|  |  |
| --- | --- |
| **­­**Изображение выглядит как эмблема, герб, нашивка, символ  Автоматически созданное описание | **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:***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*** **Походы викингов *\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

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

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

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

Студент \_\_ИУ5И-51Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Кашима \_\_\_\_\_\_\_\_\_\_**

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

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

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

# АННОТАЦИЯ

Объектом разработки является система организации и управления походами викингов. Данная программа позволяет планировать маршруты, формировать состав участников, а также управлять ресурсами и стратегией походов.

Цель работы заключается в изучении истории и особенностей походов викингов, а также создании набора программного обеспечения и сопутствующих веб-сервисов для управления походами.

В рамках работы была разработана архитектура веб-сервиса, рассчитанного на планирование маршрутов и управления ресурсами, создан пользовательский интерфейс для взаимодействия с данным веб-сервисом, разработаны и развернуты веб-сервер, нативное приложение и прогрессивное веб-приложение, обеспечивающие удобный доступ к функционалу системы.

Пояснительная записка содержит 2 приложения.

# СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc185626470)

[СОДЕРЖАНИЕ 3](#_Toc185626471)

[ВВЕДЕНИЕ 4](#_Toc185626472)

[1 ПРЕДМЕТНАЯ ОБЛАСТЬ 7](#_Toc185626473)

[2 АРХИТЕКТУРА 12](#_Toc185626474)

[3 АЛГОРИТМЫ 16](#_Toc185626475)

[4 ОПИСАНИЕ ИНТЕРФЕЙСА 18](#_Toc185626476)

[ЗАКЛЮЧЕНИЕ 27](#_Toc185626477)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 28](#_Toc185626478)

[ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ 29](#_Toc185626479)

[1. Введение 30](#_Toc185626480)

[2. Назначение разработки 30](#_Toc185626481)

[3. Стадии и этапы разработки 30](#_Toc185626482)

[4. Требования к функциональным характеристикам 31](#_Toc185626483)

[5. Требования к составу и параметрам технических средств 36](#_Toc185626484)

[6. Требования к информационной и программной совместимости 36](#_Toc185626485)

[ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ 38](#_Toc185626486)

# ВВЕДЕНИЕ

Историческая область, посвященная походам викингов, стремительно развивается, привлекая внимание историков со всего мира. В силу увеличения интереса к этой теме возникает потребность в удобном инструменте для обмена информацией и создания исторических нарративов. С учетом того, что многие историки предпочитают участвовать в обсуждениях и делиться своими исследованиями, решено создать специализированный веб-ресурс.

Целью работы является реализация классификации и предложения тем для исследовательских проектов в области викингов, включающую в себя веб-сервис, веб-приложение, мобильное приложение и выделенный сервис проверки грамотности статьи для поддержания высокого уровня качества исследовательских материалов.

Система предназначена для историков и модераторов сайта. На сайте предусмотрен ограниченный доступ к городам. Для получения доступа пользователю необходимо создать поход на получение доступа к выбранному городу. Система предоставляет автоматизированный способ создания, учета и ведения заявок. Также она позволяет модераторам принимать или отклонять заявки, редактировать существующие и создавать новые карты.

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.Ознакомиться с разработкой бэкенда с использованием фреймворка Django.
2. Разработать структуру и создать базу данных PostgreSQL, подключить её к бэкенду.
3. Создать веб-сервис со всей итоговой бизнес-логикой, кроме авторизации, в бэкенде системы для использования его в SPA.
4. Добавить авторизацию, использовать Redis для хранения сессий, а также внедрить Swagger в веб-сервис.
5. Разработать базовый интерфейс приложения для гостя на React.
6. Внедрить менеджер состояний Redux Toolkit для хранения значений фильтров, добавить адаптивность и PWA.
7. Завершить разработку интерфейса пользователя в React, использовать для обращений к методам веб-сервиса Axios.
8. Реализовать React интерфейс издатель, внедрить Real-time web.
9. Разработать десктопное приложение на Tauri.
10. Развернуть приложение на GitHub Pages.
11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта. код проекта.

# 1 ПРЕДМЕТНАЯ ОБЛАСТЬ

Города викингов были основаны в VIII-XI веках в ходе их морских экспансий [1]. Например, Бирка в Швеции и Рейкьявик в Исландии служили центрами торговли, где викинги обменивались товарами и культурными влияниями. Эти поселения были оборудованы домами, верфями и ковчегами (treasure vaults) для хранения сокровищ, что подчеркивает их разносторонний облик. Города викингов являлись не только торговыми точками, но и важными центрами социального и культурного обмена.

Система хранения данных в рамках проекта учитывает информацию о городах. Каждый город имеет следующие атрибуты: Название Описание Фотография.

Цель и результаты исследования из arXiv.org, В работе [2] ("Влияние викингов на торговлю и миграцию в средневековой Европе",: "The Impact of Vikings on Trade and Migration in Medieval Europe") исследуется, как торговые маршруты и морские походы викингов способствовали развитию городской торговли и обмена знаниями в VIII-XI веках. Согласно результатам исследования, успешные экспедиции викингов были обусловлены их высокоразвитыми морскими технологиями и умением адаптироваться к различным культурным и географическим условиям.

Эти выводы подкрепляют ключевую идею системы: использование городов как отправных точек для организации походов, что позволяет пользователям лучше понять логику экспедиций викингов.

Функционал системы, Создание черновой заявки, Пользователи выбирают города и добавляют их в черновую заявку для планирования похода. В одном походе может быть несколько городов. Редактирование черновой заявки, Добавление/удаление городов, Возможность удалить черновую заявку, Формирование похода

Когда пользователь завершает выбор, поход отправляется на модерацию. Модерация похода

Исторический модератор (historical moderator) принимает или отклоняет поход. После решения модератора статус обновляется: Принято Отклонено, Роли пользователей:

Гость (guest): может просматривать список городов и готовые походы.

Исторический исследователь (historical researcher): создает и отправляет походы на модерацию. Исторический модератор (historical moderator): проверяет походы и выносит решения. Диаграммы прецедентов в приложении иллюстрируют взаимодействие пользователей и системы. Ссылки

[1] Исторические данные о городах викингов.

[2] Preprint из arXiv.org: “The Impact of Vikings on Trade and Migration in Medieval Europe”. Более описан на диаграммах прецедентов (рис. 1).

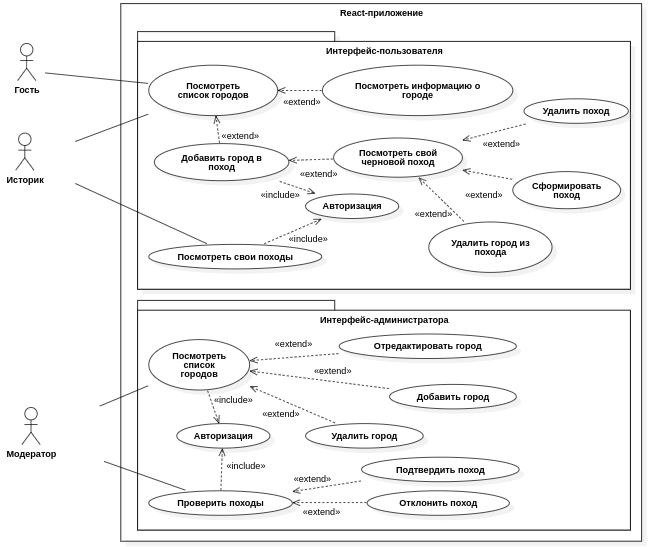


Рисунок 1 — Диаграмма прецендентов

Гостям доступен просмотр городов. Зарегистрированные гости – исторические исследователи. Они могут добавлять города в поход, росматривать список своих походов и сформировывать текущий поход. Походы обрабатываются историческим модератором. В результате обработки похода её либо одобряют, либо отклоняют. Помимо возможностей принятие/отказа, историческому модератору также доступны уникальные функции для работы с городами, а именно: просмотр всех городов, редактирование, создание и удаление городов, а также просмотр списка всех городов в табличном виде. Процесс оформления похода отражен на диаграмме бизнес-процесса (рис. 2).

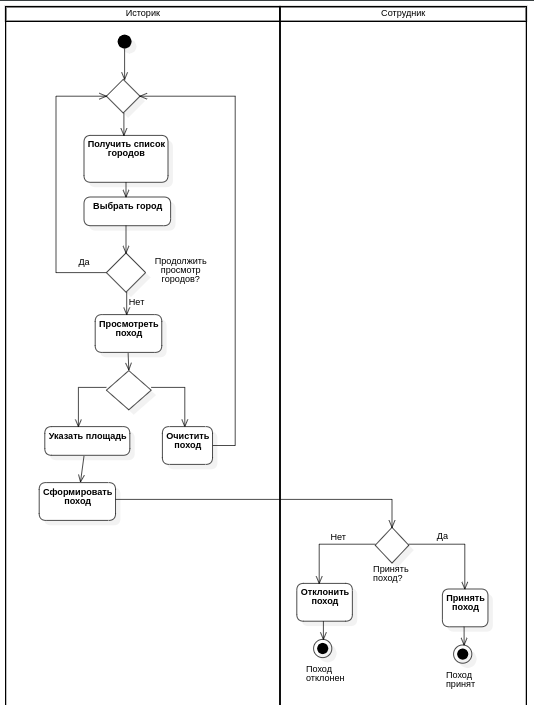
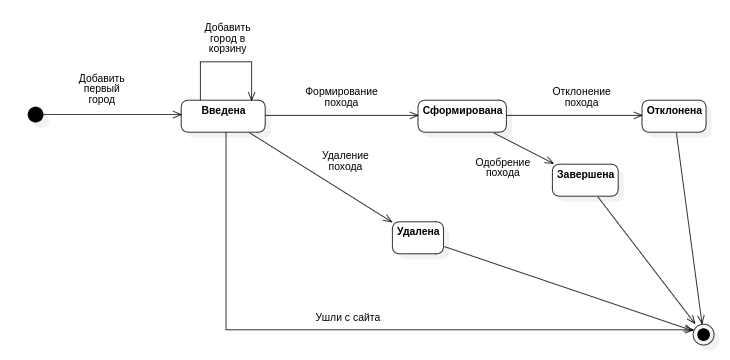


Рисунок 2 - Диаграмма бизнес-процесса

Исторический исследователь выбирает города, затем формирует на основе выбранных городов поход. Этот поход затем обрабатывает асинхронный сервис, а затем и исторический модератор. В случае одобрения похода запускается процесс проверки грамотности содержимого статьи в асинхронном сервисе, выставляющий оценку данной статьи. Возможные состояния похода отражены на диаграмме состояний (рис. 3).

 Рисунок 3 - Диаграмма состояний похода

При выборе первого города формируется. Последующие выбранные города добавляются в этот введену. Пользователь затем формирует поход, удаляет её или выходит из приложения. Сформированный поход обрабатывает исторический модератор. Он может одобрить или отклонить её.

# 2 АРХИТЕКТУРА

Архитектура системы представлена на диаграмме развертывания, изображенной на рисунке 4.

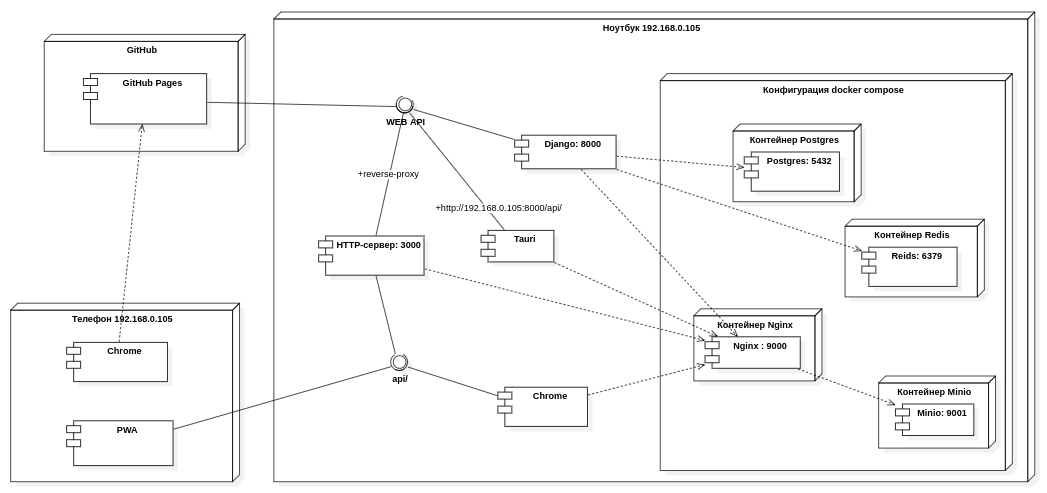


Рисунок 4 — Диаграмма развертывания

Десктопное Tauri приложение запускается вне докера, фронт тоже запускается вне докера, фронт разворачивается на Github Pages PWA устанавливается на телефон через Github Pages бэкенд запускается не на localhost, а на локальном ip адресе ноута для хранения данных была выбрана СУБД PostgreSQL. Выбранная СУБД является стандартом индустрии. Структура данных приведена на ER-диаграмме (рисунок 5). Таблица Пользователь содержит сведения о пользователях системы, таких как модераторы или участники. В ней хранятся уникальный идентификатор пользователя, информация для аутентификации, такие как имя пользователя, email и пароль. Также присутствуют булевые поля, которые определяют роль и активность пользователя: является ли он сотрудником, суперпользователем или активен ли в системе. В таблице указаны даты регистрации и последнего входа пользователя, а также имя и фамилия.

Таблица Походы хранит информацию о походах или экспедициях. Она включает уникальный идентификатор похода, текущий статус и временные метки, такие как, дата формирования похода. Также присутствуют ссылки на пользователя, отвечающего за поход (историк), и на модератора. Дополнительно хранится поле с дополнительной информацией о походе. Связь "один ко многим" с таблицей пользователей указывает на то, что один пользователь может быть историком или модератором нескольких походов.

Таблица Город содержит данные о городах, связанных с походами. В ней хранится уникальный идентификатор города, его статус, изображение, название и описание. Также присутствует поле с площадью города в числовом выражении.

Промежуточная таблица M-M реализует связь «многие ко многим» между таблицами Походы и Город. Она позволяет одному походу быть связанным с несколькими городами и наоборот, одному городу участвовать в нескольких походах. В этой таблице хранятся ссылки на идентификаторы похода и города, а также дополнительное поле для хранения значений, связанных с этой связью.

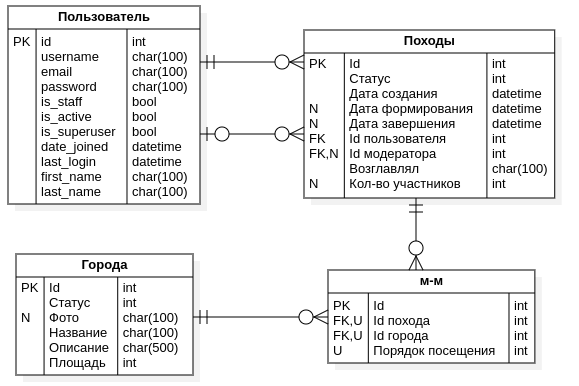


Рисунок 5 — ER-диаграмма

Устройство бэкенда приложения разработанной системы отражено на диаграмме классов бэкенда (рисунок 6). Пользователи взаимодействуют с доменами, которые, в свою очередь, связаны с моделями. Модели имеют связи с таблицами в базе данных.

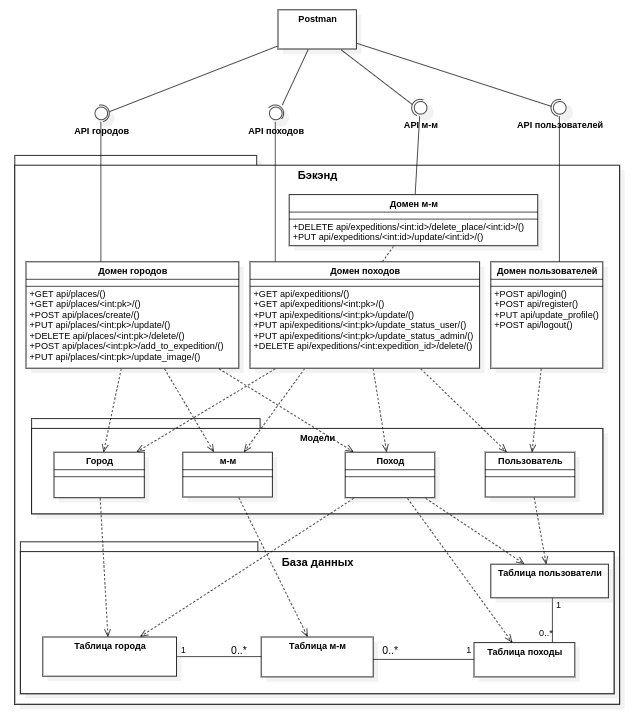


Рисунок 6 — Диаграмма классов бэкенда

Диаграмма классов фронтенда визуализирует связь фронтенда и бэкенда (рисунок 7). Каждая страница связана с API, используемым для взаимодействия с данными на соответствующей страницы.

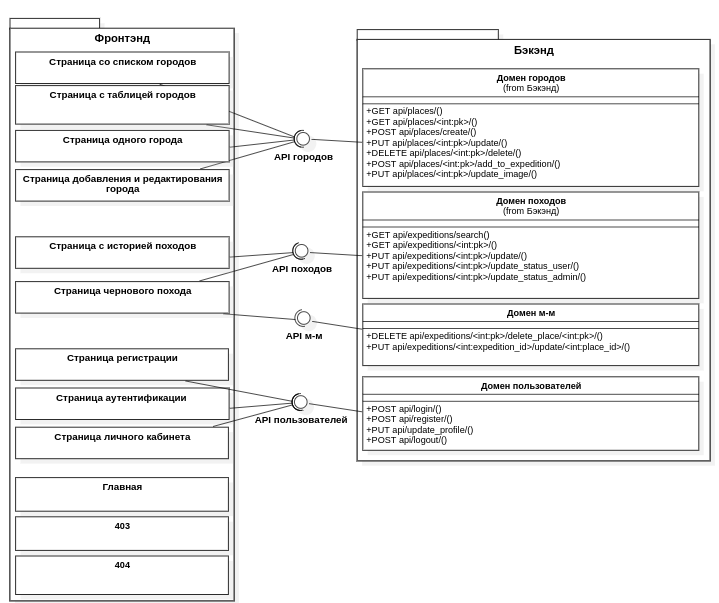


Рисунок 7 — Диаграмма классов фронтенда

# 3 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: города, походы и исторические исследователи. Методы следуют правилам REST API.

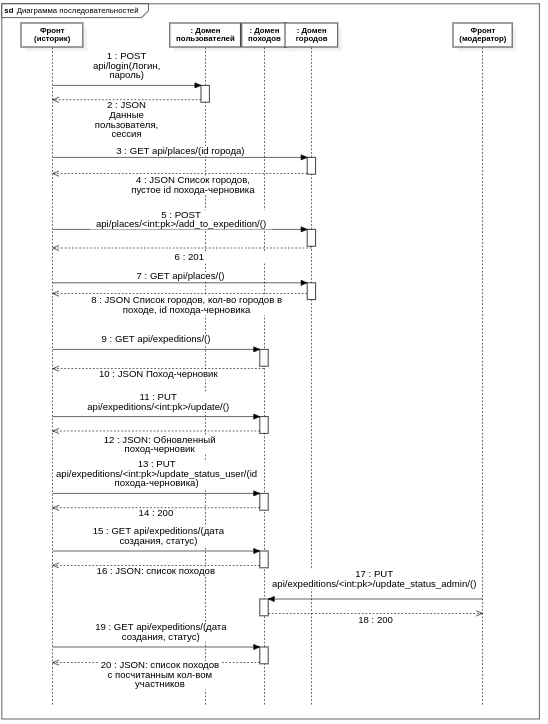


Рисунок 8 — Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация исторического исследователя. Для этого он отправляет через графический интерфейс запрос, передавая в нем имя, 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 городов, которые возвращаются в JSON формате. Пользователь выбирает город, которую хочет добавить, и, нажимая на кнопку «плюс» в графическом интерфейсе, отправляет запрос на добавление города в свою черновую заявку. Этот процесс может продолжаться несколько раз.

Когда пользователь определится с выбором, он нажимает на кнопку «сформировать» в графическом интерфейсе. После этого приложение запрашивает id черновой заявки пользователя и затем отправляет запрос на формирование этого похода. В этот момент основной веб-сервис выполняет асинхронный запрос к сервису на то, чтобы он проверил грамотность статьи. Когда поход и его города будут одобрены, пользователь сможет увидеть через некоторое время, что оценка его статьи изменилась.

Процесс рассмотрения походов происходит также через графический интерфейс. Исторический модератор может просматривать списки всех походов и, нажимая на соответствующие кнопки, отправлять запросы на одобрение или отклонение похода в основной веб-сервис. В эти запросы также можно включить фильтры по имени создателя похода и диапазону дат, в котором должны были быть созданы походы. Также через графический интерфейс они могут управлять непосредственно городами. Им доступны такие функции, как создание и редактирование города, просмотр списка городов и удаление их. Для каждой из этих функция присутствует свой метод, отправляемый на основной веб-сервис

# 4 ОПИСАНИЕ ИНТЕРФЕЙСА

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

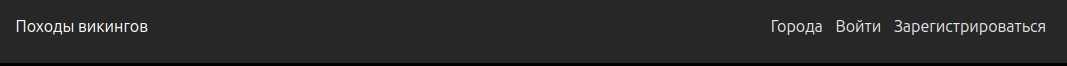


Рисунок 9 - Главное меню (неавторизированный пользователь)

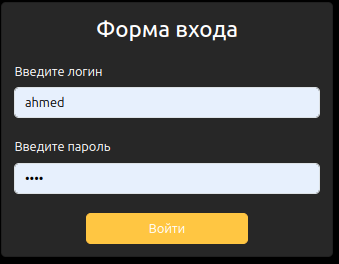


Рисунок 10 - Страница Входа

На этой странице отображается форма, через которую гость входит в свой аккаунт. При успешном вводе логина и пароля гость получает JWT, который сохраняется в cookies и используется при отправлении запросов.

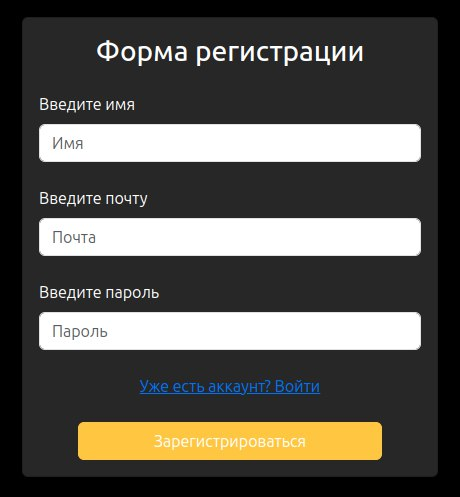


Рисунок 11 - Страница регистрации

На этой странице гости могут завести аккаунт. Для этого нужно указать имя, поча и пароль, Если введенный логин уже занят система попросит пользователя сменить его.

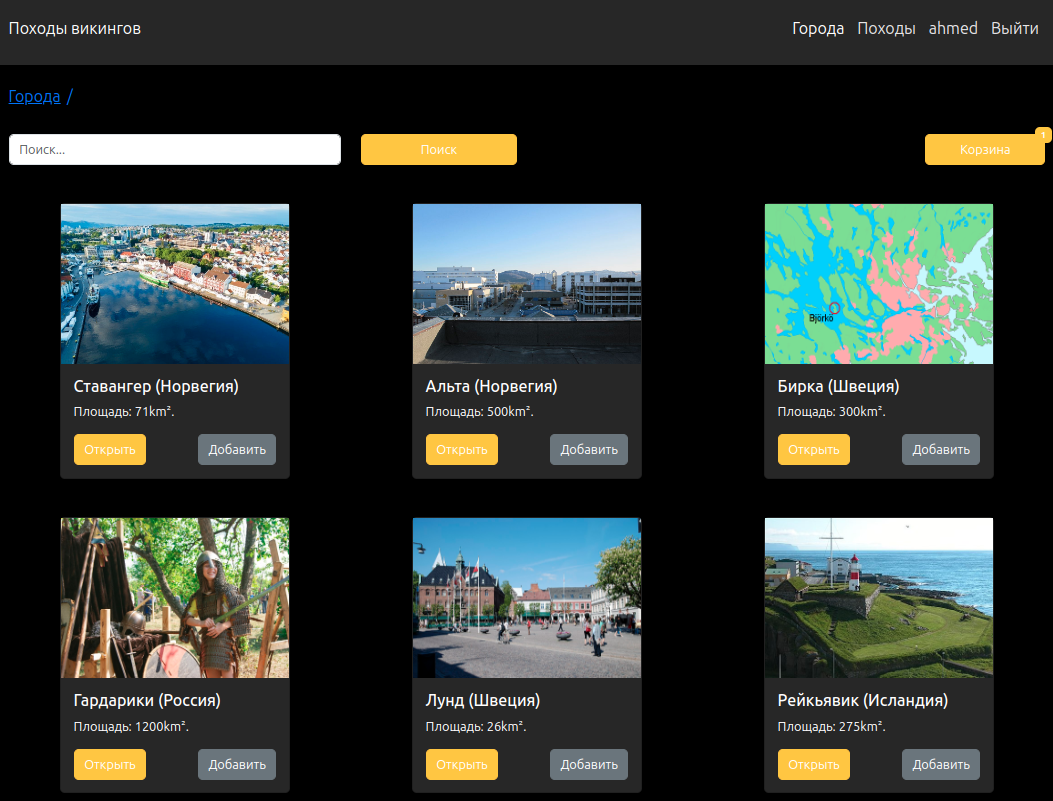


Рисунок 12 - Страница со списком городов .(исторический исследователь)

На этой странице отображается список городов в виде карточек. У каждой города есть хендлер нажатия для перехода в режим «Подробнее», переносящая пользователя на страницу с подробной информацией о городе (рисунок 12).Пользователь может добавыть города в карзину

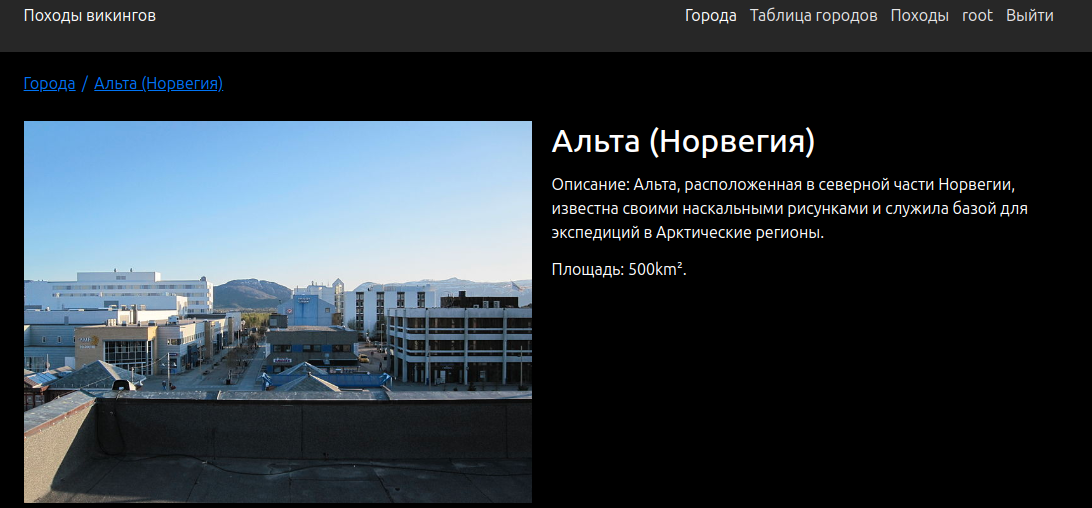


Рисунок 13 - Страница с подробным описанием города

На этой странице отображается подробная информация о городе: название, площадь и текстовое описание.

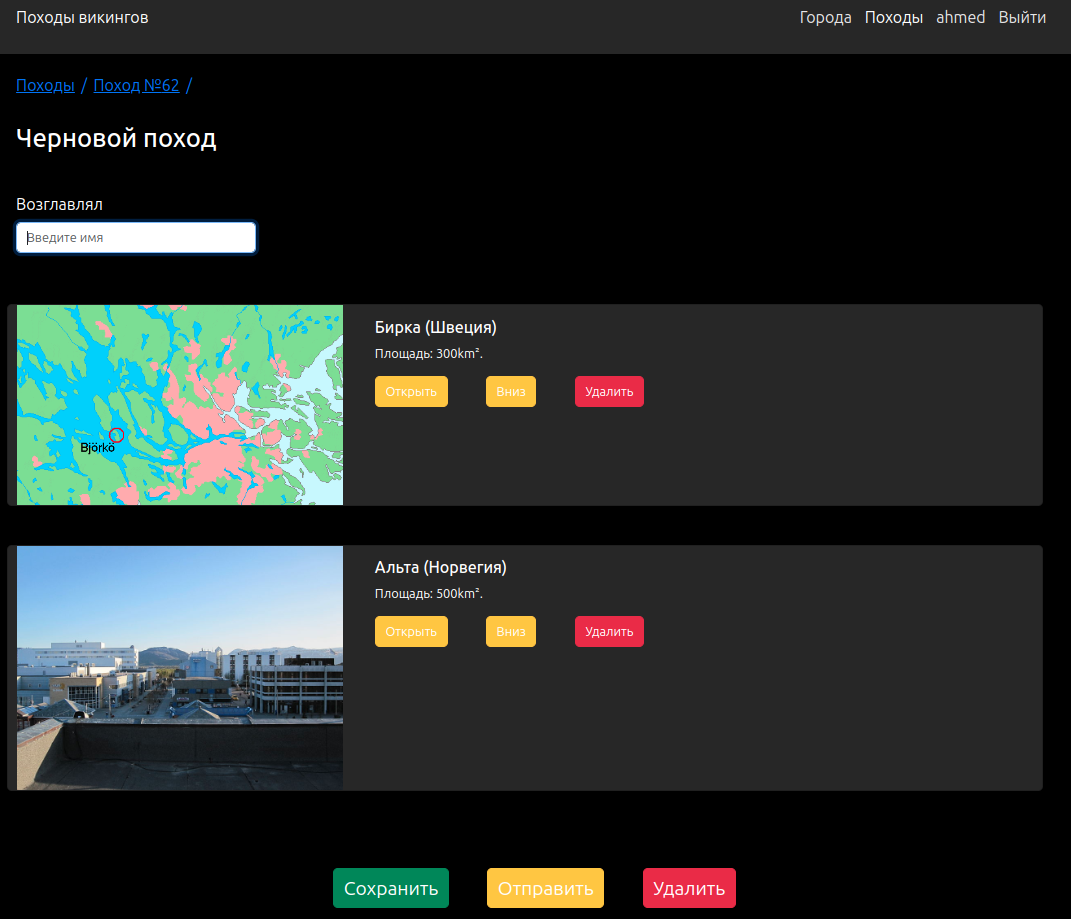


Рисунок 14 - Страница Черновой поход

На эту страницу исследователь может просматривть созданные походы или удалить поход , а также можно перейти к подробной информацию нажатия на кнопку «открыть». после того как сохранил все данные он может отправить.

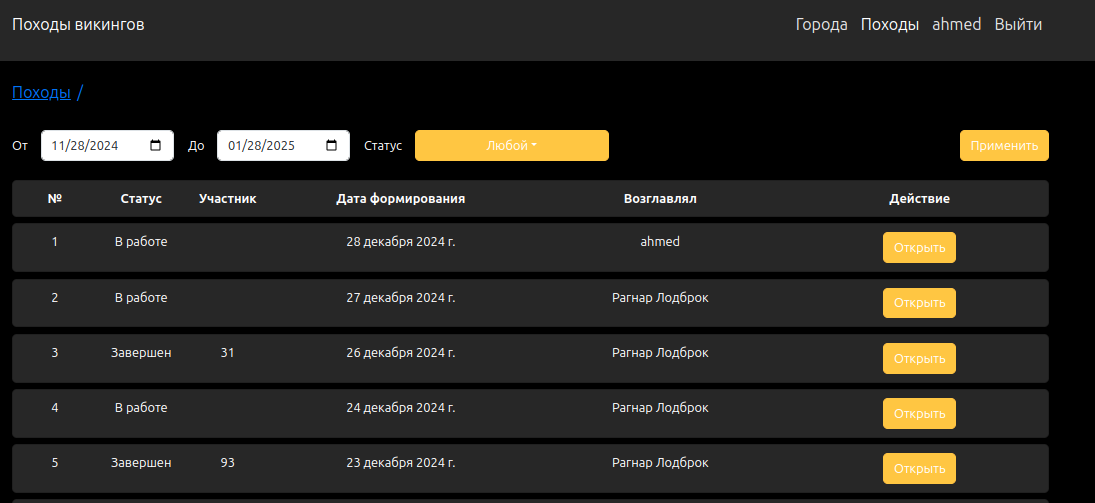


Рисунок 15 - Страница со списком заявок.(исторический исследователь)



Рисунок 16 — Меню приложения для .(исторический модератор)

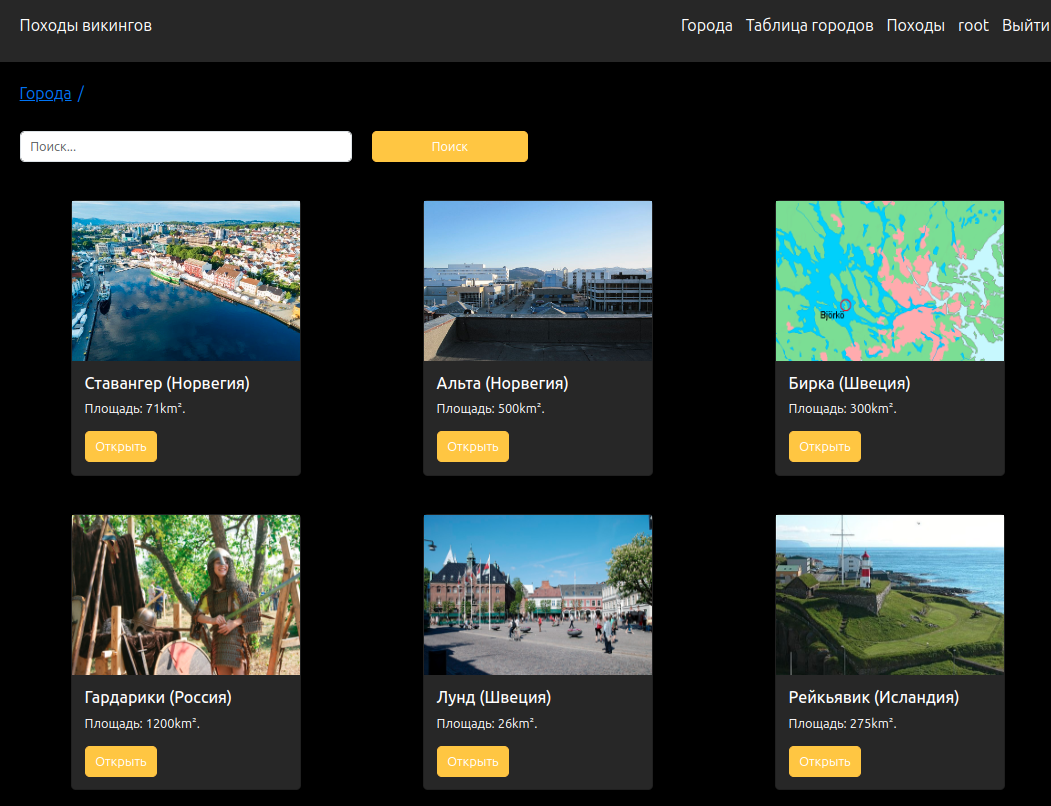


Рисунок 17 - Страница со списком городов

На этой странице отображается список городов в виде карточек. У каждой города есть хендлер нажатия для перехода в режим «Подробнее», переносящая пользователя на страницу с подробной информацией о городе

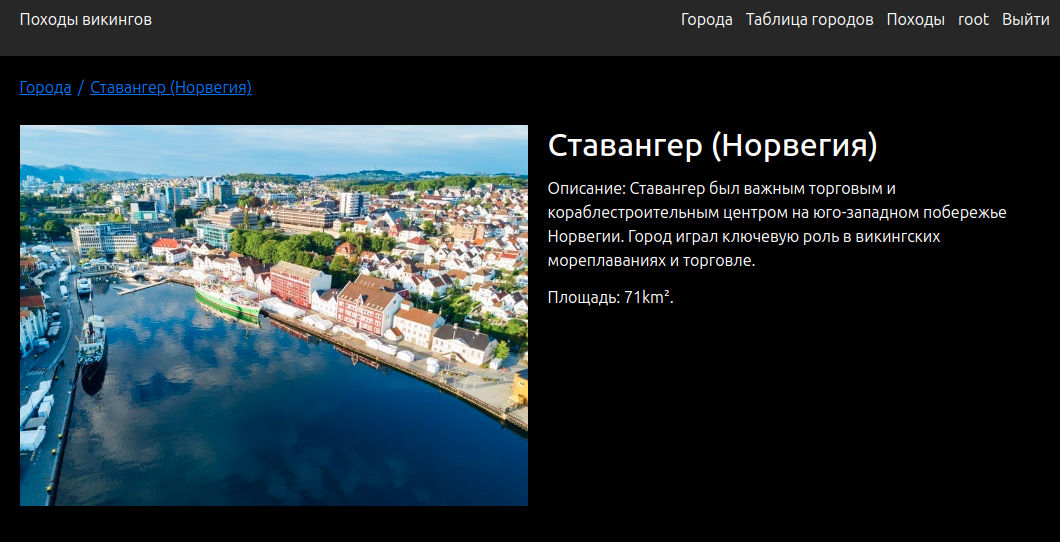


Рисунок 18 - Страница с подробным описанием города

На этой странице отображается подробная информация о городе: название, площадь и текстовое описание.



Рисунок 19 - Страница со списком заявок.(исторический модератор)

Для исторического модерато функционал этой страницы шире. Для них отображается список всех заявок всех пользователей с более подробной информацией: Дата формирования,Возглавлял, статус, участник. На странице доступна фильтрация по логину пользователя, он может принять или отклонять заявки поход нажатия соответствующих кнопок в интерфейсе.

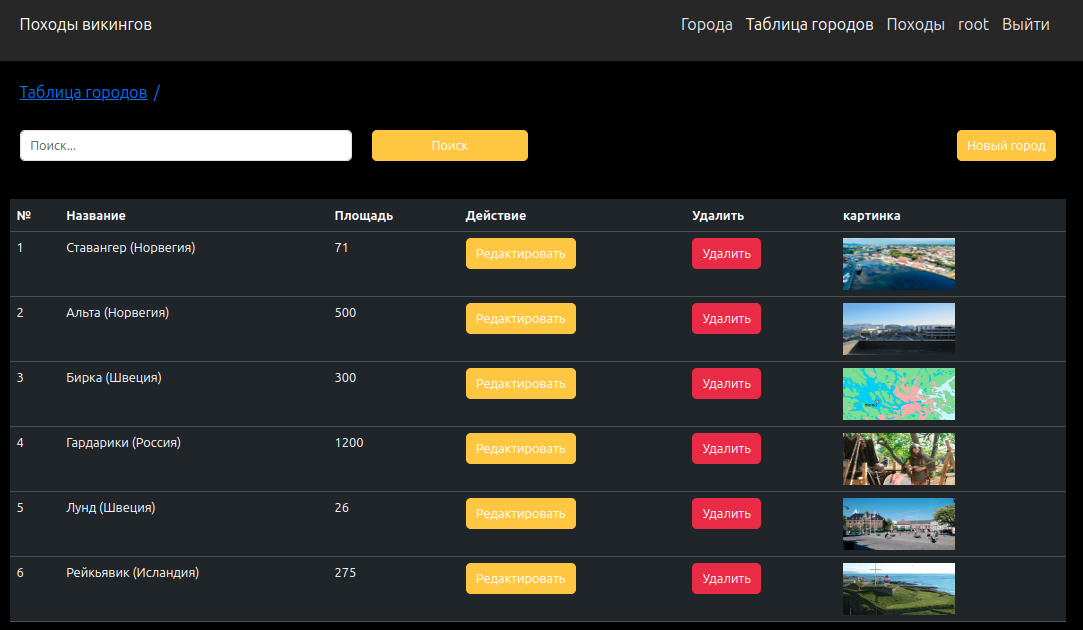


Рисунок 20 - Страница с таблицей городов

На этой странице историк может редактировть, удалить и создать новый город

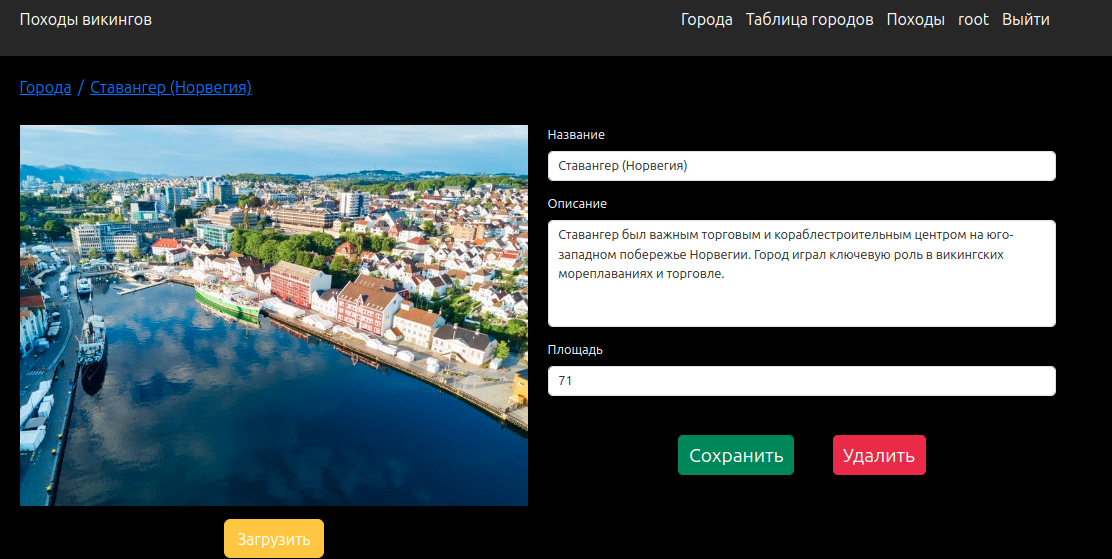


Рисунок 21 - Страница с таблицей городов в режиме редактирования города

На этой странице исторический модератор может в компактном и удобном формате просмотреть список всех городов, существующих в системе и отредактировать информацию о городе. Отображаются следующие поля: название, изображение-обложка, описание города

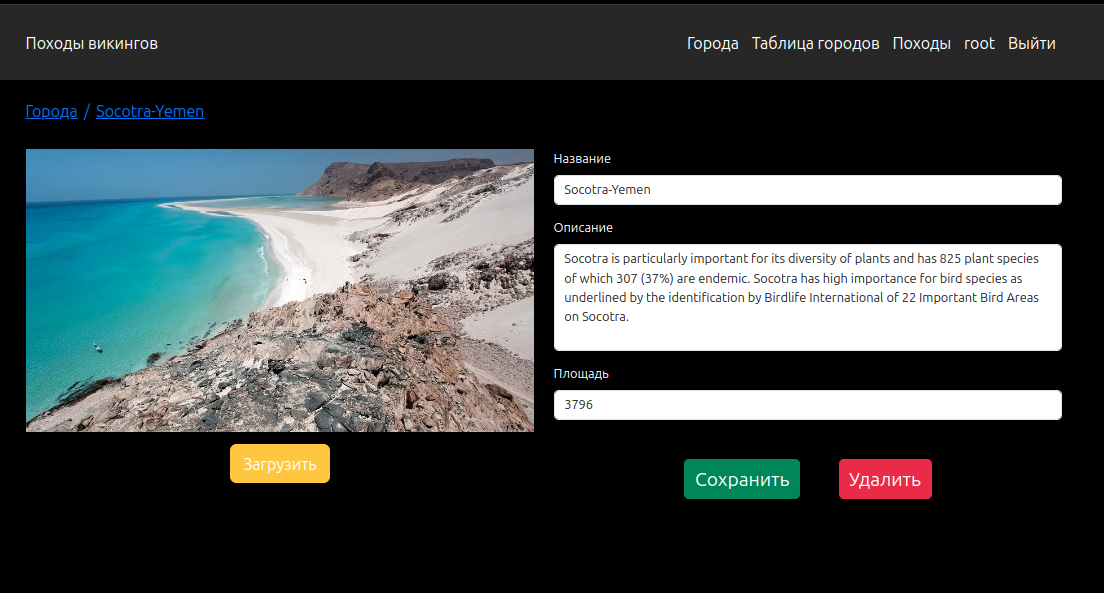


Рисунок 22 - Страница создания города

На этой странице историк может создать новый город. Для создания доступны все поля: название, описание, площадь и фотография города.



Рисунок 23 — Страница личного кабинета

на эту страницу можно изменить свои данны (паролка), личный кабинет открывается при нажатии на логин пользователя в меню

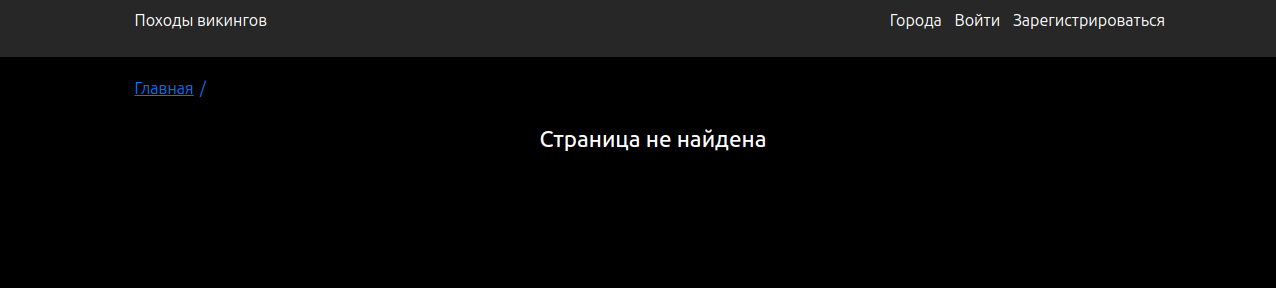


Рисунок 24 — Страница ошибки 404

В случае, если пользователь запрашивает несуществующую станицу, он перенаправляется на страницу ошибки 404

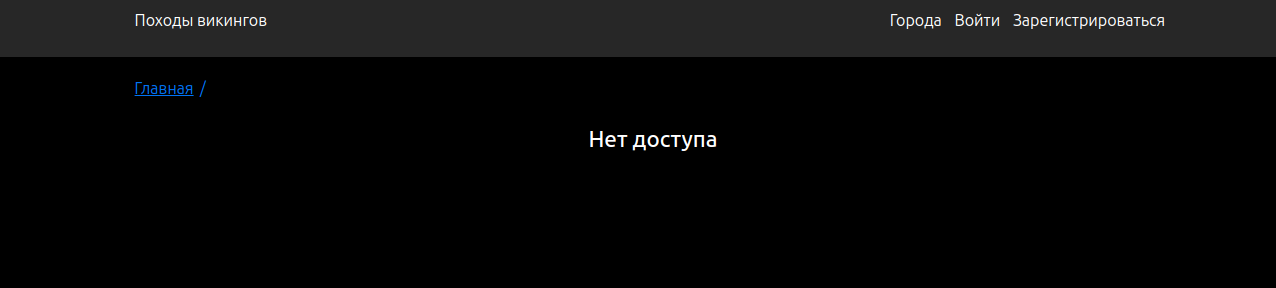


Рисунок 25 — Страница ошибки 403

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

# ЗАКЛЮЧЕНИЕ

В ходе работы были достигнуты следующие результаты:

1. Разработан дизайн приложения в Figma на основе superizdatelstvo.ru. После ознакомления с разработкой бэкенда с использованием фреймворка Django создан MVP.
2. Спроектирована и создана база данных PostgreSQL, а затем подключена к бэкенду.
3. Создан веб-сервис со всей итоговой бизнес-логикой, кроме авторизации, в бэкенде системы для использования его в SPA.
4. Добавлена авторизация с использованием Redis для хранения сессий, в веб-сервис внедрен Swagger.
5. Разработан базовый интерфейс приложения для гостя на React.
6. Внедрен менеджер состояний Redux Toolkit для хранения значений фильтров, добавлена адаптивность и PWA.
7. Завершена разработка интерфейса пользователя в React, для обращений к методам веб-сервиса использован Axios.
8. Реализован React интерфейс издателя, внедрен Real-time web.
9. Разработано десктопное приложение Tauri.
10. Приложение развернуто при помощи сервиса GitHub Pages и доступно по ссылке:https://kashima1234.github.io/WEBF\_IU5\_BMSTU
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформлен git-репозиторий на сервисе GitHub, содержащий исходный код проекта:https://github.com/kashima1234/WEBB\_IU5\_BMSTU

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Экспансия викингов (viii-xi вв.). Роль норманнов в создании единого европейского экономического пространства Роль викингов в европейской истории [Электронный ресурс] // Habr, URL: https://schoolperspektiva.ru/tolstojj-l-n-/ekspansiya-vikingov-viii-xi-vv-rol-normannov-v-sozdanii-edinogo-evropeiskogo/ (дата обращения 14.10.2024)
2. Работа с историческими источниками [Электронный ресурс] // Habr, URL: https://pikabu.ru/story/rabota\_s\_istoricheskimi\_istochnikami\_kak\_izbezhat\_oshibok\_pisatelyu\_avtorusostavitelyu\_8527824?ysclid=lrdjk7i0ys302958148 (дата обращения 04.10.2024).
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 Python [Электронный ресурс] // Python. URL: <https://docs.python.org/3/index.html/> (дата обращения: 20.10.2024).
8. Рецепты Django ORM [Электронный ресурс] // Django. URL: https://django.fun/docs/django-orm-cookbook/2.0/ (дата обращения: 10.10.2024).
9. Документация по Tauri [Электронный ресурс] // Tauri. URL: <https://v2.tauri.app/develop/> (дата обращения: 20.11.2024).
10. 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  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.И. Канев  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

\_

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Кашима Ахмед |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва - 2024

1. Введение
   1. Цель система, включающей в себя веб-сервис , веб приложение нативное приложение заключается в оперативной организации походов викингов, обеспчивающей пользователям удобные инструменты для планирования управления и участия в мероприятих.
2. Назначение разработки
   1. Система предназначена для историков и модераторов сайта. На сайте предусмотрен ограниченный доступ к городам викингов. Для получения доступа пользователю необходимо создать поход на получение доступа к выбранному городу. Система предоставляет автоматизированный способ создания, учета и ведения заявок. Также она позволяет модераторам принимать или отклонять поход. Модераторы имеют возможность редактировать существующие и создавать новые походы.
3. Стадии и этапы разработки
   1. Создание MVP и базового дизайна на основе bombardier.com;
   2. Создать базу данных для хранения информации об элементах.
   3. Создать веб-сервис в бэкенде на Django (DRF);
   4. Реализовать авторизацию и хранение сессий в Redis;
   5. Разработать базовый SPA на React для гостя;
   6. Внедрение адаптивности, разработка Tauri приложения;
   7. Интерфейс пользователя клиента в React с менеджером состояний Redux Toolkit;
   8. Реализовать интерфейс сотрудника поддержки в React;
   9. Реализовать нативное приложение;
   10. Развернуть веб-приложение React на Github Pages;
   11. Подготовка документацию к системе (РПЗ, ТЗ, набор диаграмм);
   12. Подготовка репозитория на GitHub
4. Требования к функциональным характеристикам

### Список HTTP методов

* + 1. GET Получение всех мест
    2. POST Добавление нового места
    3. GET Получить одно место
    4. PUT Изменить информацию о месте
    5. DELETE Удалить место
    6. POST Добавить место к экспедиции
    7. POST Изменение/добавление изображения места
    8. GET Получить все экспедиции
    9. GET Получить одну экспедицию
    10. PUT Изменить поля экспедиции
    11. DELETE Удалить экспедицию
    12. PUT Сохранить экспедицию
    13. PUT Модерировать экспедицию
    14. DELETE Удалить место из экспедиции
    15. PUT Изменить статус isNew для места
    16. POST Регистрировать пользователя
    17. PUT Личный кабинет
    18. POST Аутентификация
    19. POST Деавторизация

### 4.2. Меню

4.2.1. Главная **-** перенаправляет на страницу 4.6

4.2.2. Список мест **–** перенаправляет на страницу 4.7 (вызывает метод 4.1.1)

4.2.3. Список экспедиций **–** перенаправляет на страницу 4.9

4.2.4. Редактирование мест – перенаправляет на страницу 4.11, доступно только астрономам

4.2.5. Зарегистрироваться **–** перенаправляет на страницу 4.3, появляется только для гостей

4.2.6. Личный кабинет – перенаправляет на страницу 4.10, только аутентифицированным пользователям

4.2.7. Войти – перенаправляет на страницу 4.4, только для гостей

4.2.8. Выйти – перенаправляет на страницу 4.5 (вызывает метод 4.1.19), доступно только аутентифицированным пользователям

### 4.3. Регистрация

4.3.1**.** Доступно только гостям

4.3.2. Отображает форму регистрации

4.3.2.1**.** Поле имени пользователя

4.3.2.2. Поле пароля

4.3.2.3. Поле имени

4.3.2.4. Поле фамилии

4.3.3. Действия

4.3.3.1**.** Регистрация пользователя – (вызывается метод 4.1.16)

4.3.3.2. Вернуться к аутентификации – перенаправляет на страницу 4.4

### 4.4. Аутентификация

4.4.1. Доступно только гостям

4.4.2. Отображает форму аутентификации

4.4.2.1. Поле имени пользователя

4.4.2.2. Поле пароля

4.4.3**.** Действия

4.4.3.1. Войти – (вызывается метод 4.1.18)

4.4.3.2. Регистрация – перенаправляет на страницу 4.3

### 4.5. Личный кабинет

4.5.1. Доступно аутентифицированному пользователю

4.5.2. Действия

4.5.2.1**.** Изменить данные пользователя – (вызывается метод 4.1.17)

### 4.6. Главная

4.6.1**.** Доступна всем

4.6.2. Отображается статическая информация

4.6.2.1. Назначение сервиса

4.6.2.2. Контакты для связи

### 4.7. Список мест

4.7.1. Доступна всем

4.7.2. Отображаются элементы карточек с местами, (вызывается метод 4.1.1)

4.7.2.1. Название места

4.7.2.2. Краткое описание

**4.7.3.** Действия

4.7.3.1. Поиск – перенаправляет на страницу 4.6, (используется метод 4.1.1), с фильтрующим параметром

4.7.3.2. Подробнее – перенаправляет на страницу 4.8 (используется метод 4.1.3)

4.7.3.3. Добавить в экспедицию – добавляет место в экспедицию, (вызывается метод 4.1.6), только аутентифицированные пользователи.

4.7.3.4. Кнопка корзины – перенаправляет на страницу 4.9, только аутентифицированные пользователи.

### 4.8. Одно место

4.8.1. Доступно всем

4.8.2. Отображается подробная информация выбранного места, (вызывается метод 4.1.3)

### 4.9. Спсок экспедиций

4.9.1. Доступно аутентифицированному пользователю

4.9.2. Отображает список экспедиций (метод 4.1.8)

4.9.2.1. Только экспедиции, созданные данным пользователем

4.9.3. Действия

4.9.3.1. Фильтрация – фильтрует экспедиции по дате создания или статусу,

4.9.3.2. Посмотреть подробную информацию об экспедиции

4.10. Список мест таблицей

4.10.1. Доступно доверенному модератору пользователю

4.10.2. Отображаются все существующие в системе места (вызывается метод 4.1.1)

4.10.3. Действия

4.10.3.1. Удалить – удаляет место (вызывается метод 4.1.5)

4.10.3.2. Редактирование/создание – переход на страницу 4.12

### 4.11. Редактирование/создание места

4.11.1. Доступно модератору

4.11.2. Отображается информация об изменяемом/добавляемом месте (вызывается метод 4.1.3)

4.11.2.1. Название

4.11.2.2. Детальное описание

4.11.2.3. Подробное описание

4.11.2.4. Картинка

4.11.3. Действия

4.11.3.1. Сохранить – добавляет новое место, вызывается метод 4.1.2, если изменяется существующее, то вызывается метод 4.1.4

4.11.3.2. Картинка – добавляет/изменяет картинку места (вызывается метод 4.1.7)

### 4.12. 404

4.12.1. Доступно всем

4.12.2. Отображается в случае отсутствия ресурса

### 4.13. 403

4.13.1. Доступно всем

4.13.2. Отображается в случае недоступности ресурса

### 4.14. 401

4.14.1. Доступно всем

4.14.2. Отображается в случае отсутствия авторизац

1. Требования к составу и параметрам технических средств
   1. Сервер
      1. Процессор минимум 2-ядерный с частотой от 2 ГГц.
      2. Оперативная память от 2 Гб.
      3. Место на жестком диске от 10 Гб.
   2. Клиентская часть
      1. Процессор с частотой от 1ГГц.
      2. Оперативная память от 4 Гб.
2. Требования к информационной и программной совместимости
   1. Серверная часть
      1. Операционная система Linux (6.4.12 и выше)
      2. Minio (RELEASE.2022-10-15T19-57-03Z и выше)
      3. Redis (6.2.16 и выше)
      4. Python (3.10 и выше) с фреймворком Django (4.1 и выше)
      5. СУБД PostgreSQL (12.2 и выше)
      6. Node JS
      7. Docker
   2. Веб-браузер (любой из)
      1. Chrome (119.0.6045 и выше)
      2. Firefox (121.0 и выше)

**ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ**

Таблица 1 – HTTP методы разрабатываемого веб-сервиса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Метод** | **Описание** | **URL** | **Входные данные** | **Выходные данные** |
| 4.1.1 | GET | Возвращает все места  Доступно всем | /api/places/ | place\_name=string | [ { placeID: int,  name: string,  description: string,  img: string } ] |
| 4.1.2 | GET | Получает данные места по ID | /api/places/int:place\_id | place\_id: int | { placeID: int,  name: string,  description: string,  img: string } |
| 4.1.3 | PUT | Изменяет данные места (кроме картинки), доступно историк | /api/places/int:place\_id/update/ | place\_id: int,  {  name: string, | { placeID: int,  name: string,  description: string, |
| 4.1.4 | POST | Создает новое место, доступно историк | /api/places/create/ | |  | | --- | | place\_id: int,  { name: string,  description: string  } | | {name: string,  description: string,  img: string  } | | |  | | --- | | { placeID: int,  name: string,  description: string,  img: string } | | { placeID: int,  name: string,  description: string,  img: string } | |
| 4.1.5 | DELETE | Удаляет место, доступно историк | /api/places/int:place\_id/delete/ | place\_id: int | { "status": "deleted", "id": int } |
| 4.1.6 | POST | Добавляет место в экспедицию, доступно историк | /api/places/int:place\_id/add\_to\_expedition/ | place\_id: int |  |
| 4.1.7 | GET | Возвращает все экспедиции  Доступно всем | /api/expeditions/ |  | [ { expeditionID: int,  name: string,  description: string } ] |
| 4.1.8 | GET | Получает данные экспедиции по ID | /api/expeditions/int:expedition\_id/ | expedition\_id: int | { expeditionID: int,  name: string,  description: string } |
| 4.1.9 | PUT | Изменяет данные экспедиции, доступно иторик | /api/expeditions/int:expedition\_id/update/ | expedition\_id: int,  {  name: string,  description: string  } | { expeditionID: int,  name: string,  description: string } |
| 4.1.10 | PUT | Обновляет статус экспедиции для пользователя | /api/expeditions/int:expedition\_id/update\_status\_user/ | expedition\_id: int,  {  status: string  } | { expeditionID: int,  status: string } |
| 4.1.11 | PUT | Обновляет статус экспедиции для админа | /api/expeditions/int:expedition\_id/update\_status\_admin/ | expedition\_id: int,  {  status: string | { expeditionID: int,  status: string } |
| 4.1.12 | DELETE | Удаляет экспедицию, доступно историк | /api/expeditions/int:expedition\_id/delete/ | expedition\_id: int | "reject" } |
| 4.1.13 | GET | Получает место в экспедиции по ID | /api/expeditions/int:epedition\_id/places/int:place\_id/ | expedition\_id: int,  place\_id: int | { placeID: int,  name: string,  description: string  } |
| 4.1.14 | PUT | Обновляет место в экспедиции | /api/expeditions/int:expedition\_id/update\_place/int:place\_id | expedition\_id: int,  place\_id: int,  {data:  [ } | { placeID: int,  name: string, description: string  } |
| 4.1.15 | DELETE | Удаляет место из экспедиции | /api/expeditions/int:expedition\_id/delete\_place/int:place\_id/ | expedition\_id: int,  place\_id: int |  |
| 4.1.16 | POST | Регистрация пользователя. Доступно всем пользователям | /api/users/register/ | {username: string,  password: string,  } | { userID: int,  username: string,  } |
| 4.1.17 | PUT | Аутентификация пользователя. Доступно всем пользователям | /api/users/login/ | { username: string,  password: string  } | {  status: string  } |
| 4.1.18 | POST | Деавторизация, доступна любому аутентифицированному пользователю | /api/users/logout/ |  | { "status": "logged\_out" } |
| 4.1.19 | PUT | Обновляет данные пользователя, доступно аутентифицированному пользователю | /api/users/int:user\_id/update// | user\_id: int,  {  data:date | { userID: int,  } |