

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

Факультет компьютерных наук
Кафедра информационных систем

*Разработка веб-приложения для создания персонализированных путеводителей по
городу Воронеж и Воронежской области TripToVoronezh*

Курсовой проект по дисциплине «Технологии программирования»

*09.03.02 Информационные системы и технологии
6 семестр 2023/2024 учебного года*

Зав. кафедрой _____ к. т. н., доцент Д.Н. Борисов
Обучающийся _____ ст. 3 курса оч. отд. Д.В. Свиридов
Обучающийся _____ ст. 3 курса оч. отд. А.Ю. Кунахова
Руководитель _____ ст. преподаватель В.С. Тарасов

Воронеж 2024

СОДЕРЖАНИЕ

Введение.....	4
1 Постановка задачи	6
1.1 Цели работы	6
1.2 Задачи веб-приложения.....	6
1.3 Требования к веб-приложению	7
1.3.1 Требования к функциям, выполняемым в веб-приложении.....	7
1.3.2 Требования к оформлению и вёрстке страниц.....	7
1.3.3 Требование к защите информации.....	8
1.3.4 Требование к надёжности веб-приложения	8
1.3.5 Требование к численности и квалификации персонала, обрабатывающего сайт	8
1.3.6 Требования к системе администрирования.....	8
1.4 Задачи, решаемые в процессе разработки	9
2 Анализ предметной области	10
2.1 Глоссарий.....	10
2.2 Обзор аналогов.....	13
2.2.1 TripAdvisor.....	13
2.2.2 Туристер.....	14
2.2.3 VRN.Guide	16
2.2.4 TIMEWILL.....	17

2.3 Моделирование системы	19
2.3.1 Диаграмма прецедентов	19
2.3.2 Диаграмма последовательности	20
2.3.3 Диаграмма активности	22
3 Реализация	23
3.1 Средства реализации	23
3.2 Реализация базы данных	24
3.2.1 ER-диаграмма.....	24
3.3 Реализация клиентской части	25
3.3.1 Форма для поиска гидов.....	26
3.3.2 Форма экрана доступных гидов	27
3.3.3 Форма экрана авторизации	28
3.3.4 Форма экрана регистрации	29
3.3.5 Форма экрана личной страницы.....	30
3.3.6 Форма экрана сохранённых гидов	30
3.3.7 Форма главного экрана.....	31
3.4 Реализация серверной части	32
3.4.1 Архитектура серверной части приложения.....	33
4 Аналитика	34
Заключение	35
Список использованных источников	36

Введение

В современном мире, насыщенном информацией и различными активностями, потребность в персонализированных путешествиях возрастает. Туристы и местные жители стремятся к более глубокому и увлекательному погружению в культуру, достопримечательности и уникальные места, что требует тщательного планирования и организации. Поэтому возникает необходимость в инструментах, которые помогают людям создавать индивидуальные маршруты, основываясь на их интересах и предпочтениях.

Этот курсовой проект посвящен разработке веб-приложения TripToVoronezh, которое предназначено для создания персонализированных путеводителей по городу Воронеж и Воронежской области. Наше приложение предоставляет пользователям удобный инструмент для планирования и организации маршрутов, учитывая их предпочтения и интересы. С его помощью пользователи смогут формировать индивидуальные путеводители, выбирая из множества культурных, исторических и развлекательных объектов региона.

Приложение TripToVoronezh позволит пользователям указывать свои интересы, такие как история, культура, природа, гастрономия и многое другое. На основе этих данных система предложит оптимальные маршруты, включающие наиболее подходящие достопримечательности и интересные места.

Таким образом, наше веб-приложение TripToVoronezh станет незаменимым помощником для всех, кто хочет сделать своё путешествие по Воронежу и Воронежской области максимально интересным и запоминающимся.

Актуальность работы обусловлена отсутствием доступного и удобного, но при этом обширного по функционалу сервиса для создания персонализированных путеводителей по Воронежу и Воронежской области на основе интересов пользователя. Разработка веб-приложения "TripToVoronezh" отвечает актуальной потребности в инструментах, которые помогают людям находить интересные места, планировать маршруты и получать рекомендации, соответствующие их предпочтениям. Приложение "TripToVoronezh" способствует улучшению туристического опыта и повседневного досуга как жителей, так и гостей города, предоставляя пользователям удобный и функциональный инструмент для исследования местных достопримечательностей.

1 Постановка задачи

1.1 Цели работы

Целью данного проекта является разработка веб-приложения "TripToVoronezh", предоставляющего пользователям персонализированные путеводители по городу Воронеж и Воронежской области на основе их интересов.

1.2 Задачи веб-приложения

Веб-приложение позволяет пользователем решать следующие задачи:

- Систематизация информации о местах для посещения;
- Ознакомление с широким спектром досуга (рестораны, музеи, кинотеатры, памятники, фотолокации и др);
- Создание пользователями персонализированных путеводителей, на основе большого количества фильтров, представляющих собой интересные гиды;
- Возможность пользователям добавлять интересные для посещения места и гиды в специальный раздел «Мои гиды»;
- Возможность пользователям осуществлять быстрый поиск по гидам;
- Пользование удобным и интуитивно-понятным интерфейсом.

1.3 Требования к веб-приложению

1.3.1 Требования к функциям, выполняемым в веб-приложении

Веб-приложение должно обеспечить неавторизованному пользователю выполнение следующих функций:

- Создание пользовательского профиля;
- Вход в пользовательский профиль;

Веб-приложение должно обеспечить авторизованному пользователю выполнение следующих функций:

- Выход из пользовательского профиля;
- Просмотр гидов;
- Поиск гидов;
- Фильтрация параметров гидов;
- Сохранение понравившихся гидов;

1.3.2 Требования к оформлению и вёрстке страниц

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

Необходимо корректное и одинаковое отображение страниц сайта в следующих браузерах:

- Yandex Browser, версия 23.3.7.24 или новее;
- Google Chrome, версия 118.0.5993.71 или новее;
- Mozilla Firefox, версия 115 и 115 ESR или новее;
- Microsoft Edge, версия 88.0.705.68 или новее;
- Safari, версия 16.5.2.

1.3.3 Требование к защите информации

Веб-приложение должно обеспечить защиту личных данных пользователей путём хеширования паролей, хранящихся в базе данных, по алгоритму BCrypt в Spring Security.

Сервер должен быть защищён от SQL-инъекций путём применения параметризованных запросов при обращении к базе данных.

1.3.4 Требование к надёжности веб-приложения

Веб-приложение должно осуществлять бесперебойную и безошибочную работу при нагрузках до 1000 единовременных пользователей.

Основной шрифт приложения – Inter (данный шрифт можно использовать по Open Font License в коммерческой и некоммерческой деятельности).

1.3.5 Требования к численности и квалификации персонала, обслуживающего сайт

Сайт должен обслуживаться и поддерживаться как минимум 1 администратором, ознакомленным с правилами работы с сайтом. Он обязан отвечать за добавление, удаление, просматривание или редактирование данных через сервер, настраивание прав пользователей.

1.3.6 Требования к системе администрирования

Система администрирования сайта создается при помощи фреймворка Spring Security. Это мощный инструмент управления, который позволяет добавлять, удалять, просматривать или редактировать любую модель базы данных через веб-интерфейс, т.е. управлять данными по путеводителям.

1.4 Задачи, решаемые в процессе разработки

Были поставлены следующие задачи:

- Анализ предметной области;
- Обзор аналогов;
- Постановка задачи;
- Создание репозитория GitHub и досок в Trello и Miro;
- Разработка требований;
- Создание диаграмм;
- Разработка дизайна приложения;
- Написание технического задания в соответствии с ГОСТ 34.602 – 2020;
- Реализация интерфейса приложения;
- Реализация серверной части приложения;
- Развёртывание приложения;
- Написание курсовой работы.

2 Анализ предметной области

2.1 Глоссарий

В настоящей работе используются следующие термины и сокращения с соответствующими определениями:

Проект, система – разрабатываемое веб-приложение.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера

Фреймворк – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Пользователь – авторизованный в системе человек, пользующийся функционалом веб-приложения.

Личный кабинет пользователя – веб-страница, где отражаются персональная информация пользователя, введенная на этапе регистрации и/или отредактированная позже, а также переход к странице редактирования личных данных.

Администратор – пользователь, обслуживающий веб-приложение, осуществляющий связь с клиентами и данные которого были заранее занесены в систему под отличительным доменом.

Back-end – программно-аппаратная часть веб-приложения. Отвечает за функционирование внутренней части веб-приложения.

Front-end – клиентская часть приложения. Отвечает за получение информации с программно-аппаратной части и отображение ее на устройстве пользователя.

Сервер, серверная часть – компьютер, обслуживающий другие компьютеры и предоставляющий им свои ресурсы для выполнения определенных задач.

Клиентская сторона – компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой.

GitHub – веб-сервис для хостинга ИТ-проектов и их совместной разработки.

REST API (REST) – стиль архитектуры программного обеспечения для построения масштабируемых веб-приложений.

Java — это высокоуровневый язык программирования общего назначения с строгой статической типизацией и автоматическим управлением памятью посредством сборщика мусора.

Spring — это мощный Java веб-фреймворк, который позволяет создавать надежные, безопасные и поддерживаемые веб-приложения с высокой скоростью разработки.

Hibernate — это мощный Java ORM-фреймворк, который позволяет эффективно работать с базами данных, создавая надежные и поддерживаемые приложения с высокой скоростью разработки.

Система управления базами данных (СУБД) – это набор программ, которые управляют структурой БД и контролируют доступ к данным, хранящимся в БД.

PostgreSQL – это реляционная СУБД с открытым кодом.

JavaScript – это язык программирования, предназначенный для front-end-разработки и использующийся для взаимодействия с пользователем.

CSS – формальный язык описания внешнего вида веб-страницы, написанного с использованием языка разметки (HTML).

React – JavaScript-библиотека для создания пользовательских интерфейсов.

HTML – стандартизованный язык разметки для просмотра веб-страниц в браузере.

2.2 Обзор аналогов

На рынке существует множество приложений для создания путеводителей и планирования путешествий. Для определения функционала нашего веб-приложения "TripToVoronezh", мы рассмотрели наиболее популярные по рейтингам и количеству пользователей решения. Далее мы выделили их достоинства и недостатки, чтобы определить, чего может не хватать пользователям существующих решений.

2.2.1 TripAdvisor

TripAdvisor — это популярный веб-сайт для планирования путешествий, который предоставляет пользователям отзывы и рейтинги отелей, ресторанов, достопримечательностей и других туристических объектов. Он помогает путешественникам находить и бронировать лучшие варианты для их поездок, основываясь на опыте других пользователей.

Достоинства:

- Очень большая база данных по городам/странам;
- Есть связь с другими людьми и можно читать об их впечатлениях о посещении каких-либо мест/заведений.

Минусы:

- Возможность накрутки отзывов;
- Информация может быть неактуальной;
- Проблемы структурированности информации;
- Может не быть некоторых мест, не популярных или малоизвестных среди туристов.

Куда Вы едете?

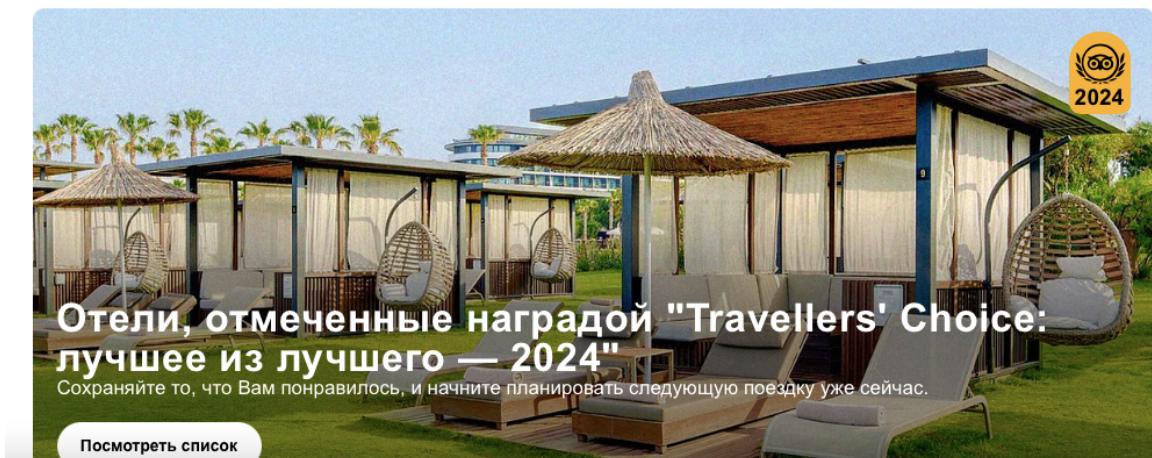
[Найти все](#) [Отели](#) [Развлечения](#) [Рестораны](#) Куда сходить, чем заняться, где остановиться...

Рисунок 1 - Главная страница в приложении TripAdvisor

2.2.2 Туристер

Туристер — это российский веб-сайт для путешественников, предлагающий информацию о различных туристических направлениях, достопримечательностях и маршрутах. Сайт также включает отзывы, советы и отчеты от пользователей, что помогает планировать поездки и находить интересные места.

Достоинства:

- Очень большая база данных по городам/странам;
- Доступ к сторонним сервисам (букинг, такси, трансферы, страховка, авиаторы, аренда такси и др.).

Минусы:

- Неудобный, перегруженный интерфейс;
- Большое количество навязчивой рекламы;
- Отсутствие регулярного обновления информации;
- Краткость и неинформативность гидов и путеводителей.

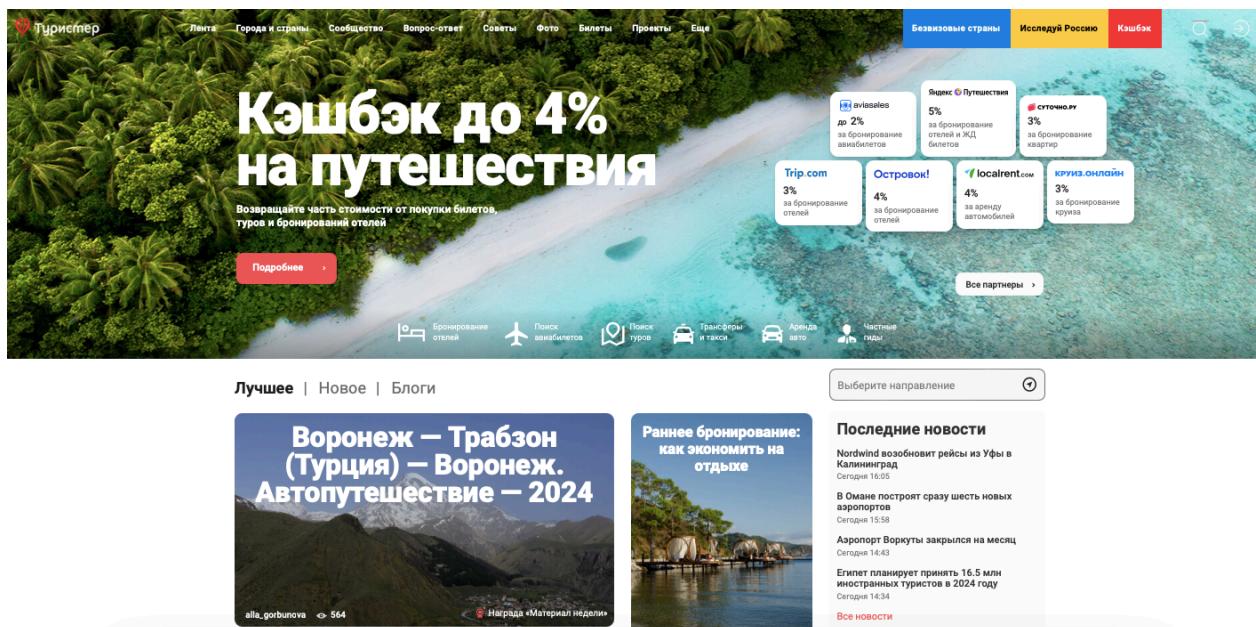


Рисунок 2 - Главный экран в приложении Туристер

2.2.3 VRN.Guide

"VRN Guide. Гид по Воронежу" — это группа в VK, предоставляющая информацию о достопримечательностях, событиях и интересных местах Воронежа. Участники группы могут получать рекомендации, читать отзывы и делиться своим опытом, что помогает лучше планировать досуг в городе.

Достоинства:

- Общедоступный паблик в соцсети VK;
- Можно общаться с другими людьми из паблика и обмениваться впечатлениями и опытом;
- Удобство интерфейса;
- Регулярное обновление информации и наличие тематических подборок.

Минусы:

- Отсутствие возможности создавать персонализированные путеводители;
- Нет возможности создавать подборки из путеводителей;
- Ограниченнность информационных возможностей ввиду выбора платформы формата соцсети.

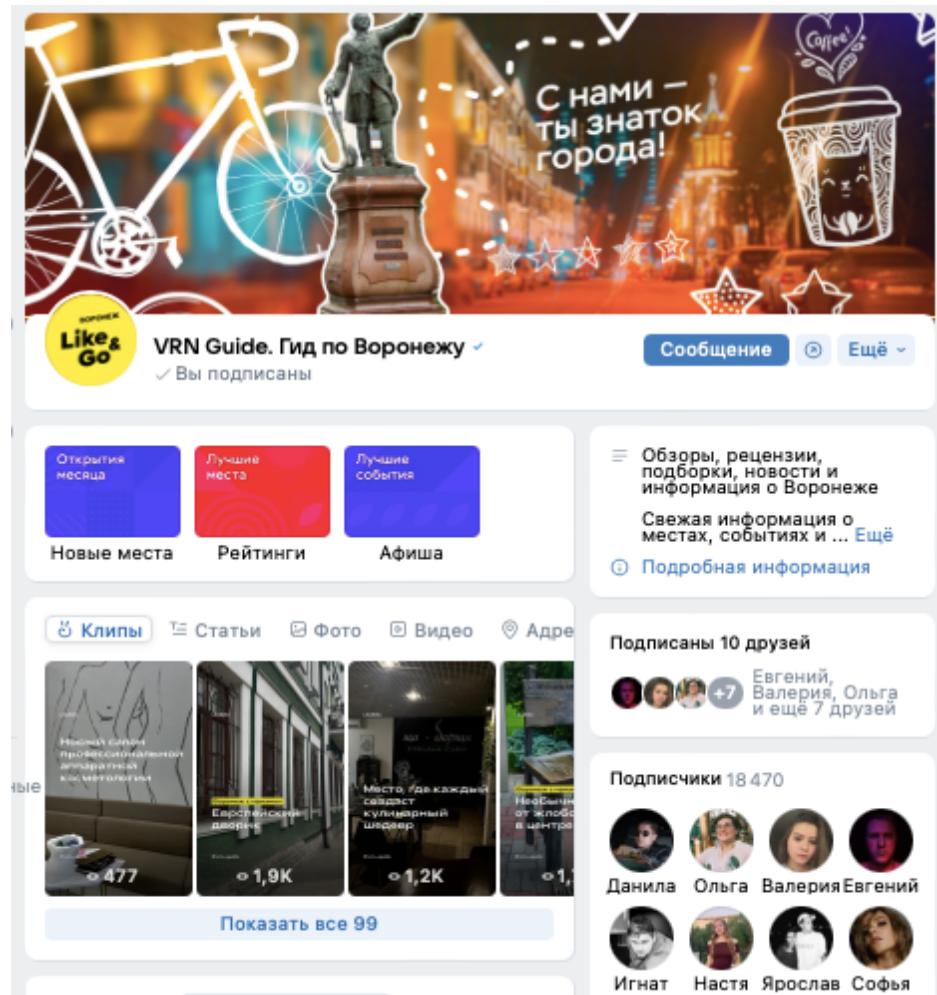


Рисунок 3 - Главный экран в группе VRN.Guide на платформе соцсети VK

2.2.4 TIMEWILL

"TIMEWILL - ваш навигатор по Воронежу" — это группа в VK, которая помогает пользователям ориентироваться в разнообразных событиях, заведениях и достопримечательностях Воронежа. Группа предоставляет актуальную информацию и рекомендации, облегчая планирование досуга и интересных мероприятий в городе.

Достоинства:

- Общедоступный паблик в соцсети VK;

- Можно общаться с другими людьми из паблика и обмениваться впечатлениями и опытом;
- Удобство интерфейса;
- Регулярное обновление информации и наличие тематических подборок.

Минусы:

- Отсутствие возможности создавать персонализированные путеводители;
- Нет возможности создавать подборки из путеводителей;
- Ограниченноть информационных возможностей ввиду выбора платформы формата соцсети.

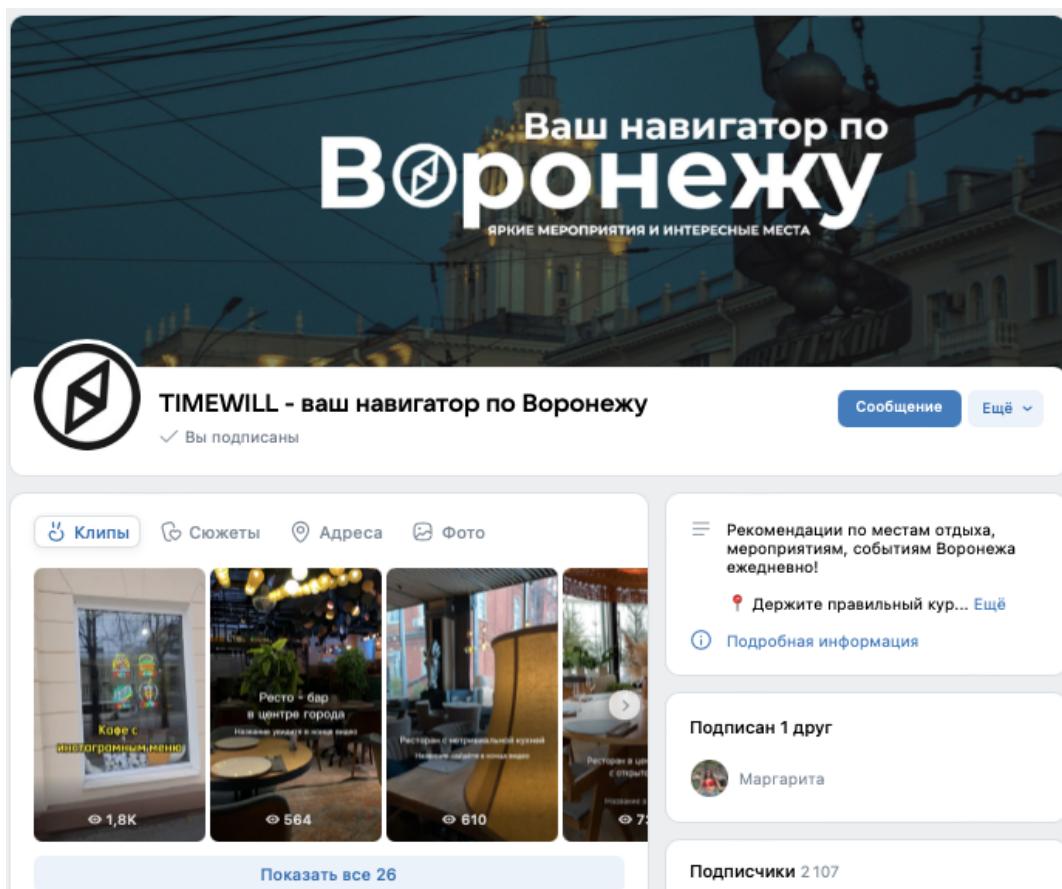


Рисунок 4 - Главный экран в группе TIMEWILL на платформе соцсети VK

2.3 Моделирование системы

2.3.1 Диаграмма прецедентов

Рассмотрим полную диаграмму для использования приложения разными типами пользователей. В данном случае необходимость составления диаграммы прецедентов продиктована прежде всего тем, что use-case диаграмма — это инструмент для моделирования системы и понимания ее функциональности и потребностей пользователей. Они помогают в определении основных действий, которые пользователь должен совершить в системе, чтобы достичь определенных целей. Они также позволяют определить возможные риски и проблемы, которые могут возникнуть в ходе использования системы. Данные диаграммы представлены на рисунках 5 и 6.

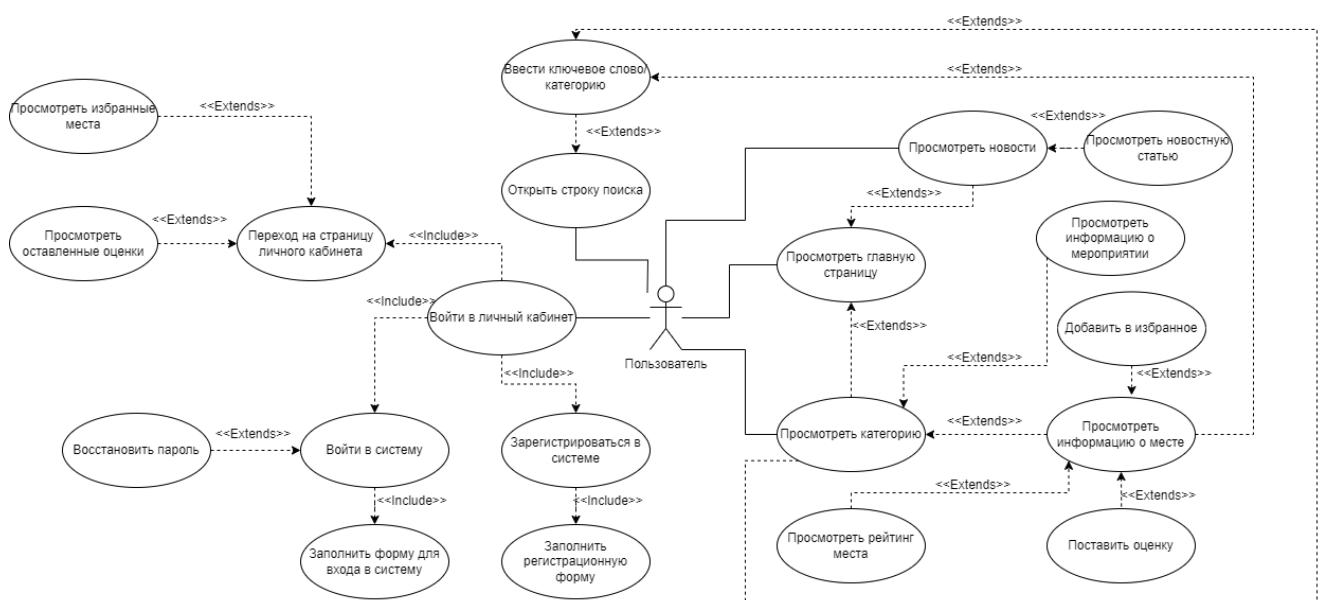


Рисунок 5 - Диаграмма use-case со стороны Клиента

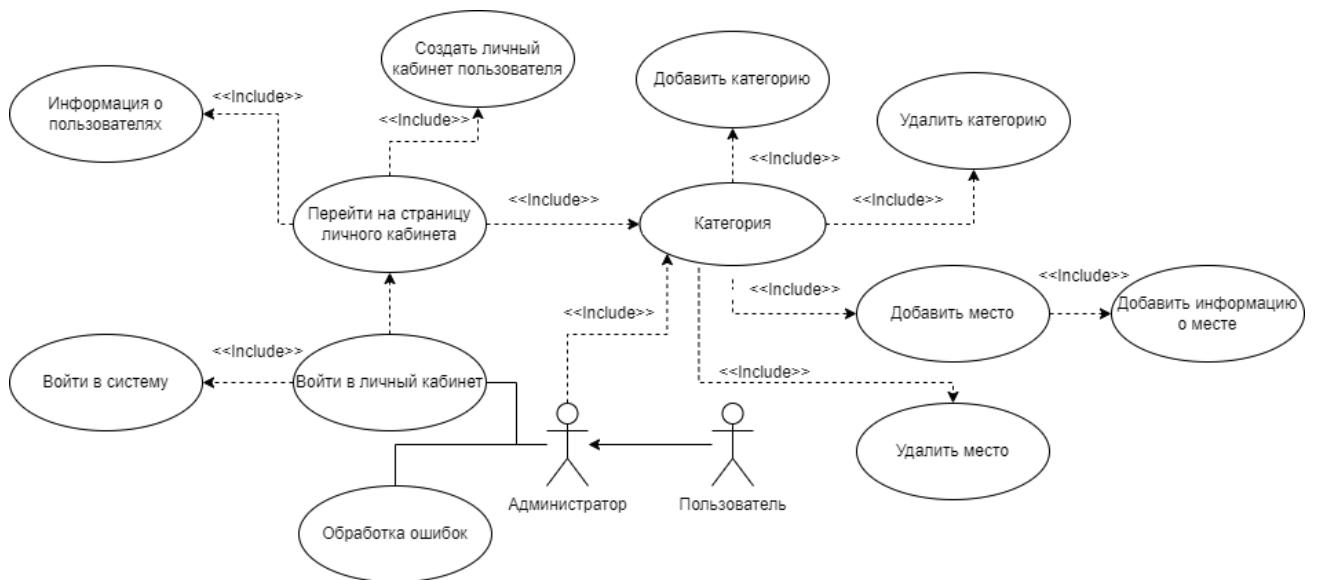


Рисунок 6 - Диаграмма use-case со стороны Администратора

2.3.2 Диаграмма последовательности

Диаграмма последовательности является важным инструментом для проекта, который помогает более глубоко понимать процесс, улучшать его эффективность и упрощать взаимодействие.

Рассмотрим диаграмму последовательности в приложении. Данная диаграмма представлена на рисунке 7.

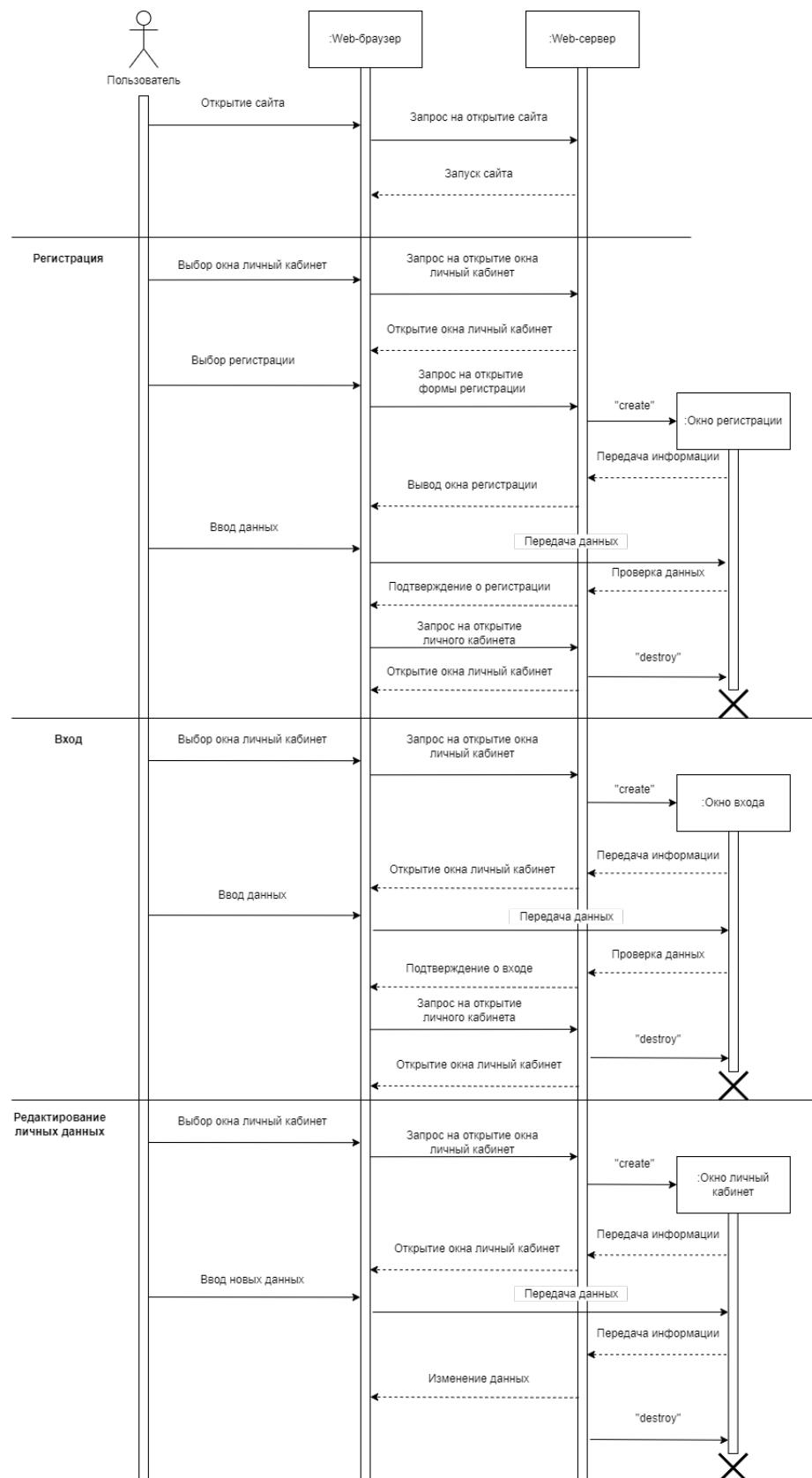


Рисунок 7 - Диаграмма последовательности

2.3.3 Диаграмма активности

Диаграмма активности помогает разработчикам лучше понимать процессы в системе, выявлять узкие места и оптимизировать их. Она также может использоваться для описания бизнес-процессов и управления проектами. Для данного проекта были спроектирована диаграмма активности, представленная на рисунке 8.

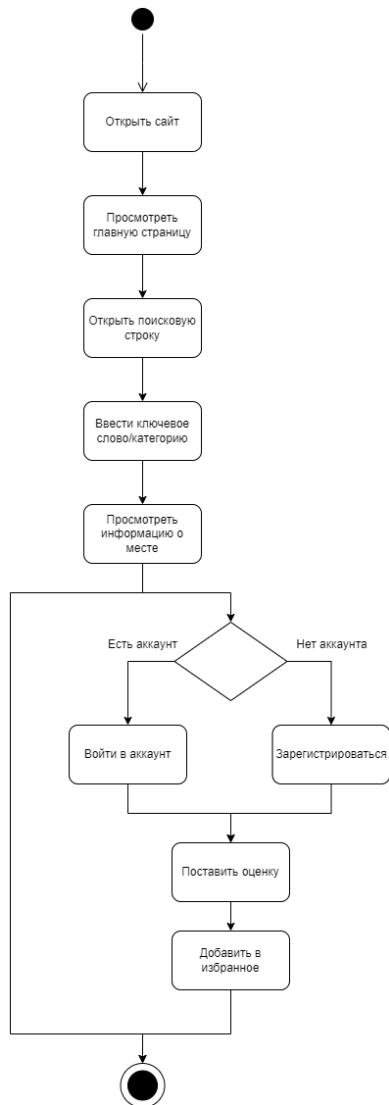


Рисунок 8 - Диаграмма активности

3 Реализация

3.1 Средства реализации

Ниже приведен перечень используемых технологий.

Backend:

- Java;
- Spring Boot Framework;
- PostgreSQL;
- Hibernate;
- Docker.

Frontend:

- Java Script;
- React;
- HTML;
- CSS.

Инструменты для ведения документации:

- Miro;
- Draw.io;
- Figma.

Дополнительный инструментарий:

- Git;
- GitHub;
- Trello.

3.2 Реализация базы данных

Для хранения данных была выбрана база данных PostgreSQL. Она является продуктом с открытым исходным кодом, который поддерживается многими серверами. PostgreSQL поддерживает множественные типы данных, такие как числа разной точности, тексты с различными кодировками, изображения, звуки, видео, XML-документы, JSON-объекты и многие другие.

3.2.1 ER-диаграмма

ER-диаграмма используется для описания структуры базы данных, проектирования новых баз данных и анализа существующих баз данных. Она позволяет разработчикам и аналитикам лучше понять логические связи и зависимости между различными сущностями и их атрибутами. Данная диаграмма представлена на рисунке 9.

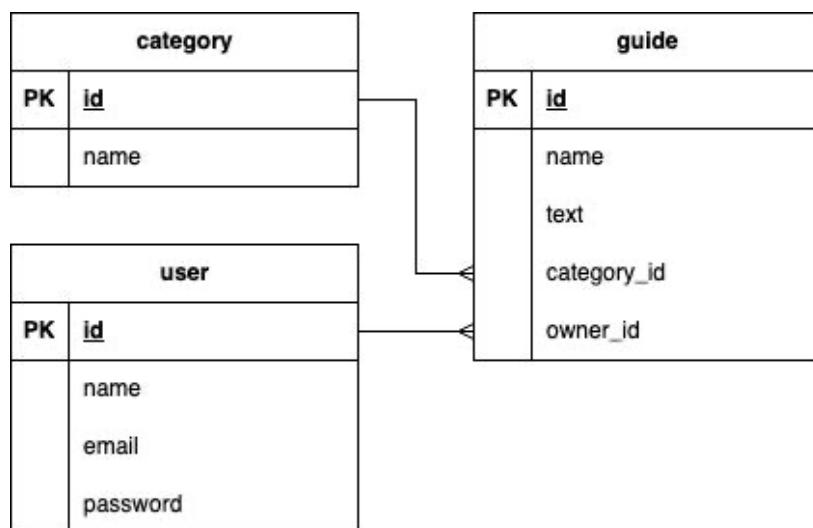


Рисунок 9 - ER-диаграмма

3.3 Реализация клиентской части

Для реализации клиентской части приложения "TripToVoronezh" было выбрано сочетание JavaScript и библиотеки React. Этот набор обеспечивает разработчикам множество возможностей для создания современных и интерактивных веб-приложений. JavaScript, как высокоуровневый язык программирования, предоставляет гибкость и широкие функциональные возможности для разработки динамичных пользовательских интерфейсов. React, в свою очередь, является мощной библиотекой для создания компонентных интерфейсов, что позволяет разработчикам создавать масштабируемые и легко поддерживаемые приложения.

Клиентская часть имеет архитектуру, основанную на модели компонентного подхода. Компоненты React обрабатывают пользовательские действия и обновляют интерфейс в соответствии с изменениями состояния приложения. Они являются интерфейсом между пользователем и бизнес-логикой приложения. Компоненты могут быть функциональными или классовыми, в зависимости от сложности и необходимости управления состоянием.

Для управления состоянием приложения используется библиотека Redux. Это позволяет централизовать состояние и облегчает управление данными в приложении. Состояние Redux хранит информацию о пользователях, интересах, маршрутах и достопримечательностях. Действия (actions) и редюсеры (reducers) обрабатывают изменения состояния и обновляют данные в хранилище (store).

Взаимодействие с сервером осуществляется с использованием библиотеки Axios для выполнения HTTP-запросов. Это позволяет получать данные с сервера и отправлять их обратно, обеспечивая функциональность приложения. Запросы могут включать получение данных о достопримечательностях, созданных маршрутах и персонализированных рекомендациях для пользователей.

Компоненты React взаимодействуют с хранилищем состояния Redux и сервисами Axios для обеспечения полной функциональности веб-приложения. Компоненты обрабатывают пользовательские действия, направляют их в Redux для управления состоянием, и при необходимости выполняют запросы к серверу через Axios.

3.3.1 Форма для поиска гидов

Пользователь (авторизованный) имеет возможность найти гид для просмотра по названию или сделать выборку на основе предложенных фильтров. В качестве ответа на его запрос будет выведен список гидов. Пользователь может нажать на гид, чтобы просмотреть его.

Зарегистрированный пользователь имеет возможность добавить понравившиеся гиды себе в “Мои гиды”.

Форма данного экрана представлена на рисунке 10.

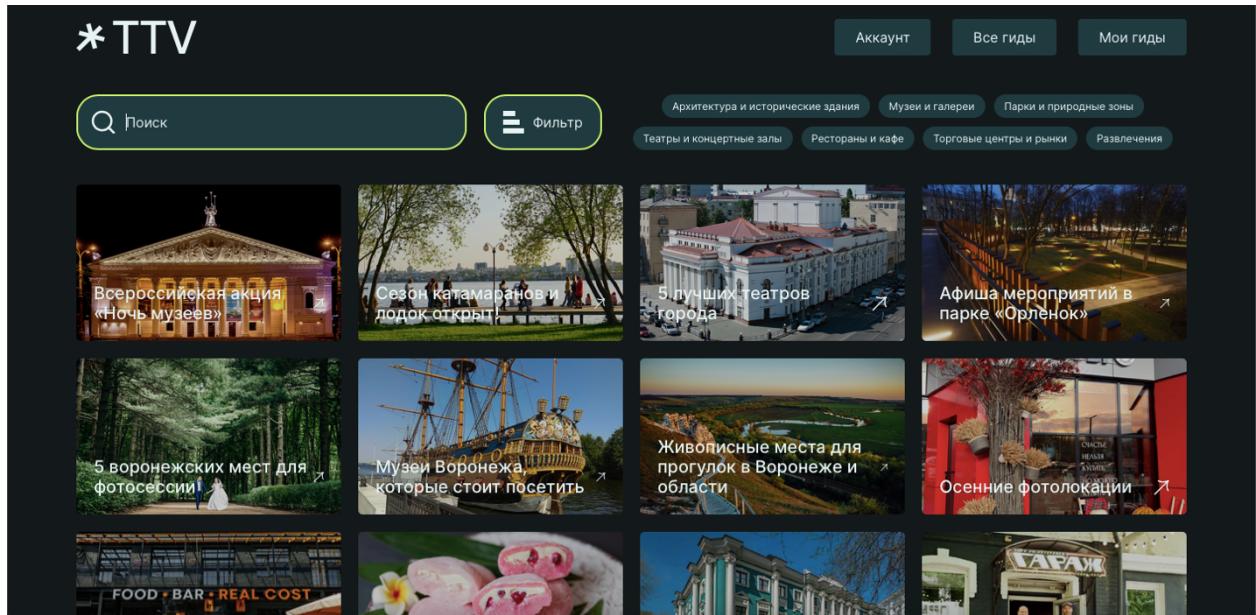


Рисунок 10 - Форма экрана поиска гидов

3.3.2 Форма экрана доступных гидов

Пользователь (авторизованный) имеет возможность увидеть все доступные для просмотра гиды, нажав на главном экране кнопку “Посмотреть все”. В качестве ответа на его запрос будет выведен список гидов. Пользователь может нажать на гид, чтобы просмотреть его.

Сверху расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: главная, аккаунт, все гиды, мои гиды.

Форма данного экрана представлена на рисунке 11.

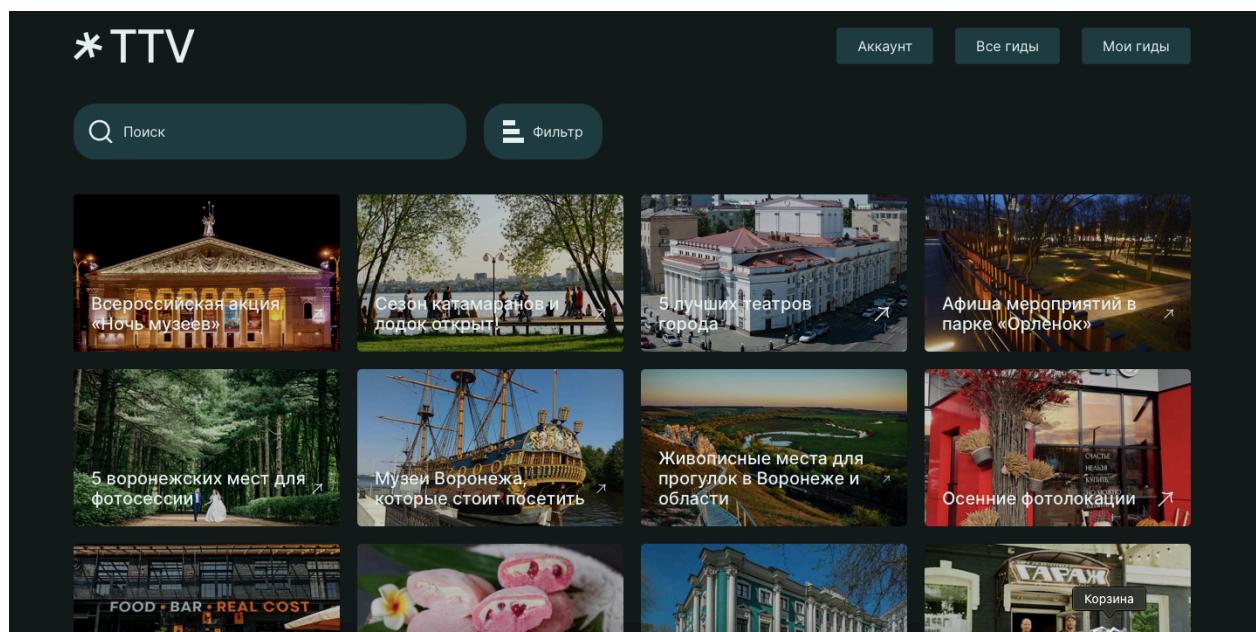


Рисунок 11 - Форма экрана просмотра доступных гидов

3.3.3 Форма экрана авторизации

На данном экране отображены поля с вводом логина (Email) и пароля к аккаунту, кнопка войти в аккаунт, а также кнопка зарегистрироваться.

После входа в аккаунт пользователь переходит на главную страницу.

Форма данного экрана представлена на рисунке 12.

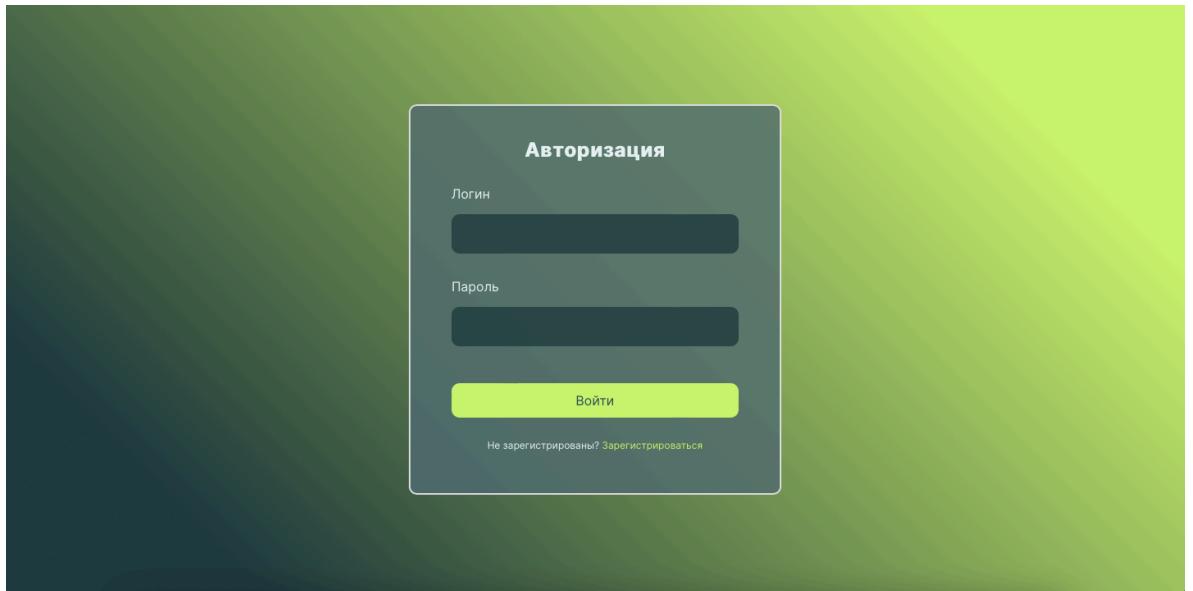


Рисунок 12 - Форма экрана авторизации

3.3.4 Форма экрана регистрации

На данном экране отображены поля для ввода имени, Email, пароля. Ниже расположены кнопки регистрации и перехода в аккаунт. Форма данного экрана представлена на рисунке 13.

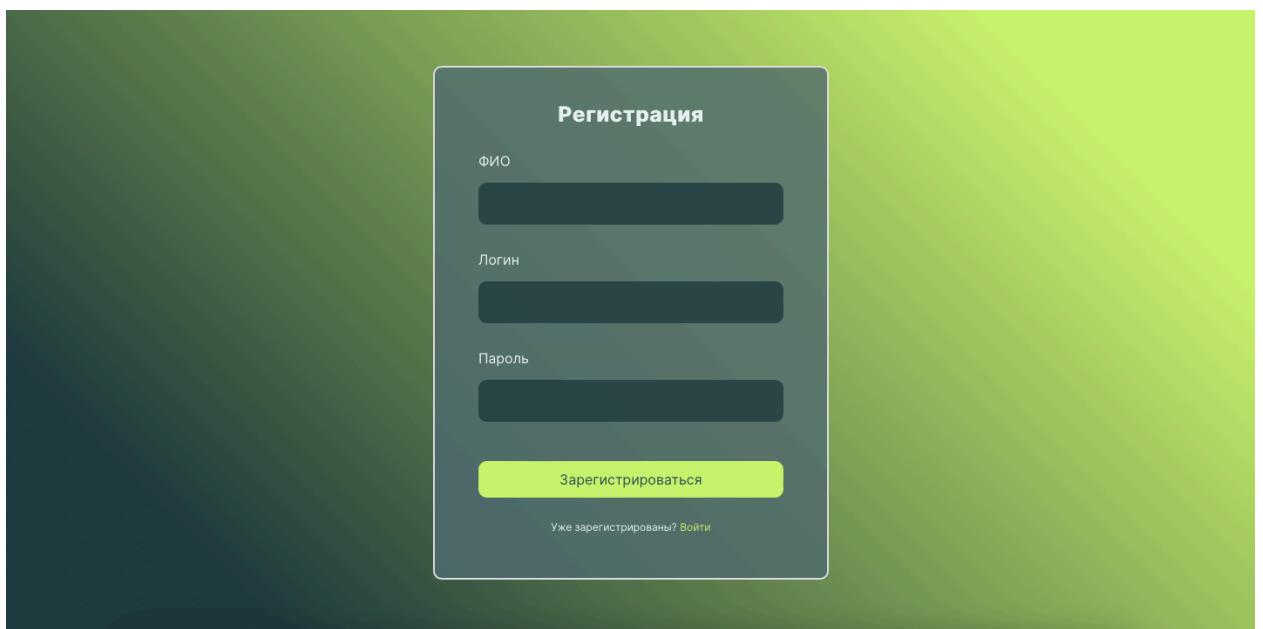


Рисунок 13 - Форма экрана регистрации

3.3.5 Форма экрана личной страницы

На данном экране указана информация о пользователе (почта). Слева расположено фото владельца аккаунта. Ниже находится кнопка для выхода из профиля.

Сверху расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: главная, , аккаунт, все гиды, мои гиды.

Форма данного экрана представлена на рисунке 14.

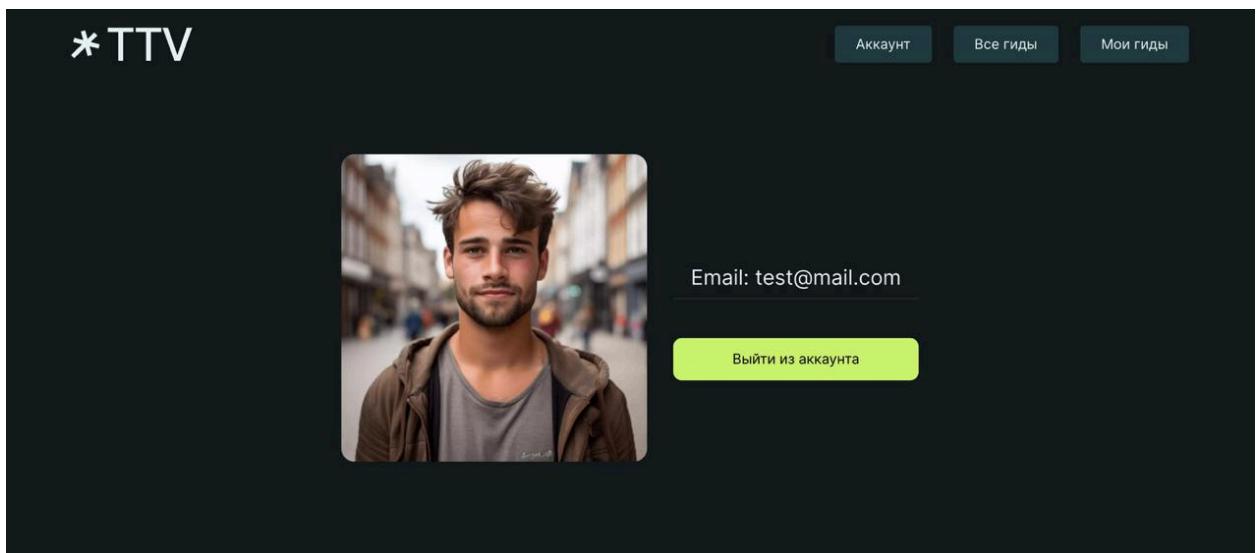


Рисунок 14 - Форма экрана личной страницы

3.3.6 Форма экрана сохранённых гидов

На данном экране расположен список гидов, сохранённых в “Мои гиды”.

Сверху расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: главная, аккаунт, все гиды, мои гиды

Форма данного экрана представлена на рисунке 15.

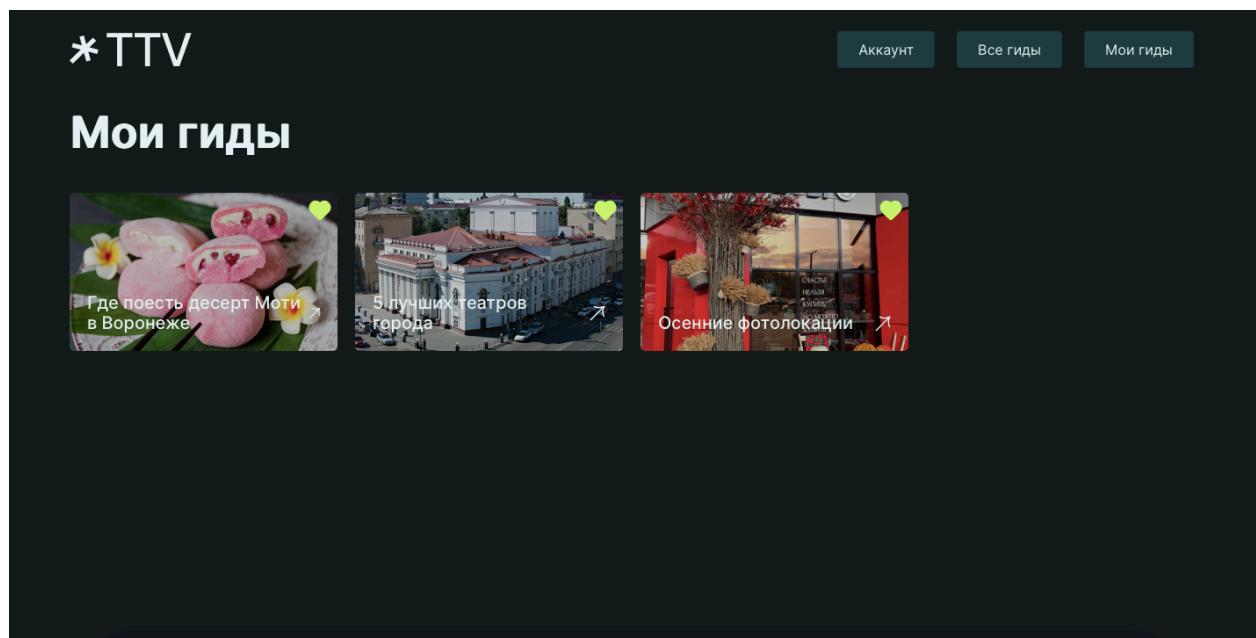


Рисунок 15 - Форма экрана сохранённых гидов

3.3.7 Форма главного экрана

На данном экране расположена главная страница.

Сверху расположена панель с кнопками, при нажатии на соответствующую кнопку пользователь переходит на страницы: главная (обновление страницы), аккаунт, все гиды, мои гиды.

В центральной части экрана располагается подборка гидов. Слева от неё располагается кнопка для переход на экран просмотра доступных гидов.

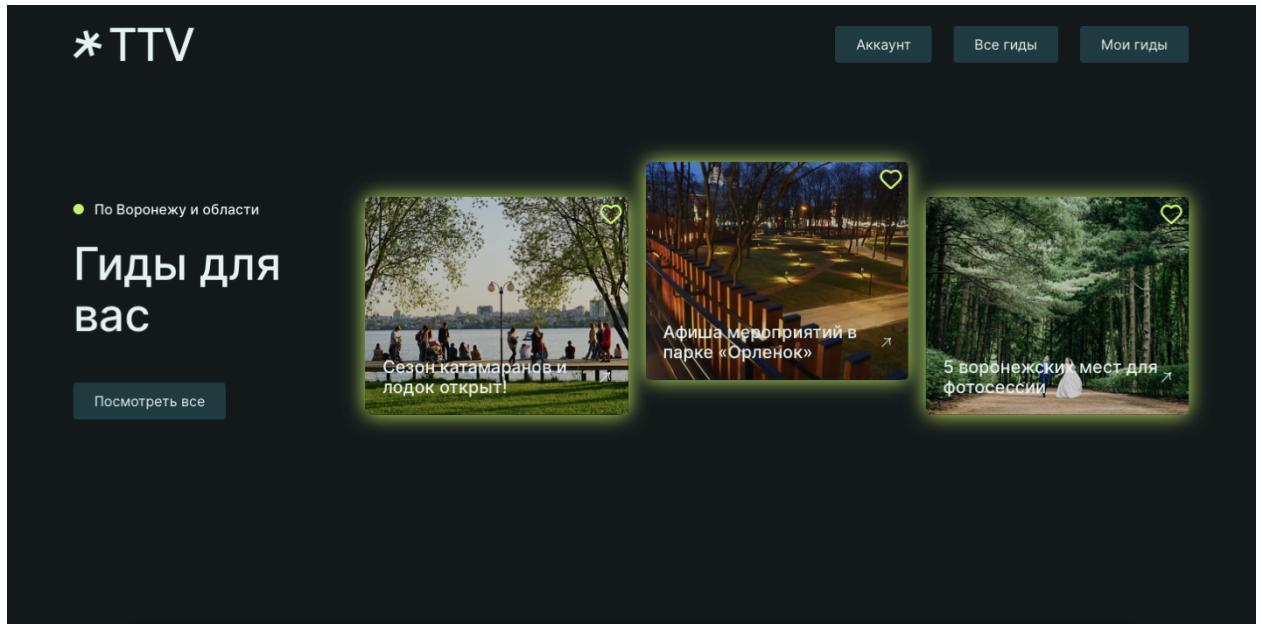


Рисунок 16 - Форма главного экрана

3.4 Реализация серверной части

В качестве языка был выбран строго типизированный объектно-ориентированный язык программирования Java. Он остается очень популярным языком программирования в разработке благодаря своим мощным возможностям и широкому спектру инструментов для разработки. К тому же существует огромное количество фреймворков и библиотек, написанных на Java, которые в перспективе можно легко интегрировать в проект.

Также применяется программная платформа Docker для тестирования и развертывания приложений.

3.4.1 Архитектура серверной части приложения

Серверная часть приложения реализована соответственно трехслойной архитектуре веб-приложения с API Rest с использованием фреймворка Spring Boot. Данный фреймворк предоставляет возможности для работы с базами данных, а также механизм внедрения зависимостей, который позволяет сделать компоненты программы слабосвязанными, а всю программу в целом более гибкой, адаптируемой и расширяемой.

4 Аналитика

Для сбора метрик для веб-приложения была выбрана Яндекс.Метрика от Яндекс. Это система для сбора данных об использовании приложения пользователями. Она отличается высокой скоростью и удобством настройки метрик для веб и мобильных приложений. Кроме того, она предлагает интуитивный интерфейс и понятное руководство по использованию.

Заключение

В ходе выполнения данного курсового проекта был выполнен анализ предметной области и аналогов разрабатываемого приложения.

Для разработки приложения были разработаны макеты интерфейса, выбрана платформа приложения, построены UML диаграммы.

Для контроля версий был создан репозиторий GitHub.

При разработке приложения были реализованы следующие задачи:

- Систематизация информации о местах для посещения;
- Обеспечение возможности пользователям ознакомиться с широким спектром досуга (рестораны, музеи, кинотеатры, памятники и др);
- Обеспечение возможности пользователям создавать персонализированные путеводители на основе разнообразных фильтров;
- Обеспечение возможности пользователям добавлять интересные для посещения места и гиды в «Мои гиды»;
- Разработка удобного и интуитивно понятного интерфейса для пользователей.

Backend часть приложения и база данных были размещены в контейнере Docker на хостинге.

Разработанное приложение удовлетворяет поставленным требованиям.

Все поставленные задачи были выполнены.

Список использованных источников

1. Петров, А. В. Разработка веб-приложений на JavaScript: основы и практическое применение. — Москва: Издательство "Наука и Техника", 2023. — 356 с.
2. Иванова, Е. С., Сидоров, М. А. Городские путеводители и их роль в туризме. // Журнал "Туризм и Культура", 2022, №4, с. 45-58.
3. Лебедев, К. П. Воронежская область: туристические маршруты и достопримечательности. — Воронеж: Издательство ВГУ, 2023. — 298 с.
4. Смирнов, В. И. Использование информационных технологий в туризме. — СПб: Издательство Политехнического университета, 2022. — 240 с.
5. TripAdvisor. (2024). Достопримечательности Воронежа. Получено 15 мая 2024 г., с https://www.tripadvisor.ru/Attractions-g798124-Activities-Voronezh_Voronezh_Oblast_Central_Russia.html
6. Туристер. (2024). Воронежская область: лучшие туристические маршруты. Получено 1 апреля 2024 г., с <https://www.tourister.ru/world/europe/russia/city/voronezh>
7. Никифорова, Т. Г. Разработка пользовательских интерфейсов для веб-приложений. — Казань: Издательство КФУ, 2023. — 310 с.
8. Воронежская область. Туристический портал. (2024). Маршруты и достопримечательности Воронежа и области. Получено 20 апреля 2024 г., с <https://www.tourism-vrn.ru/routes>
9. Культура Воронежа. (2024). Культурные мероприятия и экскурсии в Воронеже. Получено 29 мая 2024 г., с <https://www.kultura-vrn.ru/events>