|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

Расчетно-пояснительная записка

Тема: «Система организации экспедиций на Дальний восток и Сибирь»

Студент: Смирнов Д.И.

Группа ИУ5Ц-71Б

Преподаватель: Канев А.И.

2023г.

# **ВВЕДЕНИЕ**

В последние годы интерес к исследованиям и путешествиям на Дальний Восток и в Сибирь значительно возрос, открывая новые возможности для исследователей, путешественников и приключенцев. С ростом этого интереса увеличивается и число людей, желающих познакомиться с удивительной природой, культурой и историей этого региона. Учитывая разнообразие интересов путешественников, важно предоставить им удобные и доступные маршруты для путешествий.

Целью нашего проекта является создание системы, помогающей путешественникам подобрать оптимальные маршруты экспедиций на Дальний Восток и в Сибирь. Наша система будет включать в себя веб-сервис, веб-приложение, мобильное приложение и специализированный сервис оценки сложности маршрутов.

Система предназначена для путешественников и организаторов экспедиций. Наша система обеспечивает автоматизированную процедуру создания и учета экспедиций. Организаторы имеют возможность одобрять или отклонять экспедиции, а также редактировать существующие объекты и создавать новые.

Мы стремимся обеспечить путешественникам удобство и возможность легкого выбора маршрутов для исследования Дальнего Востока и Сибири, учитывая их интересы и предпочтения в этом уникальном регионе. Нефункциональные требования к разрабатываемой системе:

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

В ходе работы необходимо выполнить следующие задачи:

* 1. Разработать дизайн приложения.
  2. Создать базу данных в PostgreSQL.
  3. Создать веб-сервис на технологии dotnet 7.0.
  4. Реализовать интерфейс гостя на технологии React.
  5. Задеплоить на Github Pages.
  6. Добавить авторизацию и аутентификацию в веб-сервис.
  7. Реализовать интерфейс путешественника в React.
  8. Реализовать интерфейс организатора в React.
  9. Создать десктопное приложение на React Native.
  10. Создать выделенный сервис для проверки экспедиций в архиве.
  11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.

# **БИЗНЕС-ПРОЦЕСС**

На просторах Дальнего Востока и Сибири проложено множество захватывающих маршрутов [1], которые привлекают внимание исследователей и любителей путешествий. Чтобы избежать ручного поиска попутчиков для совместных путешествий, участники могут создавать экспедиции с определенному географическими объектами. Каждая экспедиция изначально создается как черновик, в который можно добавлять или удалять географические объекты. Один участник может указать несколько маршрутов, по которым хотел бы провести экспедицию, и при этом путешествие будет происходить последовательно по каждому маршруту.

Когда участник определится с географическими объектами [2] для путешествия, он формирует экспедицию, которая отправляется на модерацию. После этого редактирование экспедиции невозможно. Также предусмотрена возможность просмотра истории прошлых экспедиций. Организаторы обрабатывают экспедиции с учетом проверки ее в архиве, уточняя, была ли такая экспедиция в прошлом. Также участники имеют возможность отменить экспедицию, если передумают участвовать.

Экспедиция представляет собой определенные географические локации и места, предназначенные для исследования участниками. Маршруты [2] могут различаться по сложности и количеству объектов, будь то маршрут для пеших походов или масштабная экспедиция для группы из нескольких десятков людей. Характеристики объектов отображаются в их описаниях и изображениях. Для разнообразия путешествий организатор может добавлять новые географические объекты или изменять и удалять существующие. Функции пользователей с разными ролями описаны на диаграммах прецедентов (рис. 1).

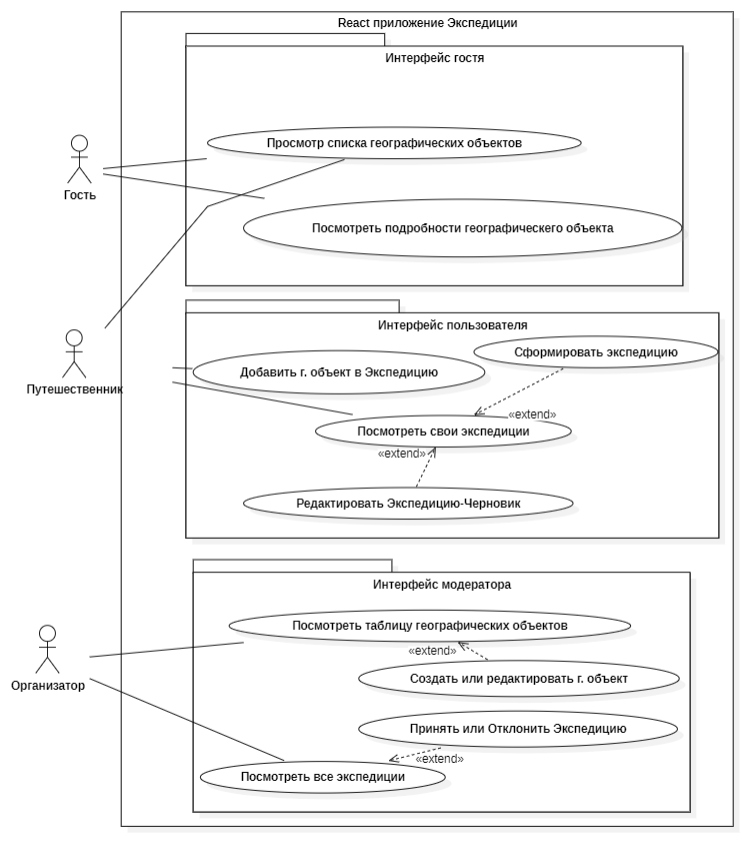


Рисунок 1 - Диаграмма прецедентов

Гостям доступна авторизация и просмотр доступных объектов. Путешественники могут добавлять географическике объекты в экспедицию, просматривать список своих экспедиций и сформировывать и редактировать текущую экспедицию. Экспедиции обрабатываются организаторами. В результате обработки экспедиции её либо одобряют, либо отклоняют. Помимо оформления экспедиции также доступны уникальные функции для работы с объектами, а именно: просмотр всех объектов, редактирование, создание и удаление объектов, а также просмотр списка всех географических объектов в табличном виде. Процесс оформления экспедиции отражен на диаграмме деятельности (рис. 2).

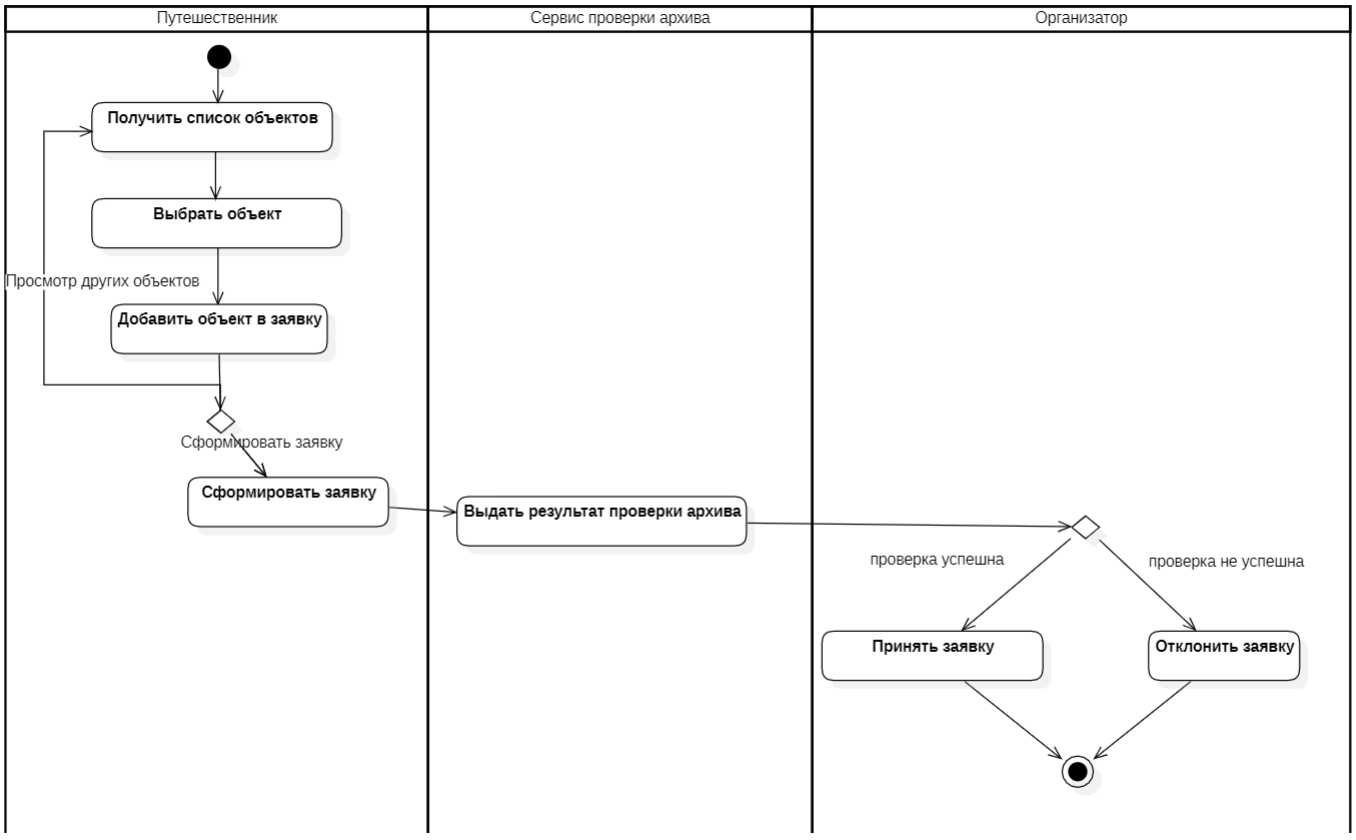


Рисунок 2 - Диаграмма деятельности

Путешественник выбирает объекты, затем формирует на основе выбранных мест экспедицию. Её затем обрабатывает асинхронный сервис, а затем и организатор. В момент формирования экспедиции выполняется проверка данной экспедиции в архиве на асинхронном сервисе. Возможные состояния экспедиции отражены на диаграмме состояний (рис. 3).

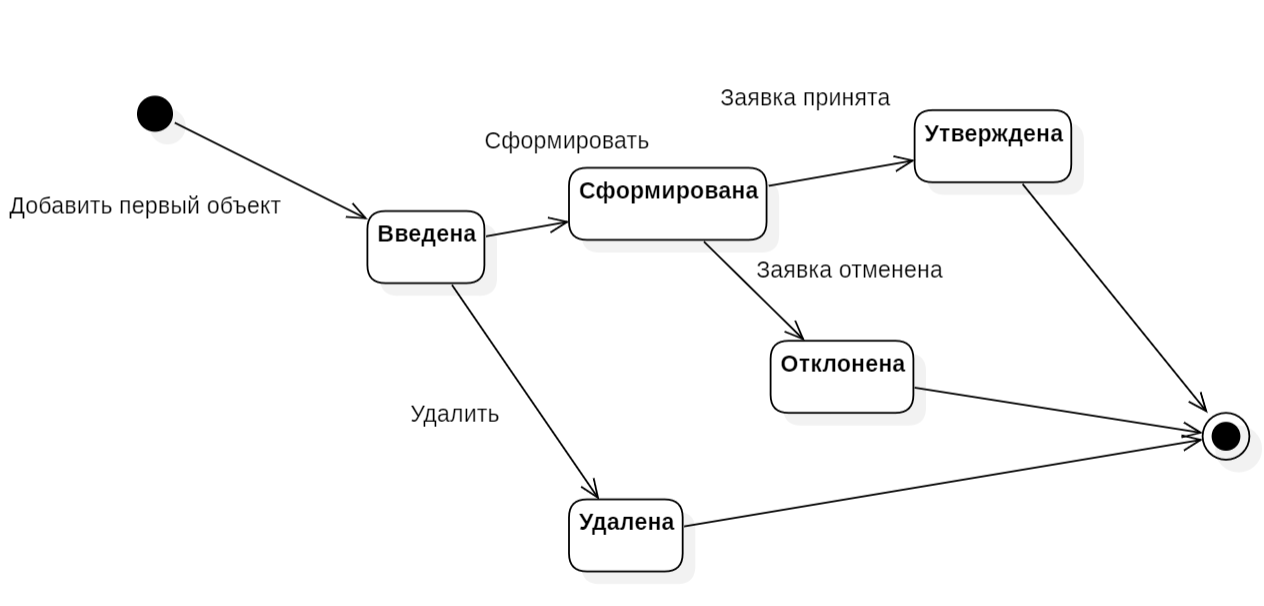


Рисунок 3 - Диаграмма состояний экспедиции

При выборе первого объекта формируется черновик. Последующие выбранные объекты добавляются в этот черновик. Далее путешественник формирует экспедицию или удаляет ее. Сформированную экспедицию обрабатывает организатор, отклоняя или одобряя ее исходя из результатов проверки в архиве.

# **АРХИТЕКТУРА**

Архитектура системы развертывания предполагает использование нескольких технологий для обеспечения функционирования различных компонентов системы. Фронтенд реализован на React [3], обеспечивая пользовательский интерфейс для нативного и браузерного приложений. React [4] используется для построения эффективных и масштабируемых интерфейсов. Веб-сервис, обрабатывающий логику бизнес-процессов, реализован на Django [5], предоставляя API для взаимодействия с фронтендом. Django был выбран из-за своей гибкости, простоты в использовании и мощных инструментов разработки.

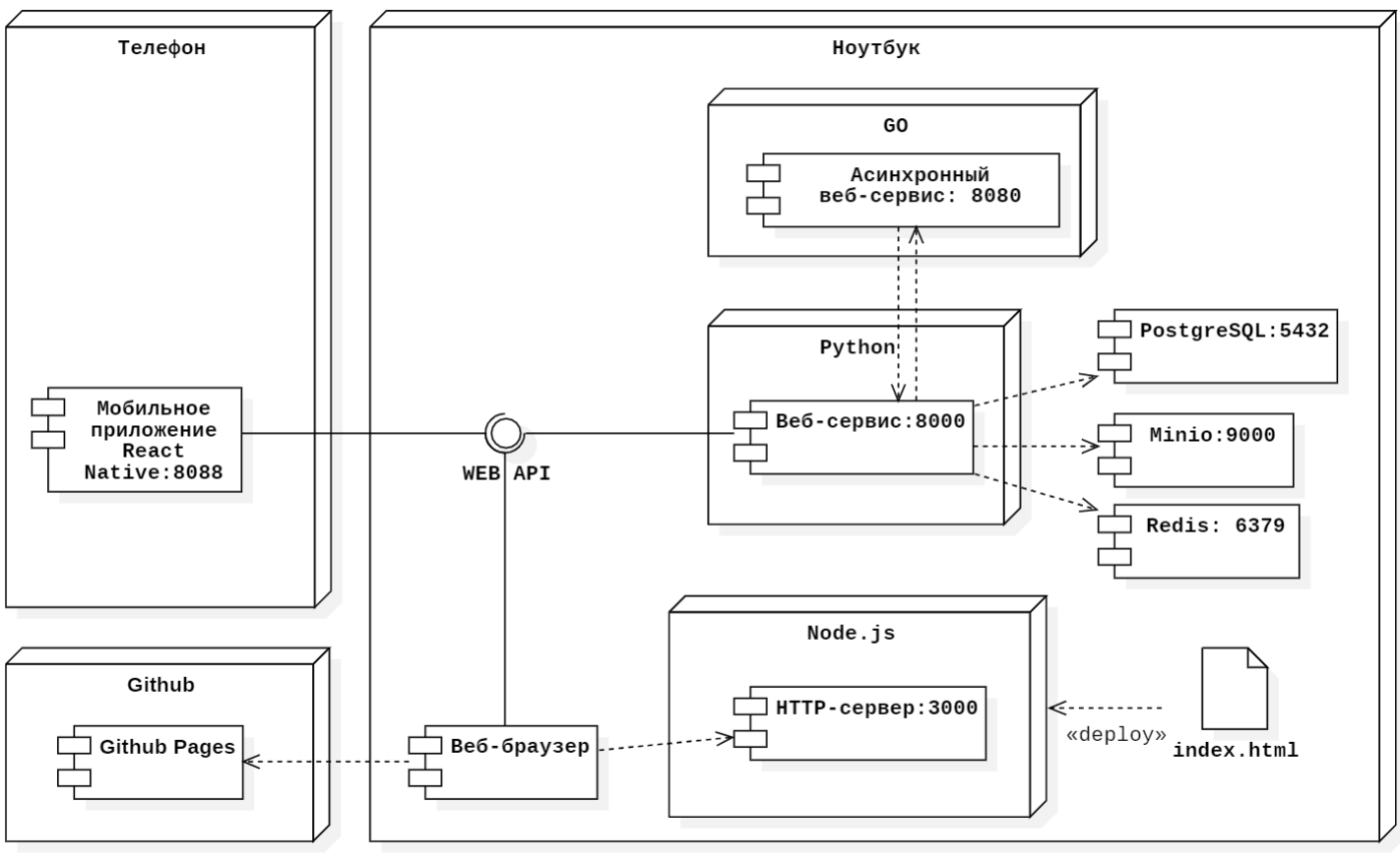


Рисунок 4 - Диаграмма развертывания

Для обработки файлов и объектов используется Minio, облачное хранилище данных, обеспечивающее эффективное хранение и управление большими объемами информации. Для асинхронной обработки задач и управления фоновыми процессами используется асинхронный сервис на языке программирования Go. Этот сервис позволяет выполнять задачи в фоновом режиме, обеспечивая более эффективную работу системы и обработку операций в реальном времени. Кроме того, для управления кэшированием и повышения производительности системы используется Redis, который обеспечивает быстрый доступ к данным и временное хранение информации для оптимизации процессов.

Данные хранятся в СУБД PostgreSQL [6], их структура отражена на ER диаграмме (рис. 5). СУБД PostgreSQL является одним из стандартов индустрии, поэтому было решено использовать её. Структура данных довольна проста. Помимо базовых полей, пользователь также имеет поле Is\_Super, отражающее его статус. Модель объектов представляет собой набор полей, необходимых исключительно для бизнес-логики. Для хранения в одной экспедиций нескольких объектов используется промежуточная таблица ObjExpeditions, реализующая связь М-М.

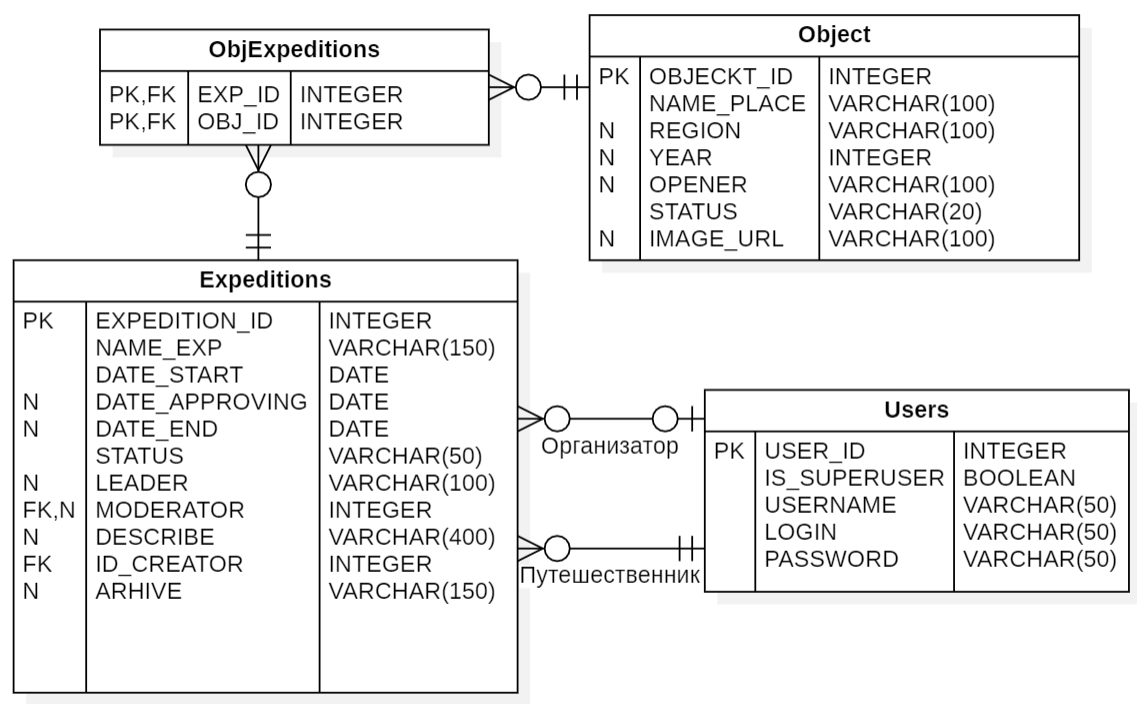


Рисунок 5 - ER диаграмма

Устройство бекенда приложения изображено на диаграмме классов бекенда (рис 6.). Модели имеют связи с таблицами в базе данных. Также модель экспедиции имеет связь с асинхронным сервисом. Объекты имеют связь с сервером Minio, т.к. в объектах хранится ссылка на их изображение, хранимое на сервере статических файлов.

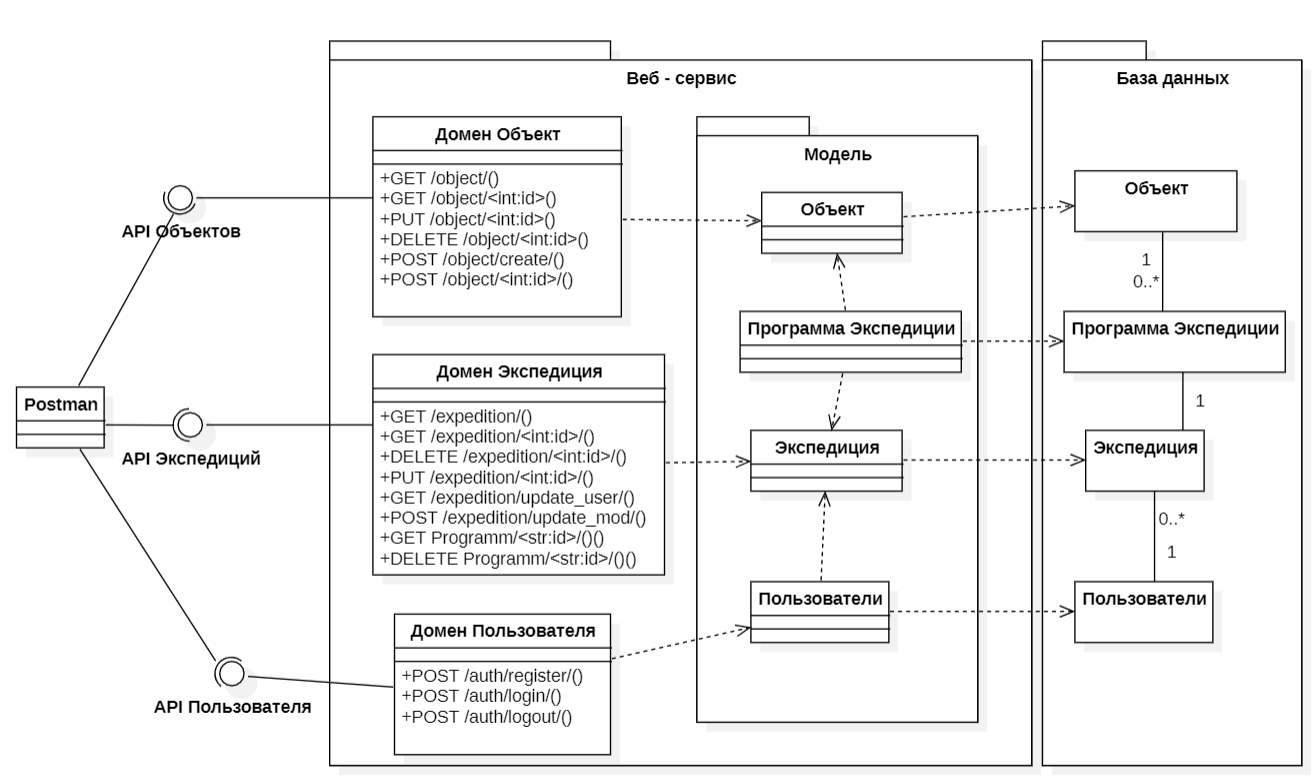


Рисунок 6 *-* Диаграмма классов бекенда

Связь фронтенда и бекенда отражена на диаграмме классов фронтенда (рис. 7). Ключевые страницы имеют связь с API аутентификации, т.к. доступ к ним осуществляется только для авторизированных пользователей с определенными правами (ролями).

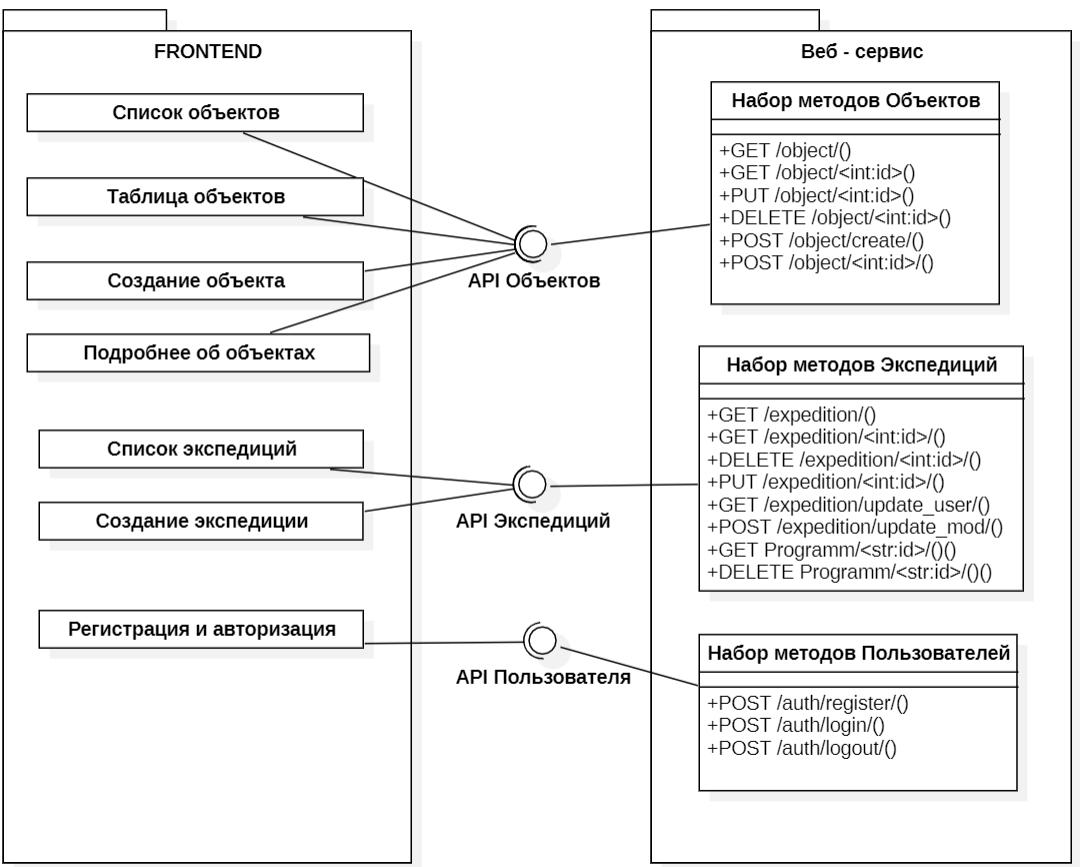


Рисунок 7 *-* Диаграмма классов фронтенда

# **АЛГОРИТМЫ**

Алгоритм работы системы отображен на диаграмме последовательности (рис. 8). В основе системы лежит веб-сервис, реализующий внутри себя всю бизнес-логику. Он предоставляет доступ к методам из следующих доменов: объекты, экспедиции, пользователи. Методы следуют правилам REST API.

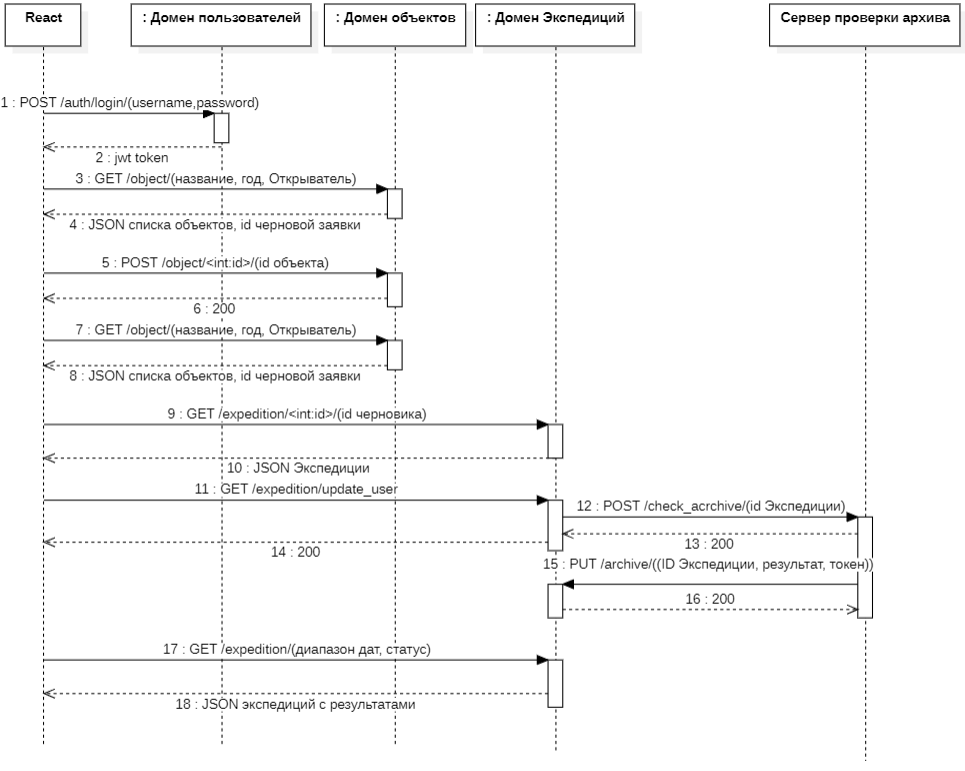


Рисунок 8 - Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. Для этого он отправляет через графический интерфейс запрос, передавая в нем username и пароль. Если аккаунт с такими данными существует, то пользователь получает jwt token 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 географических объектов, которые возвращаются в JSON формате. Путешественник выбирает объект, который хочет включить в экспедицию, и, нажимая на кнопку «добавить» в графическом интерфейсе, отправляет запрос на добавление объекта в свою черновую экспедицию. Этот процесс может продолжаться несколько раз.

Когда путешественник определится с выбором, он нажимает на кнопку «сформировать» в графическом интерфейсе. После этого приложение запрашивает id черновой экспедиции путешественника и затем отправляет запрос на формирование этой экспедиции. В этот момент основной веб-сервис выполняет асинхронный запрос к сервису archive-check на то, чтобы он проверил на существование экспедиции в архиве. Когда экспедиция будет одобрена, путешественник сможет увидеть через некоторое время результат проверки в архиве.

Процесс рассмотрения экспедиций происходит также через графический интерфейс. Организаторы могут просматривать списки всех экспедиций и, нажимая на соответствующие кнопки, отправлять запросы на одобрение или отклонение экспедиций в основной веб-сервис. В эти запросы также можно включить фильтры по диапазону дат и статусу, в котором должны были быть созданы экспедиции. Также организаторы могут управлять непосредственно объектами. Им доступны такие функции, как создание и редактирование объек ов, просмотр списка объектов и удаление их. Для каждой из этих функция присутствует свой метод, отправляемый на основной веб-сервис.

# **ОПИСАНИЕ ИНТЕРФЕЙСА**

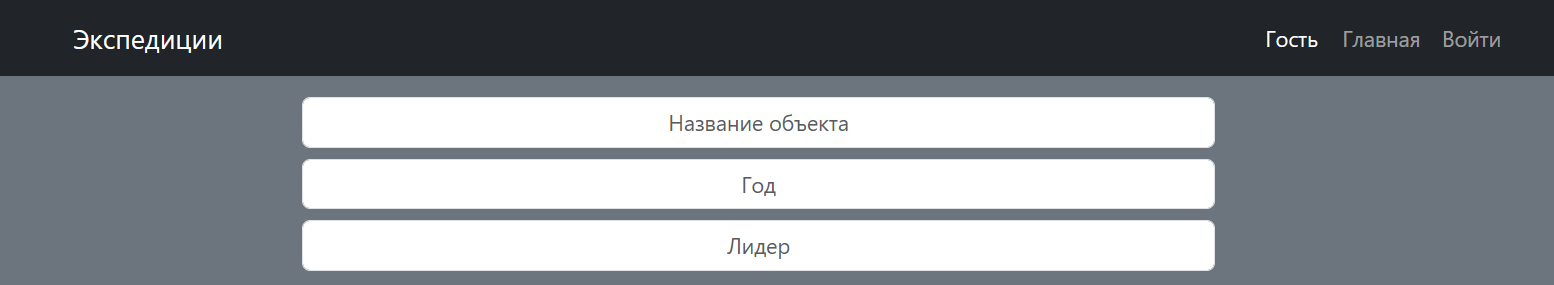
Главное меню приложения представленно в виде шапки сайта и включает пункты, которые доступны в зависимости от роли пользователя (рис. 9, 10). 

Рисунок 9 - Главное меню (неавторизированный пользователь)

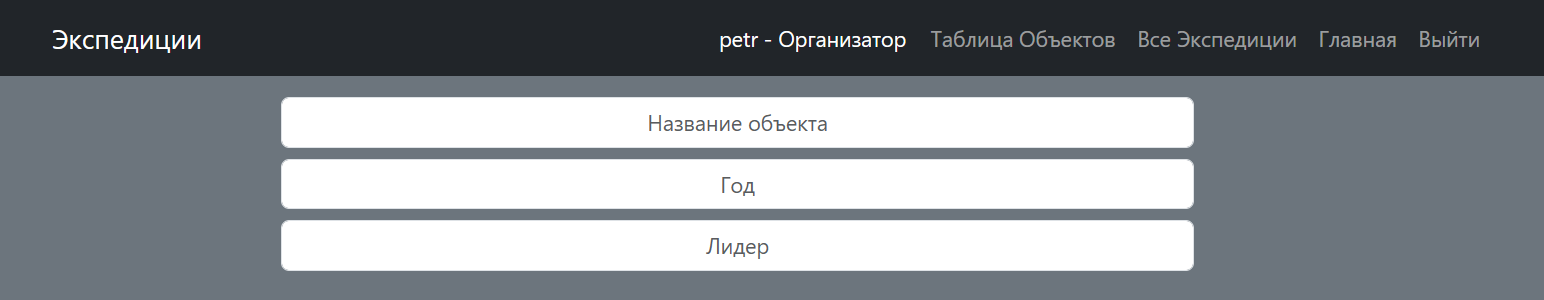
****

Рисунок 10 - Главное меню (организатор)

Первоначальная страница для всех пользователей и гостей. В зависимости от типа пользователя её содержимое меняется. Для гостей, например, там отображаются только кнопки «войти» и «зарегистрироваться», а для путешественника – «список экспедиций», «корзина».

На странице с формой авторизации (рис. 11) отображается форма, через которую гость входит в свой аккаунт. При успешном вводе логина и пароля гость получает JWT, который сохраняется в cookies и используется при отправлении запросов.

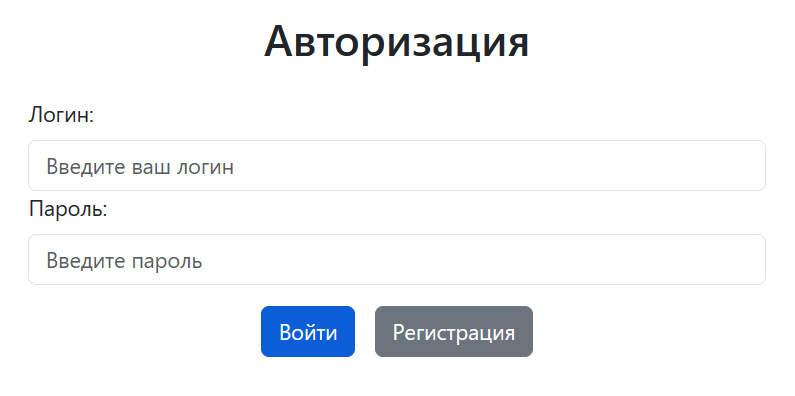
****

Рисунок 11 - Страница авторизации

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

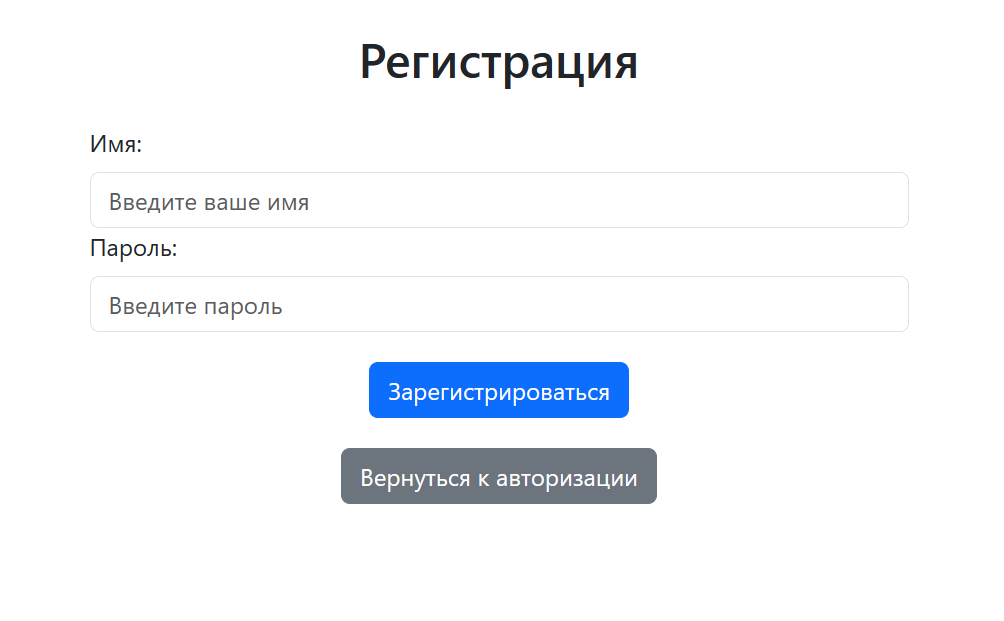
****

Рисунок 12 - Страница регистрации

На странице со списком объектов (рис. 13) отображается список объектов в виде карточек. У каждой карточки есть кнопка «Подробнее», переносящая пользователя на страницу с подробной информацией о карте, и кнопки «Добавить в экспедицию». Сверху находится фильтр, там можно найти необходимый объект по названию, году, открывателю. Страница со списком услуг под логином организатора (рис 14.)

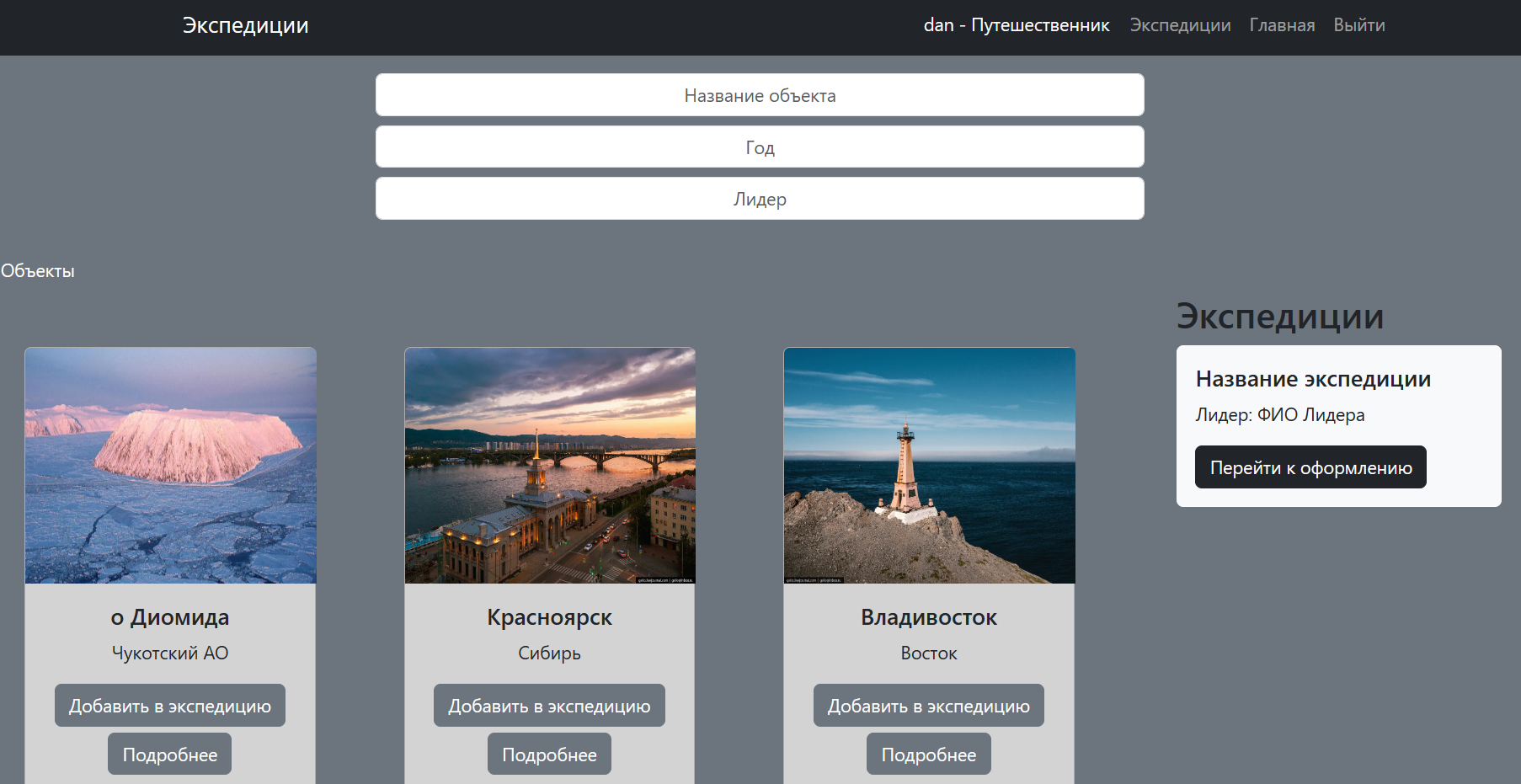
****

Рисунок 13 - Страница со списком объектов

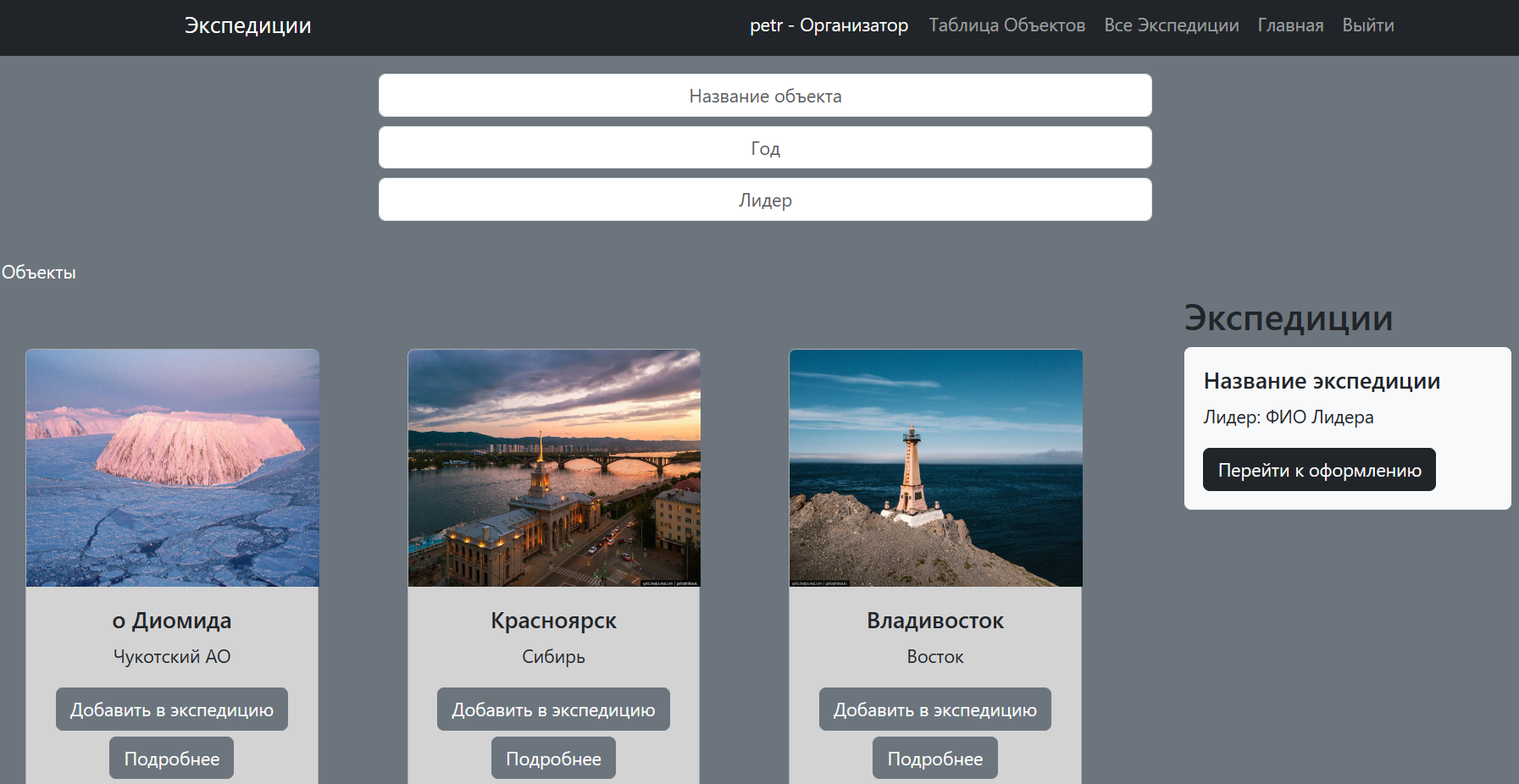
****

Рисунок 14 - Страница со списком объектов(организатор)

На странице с подробным описанием объекта (рис. 15) отображается подробная информация об объекте.

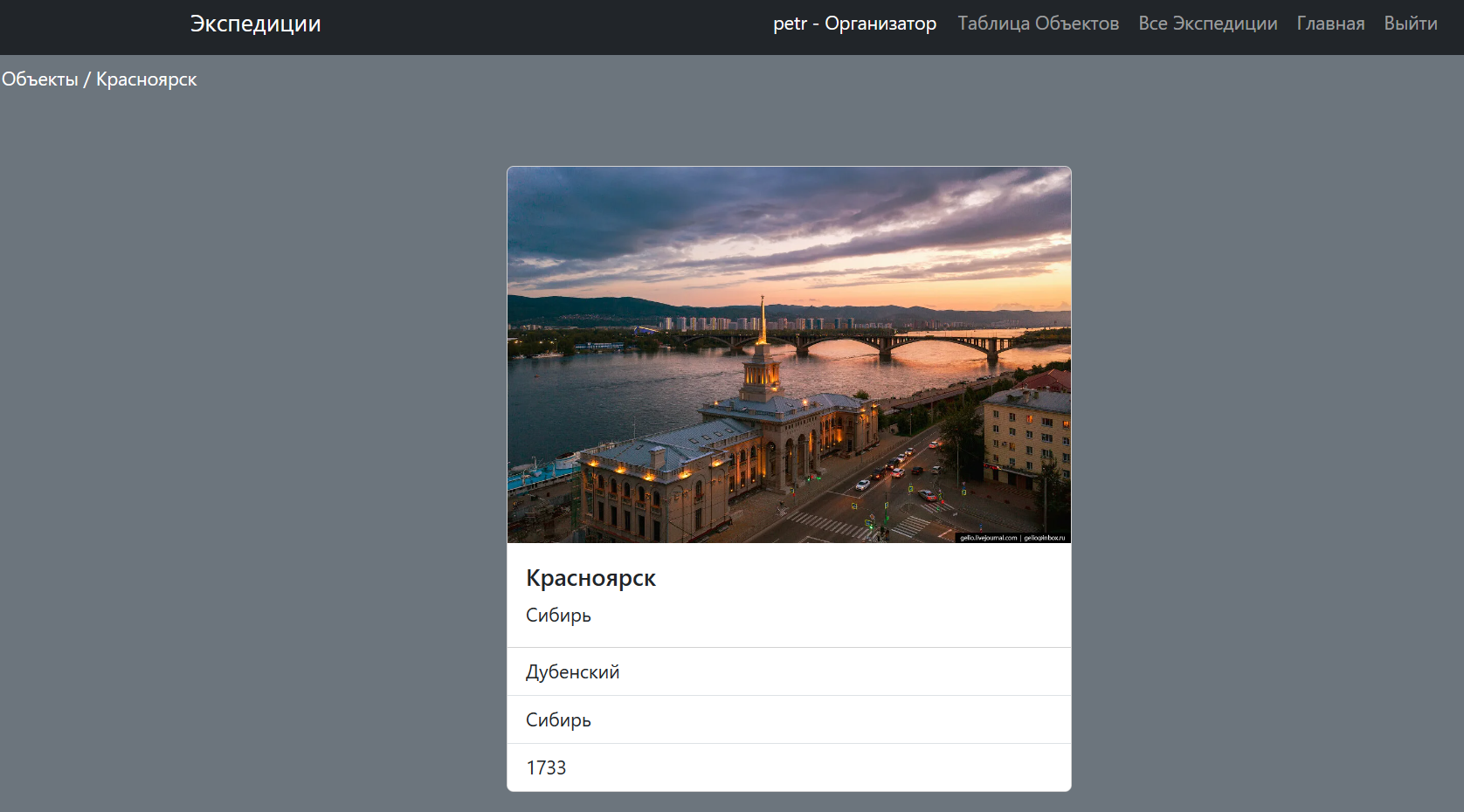
****

Рисунок 15 - Страница с подробным описанием карты

На странице со списком экспедиций (рис. 16, 17) отображается список экспедиций. В зависимости от типа пользователя этот список будет функционально отличаться. Так, для путешественников отображается список созданных ими экспедиций. При нажатии на строку открывается страница выбранной экспедиции.

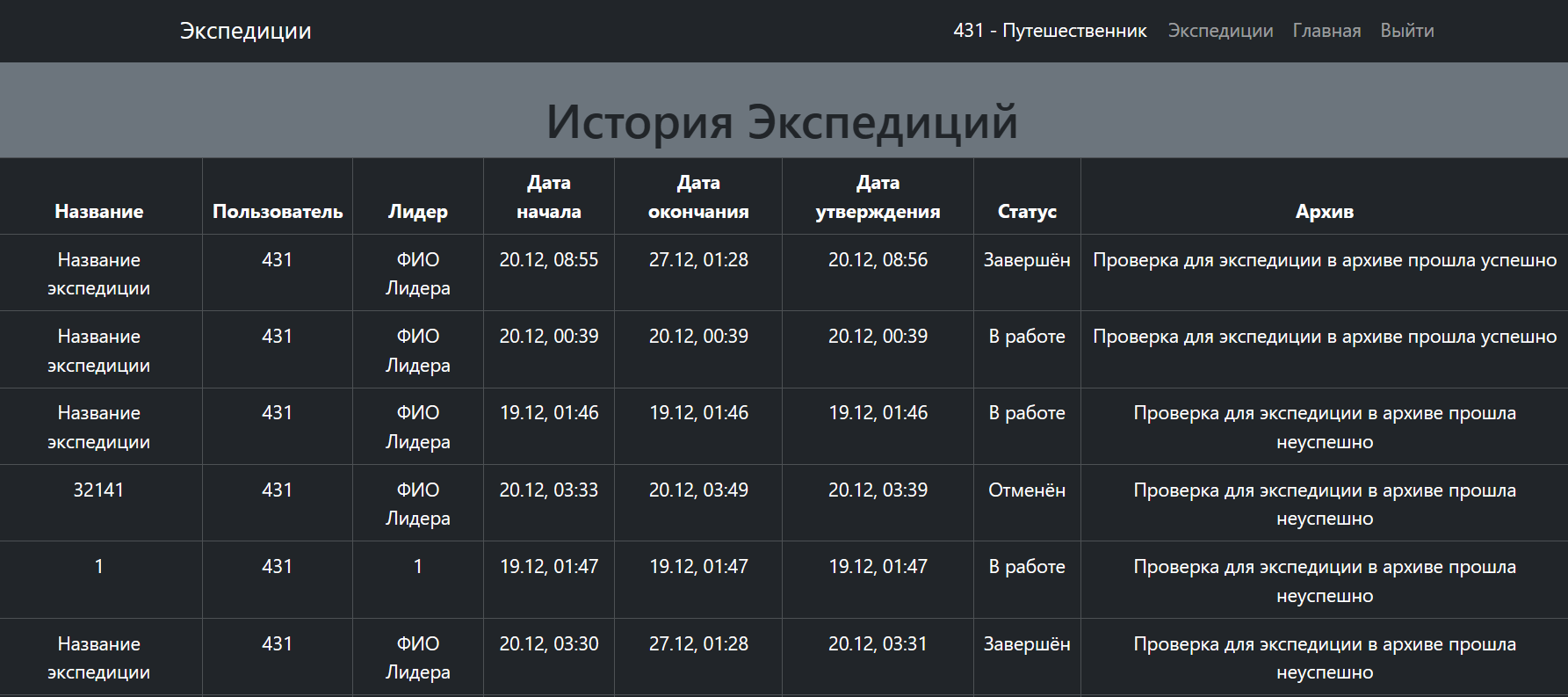
****

Рисунок 16 - Страница со списком экспедиций (путешественник)

Для организатора функции этой страницы шире. Для них отображается список всех экспедиций всех путешественников с более подробной информацией.

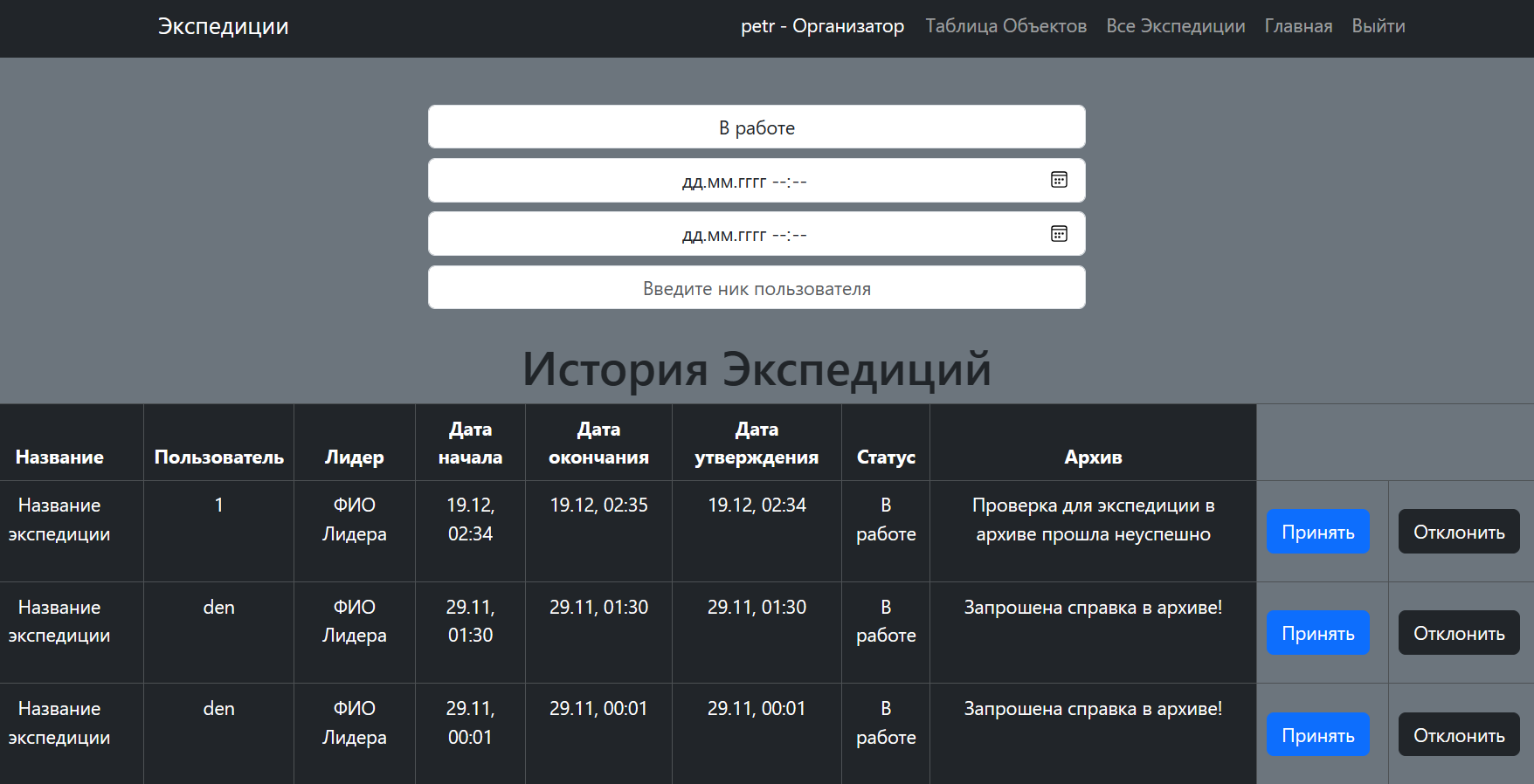
****

Рисунок 17 - Страница со списком экспедиций (организатор)

На странице с подробным описанием экспедиции (рис. 18) отображается подробная информация об экспедиции. Список выбранных карт в виде карточек, а также статус экспедиции.

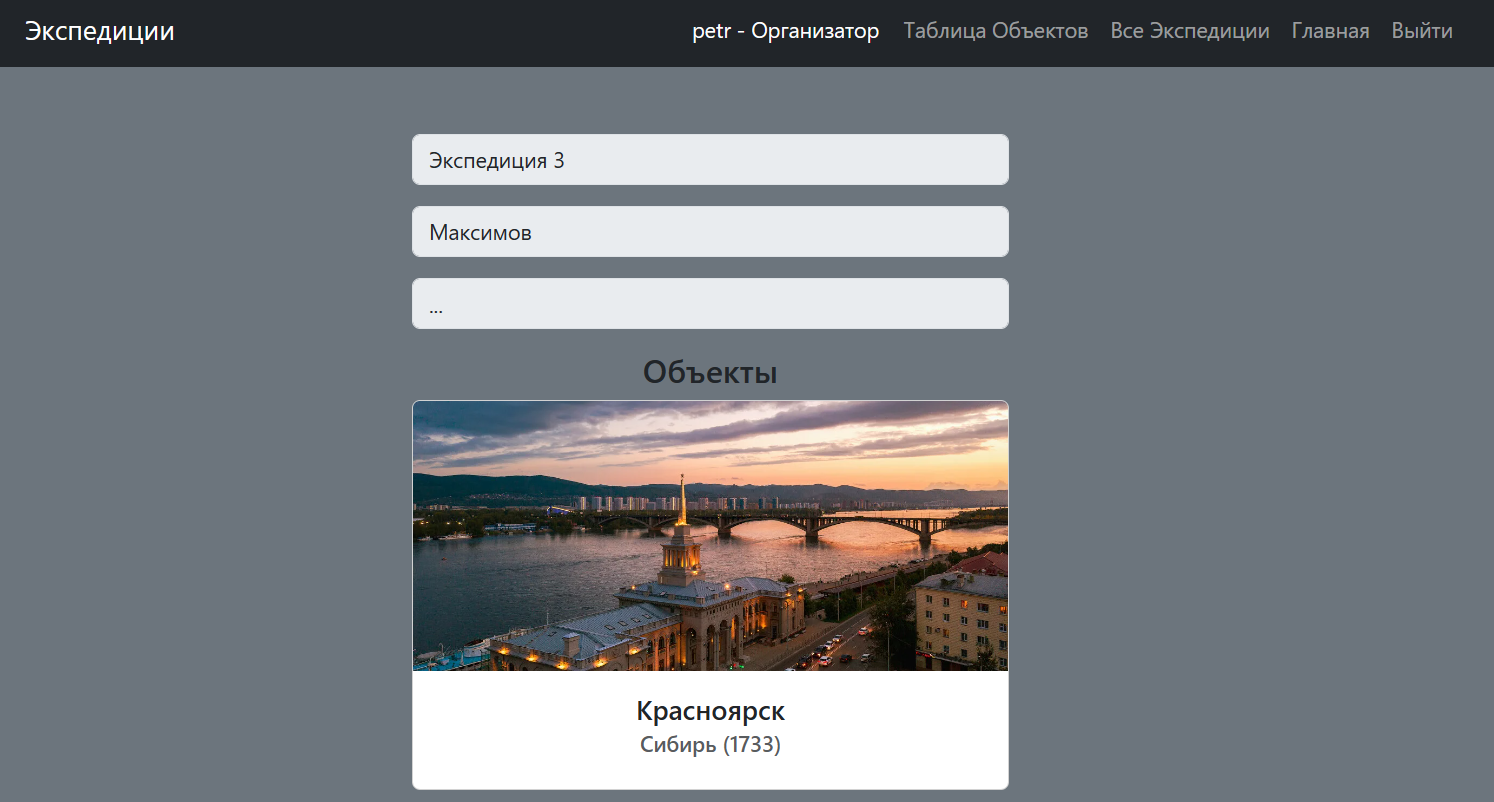
****

Рисунок 18 - Страница с подробным описанием экспедиции

На странице с таблицей объектов (рис. 19) организатор может в компактном и удобном формате просмотреть список всех объектов, существующих в системе. Также можно открыть объект или перейти на страницу создания новой (рис. 20).

****

Рисунок 19 - Страница с таблицей карт

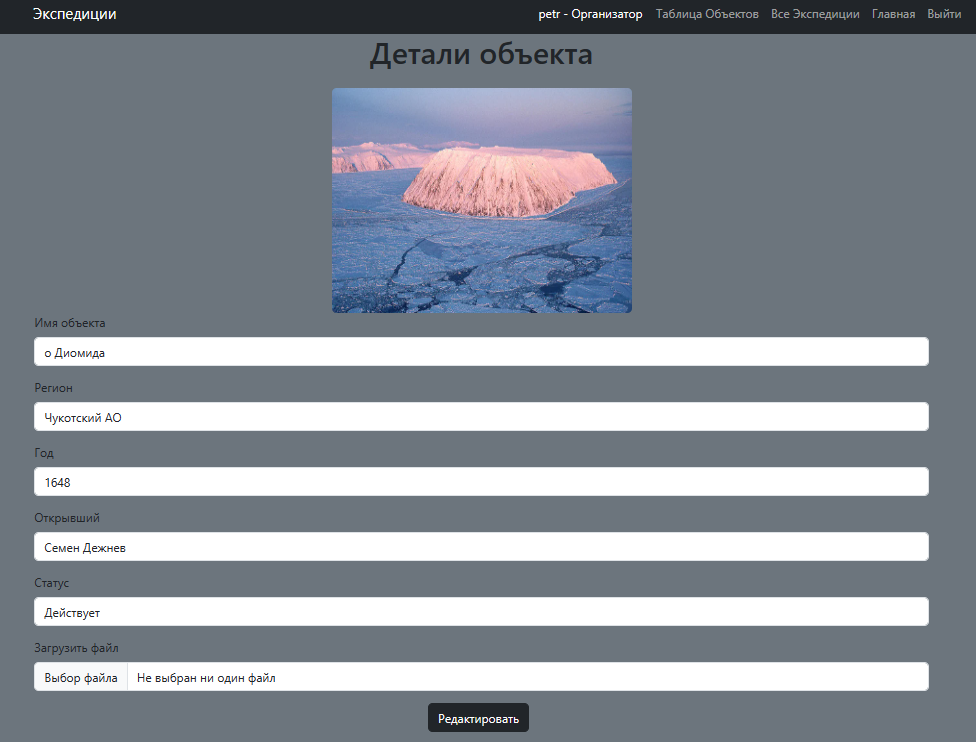
****

Рисунок 20 - Страница редактирования/создания карты

На странице редактирования/создания объекта (рис. 20) организатор может отредактировать существующий, или создать новый объект. Для редактирования доступны все поля. Также можно скрыть или отобразить объект в списке, отображаемом для пользователей.

# **ЗАКЛЮЧЕНИЕ**

В ходе работы были достигнуты следующие результаты:

1. Был разработан дизайн приложения с помощью набора стилей CSS и HTML тегов.
2. Была создана база данных на PostgreSql.
3. Был создан веб-сервис на Django.
4. Разработан интерфейс гостя с использованием технологии React Framework и подключен к веб-сервису.
5. Приложение интерфейса было развернуто на сервисе Github Pages по ссылке https://daniilsmiirnov.github.io/front/.
6. В веб-сервис добавлена авторизация через JWT, а методы задокументированы через Swagger.
7. Реализован интерфейс путешественника.
8. Реализован интерфейс организатора.
9. Выделенный сервис был разработан на Go.
10. Было реализовано десктопное приложение на React Native, повторяющее интерфейс веб-приложения на React.
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор UML диаграмм.
12. Исходный код проекта доступен в GitHub https://github.com/daniilsmiirnov/front/tree/moderator\_page.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Маршруты путешествий по Сибири [Электронный ресурс] // Nomadsclub: https://nomadsclub.ru/articles/378-marshruty-puteshestviy-po-uralu-i-sibiri.html (дата обращения: 30.10.2023).
  2. Достопримечательности Дальнего Востока [Электронный ресурс] // Kp.ru. URL: https://www.kp.ru/russia/dalnij-vostok/dostoprimechatelnosti/ (дата обращения: 30.10.2023).
  3. Руководство по React [Электронный ресурс] // Metanit. URL: https://metanit.com/web/react/ (дата обращения: 12.10.2023).
  4. Quick Start – React [Электронный ресурс] // React. URL: https://react.dev/learn (дата обращения: 12.10.2023).
  5. Django Documentation [Электронный ресурс] // Djangoproject. URL: https://docs.djangoproject.com/en/5.0/ (дата обращения: 10.09.2023).
  6. Руководство по PostgreSQL [Электронный ресурс] // Metanit. URL: https://metanit.com/sql/postgresql/ (дата обращения: 05.10.2023).
  7. Redux Documentation [Электронный ресурс] // Redux.js. URL: https://redux.js.org/tutorials/index (дата обращения: 20.10.2023).

**Приложение. Техническое задание**

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

Техническое задание

Тема: «Система организации экспедиций на Дальний восток и Сибирь»

Студент: Смирнов Д.И.

Группа ИУ5Ц-71Б

Преподаватель: Канев А.И.

2023г.

1. **Цель работы**

Целью итоговой системы является разработка веб-приложения, предоставляющего географические объекты Дальнего Востока и Сибири, а также предлагающего возможность оформления экспедиций в указанные регионы.

1. **Назначение**

Основным назначением данной системы является предоставление удобного интерфейса пользователям с авторизацией, которая позволяет путешественникам получить доступ к определенным функциям и данным в системе, обеспечивая приватность и защиту информации, с базой данных, которая отвечает за хранение и обработку различных данных, таких как экспедиции и информация об объектах, с функционалом просмотра экспедиций и географических объектов. Путешественники могут просматривать информацию о доступных объекты, а также оформлять экспедиции на их покорение.

1. **Задачи**:
   1. Разработка интерфейса системы.
   2. Подключение базы данных Postgres.
   3. Реализация авторизации и аутентификации пользователей на сервере.
   4. Реализация REST-API.
   5. Разработка функционала просмотра и управления экспедициями.
   6. Разработка функционала просмотра и управления географическими объектами.
   7. Разработка фильтрации списка объектов и экспедиций.
   8. Подключение Swagger и его настройка.
   9. Подключение Vite.
   10. Создание асинхронного сервиса для обработки запросов веб-приложения.
   11. Cоздание мобильного приложения с использованием фреймворка React Native.
   12. Создание страницы авторизации и регистрации веб-приложения.
   13. Подключение библиотеки Redux.
   14. Cоздание репозитория на платформе GitHub для хранения и управления исходным кодом проекта.
   15. Разработка документации ТЗ и РПЗ с описанием требований, функциональности, структуры и основных параметров проекта.
2. **Методы веб-сервиса:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Метод** | **Описание** | **Url** | **Входные данные** | **Выходные данные** |
| * 1. Методы авторизации и аутентификации | | | | | | |
| 4.1.1. | POST | Регистрация | /register/ | Username: string  Password: string |  |
| 4.1.2. | POST | Авторизация | /login/ | Username: string  Password: string | jwt cookies |
| 4.1.3. | POST | Выход из cистемы Доступно только путешественникам | /logout/ | jwt cookies |  |
| * 1. Методы объектов | | | | | | |
| 4.2.1. | GET | Получить список всех географических объектов | /object/ | Name\_Obj: string | {expedition\_draft:integer,  [  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: string<URL>  ]} |
| 4.2.2. | POST | Создать новый географический объект  Доступно организатору | /object/сreate/ | jwt cookies  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: File |  |
| 4.2.3. | GET | Получить географический объект по id | /object/<int:id>/ | jwt cookies  id: integer | [  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: string<URL>  ] |
| 4.2.4. | PUT | Изменить географический объект по id  Доступно организатору | /object/<int:id>/ | jwt cookies  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: File | {  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: string<URL>  } |
| 4.2.5. | POST | Добавить географический объект в экспедицию  Доступно путешественникам | /object/<int:id>/ | jwt cookies  id: integer |  |
| 4.2.6. | DELETE | Удалить объект  Доступно организаторам | object/<int:id>/ | jwt cookies  id: integer |  |
| * 1. Методы экспедиций | | | | | | |
| 4.3.1. | GET | Получить список своих экспедиций  Доступно путешественникам | /expedition/ | jwt cookies  Name\_Exp: string,  DateStart: DateTimeField  DateEnd: DateTimeField | [{  ID\_Expedition: integer,  Name\_Exp: string,  DateStart: DateTimeField,  DateEnd: DateTimeField,  DateApproving: DateTimeField ,  Status: string,  Leader: string,  Describe: string,  ID\_Creator: {  Username: string,  },  Objects: [{  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: string<URL>}]  }] |
| 4.3.2. | GET | Получить своей экспедицию по id  Доступно авторизованному пользователю | /expedition/<int:id>/ | jwt cookies  id: int | {  ID\_Expedition: integer,  Name\_Exp: string,  DateStart: DateTimeField,  DateEnd: DateTimeField,  DateApproving: DateTimeField ,  Status: string,  Leader: string,  Describe: string,  ID\_Creator: {  Username: string,  },  Objects: [{  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: string<URL>}]  } |
| 4.3.3. | GET | Изменение статуса экспедиции на сформированную  Доступно путешественникам | /expedition/update\_user/ | jwt cookies |  |
| 4.3.4. | PUT | Изменить свою экспедицию по id  Доступно путешественникам | /expedition/<int:id>/ | {  ID\_Expedition: integer,  Name\_Exp: string,  DateStart: DateTimeField,  DateEnd: DateTimeField,  DateApproving: DateTimeField ,  Status: string,  Leader: string,  Describe: string,  ID\_Creator: {  Username: string,  },  Objects: [{  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: string<URL>}]  } | {  ID\_Expedition: integer,  Name\_Exp: string,  DateStart: DateTimeField,  DateEnd: DateTimeField,  DateApproving: DateTimeField ,  Status: string,  Leader: string,  Describe: string,  ID\_Creator: {  Username: string,  },  Objects: [{  ID\_Object: integer,  Name\_Obj: string,  Region: string,  Year: integer,  Opener: string,  Status: string,  Image\_Url: string<URL>}]  } |
| 4.3.5. | PUT | Изменить статус организатором  Доступно организатору | expedition/<int:id>/update\_mod/ | jwt cookies  id: int  status: string |  |
| 4.3.6. | DELETE | Удаление своей экспедиции путешественником  Доступно путешественникам | /expedition/<int:id>/ | jwt cookies  id: int |  |
| 4.3.7. | PUT | Обновляет результаты проверки экспедиции в архиве, выполненного асинхронным сервисом | /update\_async/ | result:string,  exp\_id:integer,  token:string |  |
| 4.3.8. | DELETE | Удаление объекта из экспедиции путешественником  Доступно путешественникам | /expedition/delete/<str:id>/ | jwt cookies  id: int |  |
| * 1. Методы программы экспедиции | | | | | | |
| 4.4.1. | PUT | Изменить программу экспедиции  Доступно путешественникам | /Programm/<str:id>/ | id: str  {  ID\_Obj: integer,  ID\_Exp: integer  } |  |
| * 1. Методы внешнего сервиса Archive-check | | | | | | |
| 4.5.1. | POST | Метод внешнего сервиса archive check для проверки в архиве | /archive\_check/ | exp\_id: int |  |

1. **Функциональные требования:**
   1. Главное меню.
      1. Доступно всем пользователям.
      2. Действия
         1. Войти – переход на страницу 5.2. Только для гостей.
         2. Зарегистрироваться – переход на страницу 5.3. Только для гостей.
         3. Список экспедиций – переход на страницу 5.5. Только для путешественников.
         4. Таблица экспедиций – переход на страницу 5.5. Только для организаторов.
         5. Таблица объектов – переход на страницу 5.8. Только для организаторов.
   2. Страница авторизации. На этой странице находится форма авторизации.
      1. Доступна гостям
      2. Действия
         1. Войти – производит запрос (метод 4.1.2.), перенаправляет на страницу 5.1
         2. Регистрация – перенаправляет на страницу 5.3.
   3. Страница регистрации. На этой странице находится форма авторизации.
      1. Доступна гостям
      2. Действия
         1. Зарегистрироваться – производит запрос (метод 4.1.1.), в котором передаются введенные данные формы.
         2. Вернуться к авторизации - перенаправляет пользователя на страницу 5.2.
   4. Страница с подробным описанием географического объекта
      1. Доступна всем пользователям.
      2. Выводится информация о географическом объекте в виде карточки (метод 4.2.3).
         1. Название географического объекта.
         2. Регион.
         3. Изображение объекта.
         4. Год открытия
         5. Открыватель
      3. Действия
         1. Назад – перенаправляет на предыдущую страницу.
   5. Страница со списком Экспедиций
      1. Доступна только путешественникам.
      2. Выводится информация об экспедициях в табличном формате (метод 4.3.1).
         1. Название экспедиции.
         2. Дата создания.
         3. Дата формирования.
         4. Дата завершения.
         5. Лидер
         6. Описание
         7. Кем обработана. Только для организаторов.
         8. Список выбранных объектов.
      3. Действия
         1. Открыть экспедицию – перенаправляет на страницу 5.6.
         2. Установить экспедиции статус «отклонена» или «завершена» (метод 5.4.3.). Только для организаторов.
         3. Отфильтровать список экспедиций по диапазону дат и/или по статусу, путешественнику (метод 4.3.1). Только для организаторов.
   6. Страница с подробным описанием экспедиции. Путешественник может изменить объекты в экспедиции. Организатор может просматривать экспедиции всех путешественников.
      1. Доступна только путешeственникам.
      2. Выводится информация об экспедиции в виде карточки (метод 4.3.2).
         1. Название экспедиции.
         2. Лидер
         3. Описание
         4. Кем обработана. Только для организаторов.
         5. Список выбранных объектов.
      3. Действия
         1. Удалить объект из экспедиции (метод 4.3.8.), если она находится в статусе черновика. Только для владельца экспедиции.
         2. Сформировать экспедицию (метод 4.3.3.). Если она находится в статусе черновика. Только для владельца экспедиции.
         3. Удалить Экспедицию (метод 4.3.6.), если она находится в статусе черновика. Только для владельца экспедиции.
         4. Изменить экспедицию (метод 4.3.4).
   7. Страница редактирования/создания объекта. Организатор может изменить существующую или создать новую карту.
      1. Доступна только организаторам.
      2. Выводится информация об объекте в построчном формате (метод 4.2.3).
         1. Название географического объекта.
         2. Регион.
         3. Изображение объекта.
         4. Год открытия
         5. Открыватель
      3. Действия
         1. Изменение полей карты. Можно изменять все вышеперечисленные поля карты (методы 4.2.4.).
         2. Создание объекта(метод 4.2.2).
   8. Страница с таблицей объектов. Предоставляет организатору удобный способ отображения всех объектов.
      1. Доступна только организатору.
      2. Выводит информация об объектах в табличном виде (метод 4.2.1.).
         1. Название географического объекта.
         2. Регион.
         3. Изображение объекта.
         4. Год открытия
         5. Открыватель
      3. Действия
         1. Удалить объект (метод 4.2.6.)
         2. Редактировать объект – перенаправляет на страницу 5.7.
         3. Добавить объект – перенаправляет на страницу 5.7
   9. Страница со списком объектов.
      1. Доступна всем.
      2. Выводится информация о географическом объекте в виде карточки (метод 4.2.3).
         1. Название географического объекта.
         2. Регион.
         3. Изображение объекта.
      3. Действия
         1. Отфильтровать объекты по указанному названию и максимальному количеству игроков (метод 4.2.1.).
         2. Добавить объект в экспедицию (метод 4.2.5.). Только для путешственников.
         3. Подробнее – перенаправляет на страницу 5.4.
         4. Кнопка корзины – неактивна пока нет экспедиции-черновика, при наличии черновика переход на страницу 5.6
2. **Требования к программному обеспечению:**
   1. Серверная часть
      1. ОС: Linux Ubuntu 22.04 /Windows10 22H2
      2. redis 7.2.3-alpine
      3. nginx 1.19.2-alpine
      4. node.js v21.5.0
      5. quay.io minio RELEASE.2022-10-15T19-57-03Z
      6. postgres 14
      7. ubuntu 22.04.3 LTS
   2. Клиентская часть
      1. ОС: Windows10 22H2/MacOS Sonoma/Linux Ubuntu 22.04
      2. Веб-браузер: Safari 11.1+/Chrome 40+/Opera 27+/Firefox 44+
3. **Требования к аппаратному обеспечению:**
   1. Серверная часть
      1. Процессор минимум 2-ядерный с частотой от 2 ГГц.
      2. Оперативная память от 4 Гб.
      3. Место на жестком диске от 2 Гб.
   2. Клиентская часть
      1. Процессор с частотой от 1ГГц.
      2. Оперативная память от 512 Мб.