью пользователи (клиенты) могут формировать заявки на консультации, выбирать необходимые услуги из каталога и получать экспертные услуги по защите корпоративной сети, а консультанты — оперативно обрабатывать эти заявки, планировать график консультаций и формировать отчёты о проделанной работе. Таким образом, система способствует оптимизации внутренних процессов, контролю исполнения задач по кибербезопасности и более эффективному взаимодействию между клиентами и специалистами.

1. Стадии и этапы разработки
   1. Создание MVP и базового дизайна на основе in4security
   2. Создание базы данных для хранения информации о получателях и отправках файла в PostgreSQL
   3. Создание веб-сервиса на DRF
   4. Реализация авторизацию и хранение отозванных сессий в Redis
   5. Разработка SPA на React для гостя
   6. Внедрение адаптивности, менеджера состояний Redux Toolkit, PWА, разработка Tauri приложения
   7. Завершение разработки интерфейса отправителя на React
   8. Добавить в приложение интерфейс получателя на React
   9. Разработка десктопного приложения на Tauri
   10. Развертывание приложения при помощи GitHub Pages
   11. Подготовка всей документации (РПЗ, ТЗ и набор диаграмм)
   12. Оформление git-репозитория на сервисе GitHub
   13. Анализ сервисов-аналогов
2. Требования к функциональным характеристикам
   1. Методы HTTP
      1. GET Получение всех услуг
      2. POST Добавление услуги
      3. GET Получение услуги
      4. PUT Изменение услуги
      5. DELETE Удаление услуги
      6. POST Добавление услуги в заявку
      7. POST Изменение/добавление картинки заявки
      8. GET Получение списка всех заявок
      9. GET Получение заявки
      10. PUT Обновление полей заявки
      11. DELETE Удаление услуги из заявки
      12. PUT Формирование заявки
      13. PUT Завершение заявки
      14. DELETE Удаление заявки
      15. PUT Изменение комментария для услуги в заявке
      16. POST Регистрация пользователя
      17. PUT Изменение данных пользователя (Личный кабинет)
      18. POST Аутентификация
      19. POST Деавторизация
   2. Меню
      1. Consulting - перенаправляет на страницу 4.6
      2. Услуги – перенаправляет на страницу 4.7
      3. Заявки – перенаправляет на страницу 4.10
      4. Выпадающее меню
         1. Изменить профиль – перенаправляет на страницу 4.5
         2. Выход – (вызывается метод 4.1.19)
   3. Регистрация
      1. Доступно только гостям
      2. Отображает форму регистрации
         1. Поле логина пользователя
         2. Поле пароля
      3. Действия
         1. Регистрация пользователя – (вызывается метод 4.1.16)
         2. Войти – перенаправляет на страницу 4.4
   4. Аутентификация
      1. Доступно только гостям
      2. Отображает форму аутентификации
         1. Поле логина
         2. Поле пароля
      3. Действия
         1. Войти – (вызывается метод 4.1.18)
         2. Регистрация – перенаправляет на страницу 4.3
   5. Личный кабинет
      1. Доступно аутентифицированному пользователю
      2. Действия
         1. Изменить данные пользователя – (вызывается метод 4.1.17)
   6. Главная
      1. Доступна всем
      2. Отображается статическая информация о сервисе
         1. Назначение сервиса
   7. Список услуг
      1. Доступна всем
      2. Отображаются элементы карточек с услугами, вызывается метод 4.1.1
         1. Картинка
         2. Цена
      3. Действия
         1. Поиск – перенаправляет на страницу 4.7, (используется метод 4.1.1), с фильтрующем параметром
         2. Подробнее – перенаправляет на страницу 4.8 (используется метод 4.1.3)
         3. Добавить – добавляет получателя в заявку-черновик, (вызывается метод 4.1.2), только аутентифицированные пользователи.
         4. Корзина – перенаправляет на страницу 4.9, только аутентифицированные пользователи.
   8. Одна услуга
      1. Доступна всем
      2. Отображается подробная информация о выбранной услуге, (вызывается метод 4.1.3)
   9. Одна заявка
      1. Доступно только аутентифицированным посетителям
      2. Отображает текущий заказ-черновик пользователя, (метод 4.1.9)
         1. Список выбранных услуг
         2. Поле комментария к услуге
      3. Действия, доступны только в случае, если статус «черновик»
         1. Удалить услугу – удалят услугу из отправки, (вызывается метод 4.1.11)
         2. Сохранить – сохраняет комментарий для услуги, (вызывается метод 4.1.15)
         3. Удалить заявку – удаляет заявку-черновик, (вызывается метод 4.1.14)
         4. Заполнить – вносит данные заявки (вызывается метод 4.1.10)
   10. Заявки
       1. Доступно аутентифицированному посетителю
       2. Отображается список заявок
          1. Только заявки, созданные данным пользователем, если открыл пользователь
          2. Все оформленные заявки, если открыл администратор
       3. Действия
          1. Фильтрация – фильтрует заявки по дате создания и статусу, вызывается (метод 4.1.8)
          2. Сформировать – формирует заявку, выполняется метод 4.1.13, доступно только пользователю
          3. Отклонить – отклоняет заявку, вызывается метод 4.1.13, доступно только администратору
          4. Посмотреть подробную информацию об услуге – перенаправляет на страницу 4.9 (вызывается метод 4.1.9)
   11. Список услуг таблицей
       1. Доступно только администратору
       2. Отображаются все существующие в системе услуги (вызывается метод 4.1.1)
       3. Действия
          1. Удалить – удаляет услугу (вызывается метод 4.1.5)
          2. Редактирование/создание – переход на страницу 4.12
   12. Редактирование/создание услуг
       1. Доступно только администратору
       2. Отображается информация об услуге (вызывается метод 4.1.3)
          1. Название
          2. Описание
          3. Статус
          4. Цена
          5. Время выполнения
          6. Изображение
       3. Действия
          1. Сохранить – сохранение изменений в существующей услуге (вызывается метод 4.1.4)
          2. Добавить – добавление новой услуги (вызывается метод 4.1.2)
   13. 404
       1. Доступно всем
       2. Отображается в случае отсутствия ресурса
       3. Действия.
          1. Вернуться на главную – переход на страницу 4.6.
   14. 403
       1. Доступно всем
       2. Отображается в случае отсутствия прав на использование ресурса
       3. Действия.
          1. Вернуться на главную – переход на страницу 4.6.
3. Требования к составу и параметрам технических средств
   1. Сервер
      1. Процессор AMD Ryzen 5
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
   2. Клиент
      1. Процессор AMD Ryzen 5
      2. Оперативная память 16 Гб
      3. Свободное пространство на диске 10 Гб
4. Требования к информационной и программной совместимости
   1. Сервер
      1. ОС Ubuntu (24.10 и выше)
      2. Minio (RELEASE.2023-12-14T18-51-57Z)
      3. Redis (7.2)
      4. Node JS (1.23)
      5. PostgreSQL (17)
      6. Python 3.12
      7. Доступ к интернету
   2. Клиент
      1. 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 2 – HTTP методы разрабатываемого веб-сервиса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Тип** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| 4.1.1 | GET | / services/ | Возвращает список активных услуг Доступно всем | name=string | {  "draft\_request\_id": int, "services\_in\_draft\_request": int,  "services":  [  {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  }  ]  } |
| 4.1.2 | POST | /services/create/ | Создаёт новую услугу | {  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } | {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } |
| 4.1.3 | GET | /services/{pk}/ | Возвращает информацию об услуге по ID | pk: int | {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } |
| 4.1.4 | PUT | /services/{pk}/update/ | Обновляет данные услуги, доступно менеджеру | pk: int  {  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } | {  "id": int,  "name": string,  "description": string,  "price": float,  "status": string,  "image\_url": string  } |
| 4.1.5 | DELETE | /services/{pk}/  delete/ | Удаляет услугу | pk: int | {  "message": "Service deleted"  } |
| 4.1.6 | POST | /services/{pk}/  add-to-draft/ | Добавляет услугу в черновую заявку | pk: int | {  "request\_id": int,  "service\_id": int,  "quantity": int  } |
| 4.1.7 | POST | /services/  {service\_id}/  update-image/ | Обновляет изображение для услуги | pk: int  {  "service\_id": int,  "image": Image } | {  "detail": string,  "image\_url": string  } |
| 4.1.8 | GET | /requests/ | Возвращает список заявок | start\_date (опционально): string, end\_date (опционально): string, status (опционально): string | [  {  "id": int,  "status": string,  "creation\_date": datetime,  "total\_cost": float,  "client": string,  "services":  [  {  "id": int,  "name": string,  "price": float,  "quantity": int  }  ]  }  ] |
| 4.1.9 | GET | /requests/{pk}/ | Возвращает информацию о заявке по ID | pk: int | {  "id": int,  "status": string,  "creation\_date": datetime,  "completion\_date": datetime,  "total\_cost": float,  "client": string,  "services":  [  {  "id": int,  "name": string,  "price": float,  "quantity": int  }  ]  } |
| 4.1.10 | PUT | /requests/{pk}/  update/ | Обновляет заявку | pk: int  {  "status": string,  "total\_cost": float  } | {  "id": int,  "status": string,  "creation\_date": datetime,  "completion\_date": datetime,  "total\_cost": float,  "client": string,  "services":  [  {  "id": int,  "name": string,  "price": float,  "quantity": int  }  ]  } |
| 4.1.11 | DELETE | /requests/{pk}/  delete/ | Удаляет заявку | pk: int | {  "id": int,  "status": string,  "message": "Request successfully deleted."  } |
| 4.1.12 | PUT | /requests/{pk}/  form/ | Переводит черновую заявку в статус "Submitted" | pk: int | {  "message": "Request successfully submitted.", "status": string,  "submission\_date": datetime  } |
| 4.1.13 | PUT | /requests/{pk}/  complete-or-reject/ | Меняет статус заявки ("Completed" или "Rejected") | pk: int  {  status: string } | {  "message": "Request successfully completed/rejected.",  "status": string,  "completion\_date": datetime,  "total\_cost": float  } |
| 4.1.14 | DELETE | /request-items/  {request\_id}/  {service\_id}/  delete/ | Удаляет услугу из заявки | request\_id: int,  service\_id: int | {  "message": "Service {service\_id} removed from request {request\_id}"  } |
| 4.1.15 | PUT | /request-items/  {request\_id}/  {service\_id}/  update/ | Обновляет данные по М-М (поле comment). | request\_id: int,  service\_id: int,  {  "comment": string  } | {  "request\_id": int,  "service\_id": int,  "message": "Comment updated",  "comment": string  } |
| 4.1.16 | POST | /user / | Регистрирует пользователя, доступно всем | {  username: string,  password: string,  is\_staff: bool,  is\_superuser: bool } | {  status : string } |
| 4.1.17 | PUT | /users/update/ | Обновляет данные пользователя. Доступно пользователю. | {  "username": string,  "password": string,  "email": string  } | {  "message": string,  "username": string,  "email": string } |
| 4.1.18 | POST | /users/login/ | Аутентификация пользователя. Доступно всем | {  "username": string,  "password": string  } | {  status: string  } |
| 4.1.19 | POST | /users/logout/ | Деавторизация, доступна дюбому аутентифицированному пользователю |  | {  status: string  } |