МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Техническое задание

на разработку веб-приложения

«Сервис для планирования и учета путешествий с друзьями TravelWithFriends»

Исполнители

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.С. Воронежская

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Бондарев

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Г. Деревянко

Заказчик

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Тарасов

Воронеж 2024

**СОДЕРЖАНИЕ**

[1 Терминология 4](#_Toc161251064)

[2 Общие сведения 8](#_Toc161251065)

[2.1 Полное наименование системы и название приложения 8](#_Toc161251066)

[2.2 Разработчики и заказчик 8](#_Toc161251067)

[2.3 Перечень документов, на основании которых создается приложение 8](#_Toc161251068)

[2.4 Плановые сроки начала и окончания работы по созданию системы 9](#_Toc161251069)

[2.5 Порядок контроля и приемки автоматизированной системы 9](#_Toc161251070)

[2.6 Цели и назначение создания автоматизированной системы 10](#_Toc161251071)

[2.6.1 Цели создания сайта 10](#_Toc161251072)

[2.6.2 Назначение АС 10](#_Toc161251073)

[3 Требования к сайту и программному обеспечению 11](#_Toc161251074)

[3.1 Требования к структуре АС в целом 11](#_Toc161251075)

[3.1.1 Требования к способам и средствам обеспечения информационного взаимодействия компонентов АС 11](#_Toc161251076)

[3.1.2 Перспективы развития, модернизации АС 11](#_Toc161251077)

[3.2 Требования к функциям (задачам), выполняемым АС 11](#_Toc161251078)

[3.2.1 Требования к внесению информации о поездке 11](#_Toc161251079)

[3.2.2 Требования к администрированию **Ошибка! Закладка не определена.**](#_Toc161251080)

[3.2.3 Пользователи **Ошибка! Закладка не определена.**](#_Toc161251081)

[3.2.4 Требования к составлению плана поездки 13](#_Toc161251082)

[3.2.5 Требования к визуализации маршрута для каждого дня 14](#_Toc161251083)

[3.2.6 Требования к получению статистики расходов во время поездки 15](#_Toc161251084)

[3.3 Требования к видам обеспечения АС 15](#_Toc161251085)

[3.3.1 Требования к лингвистическому обеспечению АС 15](#_Toc161251086)

[3.3.2 Требования к программному обеспечению сайта 15](#_Toc161251087)

[3.4 Общие технические требования к АС 18](#_Toc161251088)

[3.4.1 Требования по безопасности 18](#_Toc161251089)

[3.4.2 Требования к архитектуре 18](#_Toc161251090)

[3.5 Требования к оформлению страниц 18](#_Toc161251091)

[3.5.1 Общие требования к оформлению и верстке страниц 18](#_Toc161251092)

[3.5.2 Страница регистрации 19](#_Toc161251093)

[3.5.3 Экран авторизации 19](#_Toc161251094)

[3.5.4 Главная страница 20](#_Toc161251095)

[3.5.5 Страница опубликованных путешествий 20](#_Toc161251096)

[3.5.6 Страница для связи с разработчиками 21](#_Toc161251097)

[3.5.7 Страница с профилем пользователя, историей его поездок и статистическими данными 22](#_Toc161251098)

[3.5.8 Страница создания путешествия 22](#_Toc161251099)

[3.5.9 Страница статистики расходов **Ошибка! Закладка не определена.**](#_Toc161251100)

[3.5.10 Хедер 24](#_Toc161251101)

[ПРИЛОЖЕНИЕ 26](#_Toc161251102)

1. Терминология

**.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.

**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.

1. Общие сведения
   1. Полное наименование системы и название приложения

Полное наименование: «Сервис для планирования и учета путешествий с друзьями TravelWithFriends».

Название приложения: «TravelWithFriends».

* 1. Разработчики и заказчик

Заказчик: Старший Преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Представитель заказчика: Ассистент Проскуряков Егор Дмитриевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Разработчик: 6 команда группы 7.

Состав команды разработчика:

* Воронежская Екатерина Сергеевна;
* Бондарев Максим Сергеевич;
* Деревянко Валерий Геннадьевич.
  1. Перечень документов, на основании которых создается приложение

Данный сайт будет создаваться на основании следующих документов:

* Федеральный закон "О персональных данных" от 27.07.2006 N 152–ФЗ;
* Закон РФ от 07.02.1992 N 2300-1 (ред. от 11.06.2021) "О защите прав потребителей".
  1. Плановые сроки начала и окончания работы по созданию системы

Состав и содержание работ по созданию сайта включают в себя следующие этапы:

* сбор необходимой информации, постановка целей, задач системы, которые в будущем должны быть реализованы;
* анализ предметной области, анализ конкурентов и построение структуры требований, ведущих к решению поставленных задач и целей;
* построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД;
* разработка рабочего проекта, состоящего из написания кода, отладки и корректировки кода программы;
* проведение тестирования программного обеспечения;
* процесс установки и запуска приложения на удаленном сервере. После этого этапа программное обеспечение доступно через интернет.

Работы по созданию системы должны начаться 17.03.2024 и завершиться 01.06.2024.

* 1. Порядок контроля и приемки автоматизированной системы

Предварительные отчёты по работе будет проводиться в следующие сроки:

1 аттестация (18.03.2024) — предоставлена ссылка на репозиторий проекта на GitHub, Предоставлен доступ к проекту в таск-менеджере YouTrack, предоставлен доступ к проекту Miro с общей логикой системы, предоставлено готовое техническое задание;

2 аттестация (01.05.2024) — написан программный код приложения, реализующий MVP; реализована верстка и дизайн сайта, создана БД и настроено ее взаимодействие с сервером, проведена отладка и доработка кода, проведено тестирование по работе системы, результаты работ и исходный код предоставлены на репозитории проекта на GitHub;

3 аттестация (10.06.2024) — предоставлен курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовая система.

* 1. Цели и назначение создания автоматизированной системы
     1. Цели создания сайта

Целями создания сайта являются:

* получить прибыль 5000 рублей от продаж подписки туристическим организациям и пользователям к 10.07.2024;
* число зарегистрированных пользователей к 20.06.2024 должно составить не менее 20 человек.
  + 1. Назначение АС

АС предназначена 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. Требования к сайту и программному обеспечению
   1. Требования к структуре АС в целом
      1. Требования к способам и средствам обеспечения информационного взаимодействия компонентов АС

Информационное взаимодействие между компонентами АС должно проходить в рамках системы, расположенной на онлайн-хостинге по протоколу HTTPS.

* + 1. Перспективы развития, модернизации АС

АС должна предоставлять возможность развития своих функций в рамках следующих направлений:

* интеграция с сервисами бронирования отелей;
* возможность выбрать маршрут на день из маршрутов, предлагаемых системой.
  1. Требования к функциям (задачам), выполняемым АС
     1. Пользователи

Система должна позволять учитывать следующие сведения о пользователях:

* имя пользователя;
* адрес электронной почты;
* пароль.

При регистрации указываются все выше перечисленные сведения о пользователе, при авторизации лишь адрес электронной почты и пароль.

Система должна позволять неавторизированным пользователям (гостям) следующие возможности:

* просмотр главной страницы;
* просмотр страницы опубликованных поездок;
* регистрацию;
* авторизацию.

Система должна предоставлять авторизованным пользователям (пользователям) следующие возможности:

* просмотр главной страницы;
* просмотр страницы опубликованных поездок;
* создание путешествия;
* формирование списка активностей;
* получение статистики расходов;
* просмотр информации на странице личного кабинета.
  + 1. Требования к администрированию

Система должна позволять администратору удалять с сайта поездки, нарушающие правила сайта. Также у администратора должна быть возможность управлять подпиской пользователя; удалять пользователей, нарушающих правила сайта.

* + 1. Требования к созданию поездки

Система должна позволять указывать следующие сведения о поездке:

* название поездки;
* количество людей;
* логины участников поездки, при условии, что они зарегистрированы в системе;
* даты начала и окончания поездки;
* город, в котором будет проходить путешествие;
* название отеля для проживания.

Система не должна предоставлять возможность изменить данную информацию.

Система накладывает следующие ограничения:

* пользователь с приобретенной подпиской может добавить до 15 участников к поездке, пользователь без подписки — до 7;
* общее количество поездок пользователя с подпиской составляет не более 15, пользователя без подписки — не более 5.
  + 1. Требования к составлению плана поездки

После внесения информации о поездке система должна позволять составить план поездки. Для этого в каждый день поездки можно внести следующую информацию:

* активность;
* категорию активности;
* отметить участников в каждой активности;
* внести цену за одного человека, либо общую стоимость активности. В случае внесения цены за одного человека система посчитает общую стоимость по формуле, умножив цену за одного человека на количество участников. В случае внесения общей стоимости система посчитает стоимость за одного человека, разделив общую стоимость на количество участников;
* выбрать участника, который оплачивает активность, либо выбрать пункт «каждый сам». В случае выбора участника итоговая стоимость активности записывается на него. В случае выбора пункта «каждый сам» стоимость активности делится между всеми участниками активности.

Система должна позволять:

* ввести в строку поиска название места или достопримечательности. Из списка предложенных вариантов выбрать нужный пункт;
* ввести активность, не используя строку поиска;
* удалить любую активность.

Система накладывает следующие ограничения:

* активностей, введенных через строку поиска не более 5 на один день;
* активностей, введенных без использования строки поиска не более 15 на один день, если у создателя поездки оформлена подписка; и не более 10, если у создателя нет подписки;

После внесения желаемых активностей пользователь может визуализировать маршрут для каждого дня на карте.

* + 1. Требования к визуализации маршрута для каждого дня

Система должна позволять визуализировать маршрут следующим образом:

* рядом с каждым днем поездки находится кнопка «Отобразить на карте»;
* при нажатии на данную кнопку на карте отобразятся маркеры каждой активности, которая была внесена посредством выбора из предложенных вариантов после использования строки поиска;
* активности, внесенные пользователем самостоятельно, не будут отображены на карте;
* маркеры будут соединены маршрутами;
* построение маршрутов будет осуществлено по следующему принципу: от активности, указанной первой, будет построен маршрут до активности, указанной второй. От нее будет построен маршрут до активности указанной третьей, и так далее.

3.2.6 Требования к получению статистики расходов во время поездки

Система должна позволять просматривать статистику следующим образом: при нажатии на кнопку «отобразить статистику» формируются диаграммы расходов:

* диаграмма распределения расходов на каждую категорию в процентном соотношении;
* диаграмма распределения расходов на каждого участника поездки в процентном соотношении.

Система должна подсчитать и отобразить общую сумму расходов, а также сумму расходов каждого участника поездки.

* 1. Требования к видам обеспечения АС
     1. Требования к лингвистическому обеспечению АС

Система должна поддерживать интерфейс на русском языке.

* + 1. Требования к программному обеспечению сайта

Сайт должен иметь архитектуру, соответствующую модели клиент-серверного взаимодействия на основе REST API.

Для реализации серверной части сайта будут использоваться следующие средства:

* язык программирования C# —объектно-ориентированный язык программирования общего назначения, позволяет писать быстрый и эффективный код благодаря использованию JIT-компиляции (Just-In-Time) и оптимизациям в рамках платформы .NET;
* ASP.Net Core 8 — кроссплатформенный фреймворк для создания веб-приложений на платформе .NET. 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;
* СУБД PostgreSQL 16.2 — объектно-реляционная система управления базами данных. Мощная и надежная БД с открытым исходным кодом, поддерживает множество типов данных.

Для реализации клиентской части сайта будут использоваться следующие средства:

* язык гипертекстовой разметки HTML5 — язык разметки для создания веб-страниц. Является стандартом для создания структуры веб-страниц и обеспечивает семантическую разметку;
* формальный язык описания внешнего вида документа CSS3 — каскадные таблицы стилей, используемые для оформления веб-страниц. Позволяет создавать дизайны для веб-страниц, а также обеспечивает адаптивность;
* язык программирования TypeScript 4.9.5 — строго типизированный язык программирования, расширяющий возможности JavaScript. Обеспечивает удобство разработки, добавляет статическую типизацию и возможности объектно-ориентированного программирования;
* React 18.2.0 — библиотека JavaScript с открытым исходным кодом. Обладает высокой производительностью, модульностью и удобством в использовании компонентов.

Инструменты для ведения документации:

* YouTrack — это средство для управления проектами и задачами, которое обеспечивает удобный интерфейс, гибкую настройку и возможность отслеживать прогресс работы над проектом;
* Miro — это инструмент для совместной работы и визуализации идей, который позволяет создавать диаграммы, макеты, схемы и другие элементы проекта;
* Draw.io — онлайн-инструмент для создания диаграмм, схем и других визуальных элементов;
* Figma — инструмент для дизайна интерфейсов, который обеспечивает возможность создания прототипов, макетов и дизайнов веб-приложений.

Дополнительные инструменты:

* Git — распределенная система управления версиями, которая обеспечивает контроль изменений в коде, возможность ветвления и слияния кода;
* GitHub — это платформа для хостинга проектов на базе Git, которая обеспечивает возможность хранения кода, управления задачами, рецензирования кода и совместной работы над проектами;
* платформа Docker 25.0.3 — открытая платформа для разработки, доставки и эксплуатации приложений. Позволяет создавать среды разработки и развертывать приложения с минимальными затратами на конфигурацию и совместимость.
  1. Общие технические требования к АС
     1. Требования по безопасности

Система должна обеспечить хранение паролей в зашифрованном виде. Шифрование производится по алгоритму хеширования SHA256.

* + 1. Требования к архитектуре

Созданная система должна иметь архитектуру, соответствующую шаблону клиент-серверного приложения, а также иметь разделение на backend (серверную) и frontend (клиентскую) часть, взаимодействие между которыми должно происходить с помощью REST API.

* 1. Требования к оформлению страниц
     1. Общие требования к оформлению и верстке страниц

Сайт должен быть оформлен в одной цветовой палитре с использованием ограниченного набора шрифтов. У страниц сайта должен быть единый стиль. В оформлении приложения должно присутствовать разработанное название. На каждой странице должен присутствовать хедер.

Необходимо корректное и одинаковое отображение страниц сайта в следующих браузерах:

* Google Chrome 122.0.6261.111 и выше;
* Yandex Browser 23.9.0.2325 и выше.
  + 1. Страница регистрации

Инициализация экрана должна выглядеть следующим образом:

* отобразить экран с формой для регистрации пользователя.
* если пользователь не был зарегистрирован ранее, он должен ввести свои личные данные и перейти на экран авторизации;
* если желания входить в систему нет, можно пользоваться сайтом как неавторизованный пользователь.

Должны быть реализованы следующие элементы экрана:

* форма для заполнения полей личными данными;
* кнопка «Зарегистрироваться»;
* кликабельная ссылка для перехода на экран авторизации;
* поле для вывода результата, которое возникает в случае успешной регистрации или некорректно введенных данных.

Компоновка и логика заключается в том, что этот экран необходим для осуществления регистрации пользователя в системе.

* + 1. Экран авторизации

Инициализация экрана должна выглядеть следующим образом:

* отобразить экран с формой для авторизации пользователя.
* если пользователь был зарегистрирован ранее, он может ввести email, пароль и зайти в личный кабинет;
* если пользователь не был зарегистрирован ранее, он должен перейти на форму регистрации;
* если желания входить в систему нет, можно пользоваться сайтом как неавторизованный пользователь.

Должны быть реализованы следующие элементы экрана:

* форма для заполнения полей личными данными. Сделать их кликабельными;
* кнопка «Войти»;
* кликабельная ссылка для перехода на экран регистрации;
* поле для вывода результата, которое возникает в случае успешной авторизации, некорректно введенных данных или если такого пользователя не существует в системе.

Компоновка и логика заключается в том, что этот экран необходим для осуществления входа пользователя в систему.

* + 1. Главная страница

Инициализация экрана должна выглядеть следующим образом:

* отобразить экран с основной информацией о сайте.

Должны быть реализованы следующие элементы экрана:

* навигационная панель (хедер) с основными категориями сайта. Сделать их кликабельными;
* заголовки, описание и изображения, отображающие этапы работы с приложением. Сделать их некликабельными;
* кнопка «Создать путешествие». При нажатии на нее происходит переход на экран авторизации, если пользователь не авторизован, и на экран создания путешествия, если пользователь авторизован.

Компоновка и логика заключается в том, что этот экран информирует пользователя о деятельности сайта. Он является экраном-приветствием.

* + 1. Страница опубликованных путешествий

Инициализация экрана должна выглядеть следующим образом:

* отобразить экран с карточками опубликованных пользователями путешествий;
* экран скролится.

Должны быть реализованы следующие элементы экрана:

* навигационная панель (хедер) с основными категориями сайта. Сделать их кликабельными;
* карточка завершенного опубликованного путешествия пользователя содержит имя создателя, краткое описание;
* кнопка «Начать путешествие» под каждой карточкой для авторизованного пользователя для перехода на экран создания путешествия с заданными параметрами.

Компоновка и логика заключается в том, что этот экран необходим для вывода информации по путешествиям разных пользователей, разрешенных к просмотру.

* + 1. Страница для связи с разработчиками

Инициализация экрана должна выглядеть следующим образом:

* отобразить экран с общей информацией о разработчиках;
* экран не скролится.

Должны быть реализованы следующие элементы экрана:

* навигационная панель (хедер) с основными категориями сайта. Сделать их кликабельными;
* форма с кликабельными ссылками для возможности связаться с разработчиками в VK;

Компоновка и логика заключается в том, что этот экран необходим для связи с разработчиками.

* + 1. Страница с профилем пользователя и историей его поездок

Инициализация экрана должна выглядеть следующим образом:

* отобразить экран с информацией о пользователе, совершенных путешествиях;
* экран скролится.

Должны быть реализованы следующие элементы экрана:

* навигационная панель (хедер) с основными категориями сайта. Сделать их кликабельными;
* форма с информацией о пользователе;
* форма с карточками, содержащими информацию о совершенных путешествиях. Если их нет, уведомляющая надпись «Список пуст». Каждая карточка должна быть кликабельной для перехода к экрану создания путешествия.

Компоновка и логика заключается в том, что этот экран необходим для вывода информации о пользователе.

* + 1. Страница создания путешествия

Инициализация экрана должна выглядеть следующим образом:

* отобразить экран с формой для создания, настройки и просмотра поездки;
* экран скролится.

Должны быть реализованы следующие элементы экрана:

* навигационная панель (хедер) с основными категориями сайта. Сделать их кликабельными;
* форма для указания даты начала и даты окончания путешествия, города, добавления членов команды;
* поле для вывода результата, которое возникает в случае успешного добавления члена команды, некорректно введенных данных пользователя или невозможности добавить пользователя;
* поле для выбора режима доступа (публичное, приватное) к путешествию;
* форма, для составления плана на каждый день поездки, с указаниями активностей, трат, участников;
* строка поиска, которая позволит ввести название активности и выбрать нужную из выпадающего списка. Такие активности должны отображаться на карте по кнопке;
* карта с отображающимися маркерами активностей и маршрутами между ними;
* кнопка «сохранить путешествие»;
* кнопка «показать статистику»;
* форма с диаграммой распределения расходов на каждую категорию в процентном соотношении и диаграммой распределения расходов на каждого участника поездки в процентном соотношении;
* форма общей суммой расходов, а также сумму расходов каждого участника поездки.

Компоновка и логика заключается в том, что этот экран необходим для заполнения информации о поездке, просмотра маршрута и просмотра статистики расходов.

* + 1. Хедер

Данный раздел должен быть представлен в виде навигационного меню в верхней части экрана (хедер) с определенными подразделениями. При нажатии на кнопки меню происходит переход на соответствующую страницу сайта.

Для неавторизованного пользователя меню содержит следующие подразделения:

* на главную страницу (при нажатии на наименование сайта);
* опубликованные путешествия;
* о нас;
* войти.

Для авторизованного пользователя меню содержит следующие подразделения:

* на главную страницу (при нажатии на наименование сайта);
* опубликованные путешествия;
* о нас;
* создать путешествие;
* личный кабинет;
* выйти.

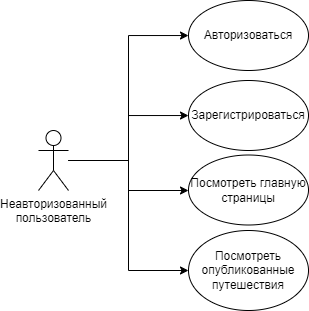
Для администратора меню содержит следующие подразделения:

* на главную страницу (при нажатии на наименование сайта);
* опубликованные путешествия;
* личный кабинет;
* для администратора;
* выйти.

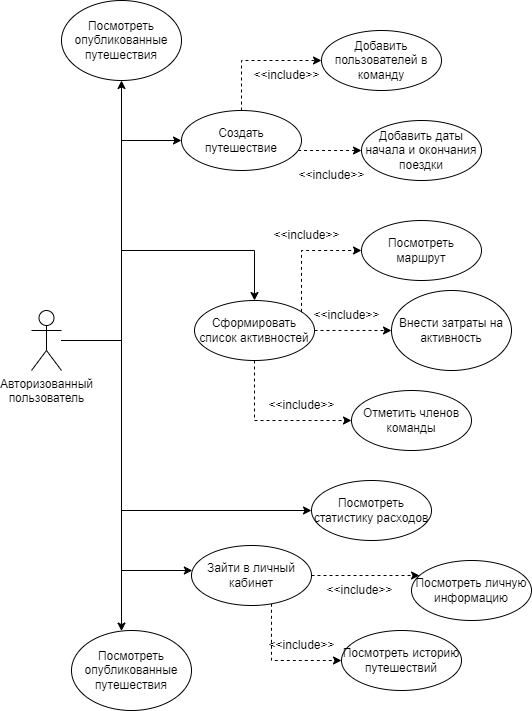
ПРИЛОЖЕНИЕ А USE Case diagramm



1. Диаграмма Use Case со стороны Администратора

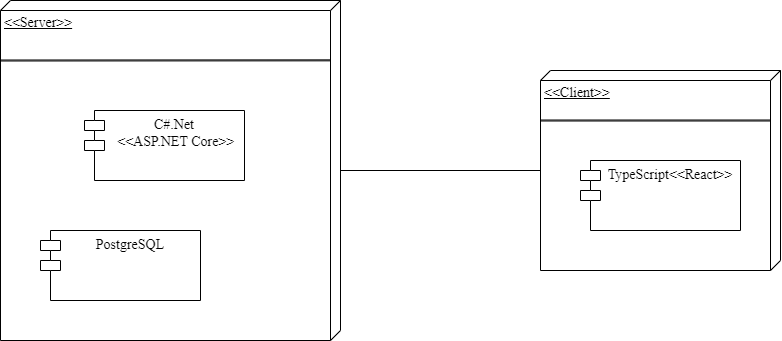


1. Диаграмма Use Case со стороны Неавторизованного пользователя



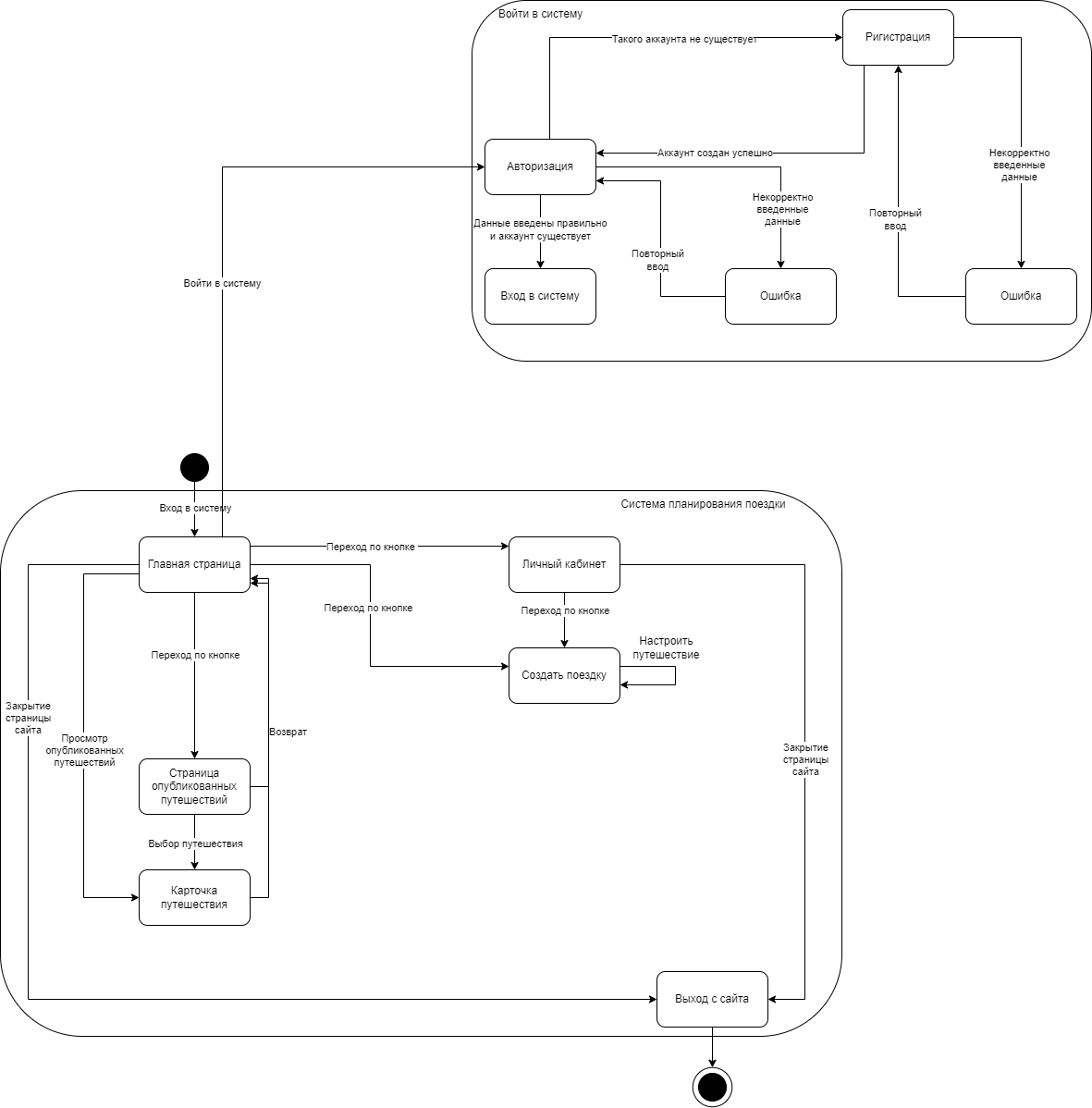
1. Диаграмма Use Case со стороны Авторизованного пользователя

приложение б Deployment diagramm



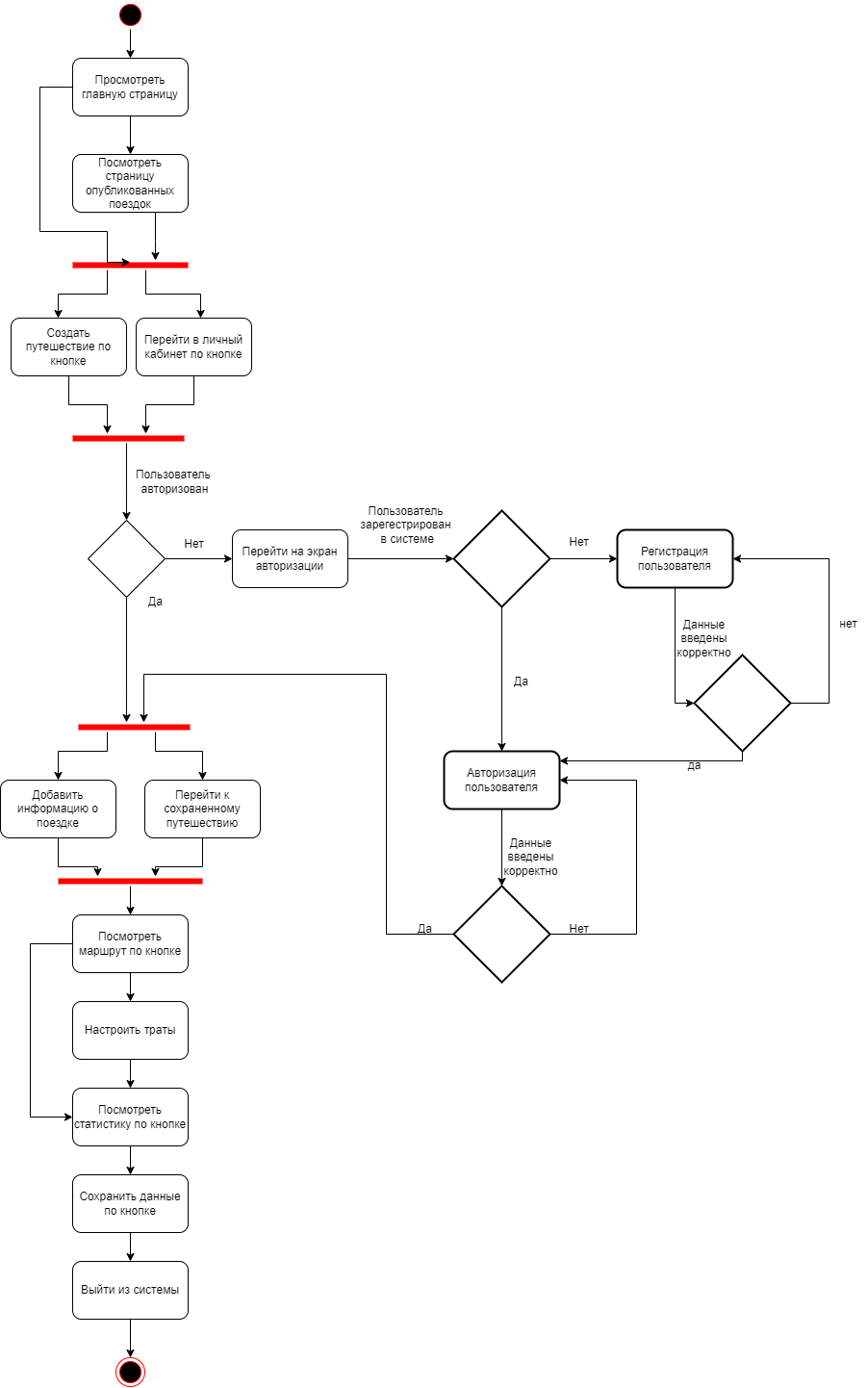
1. Диаграмма развертывания

приложение в statechart diagramm



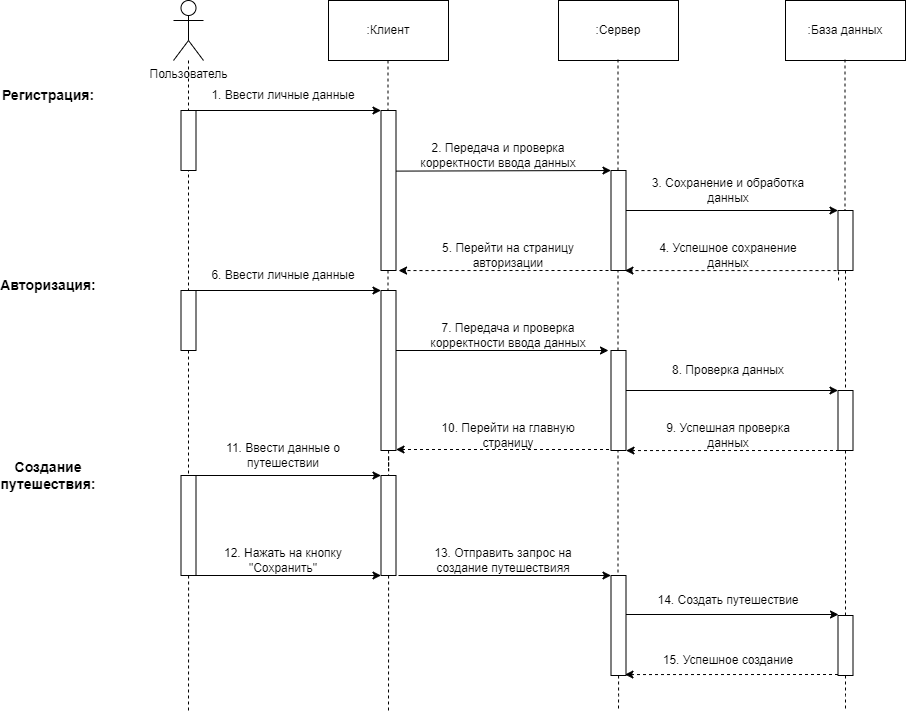
1. Диаграмма состояний

приложение г activity diagramm



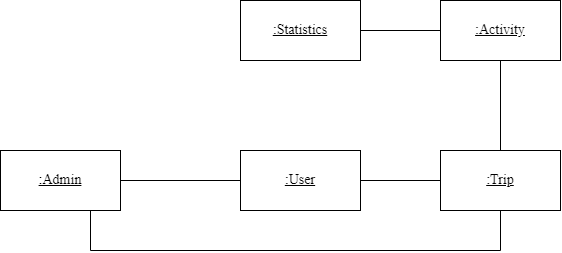
1. Диаграмма активности

приложение д sequence diagramm



1. Диаграмма последовательностей

приложение е object diagramm



1. Диаграмма объектов