



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:**

***Заявки на подключение провайдера***

Студент ИУ5-53Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата) Нагапетян В. С.  
(И.О.Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата) А.И. Канев  
(И.О.Фамилия)

2024 г.

## **АННОТАЦИЯ**

Расчётно-пояснительная содержит 27 страниц. С приложениями объем составляет 42 страницу. Работа включает в себя 8 диаграмм и 16 изображений системы. В процессе выполнения было использовано 10 источников.

Объектом разработки является система подключения интернет-услуг провайдера. Данный программный модуль позволяет выбирать интернет-услуги и оставлять заявки на их подключение.

Цель работы заключается в создании набора программного обеспечения и сопутствующих веб сервисов для сервиса интернет-услуг.

В работе была разработана архитектура веб сервиса, рассчитанного на подключение интернет-услуг, разработан интерфейс для общения с данным веб сервисом, разработан и развернут веб сервер, нативное приложение и прогрессивное веб приложение, способные общаться с данным веб сервисом.

Пояснительная записка содержит 2 приложения.

## СОДЕРЖАНИЕ

АННОТАЦИЯ.....	1
СОДЕРЖАНИЕ .....	2
ВВЕДЕНИЕ.....	3
1 ПРЕДМЕТНАЯ ОБЛАСТЬ.....	5
2 АРХИТЕКТУРА .....	10
3 АЛГОРИТМЫ.....	14
4 ОПИСАНИЕ ИНТЕРФЕЙСА .....	16
ЗАКЛЮЧЕНИЕ .....	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	25
ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ .....	26
1 Введение.....	27
2 Назначение разработки.....	27
3 Стадии и этапы разработки .....	27
4 Требования к функциональным характеристикам .....	28
5 Требования к составу и параметрам технических средств.....	28
6 Требования к составу и параметрам технических средств.....	33
ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ .....	0

## ВВЕДЕНИЕ

Интернет сегодня является основой современной информационной среды, объединяя миллиарды людей по всему миру. По данным Statista, в октябре 2024 года количество пользователей интернета составило 5,52 миллиарда человек, что эквивалентно 67,5% мирового населения.

С развитием цифровизации и доступности интернет-услуг увеличивается потребность в качественных сервисах, способных обеспечивать стабильное подключение, высокую скорость и широкий спектр возможностей для пользователей. Именно в этих условиях сервисы интернет-провайдеров становятся важным звеном, соединяющим людей с цифровым миром и создающим основу для их повседневной жизни и работы.

Разработанная система предназначена для организации удобного и эффективного процесса подачи и управления заявками на подключение интернет-услуг. Он позволяет пользователям интернет-услуг легко создавать заявки, добавляя в них необходимые услуги из каталога, и отслеживать статус обработки своей заявки в реальном времени. Менеджеры по интернет-услугам, в свою очередь, могут управлять доступными услугами, редактировать их характеристики и следить за активными заявками пользователей интернет-услуг, принимая решения о завершении или отклонении заявок, что обеспечивает оперативное взаимодействие между клиентом и провайдером.

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 базовый дизайн на основе [moskva.beeline.ru](https://moskva.beeline.ru)
2. Создать базу данных для хранения информации происшествий и обращениях
3. Создать веб-сервис на базе Java+Spring Boot для использования его в SPA

4. Добавить авторизацию и хранение JWT-токенов в Redis
5. Разработать базовый SPA на React для гостя
6. Внедрить адаптивность, разработать Tauri приложение
7. Интерфейс пользователя интернет-услуг в React с менеджером состояний Redux Toolkit
8. Реализовать интерфейс менеджера по интернет-услугам в React
9. Разработать нативное приложение
10. Развернуть веб-приложение в GitHub Pages
11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм
12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта

В результате работы были реализованы все пункты кроме 8.

# 1 ПРЕДМЕТНАЯ ОБЛАСТЬ

В современном интернете задержка сети (delay) остается значительной проблемой, особенно для приложений, где важна скорость реакции, таких как онлайн-игры, поисковые системы и финансовые транзакции. В своей работе «cISP: A Speed-of-Light Internet Service Provider» Дебопам Бхаттачерджи (Deboram Bhattacharjee), Сангита Абду Джьоти (Sangeetha Abdu Jyothi) и соавторы предлагают инновационную концепцию интернет-провайдера (ISP – internet service provider), способного обеспечить передачу данных с минимальной задержкой, близкой к скорости света [1].

Ключевой проблемой, которую решают авторы, является одновременное обеспечение низкой задержки и масштабируемой пропускной способности (bandwidth). При этом учитываются многочисленные практические факторы, от доступности микроволновых башен связи до очередей пакетов.

Для достижения этих целей авторы предлагают использовать комбинацию таких технологий, как оптика свободного пространства, микроволны, миллиметровые волны и существующей оптоволоконной (optical fiber). Микроволновая связь обеспечивает прямые пути передачи данных между башнями связи, в то время как оптоволоконные каналы используются для обеспечения высокой пропускной способности там, где это необходимо.

Согласно исследованию [1], применение представленной технологии на территории США и Европы позволила бы достичь средних задержек в пределах 5% от теоретического минимума, который возможен при прямой передаче данных со скоростью света. В приложении к онлайн-играм для толстых клиентов удалось снизить задержку в 3–4 раза, а для тонких клиентов, использующих больше трафика (traffic), предлагаемый метод предсказательной передачи данных позволил существенно уменьшить «время кадра». Кроме того, в тестах веб-серфинга использование cISP обеспечило сокращение времени загрузки страниц на 31% по сравнению с традиционными сетями, а для небольших объектов размером до 1460 байт задержка уменьшилась на 59%. Например, для

Amazon ускорение загрузки страниц на 100 мс может увеличить годовую прибыль на \$78.7 – \$551 млн [1].

Таким образом, предложенная концепция cISP не только подтверждает свою техническую жизнеспособность, но и демонстрирует экономическую эффективность.

Разработанная система, представляющая сервис провайдера интернет-услуг (ISP), имеет каталог доступных интернет-услуг (internet services). Пользователи (users) могут подключить эти услуги, оставив заявку на подключение (connection request).

Пользователь интернет-услуг (user) сервиса интернет-услуг (ISP), может выбрать интернет-услугу для подключения, указать ФИО/организацию (customer), телефон для связи (phone number) и отправить заявку на рассмотрение менеджера по интернет-услугам. Каждая заявка в начале создается как черновик, в который можно добавлять и из которого можно удалять интернет-услуги. Одна заявка может содержать несколько услуг; тогда, в случае согласования заявки, будет оформлен заказ на подключение всех интернет-услуг из списка. Для каждой интернет-услуги в заявке можно указать количество (amount), которое по умолчанию равно единице.

Пользователь может посмотреть список всех созданных им заявок. В случае, если заявку составлена корректно, менеджер утверждает её. В противном случае – отклоняет. Менеджер (manager) может только либо согласовать, либо отклонить заявку.

Менеджеры сервиса интернет-услуг могут редактировать список интернет-услуг, доступных для подключения: добавлять новые интернет-услуги, удалять устаревшие, а также редактировать информацию о каждой отдельно взятой услуге. Менеджеры имеют возможность просматривать все заявки, созданные пользователями интернет-услуг, а также фильтровать их по заказчику, дате формирования (formation datetime) и статус (status). После того, как заявка была согласована или отклонена, менеджер не может изменять её статус.

Функции пользователей с различными ролями описаны на диаграмме прецедентов (рисунок 1).

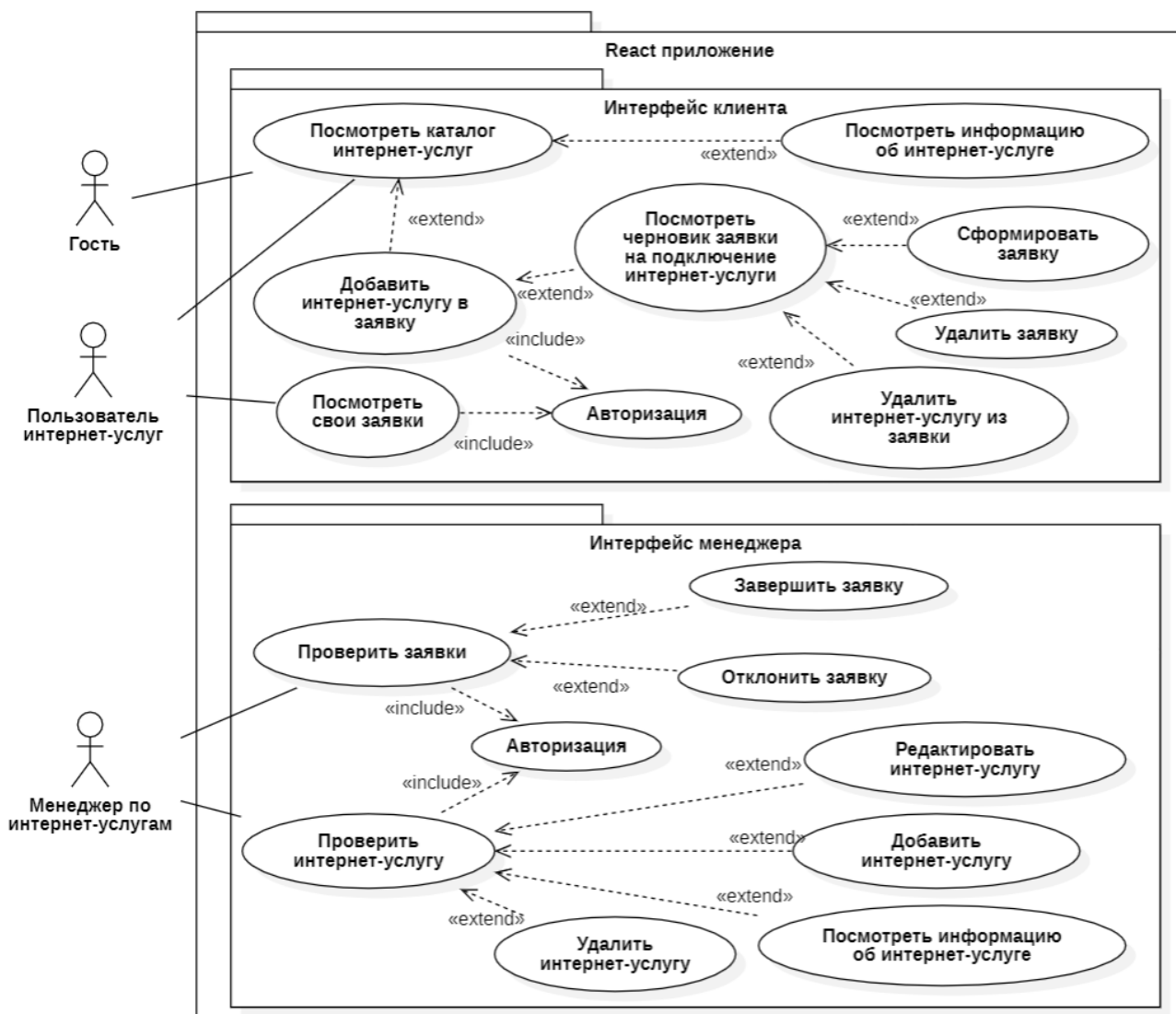


Рисунок 1 – Диаграмма прецедентов

Гостям доступен просмотр списка доступных для подключения интернет-услуг. Гости, прошедшие этап регистрации, являются клиентами. Клиенты могут добавлять интернет-услуги в заявку на подключение, формировать заявку из черновика, а также просматривать список своих заявок.

Клиент выбирает интернет-услугу, которую хочет подключить, затем на основе выбранного списка интернет-услуг формирует заявку. После чего менеджер может просмотреть её, а также согласовать или отклонить. В случае некорректного составления заявки клиент имеет возможность удалить заявку.



Возможные состояния статуса заявки отображены на диаграмме состояний (рисунок 2).



Рисунок 2 – Диаграмма состояний заявок на подключение

Заявки обрабатываются менеджерами компании. В результате обработки заявки её либо одобряют, либо отклоняют. Менеджеру доступны следующие операции для работы с интернет-услугами: просмотр всех услуг, редактирование, создание и удаление услуг, а также просмотр списка всех услуг табличном виде. Процесс оформления заявки отражен на диаграмме деятельности (рисунок 3).

В начале взаимодействия с системой оформления заявок на подключение интернет-услуг клиент запрашивает список доступных для подключения интернет-услуг. При необходимости клиент может добавить конкретную услугу в текущую черновую заявку. После чего может продолжить выбор интернет-услуг из каталога или перейти к оформлению заявки. В случае, если клиент выбрал оформление заявки на подключение интернет-услуг, он может указать количество для всех услуг в списке (по умолчанию 1). После чего указать ФИО заказчика/организацию и номер для связи. По нажатии кнопки «Сформировать заявку» заявка формируется и становится доступна в интерфейсе менеджера.

Менеджер имеет возможность просматривать заявки пользователей, а также принимать решение об одобрении заявки, если она может быть выполнена компанией или отклонении заявки, если она не может быть выполнена компанией.

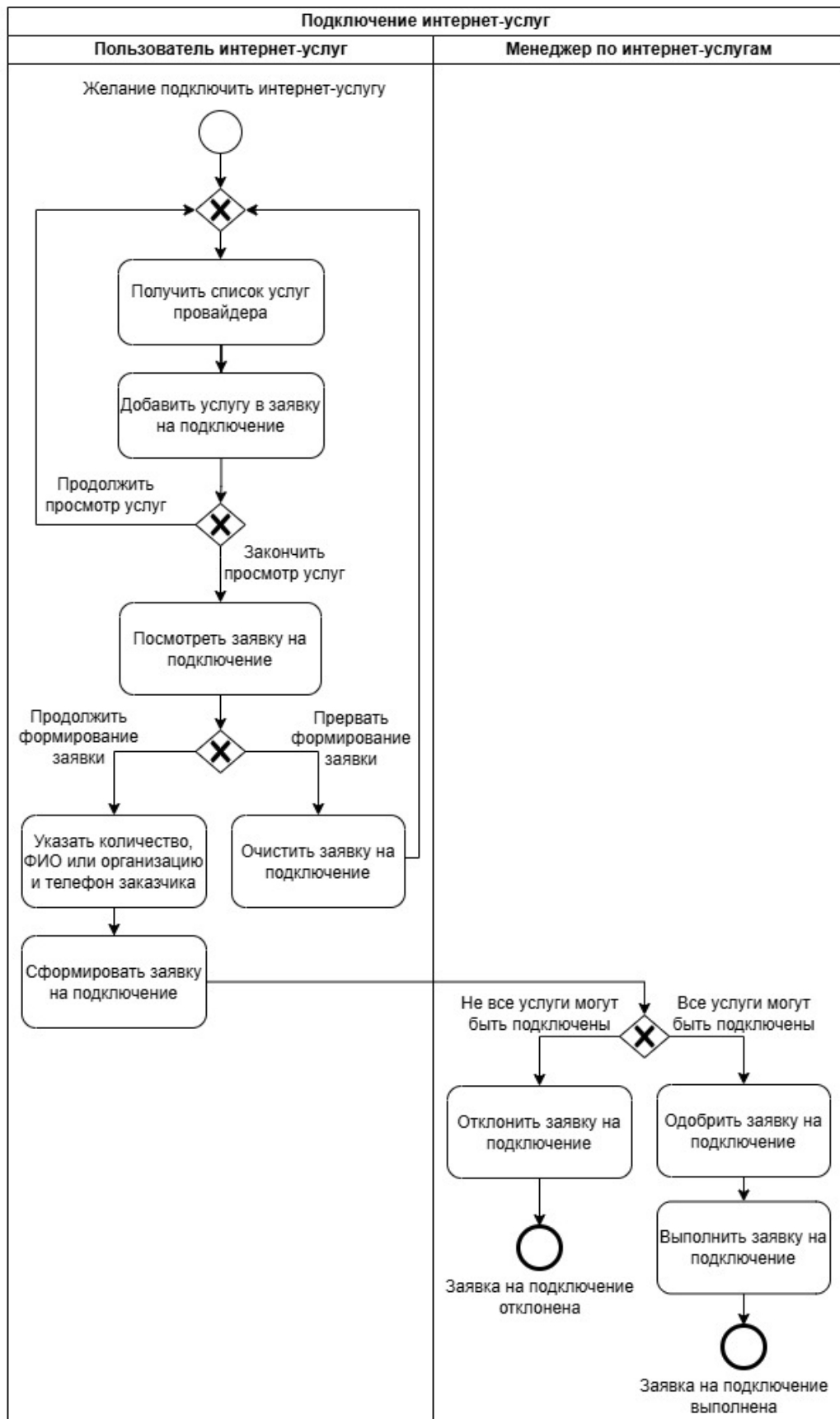


Рисунок 3 – Диаграмма взаимодействия в BPMN2.0

## 2 АРХИТЕКТУРА

Архитектура системы отображена на диаграмме развертывания (рисунок 4).

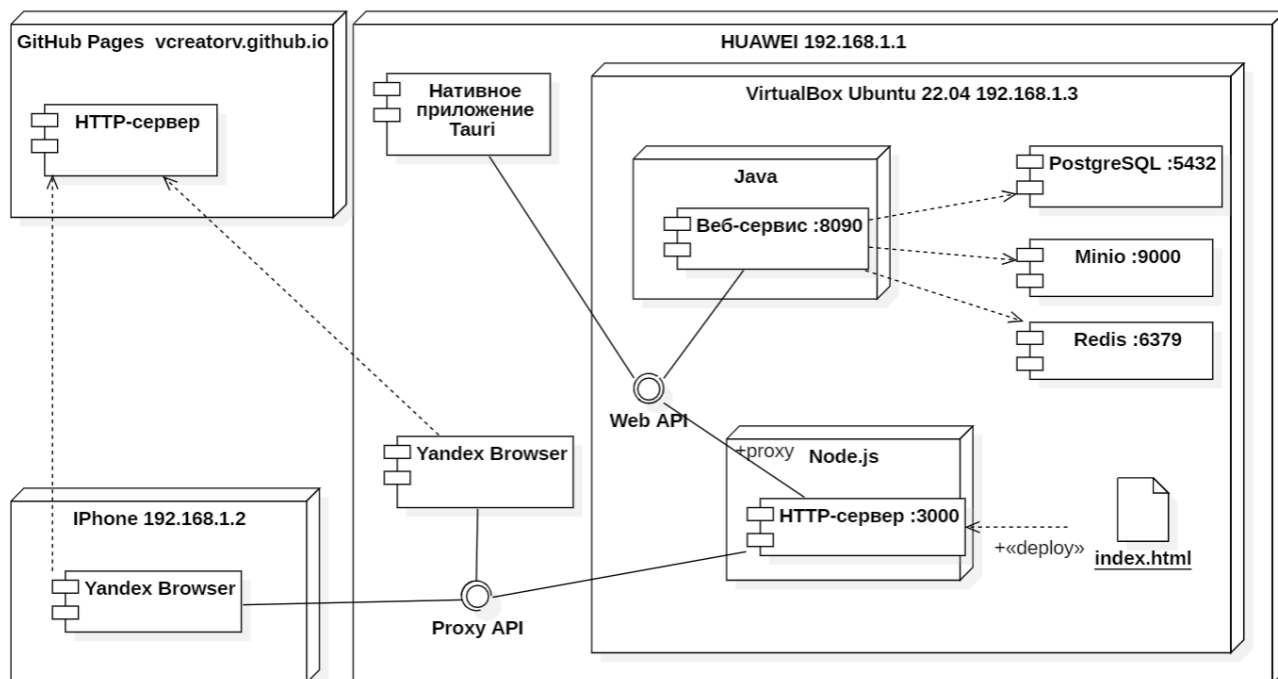


Рисунок 4 – Диаграмма развертывания

Веб-сервис, реализованный на фреймворке Spring [2] с использованием Spring Boot [3] связан с серверами Minio [4], Redis [5] и СУБД PostgreSQL [6]. В Redis хранятся jwt-токены пользователей, которые вышли из аккаунта; по истечении фиксированного времени - удаляются.

Язык программирования Java [7] был выбран благодаря его надежности, масштабируемости и широкому применению в разработке корпоративных решений. Использование Spring Boot обусловлено его мощной экосистемой, упрощающей процесс разработки и благодаря тому, что этот фреймворк является стандартом индустрии для создания масштабируемых веб-приложений. Веб-сервис на Spring Boot является общим для веб-сервера и десктопного приложения Tauri [8].

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

Структура данных отражена на ER диаграмме (рисунок 5). Модель услуг провайдера представляет собой набор полей, отражающих свойства интернет-услуг. Данные об интернет-услугах хранятся в таблице `provider_dutes`. Для хранения в одной заявке на подключение нескольких интернет-услуг используется промежуточная таблица связи `duties_requests`, которая реализует связь М-М. Таблица `connection_requests` представляет собой список заявок на подключение интернет-услуг. Данные о пользователях система – клиентах и менеджерах – хранятся в таблице `users`.

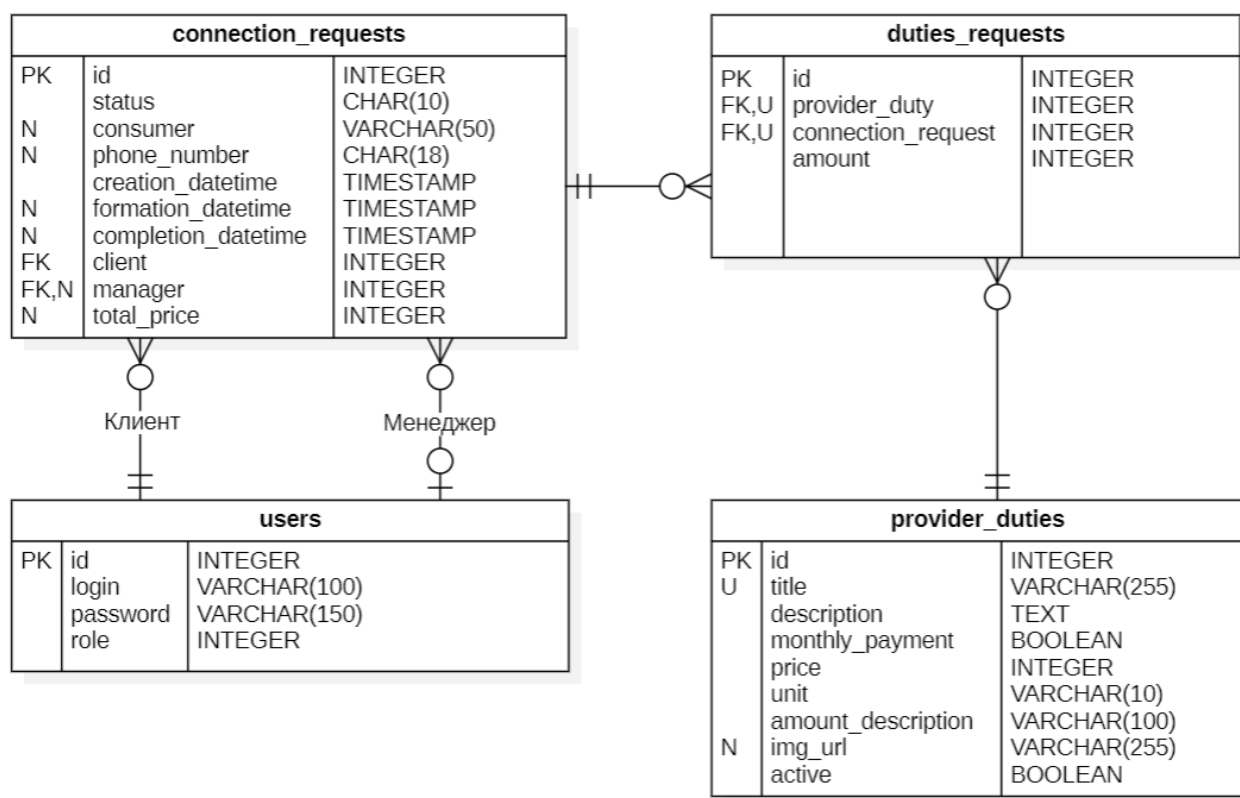


Рисунок 5 – ER-диаграмма

Веб-сервер реализован с использованием языка TypeScript и фреймворка React [9]. 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.

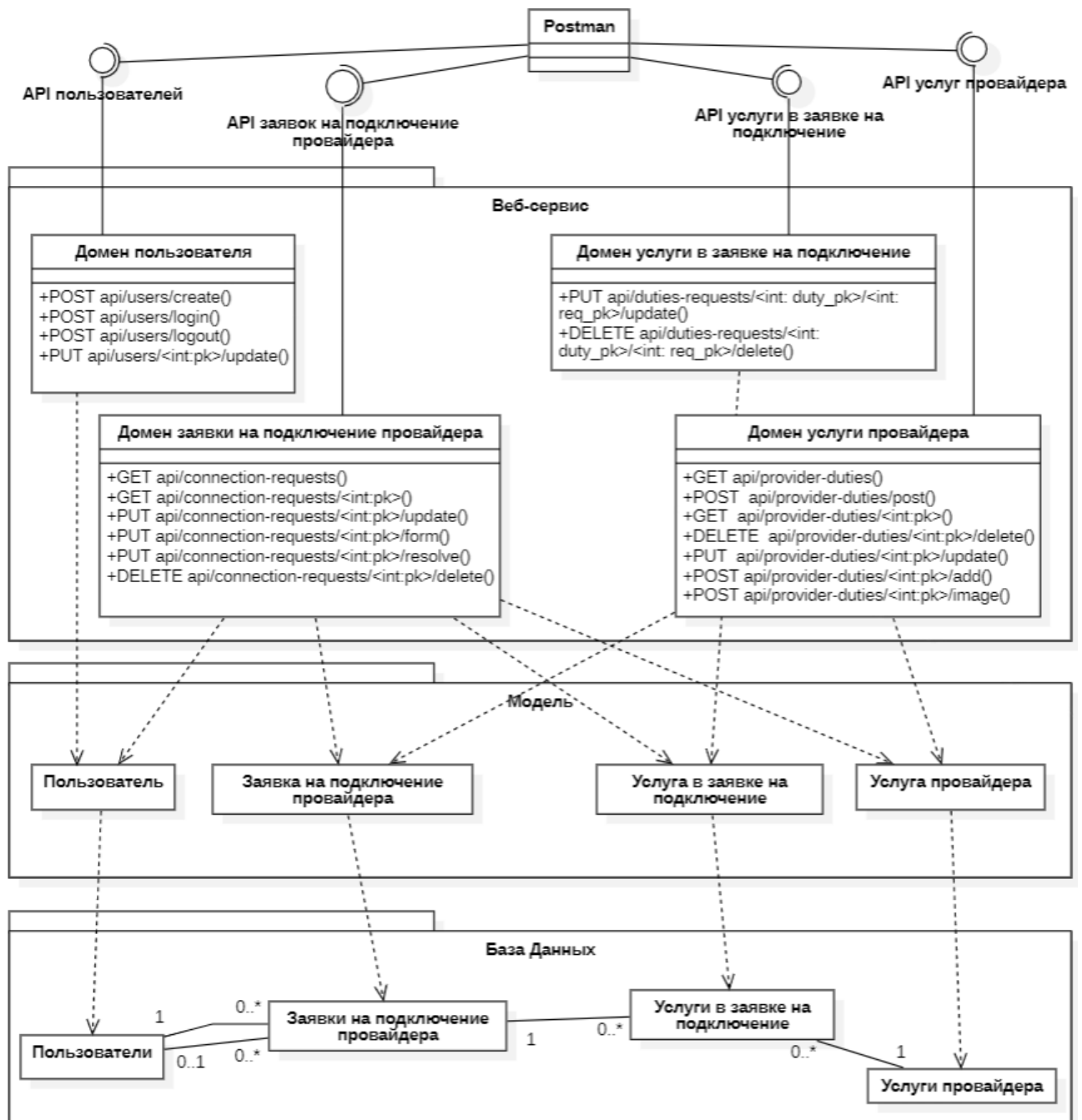


Рисунок 6 – Диаграмма классов бэкенда

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.

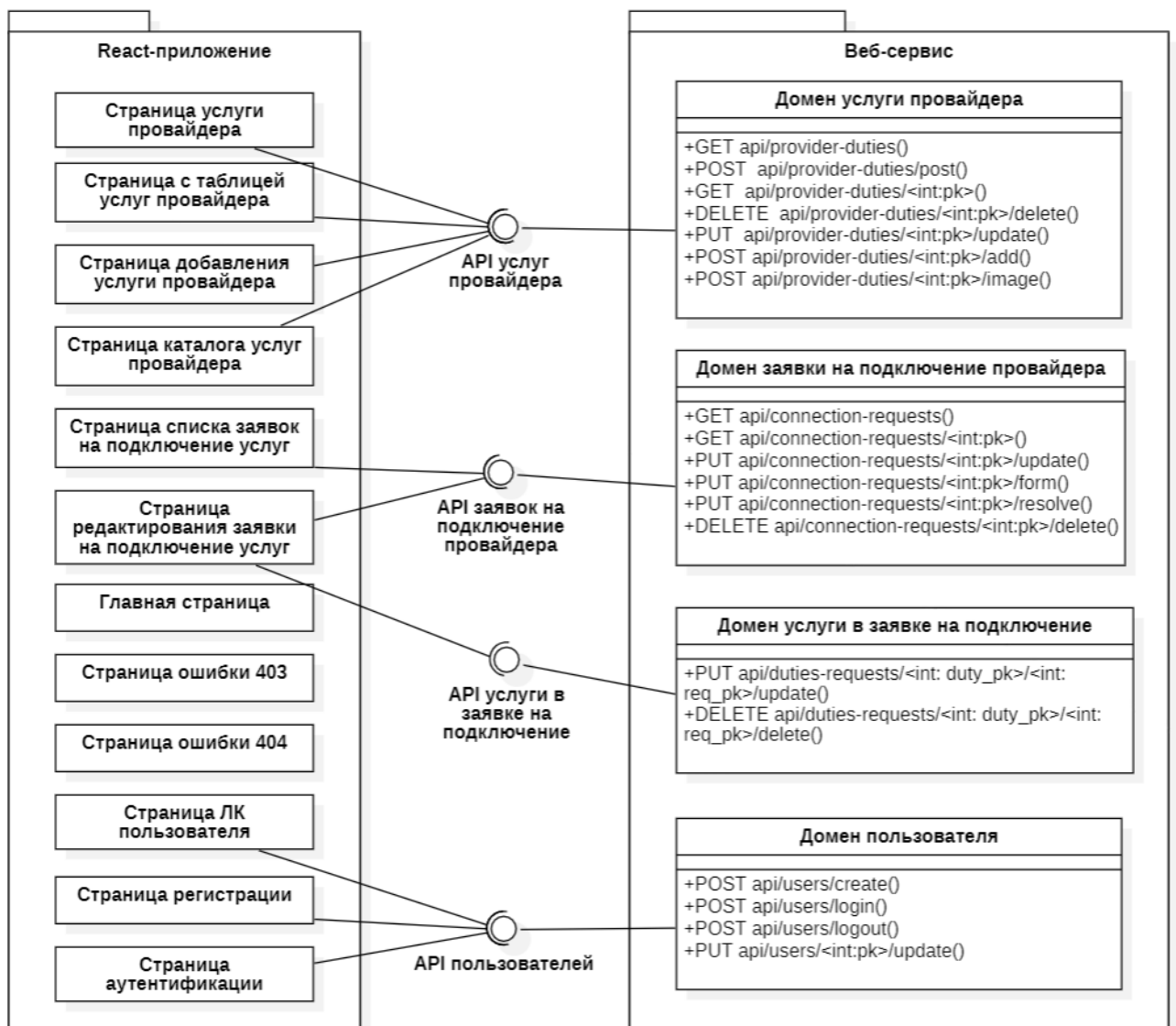


Рисунок 7 – Диаграмма классов фронтенда

### 3 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: услуги, заявки, пользователи. Методы следуют правилам REST API.

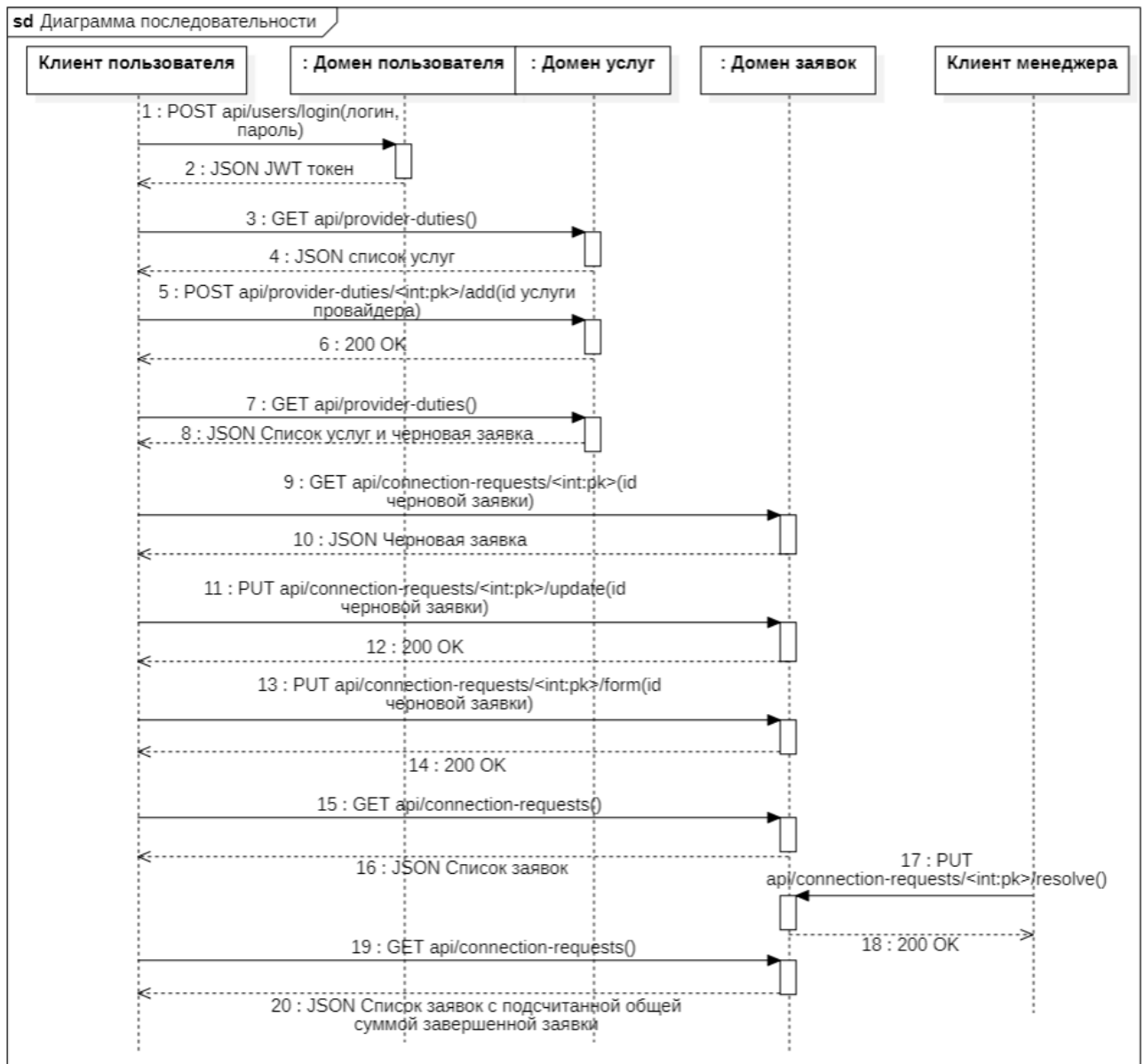


Рисунок 8 – Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. При помощи графического интерфейса пользователь отправляет запрос, передавая в нем логин и пароль. Если аккаунт с указанными данными существует в базе, на

клиент возвращается информация о пользователе и устанавливается JWT-токен в local storage при ответном запросе. Если же аккаунта с введенным логином не существует, или пароль введен неверно, на клиент возвращается ошибка. В таком случае пользователю надо пройти регистрацию, либо ввести верный пароль. На этом же этапе происходит проверка: является ли пользователь клиентом или менеджером. Графический интерфейс пользователя запрашивает у веб-сервиса список интернет-услуг, которые возвращаются в JSON формате. Клиент выбирает интернет-услугу, которую хочет добавить в заявку на подключение, и, нажимая на кнопку «добавить» в графическом интерфейсе, отправляет запрос на добавление интернет-услуги в свою черновую заявку. Этот процесс может продолжаться несколько раз.

Когда пользователь определится со списком необходимых для подключения интернет-услуг, а также указал их количество, ФИО заказчика/организацию и номер для связи, он нажимает кнопку «сформировать» в графическом интерфейсе. После чего приложение отправляет на веб-сервис запрос на формирование заявки. Клиент может отслеживать статус сформированных заявок на соответствующей странице в графическом интерфейсе.

Процесс рассмотрения заявок менеджерам так же происходит при помощи графического интерфейса. Менеджеры могут просматривать список всех созданных пользователями заявок; согласовывать или отклонять их при помощи соответствующих кнопок. Доступна фильтрация по логинам пользователей, а также статусам и дате формирования заявок. Менеджеры имеют возможность редактировать список интернет-услуг, доступных для подключения, а также изменять данные о каждой услуге. Для каждой из этих возможностей существует соответствующий метод, отправляемый на веб-сервис.



## 4 ОПИСАНИЕ ИНТЕРФЕЙСА

Главное меню приложения включает пункты, которые доступны в зависимости от роли пользователя (рисунки 9, 10, 11).

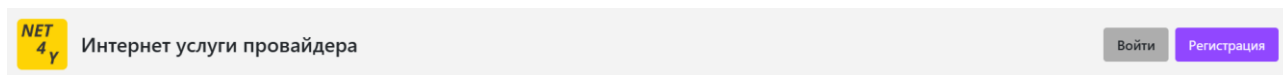


Рисунок 9 – Меню приложения (для гостя)

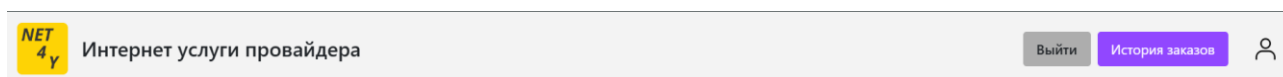


Рисунок 10 – Меню приложения (для клиента)

На странице с формой регистрации (рисунок 11) отображается форма, при помощи которой гость может создать новый аккаунт. После успешной регистрации открывается форма аутентификации.

The screenshot shows the registration form. At the top, there is a header with the logo 'NET 4 Y' and the text 'Интернет услуги провайдера'. On the right, there are buttons for 'Войти' (Login) and 'Регистрация' (Registration). The main content area is titled 'Регистрация' (Registration). It contains four input fields: 'Имя' (Name), 'Логин' (Login), 'Пароль' (Password), and 'Повторите пароль' (Repeat password). Below these fields is a blue button labeled 'Зарегистрироваться' (Register).

Рисунок 11 – Страница регистрации

На странице с формой входа (рисунок 12) отображается форма, через которую пользователь может войти в свой аккаунт. При успешном вводе данных аккаунта на клиент приходит JWT-токен и сохраняется в local storage.

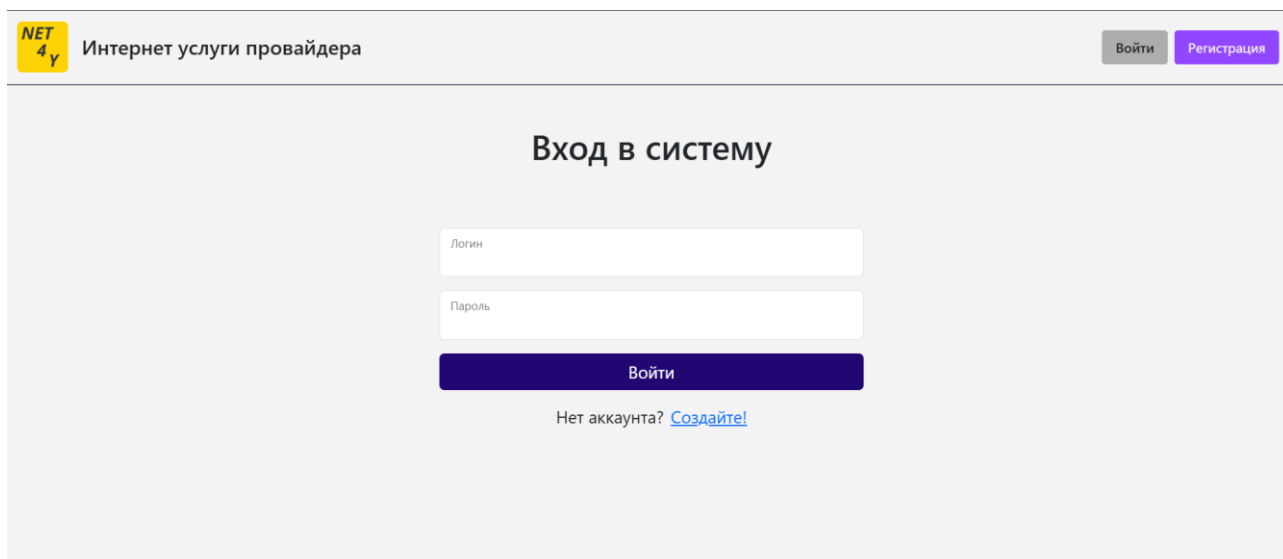


Рисунок 12 – Страница входа

После входа в аккаунт пользователь попадает на главную страницу (рисунок 13). На этой странице располагается описание разработанной системы.



Рисунок 13 – Главная страница

Страница каталога интернет-услуг (рисунок 14) содержит список услуг в виде карточек. При нажатии на кнопку «Подробнее» на любой карточке, открывается страница с подробным описанием выбранной интернет-услуги. Также с помощью нажатия на кнопку «Добавить» клиент может добавлять интернет-услуги в черновую заявку. Вверху страницы находится строка для поиска интернет-услуг по названию.

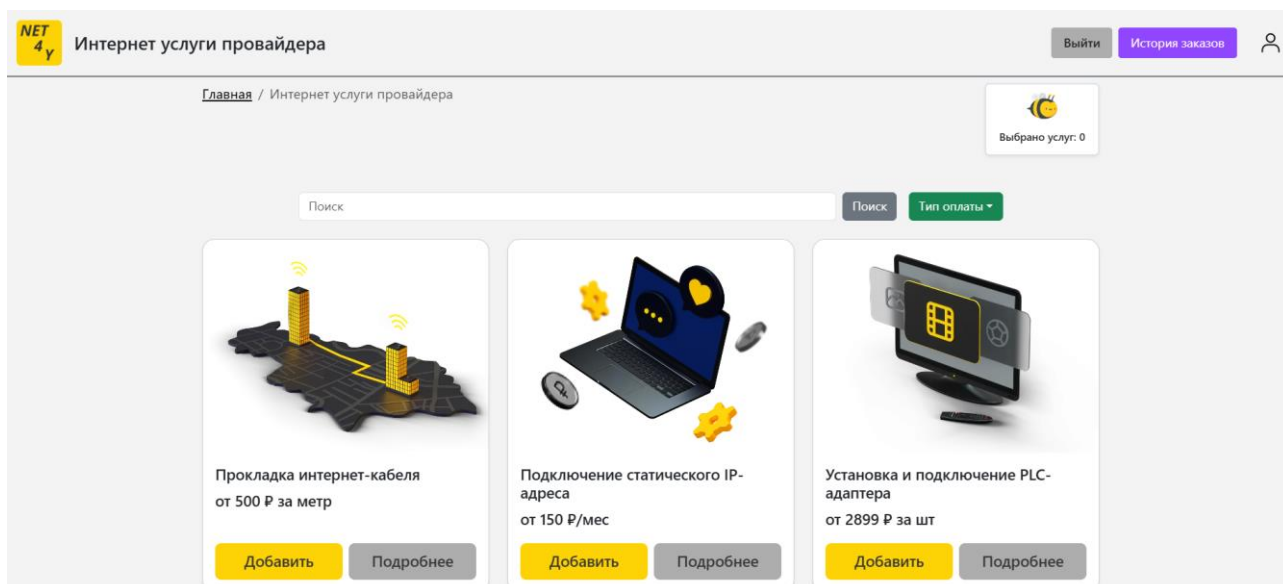


Рисунок 14 – Страница каталога интернет-услуг

На странице с подробным описанием интернет-услуги выводится основная информация о ней (рисунок 15).

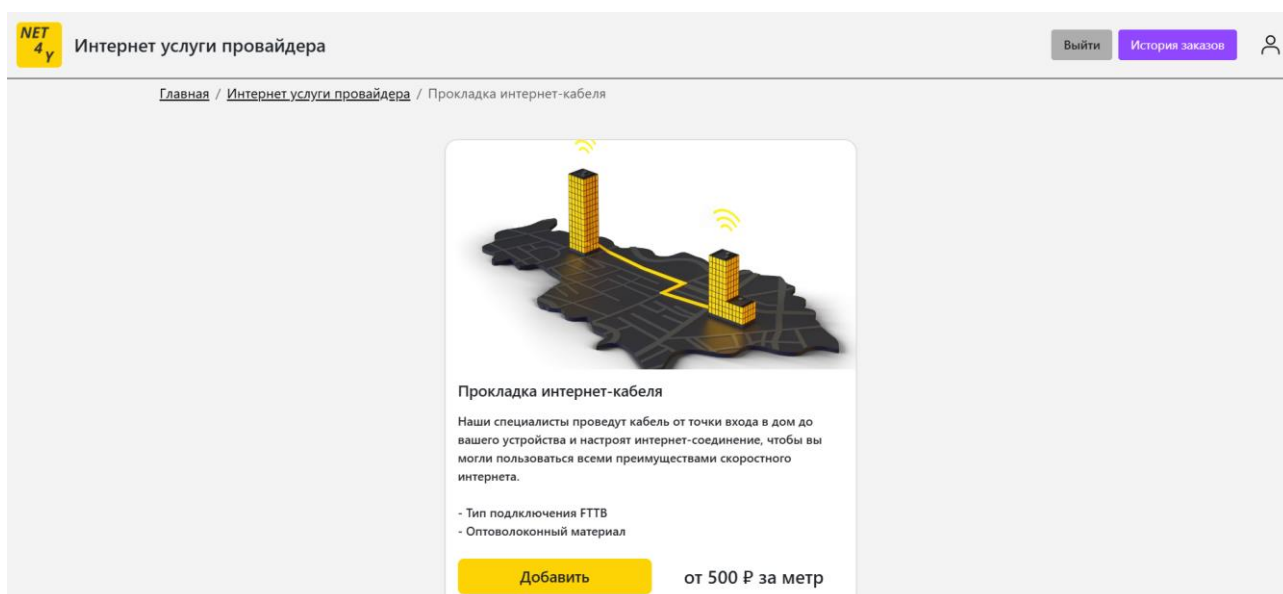


Рисунок 15 – Страница с описанием интернет-услуги

На странице одной заявки (рисунок 16) отображается текущая заявка пользователя в виде таблицы. У пользователя есть возможность удалить интернет-услугу из заявки, изменить количественный параметр услуги, а также отправить заявку на проверку или удалить ее. Редактирование оформленных заявок недоступно.

**Текущая заявка**

Организация/ФИО: \_\_\_\_\_ Номер для связи: \_\_\_\_\_

**Прокладка интернет-кабеля**  
 количество метр:  500 Р

**Подключение статического IP-адреса**  
 количество шт:  150 Р/мес

**Итого:** 650 Р

Очистить заявку Оформить подключение

Рисунок 16 – Страница заявки на подключение интернет-услуг

На странице списка заявок (рисунок 17) клиенты могут просматривать созданные ими заявки в виде таблицы. На этой странице можно посмотреть подробную информацию о заявке, нажав на её номер.

**Заявки**

Статус:  Дата оформления с:  Дата оформления по:  Применить

№	Статус	Создано	Оформление	Завершение	Стоимость	Заказчик
18	В обработке	26.11.2024	26.11.2024	---	---	Артемис Смирнов
20	В обработке	26.11.2024	26.11.2024	---	---	Артем
76	В обработке	01.12.2024	01.12.2024	---	---	Artem
77	В обработке	01.12.2024	01.12.2024	---	---	Artem

Рисунок 17 – Страница списка заявок на подключение интернет-услуг

Пользователь интернет-услуг может изменить свои данные (логин и пароль) на странице редактирования данных пользователя интернет-услуг (рисунок 18). Она открывается при нажатии на иконку пользователя в меню.

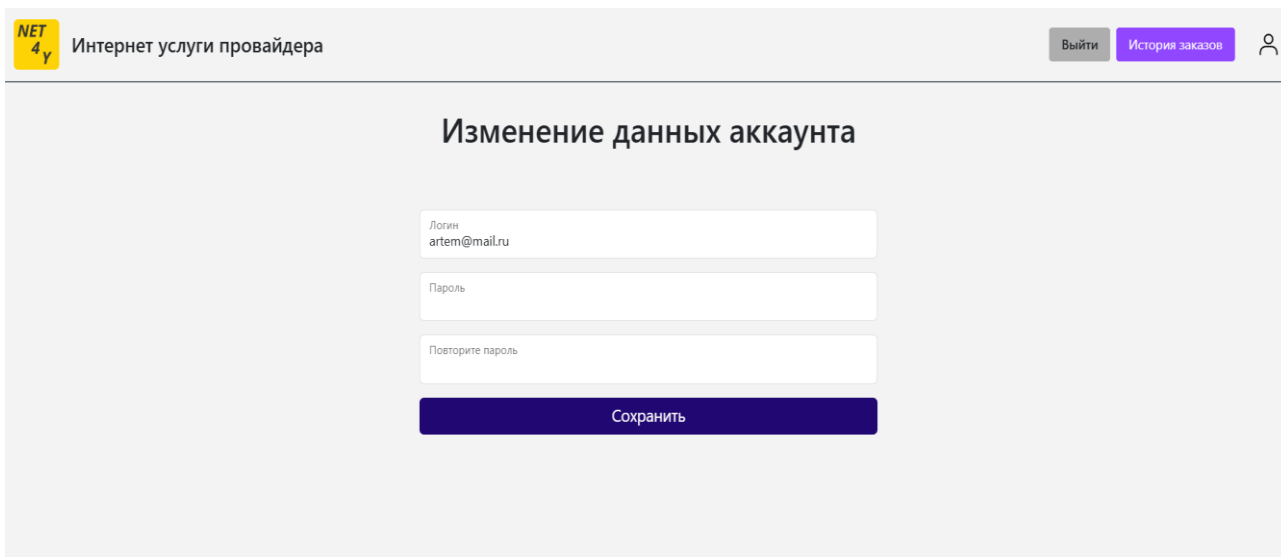


Рисунок 18 – Страница редактирования информации о пользователе интернет-услуг



Рисунок 19 – Главная страница в PWA

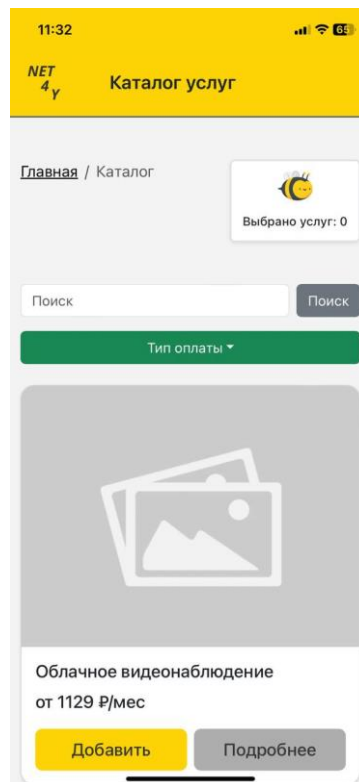


Рисунок 20 – Страница каталога интернет-услуг в PWA

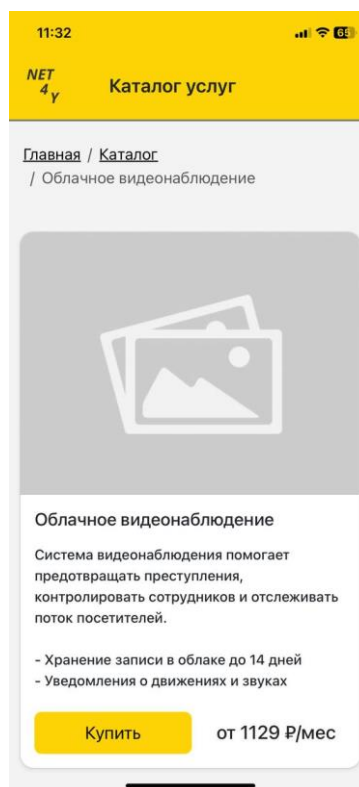


Рисунок 21 – Страница с описанием интернет-услуги в PWA

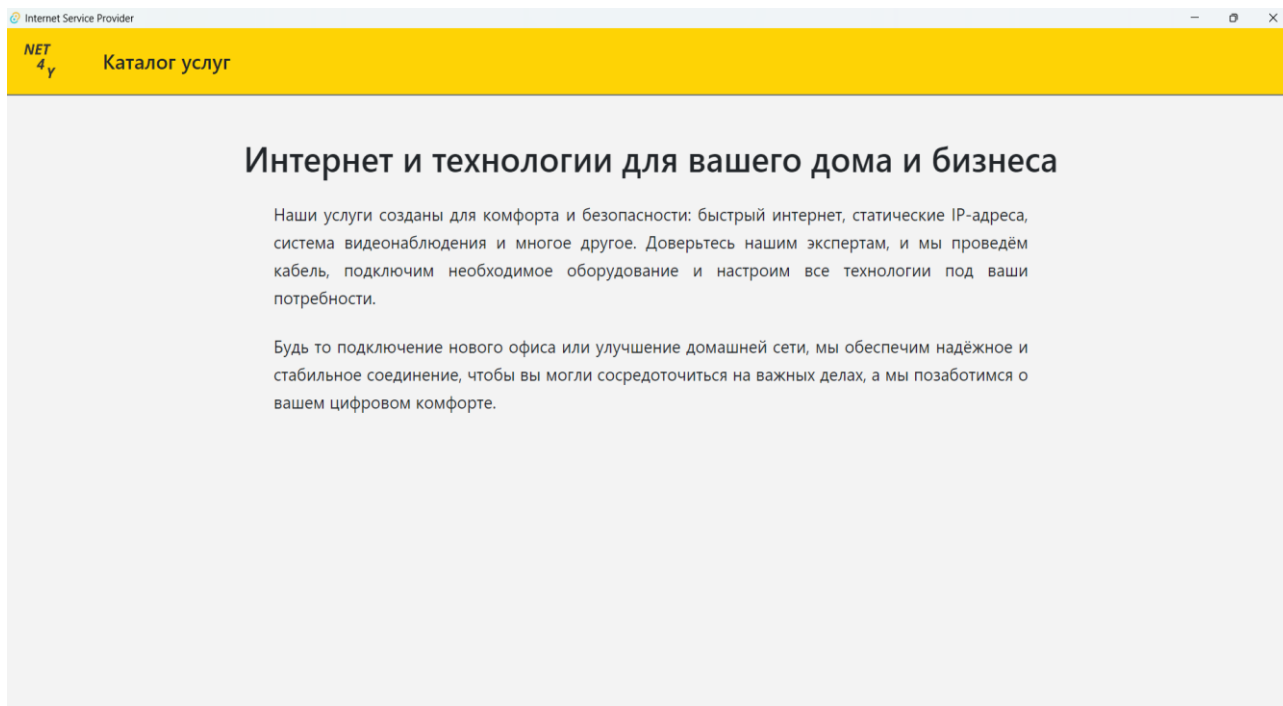


Рисунок 22 – Главная страница в Таугі

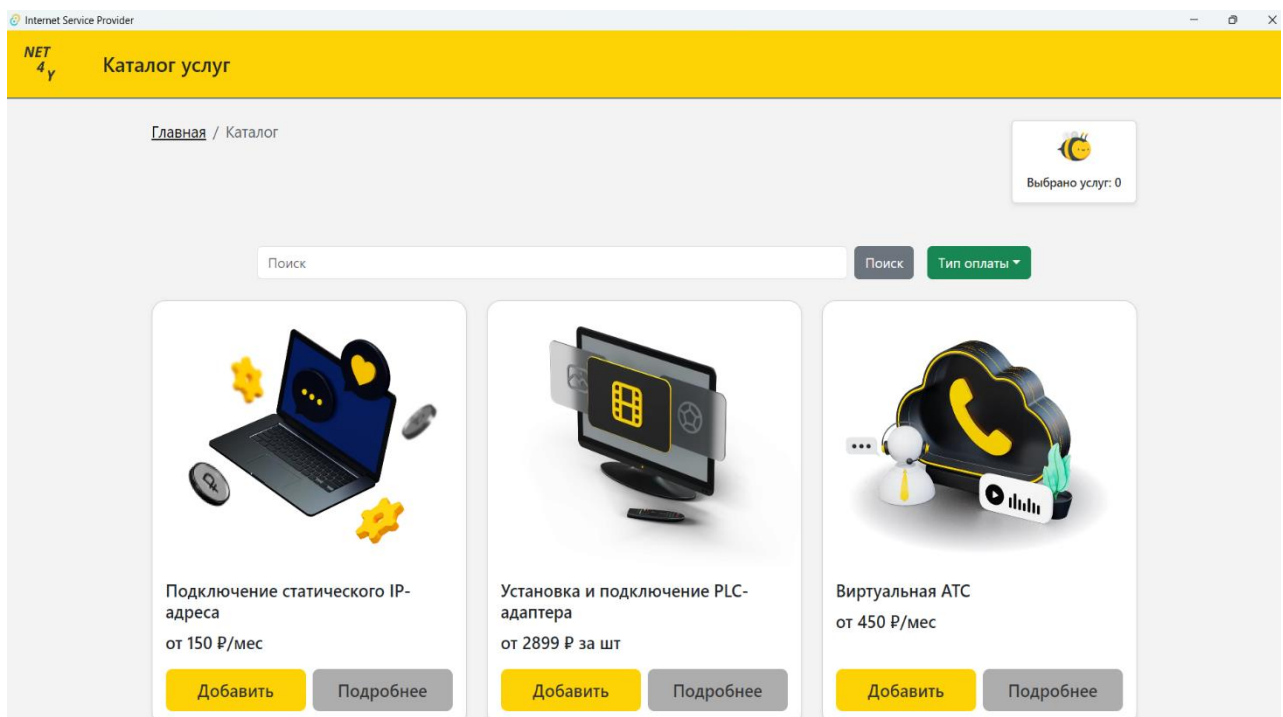


Рисунок 23 – Страница каталога интернет-услуг в Таугі

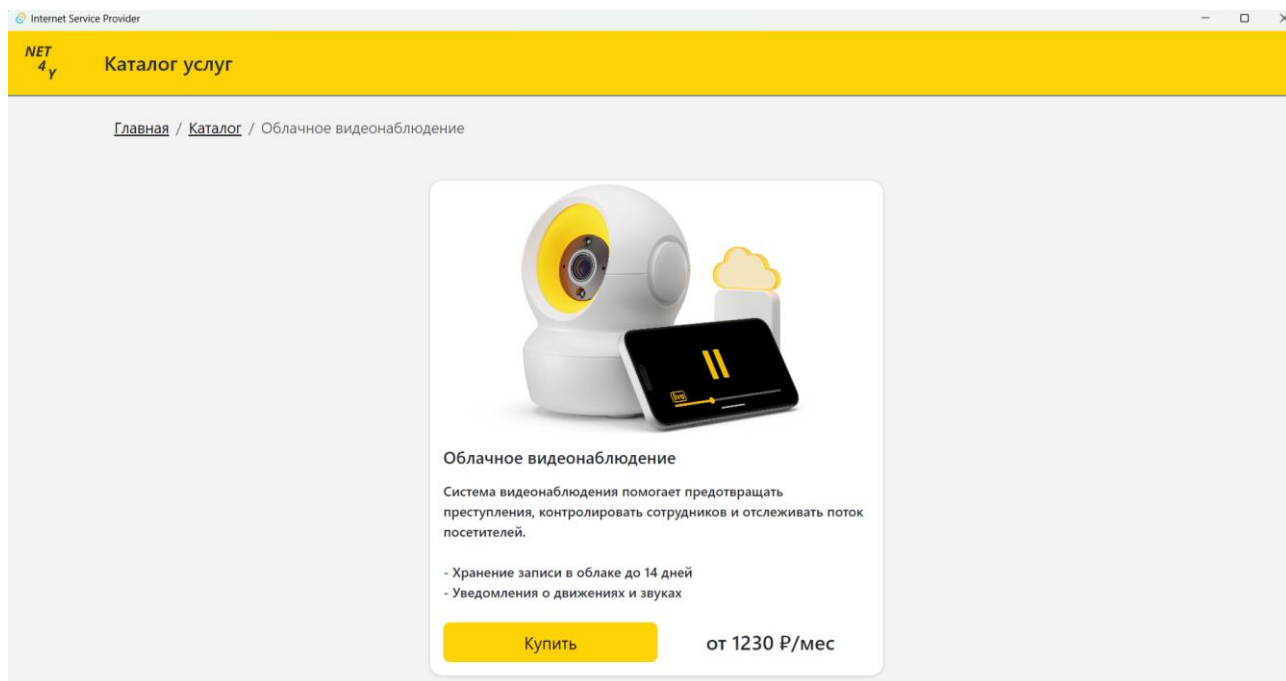


Рисунок 24 – Страница интернет-услуги в Таугі



## ЗАКЛЮЧЕНИЕ

В ходе работы были достигнуты следующие результаты:

1. Создан MVP и базовый дизайн приложения на основе [moskva.beeline.ru](https://moskva.beeline.ru)
2. Создана база данных для хранения информации об интернет-услугах и заявках на их подключение
3. Создан веб-сервис на Spring Boot
4. Реализована авторизация и хранение JWT в Redis
5. Разработан базовый SPA на React для гостя
6. Внедрена адаптивность, менеджер состояний Redux Toolkit, PWA, разработано Tauri приложение
7. Завершена разработка интерфейса пользователя интернет-услуг в React
8. Реализован интерфейс менеджера по интернет-услугам
9. Разработано десктопное приложение Tauri
10. Приложение развернуто при помощи сервиса GitHub Pages и доступно по ссылке: <https://vcreatorv.github.io/internet-service-provider-frontend/>
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформлен git-репозиторий на сервисе GitHub, содержащий исходный код проекта: <https://github.com/vcreatorv/internet-service-provider-backend>

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. cISP: A Speed-of-Light Internet Service Provider [Электронный ресурс] // Arxiv. URL: <https://arxiv.org/abs/1809.10897> (дата обращения: 20.11.2024);
2. Документация по Spring [Электронный ресурс] // Spring Framework. URL: <https://spring.io/projects/spring-framework> (дата обращения: 15.09.2024);
3. Документация по Spring Boot [Электронный ресурс] // Spring Boot. URL: <https://spring.io/projects/spring-boot> (дата обращения: 15.09.2024);
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 Java [Электронный ресурс] // Java. URL: <https://docs.oracle.com/en/java/javase/17/> (дата обращения: 13.09.2024);
8. Документация по Tauri [Электронный ресурс] // Tauri. URL: <https://v2.tauri.app/develop/> (дата обращения: 20.11.2024);
9. 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  
\_\_\_\_\_ В.И.Терехов  
" \_\_ " \_\_\_\_\_ 2024 г.

Согласовано  
Научный руководитель  
\_\_\_\_\_ А.И. Канев  
" \_\_ " \_\_\_\_\_ 2024 г.

## Заявки на подключение провайдера

Техническое задание  
(вид документа)

писчая бумага  
(вид носителя)

16  
(количество листов)

ИСПОЛНИТЕЛЬ:

\_\_\_\_\_ Нагапетян Валерий Сергеевич  
" \_\_ " \_\_\_\_\_ 2024 г.

Москва - 2024

## **1 Введение**

Реализовать систему для учета заявок на подключение провайдера, включающую в себя веб-сервис, веб-приложение, мобильное приложение и выделенный сервис обработки заказов.

## **2 Назначение разработки**

Система заявок на подключение провайдера предназначена для организации удобного и эффективного процесса подачи и управления заявками на подключение интернет-услуг. Она позволяет пользователям интернет-услуг легко создавать заявки, добавляя в них необходимые услуги из каталога, и отслеживать статус обработки своей заявки в реальном времени. Менеджеры по интернет-услугам, в свою очередь, могут управлять доступными услугами, редактировать их характеристики и следить за активными заявками пользователей интернет-услуг, принимая решения о завершении или отклонении заявок, что обеспечивает оперативное взаимодействие между клиентом и провайдером.

## **3 Стадии и этапы разработки**

- 3.1. Создать MVP и базовый дизайн на основе [moskva.beeline.ru](https://moskva.beeline.ru)
- 3.2. Создать базу данных для хранения информации происшествий и обращениях
- 3.3. Создать веб-сервис на базе Java+Spring Boot для использования его в SPA
- 3.4. Добавить авторизацию и хранение JWT-токенов в Redis
- 3.5. Разработать базовый SPA на React для гостя
- 3.6. Внедрить адаптивность, разработать Tauri приложение
- 3.7. Интерфейс пользователя интернет-услуг в React с менеджером состояний Redux Toolkit
- 3.8. Реализовать интерфейс менеджера по интернет-услугам в React
- 3.9. Разработать нативное приложение
- 3.10. Развернуть веб-приложение в GitHub Pages

3.11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм

3.12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта

## **4 Требования к функциональным характеристикам**

### **4.1. Методы HTTP**

4.1.1. POST Регистрация

4.1.2. POST Вход в аккаунт

4.1.3. POST Выход из аккаунта

4.1.4. PUT Изменение данных пользователя интернет-услуг

4.1.5. GET Получение списка заявок на подключение услуг

4.1.6. GET Получение одной заявки на подключение услуг

4.1.7. PUT Изменение данных о заявке на подключение услуг

4.1.8. PUT Формирование заявки на подключение услуг

4.1.9. PUT Разрешение заявки на подключение услуг (отклонение или завершение)

4.1.10. DELETE Удаление заявки на подключение услуг

4.1.11. GET Получение списка доступных услуг провайдера

4.1.12. GET Получение информации об услуге провайдера

4.1.13. POST Добавление информации об услуге провайдера

4.1.14. PUT Изменение информации об услуге провайдера

4.1.15. PUT Добавление услуги провайдера в заявку

4.1.16. POST Добавление логотипа услуги провайдера

4.1.17. DELETE Удаление услуги провайдера

4.1.18. PUT Изменение информации об услуге провайдера заявке

4.1.19. DELETE Удаление услуги провайдера из заявки

### **4.2. Меню**

4.2.1. Главная. Ссылка на страницу 4.3. Доступно всем пользователям

- 4.2.2. Каталог услуг по подключению. Ссылка на страницу 4.6. Доступно всем пользователям
- 4.2.3. Заявки. Ссылка на страницу 4.9. Доступно только авторизованным пользователям интернет-услуг
- 4.2.4. Таблица услуг провайдера. Ссылка на страницу 4.14. Доступно только менеджеру
- 4.2.5. Войти. Ссылка на страницу 4.4. Доступно только неавторизованному пользователю
- 4.2.6. Зарегистрироваться. Ссылка на страницу 4.5. Доступно только неавторизованному пользователю
- 4.2.7. Личный кабинет. Ссылка на страницу 4.11. Доступно только авторизованному пользователю интернет-услуг
- 4.3. Главная страница. На этой странице находится текст с приветствием
  - 4.3.1. Доступна всем пользователям.
  - 4.3.2. Действия
    - 4.3.2.1. Войти – переход на страницу 4.4. Только для неавторизованных пользователей
    - 4.3.2.2. Регистрация - переход на страницу 4.5. Только для неавторизованных пользователей
    - 4.3.2.3. Полный список услуг провайдера – переход на страницу 4.6. Доступно всем пользователям
    - 4.3.2.4. Список услуг на подключение в текущей заявке – переход на страницу 4.12. Только для авторизованных пользователей интернет-услуг
    - 4.3.2.5. Список заявок – переход на страницу 4.9. Только для авторизованных пользователей интернет-услуг
    - 4.3.2.6. Добавить услугу провайдера – переход на страницу 4.8. Только для менеджеров
    - 4.3.2.7. Таблица заявок на подключение услуг провайдера – переход на страницу 4.14. Только для менеджеров

4.3.2.8. Выйти – переход на страницу 4.3., выход из аккаунта (метод 4.1.3)

#### 4.4. Страница с формой авторизации

4.4.1. Доступна неавторизованным пользователям

4.4.2. Действия

4.4.2.1. Войти – метод 4.1.2. с указанными данными

4.4.2.2. Зарегистрироваться – переход на страницу 4.5

#### 4.5. Страница с формой регистрации

4.5.1. Доступна неавторизованным пользователям

4.5.1.1. Действия

4.5.1.2. Зарегистрироваться – метод 4.1.1. с указанными в форме данными

4.5.1.3. Войти – переход на страницу 4.4

#### 4.6. Страница с каталогом услуг, доступных для подключения. На этой странице располагается каталог доступных для подключения услуг и форма для поиска услуг по названию

4.6.1. Доступна всем пользователям

4.6.2. Выводится информация об услугах провайдера в виде карточку. (метод 4.1.11)

4.6.2.1. Логотип услуги провайдера

4.6.2.2. Название

4.6.2.3. Стоимость подключения

4.6.3. Действия

4.6.3.1. Отфильтровать услуги провайдера по названию (метод 4.1.11)

4.6.3.2. Добавить услугу провайдера в заявку на подключение (метод 4.1.16). Только для авторизованных пользователей

4.6.3.3. Подробнее – перенаправляет на страницу 4.7

#### 4.7. Страница с подробным описанием услуги провайдера

4.7.1. Доступно всем пользователям

4.7.2. Выводится подробная информация о конкретной услуге провайдера (метод 4.1.12)

4.7.2.1. Логотип услуги провайдера

4.7.2.2. Название

4.7.2.3. Стоимость подключения

4.7.2.4. Описание

4.7.3. Действия

4.7.3.1. Редактировать – переход на страницу 4.8. Доступно только менеджерам

4.8. Страница для добавления/редактирования услуг провайдера

4.8.1. Доступно только менеджерам

4.8.2. Выводится информация об услуге провайдера в построчном виде (метод 4.1.11)

4.8.3. Действия

4.8.3.1. Сохранить – сохранение изменений существующей услуги провайдера (методы 4.1.15 и 4.1.17)

4.8.3.2. Добавить – добавление услуги провайдера в каталог (методы 4.1.13 и 4.1.17)

4.9. Страница со списком заявок

4.9.1. Доступна только авторизованным пользователям. Пользователям доступны только их заявки, менеджерам доступны все заявки

4.9.2. Выводится информация о заявках в табличном виде (метод 4.1.5)

4.9.2.1. ID заявки

4.9.2.2. Статус заявки

4.9.2.3. Дата создания

4.9.2.4. Дата формирования

4.9.2.5. Дата завершения

4.9.3. Действия



4.9.3.1. Открыть заявку – переход на страницу 4.10

4.9.3.2. Завершить заявку (метод 4.1.9). Доступно только менеджерам

4.9.3.3. Отклонить заявку (метод 4.1.9). Доступно только менеджерам

4.10. Страница с подробным описанием заявки на подключение. Пользователь может изменять данные заявки. Менеджер может просматривать заявки пользователей

4.10.1. Доступна только авторизованным пользователям

4.10.2. Выводится информация об услугах провайдера в виде горизонтальных карточек

4.10.3. Действия

4.10.3.1. Удалить услугу провайдера из заявки. Доступно только создателю заявки (метод 4.1.18)

4.10.3.2. Изменить данные услуги провайдера в заявке. Доступно только создателю заявки (метод 4.1.19)

4.10.3.3. Изменить данные заявки на подключение. Доступно только создателю заявки (метод 4.1.7)

4.10.3.4. Сформировать заявку. Доступно только владельцу заявки, если заявка находится в статусе черновика (метод 4.1.8)

4.10.3.5. Удалить заявку. Доступно только владельцу заявки, если заявка находится в статусе черновика (метод 4.1.10)

4.10.3.6. Открыть услугу провайдера – переход на страницу 4.7

4.11. Страница личного кабинета пользователя интернет-услуг

4.11.1. Доступна только авторизованным пользователям интернет-услуг.

4.11.2. Действия

4.11.2.1. Изменение информации о пользователе интернет-услуг (метод 4.1.4)

4.11.2.2. Выйти – переход на страницу 4.3, выход из аккаунта (метод 4.1.3)

4.12. Страница статуса «403 Forbidden». Отображается, если у пользователя нет доступа к запрашиваемой странице

4.13. Страница статуса «404 Not found». Отображается, если запрашиваемая страница не найдена

4.14. Страница с таблицей услуг провайдера

4.14.1. Доступна только менеджерам

4.14.2. Выводиться информация об услугах провайдера в табличном виде (метод 4.1.11)

4.14.2.1. Название услуги провайдера

4.14.2.2. Описание

4.14.2.3. Стоимость подключения

4.14.3. Действия

4.14.3.1. Открыть услугу провайдера – переход на страницу 4.7

4.14.3.2. Добавить услугу провайдера – переход на страницу 4.8

## **5 Требования к составу и параметрам технических средств**

5.1. Серверная часть

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

5.1.2. Оперативная память от 4 Гб

5.1.3. Место на жестком диске от 15 Гб

5.2. Клиентская часть

5.2.1. Процессор с частотой от 2ГГц

5.2.2. Оперативная память от 2 Гб

5.2.3. Место на жестком диске от 20 Гб

## **6 Требования к составу и параметрам технических средств**

6.1. Серверная часть

6.1.1. ОС: Linux Ubuntu

6.1.2. JDK Development Kit 17.0.12

6.1.3. Redis 7.2.3

6.1.4. Minio RELEASE.2024-09-16T04-13-10Z

6.1.5. Postgres 16.0

6.2. Клиентская часть

6.2.1. ОС: Windows/Linux

6.2.2. Веб-браузер: Chrome (129 и выше)

## ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ

Таблица 1 – HTTP методы разрабатываемого веб-сервиса

№	Тип	URL	Описание	Входные данные	Выходные данные
1.1. Авторизации и аутентификации					
4.1.1.	POST	/api/users/create	Регистрация	{ login: string, password: string, role: string }	
4.1.2.	POST	api/users/login	Вход в аккаунт	{ login: string password: string }	auth JWT { accessToken: string, expiresIn: long }
4.1.3.	POST	/api/users/logout	Выход из аккаунта. Доступно только авторизованным пользователям	auth JWT	
4.1.4.	POST	/api/users/update	Изменение данных пользователя	auth JWT { login: string, password: string, role: string }	auth JWT { login: string, password: string, role: string }

№	Тип	URL	Описание	Входные данные	Выходные данные
				}	}
1.2. Методы заявок на подключение услуг провайдера					
4.1.5.	GET	/api/connection-requests	Получение всех заявок на подключение услуг провайдера. Авторизованному пользователю доступны только его заявки. Менеджеру доступны все заявки	{ creationDatetime: string, completionDatetime: string, status: string }	[[ id: number, status: string consumer: string, phoneNumber: string, creationDatetime: string, formationDatetime: string, completionDatetime: string, client: string, manager: string ]]
4.1.6.	GET	/api/connection-requests /id	Получение заявки на подключение по id	-	{ status: string consumer: string, phoneNumber: string, creationDatetime: string, formationDatetime: string, completionDatetime: string, client: string,

№	Тип	URL	Описание	Входные данные	Выходные данные
					manager: string, providerServiceList: [{ id: number, title: string, price: number, monthlyPayment: boolean, imgUrl: string, unit: string, amountDescription: string, amount: number }, ] }
4.1.7.	PUT	/api/connection-requests/id/put	Изменение данных заявки на подключение. Доступно только авторизованному пользователю	{ consumer: string, phoneNumber: string } 	-
4.1.8.	PUT	/api/connection-requests/id/form	Формирование заявки на подключение. Доступно только авторизованному пользователю	-	-

№	Тип	URL	Описание	Входные данные	Выходные данные
4.1.9.	PUT	/api/connection-requests/id/resolve	Завершение заявки. Доступно только менеджеру	{ status: string }	{ id: number, status: string consumer: string, phoneNumber: string, creationDatetime: string, formationDatetime: string, completionDatetime: string, client: string, manager: string }
4.1.10.	DELETE	/api/connection-requests/id/delete	Удаление заявки на подключение. Доступно только авторизованному пользователю	-	-
1.3. Методы услуг провайдера					
4.1.11.	GET	/api/provider-duties	Получение списка услуг провайдера и информации о текущей заявке-черновике.	{ title: string, monthlyPayment: boolean }	{ itemsInCart: number, connectionRequestId: number, providerServiceList: [ {

№	Тип	URL	Описание	Входные данные	Выходные данные
					id: number, title: string, price: number, monthlyPayment: boolean, imgUrl: string, unit: string, amountDescription: string } ]
4.1.12.	GET	/api/provider-duties/id	Получение информации об услуге провайдера по id	-	{ id: number, title: string, description: string, price: number, monthlyPayment: boolean, imgUrl: string, unit: string, amountDescription: string }



№	Тип	URL	Описание	Входные данные	Выходные данные
4.1.13.	POST	/api/provider-duties/post	Добавление новой услуги провайдера в каталог. Доступно только менеджеру.	{ title: string, description: string, price: number, monthlyPayment: boolean, imgUrl: string, active: true, unit: string, amountDescription: string }	{ id: number, title: string, description: string, price: number, monthlyPayment: boolean, imgUrl: string, unit: string, amountDescription: string }
4.1.14.	DELETE	/api/provider-duties/id/delete	Удаление услуги провайдера из каталога. Доступно только менеджеру	-	-
4.1.15.	PUT	/api/provider-duties/id/update	Изменение информации об услуге провайдера по id. Доступно только менеджеру	{ title: string, description: string, price: number, monthlyPayment: boolean, active: true, unit: string, amountDescription: string }	{ id: number, title: string, description: string, price: number, monthlyPayment: boolean, imgUrl: string, active: true, }

№	Тип	URL	Описание	Входные данные	Выходные данные
				}	unit: string, amountDescription: string }
4.1.16.	POST	/api/provider-duties/id/add	Добавление услуги провайдера в заявку-черновик. Доступно только авторизованному пользователю.	-	-
4.1.17.	POST	/api/provider-duties/id/image	Добавление логотипа услуги провайдера в заявку. Доступно только менеджеру	File: Multipart/Formdata	-
1.4. Методы услуги провайдера в заявке на подключение					
4.1.18.	DELETE	/api/duties-requests/duty_id/re q_id/delete	Удаление услуги провайдера из заявки-черновика. Доступно только для авторизованных пользователей.	-	-
4.1.19	PUT	/api/duties-requests/duty_id/re q_id/update	Указание количественной характеристики услуги в заявке	{ amount: number }	-