

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Сервис для планирования и учета путешествий с друзьями  
«TravelWithFriends»  
«Технологии программирования»

09.03.04 Программная инженерия

Информационные системы и сетевые технологии

Зав. кафедрой \_\_\_\_\_ С.Д. Махортов, д. ф.-м. н., профессор

Обучающийся \_\_\_\_\_ Е.С. Воронежская, 3 курс, д/о

Обучающийся \_\_\_\_\_ М.С. Бондарев, 3 курс, д/о

Обучающийся \_\_\_\_\_ В.Г. Деревянко, 3 курс, д/о

Руководитель \_\_\_\_\_ В.С. Тарасов, ст. преподаватель

Руководитель \_\_\_\_\_ Е.Д. Проскуряков, преподаватель

## Содержание

Содержание.....	2
Введение.....	3
1 Постановка задачи.....	4
1.1 Требования к разрабатываемой системе .....	4
1.1.1 Функциональные требования .....	4
1.1.2 Нефункциональные требования .....	5
1.2 Требования к архитектуре.....	5
1.3 Задачи, решаемые в процессе разработки .....	6
2 Анализ предметной области .....	7
2.1 Терминология (гlossарий) предметной области .....	7
2.2 Обзор аналогов.....	11
2.2.1 Troupe .....	11
2.2.2 Plan Harmony .....	11
2.2.3 MiTravel.....	12
2.2.4 Сравнительная таблица аналогов.....	13
2.3 Диаграммы, иллюстрирующие работу системы.....	13
2.3.1 Диаграмма прецедентов (Use case) .....	13
2.3.2 Диаграмма последовательности (Sequence diagram).....	17
2.3.3 Диаграмма состояний (Statechart diagram) .....	18
2.3.4 Диаграмма деятельности (Activity diagram).....	19
2.3.5 Диаграмма объектов (Object diagram).....	21
2.3.6 Диаграмма сотрудничества (Collaboration diagram).....	21
2.3.7 Диаграмма развертывания (Deployment diagram).....	22
2.3.8 ER-диаграмма .....	22

## **Введение**

В современном мире путешествия становятся все более доступным способом проведения свободного времени.

Согласно отчету World Tourism Organization, в 2023 году количество международных поездок превысило 1,4 миллиарда, демонстрируя восстановление туристической индустрии после пандемии COVID-19. В 2024 году количество поездок по России должно составить 80 миллионов. 69% путешественников предпочтут приобретать билеты и бронировать отели без посредников. При этом 59% россиян будут ездить только по России. К 2027 году количество путешествий по стране должно достигнуть отметки в 103 миллиона.

Совместные поездки могут быть сложным организационным испытанием, особенно когда речь идет о планировании маршрутов и учете финансов. Статистика свидетельствует о том, что все больше людей обращаются к путешествиям в кругу друзей как к предпочтительному варианту отпуска, особенно в летнее время. Проблема эффективного планирования и учета финансов при совместных путешествиях часто ведет к напряженным отношениям между участниками и даже к разногласиям после возвращения домой.

В данной курсовой работе был реализован сервис для организации и планирования совместных путешествий, что включает в себя составление плана поездки, визуализацию маршрутов на карте, ведение расходов и построение статистики на их основе.

## **1 Постановка задачи**

Целью данного проекта является создание сервиса совместных путешествий. Этот сервис предоставит клиентам возможность составить оптимальный маршрут, доступный каждому участнику поездки. Каждый участник поездки сможет внести в список трату, а также указать участников этой траты. После чего каждый участник сможет посмотреть наглядную статистику расходов.

### **1.1 Требования к разрабатываемой системе**

#### **1.1.1 Функциональные требования**

К разрабатываемому сервису выдвигаются следующие функциональные требования:

- просмотр общей информации опубликованных поездок;
- создание и планирование поездки авторизованным пользователем;
- отслеживание маршрута, проходящего через ключевые точки авторизованным пользователем;
- добавление расходов в путешествие авторизованным пользователем;
- просмотр статистики расходов за путешествие авторизованным пользователем;
- сохранение истории путешествий авторизованным пользователем;
- оформление подписки авторизованным пользователем;
- управление подпиской администратором;
- удаление поездок и пользователей администратором.

### 1.1.2 Нефункциональные требования

К разрабатываемому сервису выдвигаются следующие нефункциональные требования:

- сервис должен обладать интерфейсом, выполненном в едином стиле со всем необходимым набором функций;
- сервис должен использовать современные технологии и инструменты разработки.

### 1.2 Требования к архитектуре

Список требований к архитектуре:

- Приложение должно быть построено с использованием протоколов HTTP.
- Для хранения информации необходимо использовать реляционную базу данных.
- Клиентская часть приложения должна быть написана с использованием технологий frontend разработки, таких как HTML, CSS, TypeScript с фреймворком React. Выбор этого фреймворка объясняется тем, что он обладает простым синтаксисом, позволяет обновлять только те элементы, которые требуют изменений и использовать повторно уже существующие элементы.
- Серверная часть приложения должна быть написана с использованием технологий backend разработки, таких как язык программирования C# и фреймворк ASP.Net Core [1]. Этот кроссплатформенный фреймворк обладает высокой производительностью и эффективностью благодаря использованию языка C# и оптимизациям внутри платформы. Имеет

интегрированную поддержку для разработки RESTful API. Обеспечивает широкий спектр инструментов и библиотек для разработчиков, включая Entity Framework для работы с базами данных, инструменты автоматизации развертывания.

### **1.3 Задачи, решаемые в процессе разработки**

Процесс организации данного веб-приложения построен на основе гибкой методологии Kanban.

В процессе разработки сервиса планирования совместных путешествий будут решаться следующие задачи:

- Анализ предметной области: необходимо изучить специфику планирования совместных путешествий.
- Проектирование базы данных: на основе полученных требований необходимо разработать структуру базы данных, которая будет использоваться в приложении.
- Разработка серверной части приложения: на этом этапе необходимо разработать серверную часть приложения, которая будет отвечать за обработку запросов клиента и взаимодействие с базой данных. Для этого используется фреймворк ASP.Net Core.
- Разработка клиентской части приложения: клиентская часть приложения должна быть написана с использованием современных технологий frontend разработки, таких как HTML, CSS, TypeScript.
- Тестирование и отладка: на этом этапе необходимо провести тестирование и отладку приложения, чтобы убедиться, что оно соответствует требованиям, определенным в начале проекта.

## 2 Анализ предметной области

### 2.1 Терминология (гlossарий) предметной области

**.NET** — модульная платформа для разработки программного обеспечения с открытым исходным кодом.

**ASP.NET Core** — свободно-распространяемый кроссплатформенный фреймворк для создания веб-приложений на платформе .NET с открытым исходным кодом.

**Backend** — часть программного обеспечения, отвечающая за обработку данных и взаимодействие с базой данных.

**C#** — объектно-ориентированный язык программирования общего назначения.

**Frontend** — часть программного обеспечения, отвечающая за визуальное представление данных и взаимодействие с пользователем.

**TypeScript** — скриптовый (сценарный) язык программирования, используемый для создания интерактивных веб-страниц [2].

**HTTP** — протокол передачи данных в сети Интернет, который используется для передачи информации между клиентом и сервером.

**HTTPS** — защищенная версия протокола HTTP, использующая шифрование для безопасной передачи данных.

**REST API** — архитектурный стиль веб-служб, который использует протокол HTTP для передачи данных между клиентом и сервером.

**SQL** — язык структурированных запросов, используемый для взаимодействия с базами данных.

**Автоматизированная система** — программа или набор программ, предназначенных для выполнения задач без прямого участия человека.

**Авторизация** — процесс предоставления пользователю или группе пользователей определенных разрешений, прав доступа и привилегий в компьютерной системе.

**Аутентификация** — процесс проверки подлинности пользователя, предоставляющего учетные данные (логин и пароль) для доступа к системе или сервису.

**Администратор** — лицо, ответственное за управление и обслуживание системы.

**Адрес электронной почты** — уникальный идентификатор, используемый для регистрации пользователя в системе.

**Аккаунт** — персональная учетная запись пользователя, которая позволяет ему получить доступ к определенным ресурсам или функциям в рамках системы или сервиса.

**Активность** — любое действие или занятие, которое человек совершает или планирует совершить во время своего пребывания в поездке. Может включать в себя различные виды деятельности, такие как экскурсии, посещение достопримечательностей, пешие или велосипедные прогулки, занятия спортом, походы, культурные мероприятия и другие формы отдыха или развлечений.

**База данных (БД)** — организованное совокупность данных, обычно хранящихся и обрабатываемых с использованием компьютерных систем.

**Библиотека** — набор функций или классов, предназначенных для решения определенной задачи или облегчения разработки программного обеспечения.



**Браузер** — программное обеспечение для просмотра веб-страниц и других ресурсов в Интернете.

**Веб-приложение** — программное обеспечение, доступное через браузер и предназначенное для выполнения определенных функций через Интернет.

**Денежные траты** — сумма денег, которую человек использует для покупки товаров или услуг.

**Маршрут** — путь, который следует пройти или проехать от одной точки (начальной) к другой (конечной).

**Отладка** — процесс поиска и исправления ошибок в программном коде.

**Пароль** — секретная комбинация символов, используемая для аутентификации пользователя и обеспечения безопасности данных.

**Поездка** — путешествие с целью отдыха, туризма, включает в себя формирование плана, определение дат начала и окончания.

**Программное обеспечение (ПО)** — совокупность программных инструкций, данные и документации, предназначенных для работы компьютерной системы или устройства.

**Пользователь** — авторизованный в системе человек, использующий веб-приложение.

**Гость** — неавторизованный в системе человек, использующий часть функционала веб-приложения.

**Расходы** — денежные траты, совершенные с той или иной целью.

**Сервер** — компьютер или программа, предоставляющая ресурсы или услуги другим компьютерам или программам.

**Система управления базами данных (СУБД)** — программное обеспечение для управления базами данных.

**Участник поездки** — пользователь, который принимает участие в поездке вместе с другими пользователями.

**Фреймворк** — набор библиотек и инструментов, облегчающих разработку программного обеспечения.

**Header (хедер)** — это верхняя часть веб-страницы, которая содержит логотип компании, навигационные ссылки и другие элементы для управления и навигации по сайту.

## 2.2 Обзор аналогов

### 2.2.1 Troupe

Troupe – приложение для планирования групповых поездок. В troupe есть различные разделы для облегчения планирования, такие как опросы, проживание, мероприятия, маршрут. Интерфейс удобный и понятный. Есть интеграция с рядом отелей за рубежом. Из недостатков можно выделить отсутствие учета расходов и отсутствие истории путешествий. Интерфейс приложения представлен на Рисунке 1.

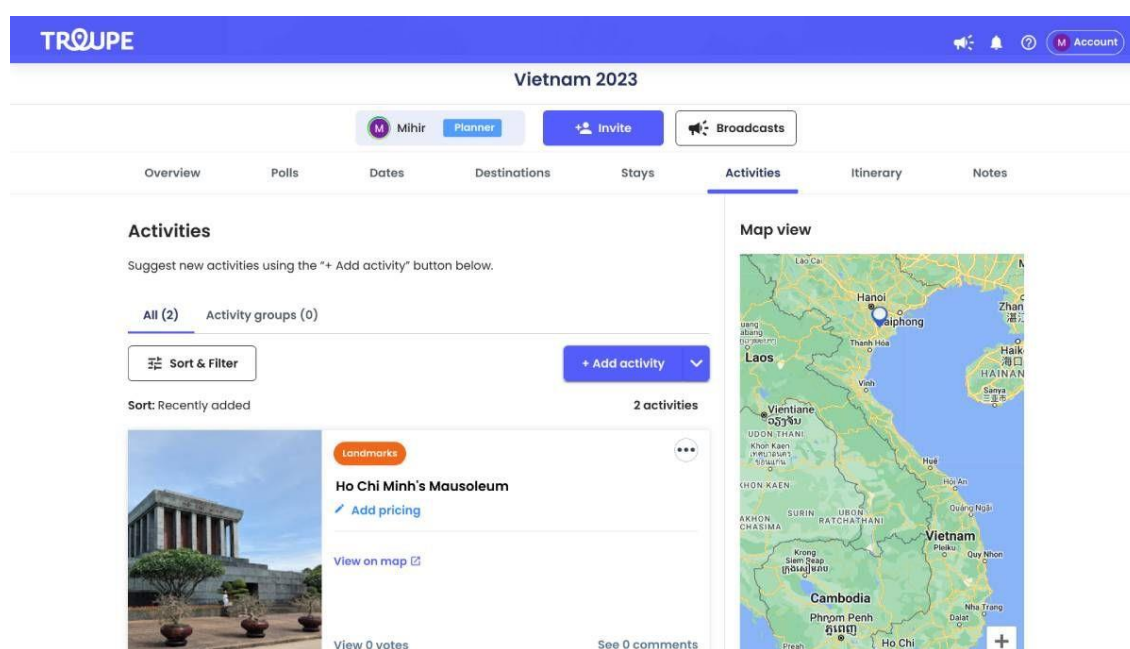


Рисунок 1 - Внешний вид добавления мероприятия «Troupe»

### 2.2.2 Plan Harmony

Plan Harmony – календарное планирование и отслеживание бюджета для групповых поездок. Plan Harmony позволяет вам создавать несколько поездок, в которые можно пригласить друзей, просто поделившись ссылкой, после чего вы можете совместно их спланировать. После того как вы установили даты поездки, по умолчанию отображается календарь, поэтому вы планируете, как если бы вы использовали общий Календарь Google. Из

преимуществ, это приложение позволяет вести совместный учет расходов, предоставляет статистику расходов. Из недостатков, нет визуализации маршрутов. Интерфейс приложения представлен на Рисунке 2.

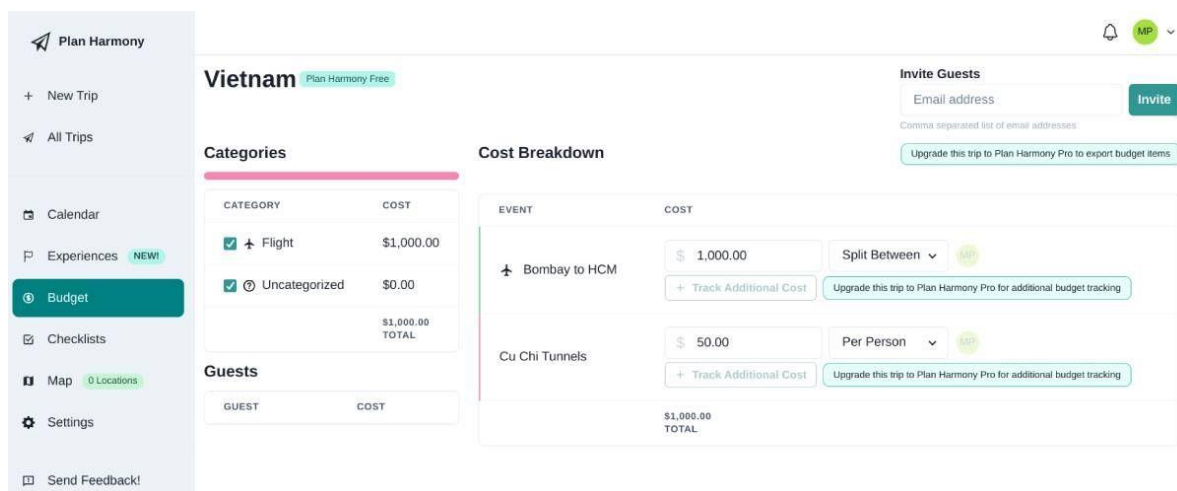


Рисунок 2 - Внешний вид добавления мероприятия «Plan Harmony»

### 2.2.3 MiTravel

MiTravel – приложение, использующее систему канбан-досок для планирования путешествий, на которой вы вместе с другими добавляете различные элементы для вашей совместной поездки и даже проводите опросы для быстрого принятия решений. Лучше всего? Это совершенно бесплатно, без каких-либо скрытых затрат. Интерфейс приложения представлен на Рисунке 3.

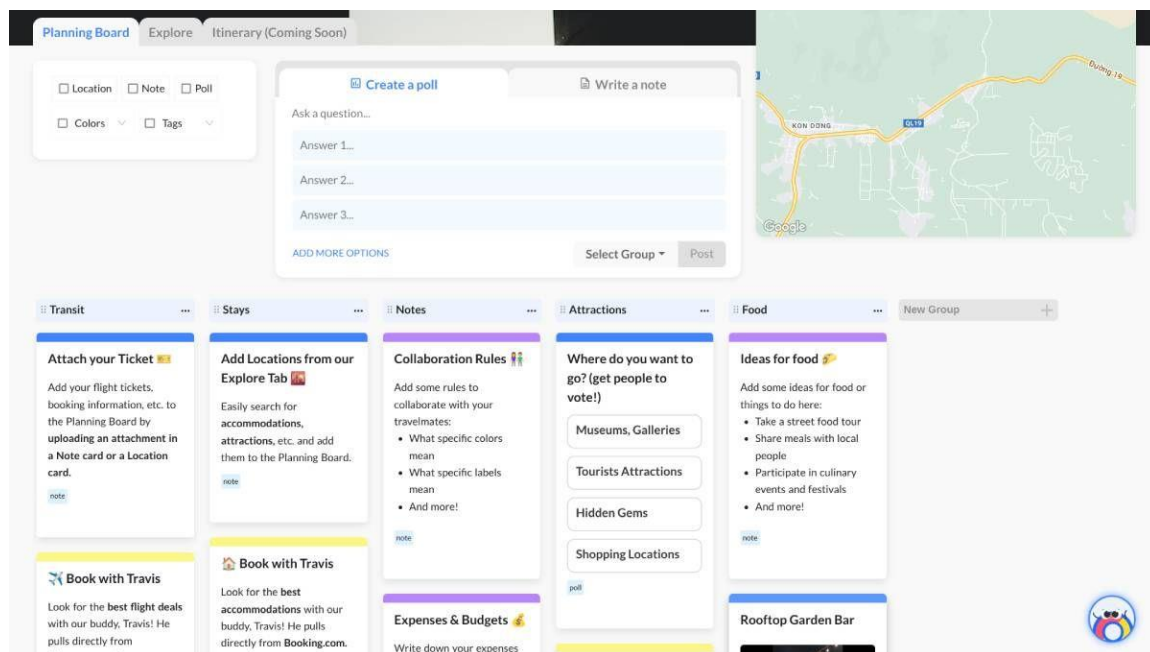


Рисунок 3 - Внешний вид доски «MiTravel»

## 2.2.4 Сравнительная таблица аналогов

Для удобства восприятия здесь представлена сравнительная таблица аналогов.

	TRUPE	Plan Harmony	mitrovel Beta
Карта, для отображения маршрута	+	-	+
Учет и статистика расходов	-	+	-
Встроенный чат	+	+	-
История путешествий	-	+	-
Удобный интерфейс	+	+	-

Рисунок 4 - Обзор аналогов

## 2.3 Диаграммы, иллюстрирующие работу системы

### 2.3.1 Диаграмма прецендентов (Use case)

Диаграмма прецедентов (Use case) представлена для трёх типов акторов: неавторизованного пользователя, авторизованного пользователя, администратора. У каждого из них своя модель поведения, которую можно проследить на Рисунках 5-7.

Неавторизованный пользователь может:

- зарегистрироваться в системе;
- авторизоваться в системе;
- просматривать опубликованные путешествия;
- просматривать главную страницу.

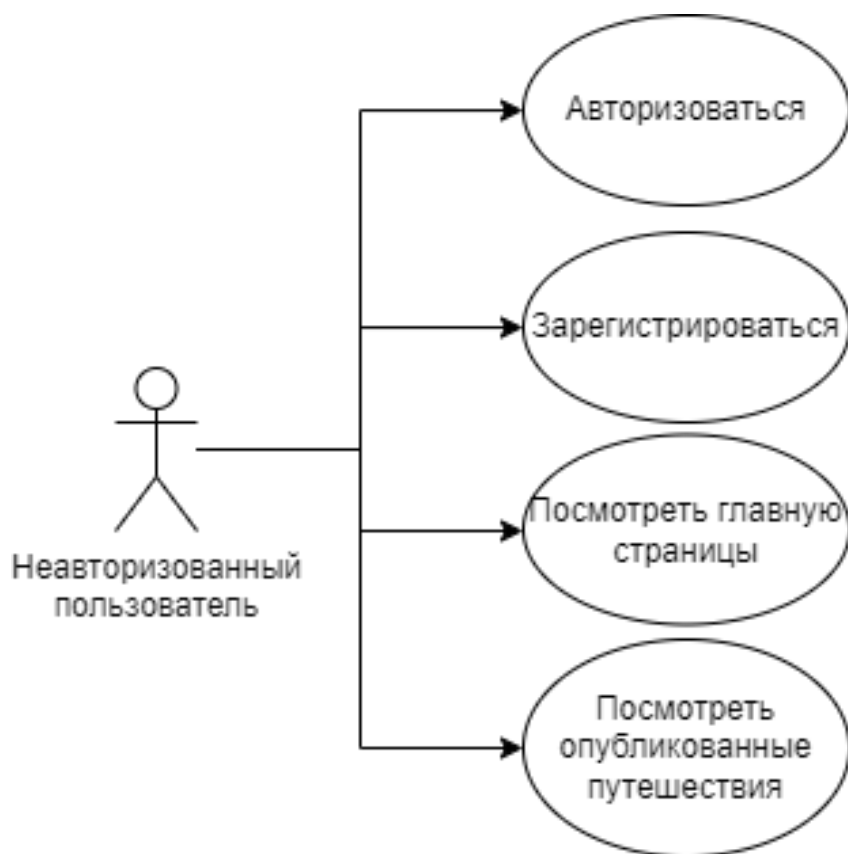


Рисунок 5 - Диаграмма прецедентов (Use case) для неавторизованного пользователя

Авторизованный пользователь помимо функций, доступных неавторизованному пользователю, может:

- создавать путешествия: добавлять пользователей в команду, добавлять даты начала и окончания поездки;
- формировать список активностей, включая просмотр маршрута между точками активностей, внесение трат на активность, указание членов команды, участвовавших в данной активности;
- просмотреть статистику расходов за поездку;
- взаимодействовать с личным кабинетом: посмотреть личную информацию, указанную на сайте, посмотреть историю путешествий;
- оформить подписку.

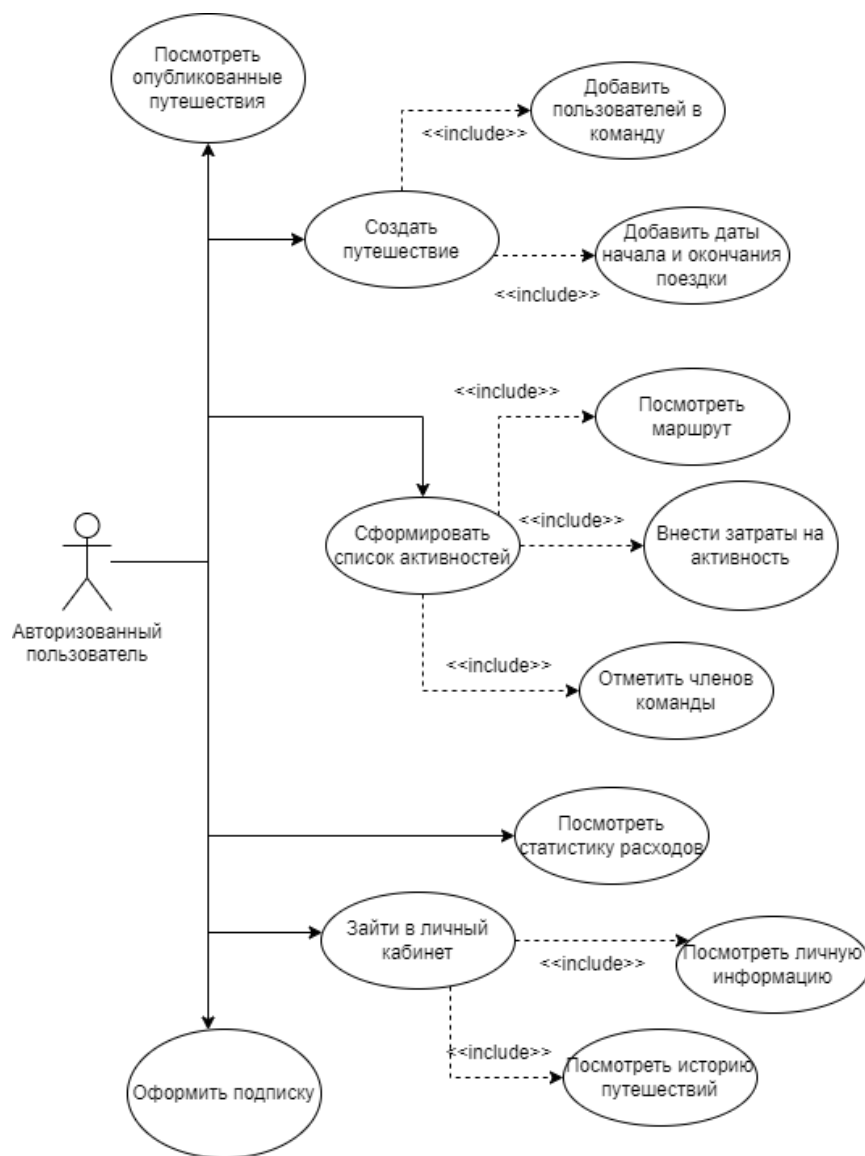


Рисунок 6 - Диаграмма прецедентов (Use case) для авторизованного пользователя

Существуют следующие функции, доступные администратору:

- удалить пользователей;
- управлять подпиской пользователей;
- удалить любое путешествие;
- просматривать страницу опубликованных путешествий.



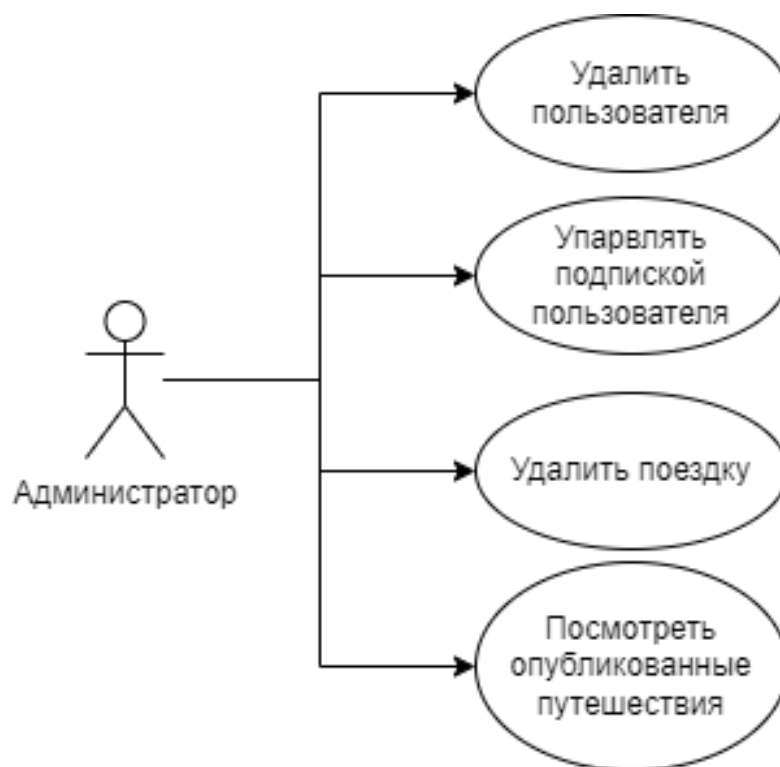


Рисунок 7 - Диаграмма прецедентов (Use case) для администратора

### 2.3.2 Диаграмма последовательности (Sequence diagram)

Существует также диаграмма последовательностей (Рисунок 8), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта и взаимодействие актеров информационной системы в рамках прецедента. Участником данной системы является пользователь, а объектами – клиент, сервер и база данных.

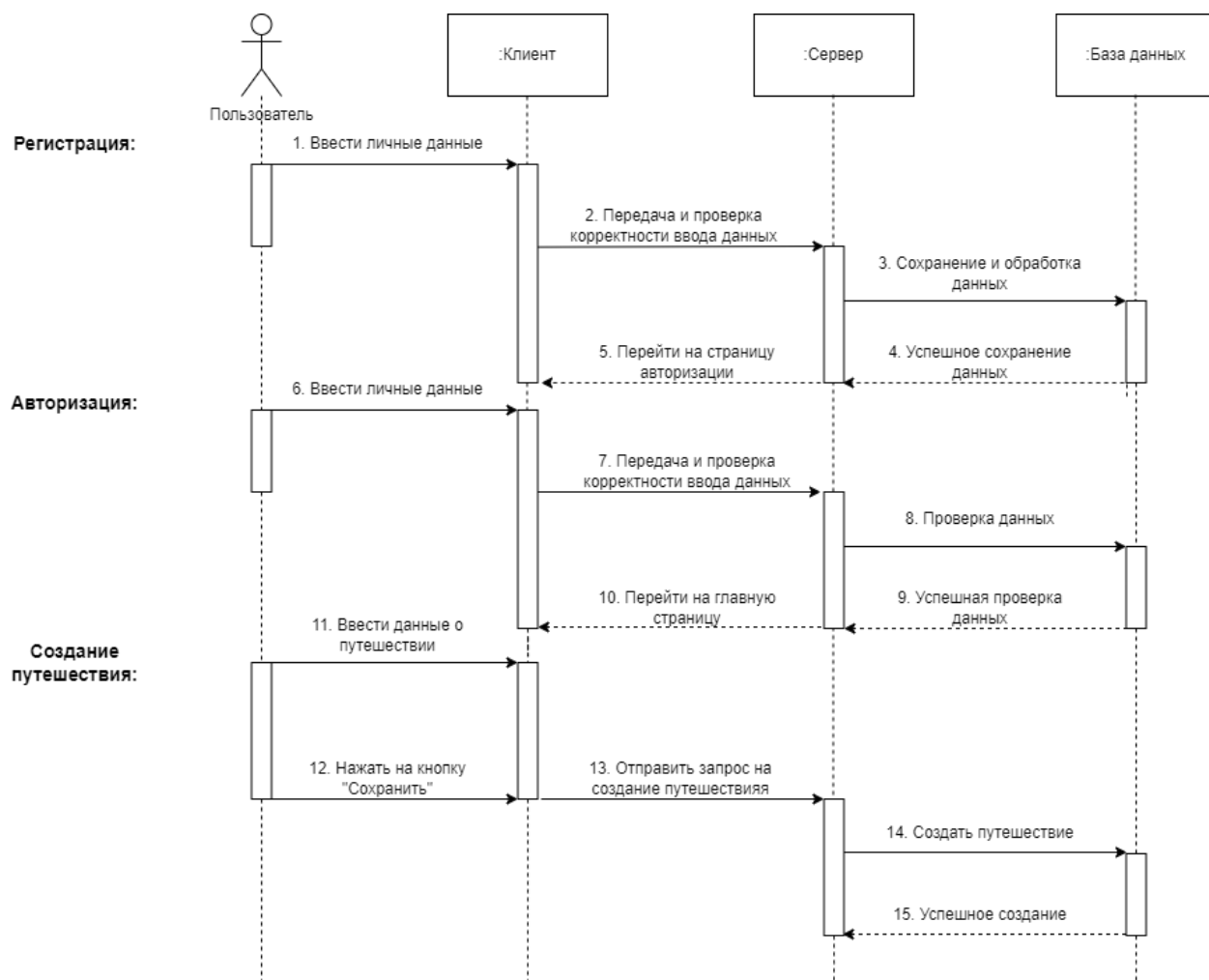


Рисунок 8 - Диаграмма последовательности

### 2.3.3 Диаграмма состояний (Statechart diagram)

Диаграмма состояний (Рисунок 9) отражает внутренние состояния объекта в течение его жизненного цикла от момента создания до разрушения. На данной диаграмме рассмотрены состояния от момента входа в систему до полного выхода из нее.

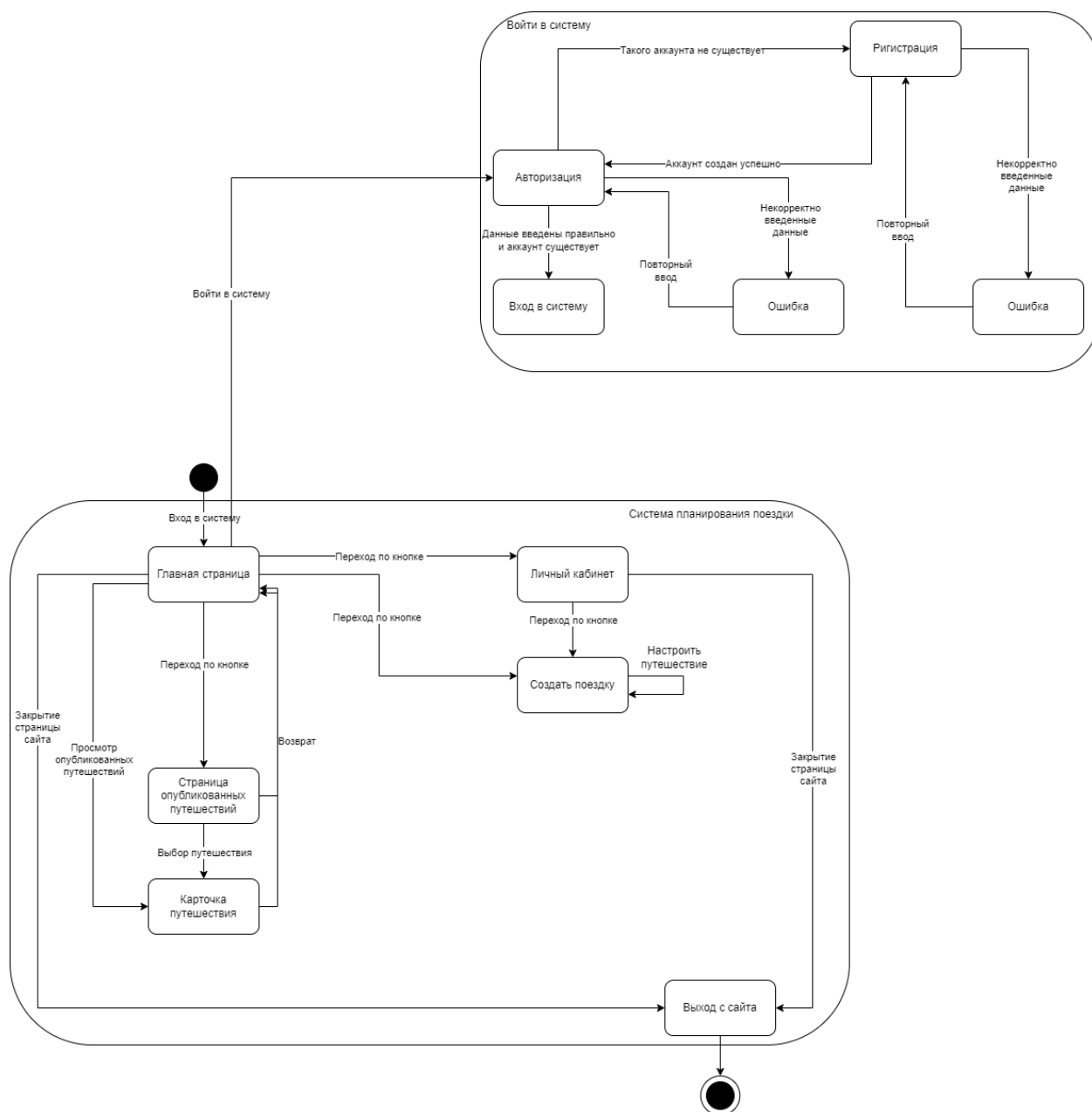


Рисунок 9 - Диаграмма состояний

### 2.3.4 Диаграмма деятельности (Activity diagram)

Диаграмма деятельности (Рисунок 10) представляет собой диаграмму, на которой показаны действия, состояния которых описаны на диаграмме состояний. Она описывает действия системы или людей, выполняющих действия, и последовательный поток этих действий.

В данном случае рассмотрен путь действий пользователя.

Диаграмма показывает, что пользователь может использовать основные функции сайта только будучи авторизованным.

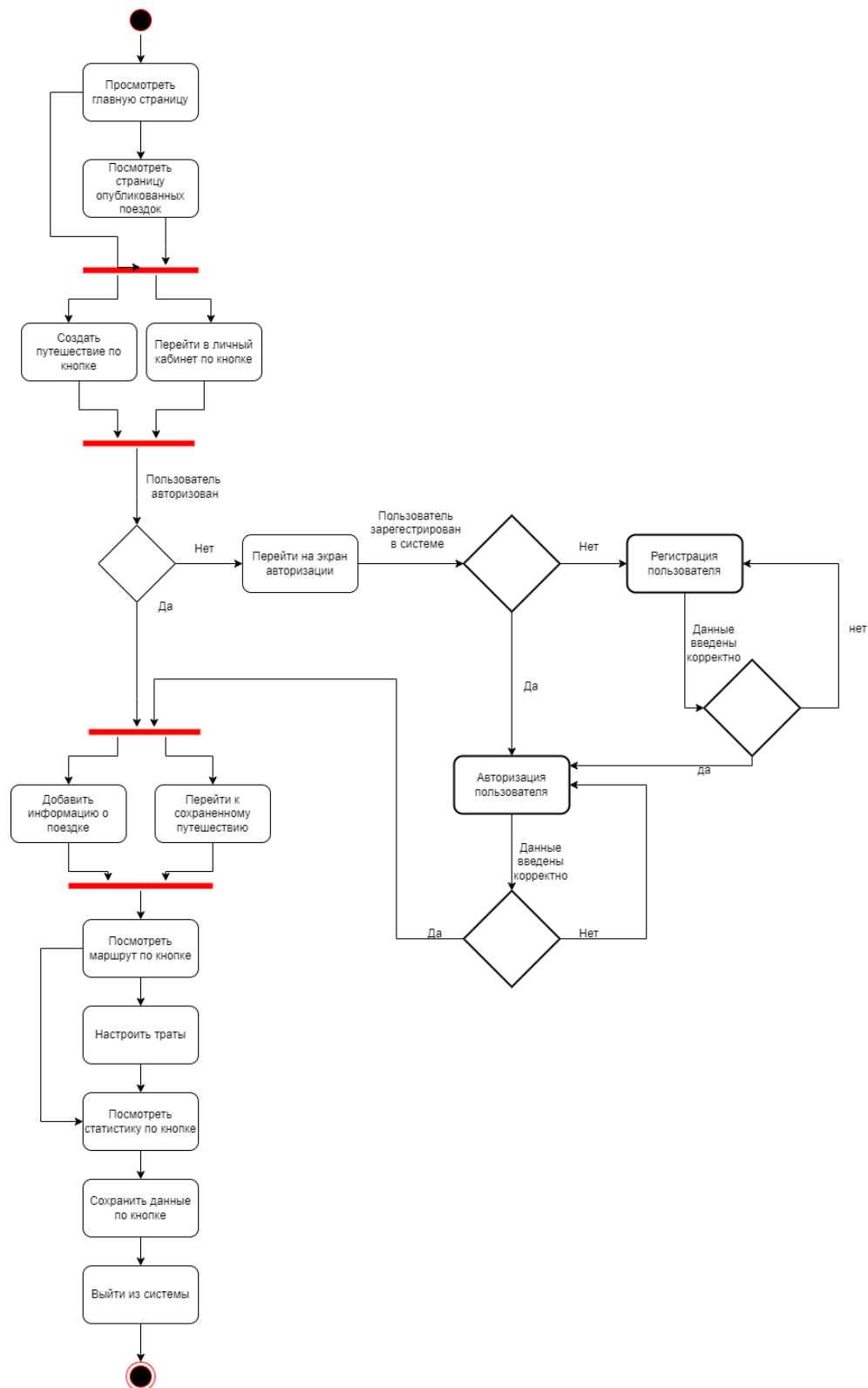


Рисунок 10 - Диаграмма деятельности (активности)

### 2.3.5 Диаграмма объектов (Object diagram)

Была создана диаграмма объектов. (Рисунок 11)

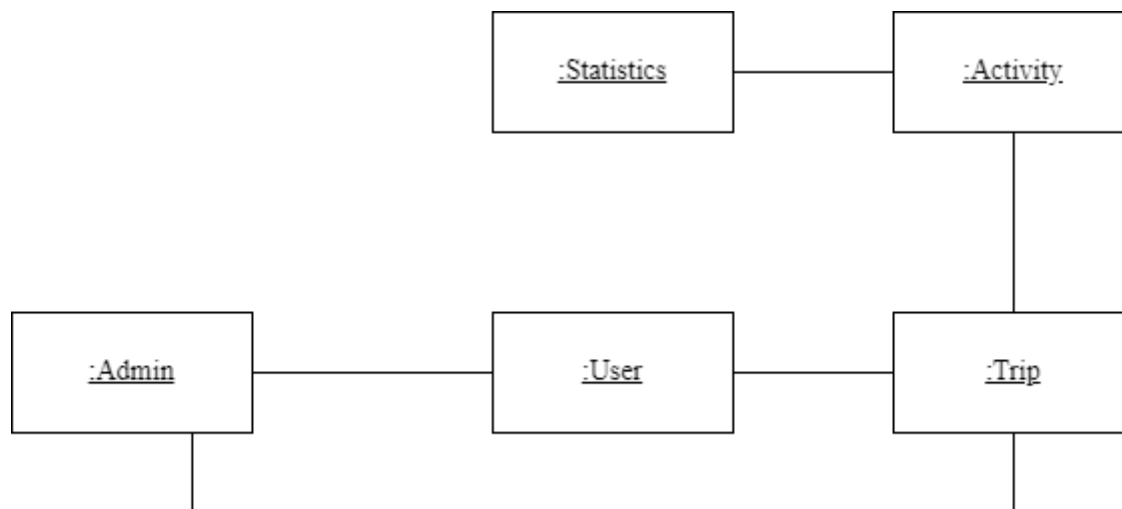


Рисунок 11 - Диаграмма объектов

### 2.3.6 Диаграмма сотрудничества (Collaboration diagram)

Диаграмма сотрудничества (Рисунки 12-13) — это вид диаграммы взаимодействия, в котором основное внимание сосредоточено на структуре взаимосвязей объектов, принимающих и отправляющих сообщения.

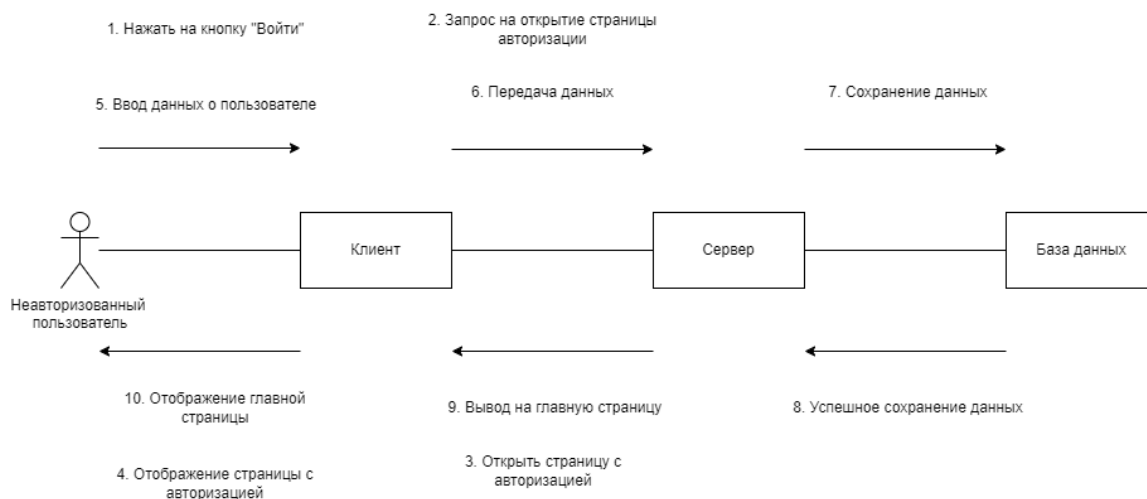


Рисунок 12 - Диаграмма сотрудничества при авторизации

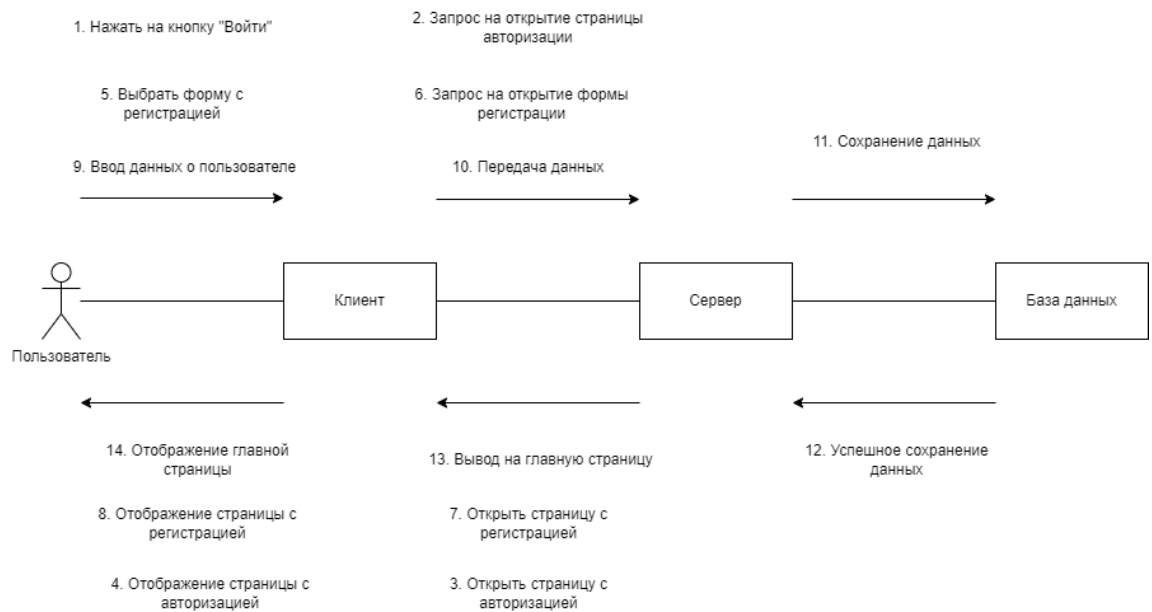


Рисунок 13 - Диаграмма сотрудничества при регистрации

### 2.3.7 Диаграмма развертывания (Deployment diagram)

Диаграмма развертывания (Рисунок 14) предназначена для представления общей конфигурации или топологии распределенной программной системы.

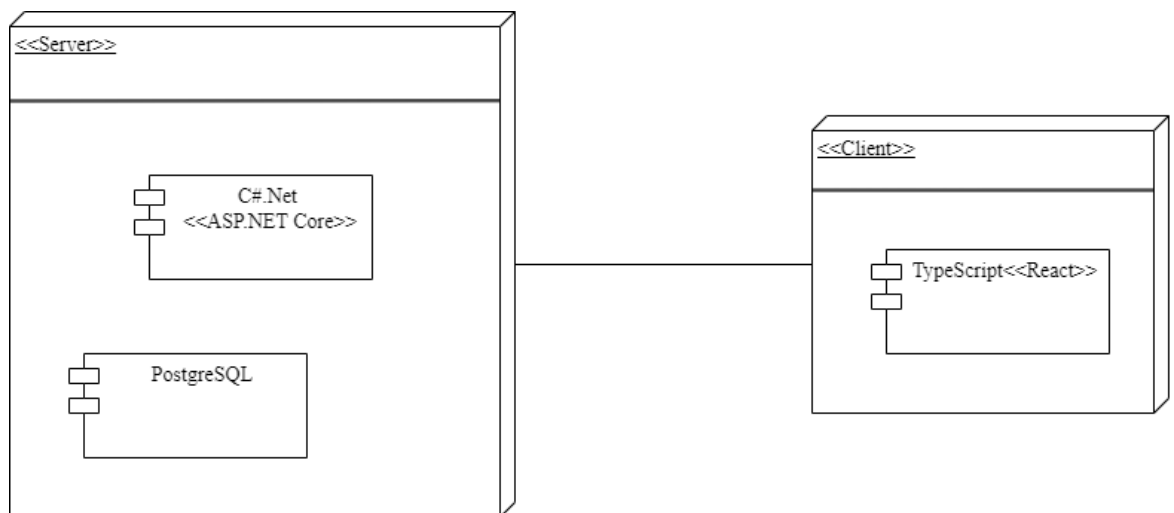


Рисунок 14 - Диаграмма развертывания

### 2.3.8 ER-диаграмма

ER-диаграмма — это графическое представление модели данных, которая используется для описания концептуальной структуры базы данных. В такой диаграмме есть сущности, которые представляют объекты, с которыми работает система, и связи между сущностями, описывающие их взаимодействия. ER-диаграмма помогает наглядно описать структуру базы данных и увидеть связи между ее элементами (Рисунок 15).

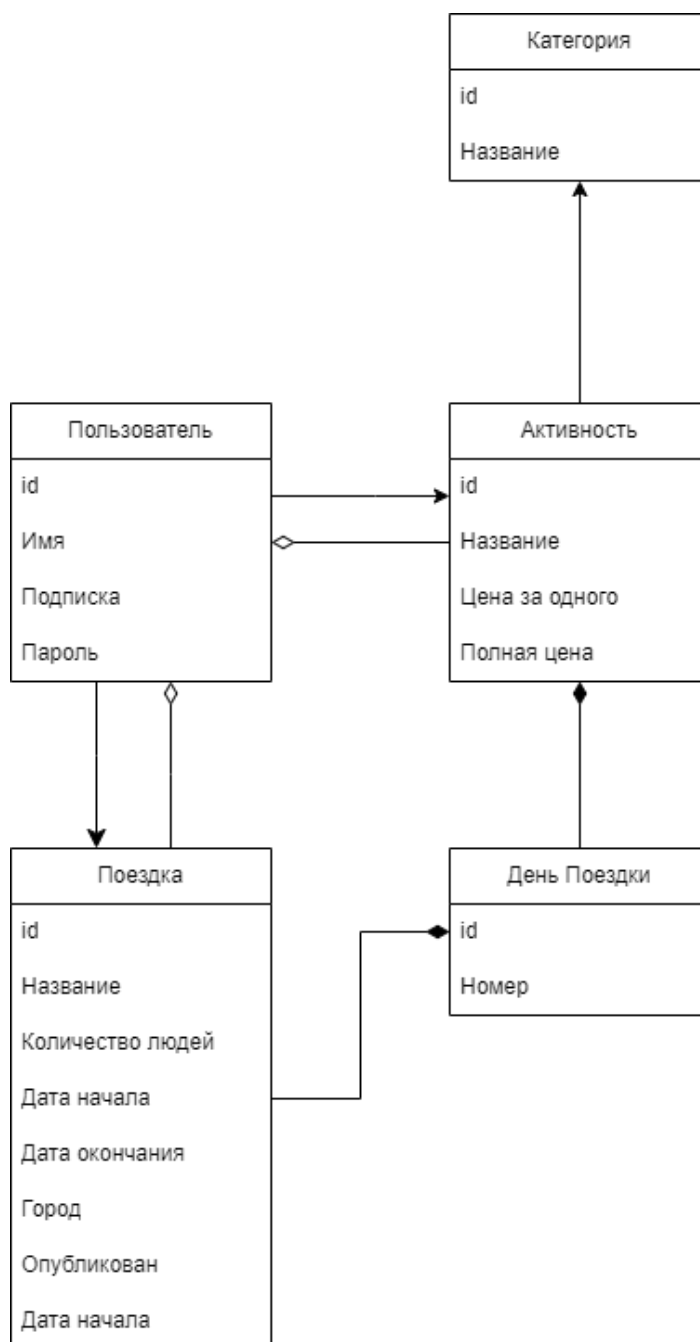


Рисунок 15 - ER-диаграмма