

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ВЕБ-
ПРИЛОЖЕНИЯ ДЛЯ ТУРИСТИЧЕСКОЙ ПЛАТФОРМЫ «ОТКРОЙ
САРАТОВ»**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы

направления 02.03.02 — Фундаментальная информатика и информационные
технологии

факультета КНиИТ

Блохиной Анастасии Дмитриевной

Научный руководитель

зав. каф. техн. пр., к. ф.-м. н., доцент _____

И. А. Батраева

Заведующий кафедрой

к. ф.-м. н. _____

С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

В условиях стремительного развития информационных технологий наблюдается значительный рост популярности веб-приложений, особенно в сфере туризма и путешествий. Современные пользователи ожидают интуитивно понятных, удобных и эстетически привлекательных интерфейсов. Сложные и неудобные решения могут отпугнуть пользователей, снижая их вовлеченность и удовлетворенность сервисом. Поэтому проектирование и разработка эффективного пользовательского интерфейса (UI), обеспечивающего комфортное взаимодействие, становятся ключевыми факторами успеха любого цифрового продукта.

В рамках учебной программы «Стартап как диплом» была инициирована разработка туристической платформы «Открой Саратов». Этот проект реализован командой разработчиков и представляет собой единую агрегатную платформу, которая позволит пользователям — как туристам, так и жителям города — быстро и эффективно находить интересные места, маршруты и мероприятия. Кроме того, данное решение будет полезно для владельцев организаций и экскурсоводов, предоставляя функционал для размещения информации, что поможет привлечь новых клиентов.

Целью данной дипломной работы является разработка интуитивно понятного и привлекательного пользовательского интерфейса для туристического веб-приложения «Открой Саратов», который улучшит пользовательский опыт и повысит интерес к туристическим возможностям города.

Для достижения поставленной цели, необходимо выполнить следующие задачи:

- провести анализ предметной области и анализ существующих решений;
- разработать техническую концепцию (структуру) для интерфейса;
- сформулировать требования к проекту;
- разработать дизайн веб-приложения;
- осуществить анализ и выбор инструментов для разработки;
- осуществить программную реализацию пользовательского интерфейса.

Структура и объем работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 5 приложений. Общий объем работы — 64 страниц, из них 42 страниц — основное содержание, включая 6 рисунков, цифровой носитель в качестве приложения, список

использованных источников информации — 20 наименований.

Первый раздел «Анализ предметной области» посвящен исследованию актуальности и проблематики внутреннего туризма в России, а также анализу существующих решений в данной сфере. В данном разделе были также выявлены и сформулированы требования к проекту, рассмотрены разные виды архитектурных стилей и выявлен технологический стек для разработки платформы.

Второй раздел «Разработка программной части» посвящен реализации пользовательского интерфейса веб-приложения «Открой Саратов». В разделе подробно описывается процесс разработки ключевых страниц: регистрации, авторизации, поиска, личного кабинета пользователя, а также слоя работы с API и системы управления сессиями.

Основное содержание работы

Первый раздел «Анализ предметной области» посвящен исследованию актуальности и проблематики внутреннего туризма в России, а также анализу существующих решений в данной сфере. В разделе рассматриваются недостатки текущих платформ, таких как TripAdvisor, социальные сети и картографические сервисы, и обосновывается необходимость создания специализированной туристической платформы для Саратова. Приводятся функциональные и нефункциональные требования к продукту, включая требования к пользовательскому интерфейсу. Также анализируются различные архитектурные стили (монолитная, микросервисная, клиент-серверная, MVC) и обосновывается выбор архитектуры MVC для проекта. Рассматривается технологический стек, включая инструменты для дизайна (Figma), веб-разработки (React, TypeScript) и библиотеки для реализации функционала (Zod, Zustand, TanStack Router).

Постановка задачи. Концепция проекта «Открой Саратов» — создание платформы-агрегатора, которая будет собирать и структурировать информацию о туристических услугах и мероприятиях в Саратове. Ключевыми преимуществами данной платформы станут удобный интерфейс, структурированная база данных и возможность экономии времени пользователей при поиске нужной информации. Кроме того, платформа предоставит новые возможности для бизнеса, включая заведения и организаторов мероприятий, которые смогут размещать свои предложения на едином ресурсе.

Целевыми группами данного проекта являются как туристы и местные жители, так и заведения и организаторы мероприятий, которые выступают в роли поставщиков контента. Ожидаемый эффект от реализации проекта включает развитие туристической сферы в регионе, культурный рост и экономические выгоды для города.

Анализ существующих решений. В разделе «Анализ конкурентов» исследуются существующие платформы, предоставляющие туристические услуги, и выявляются их слабые стороны. Рассматриваются три категории решений: универсальные агрегаторы («TripAdvisor»), социальные сети (Telegram-каналы) и картографические сервисы («Яндекс Карты»). Несмотря на их популярность, выявлены ключевые недостатки: отсутствие структурированных данных, ориентация на массовый туризм, избыток рекламы и отсутствие специализированных функций для локальных маршрутов. Эти ограничения создают проблемы для

пользователей, ищущих актуальную информацию о Саратове.

На основе анализа определены конкурентные преимущества платформы «Открой Саратов»: глубокая локализация контента, удобный интерфейс и интеграция лучших функций аналогов (фильтрация, сохранение избранного, визуальная подача). Также проект фокусируется на уникальных местах города, предлагая как туристам, так и бизнесу единый инструмент для планирования и продвижения. Также рассматриваются потенциальные риски (нехватка контента, низкая активность пользователей) и способы их минимизации через партнерства и геймификацию.

Описание требований для продукта. В данном разделе сформулированы ключевые критерии, которым должна соответствовать туристическая платформа «Открой Саратов». Требования разделены на две категории: функциональные, которые определяют, что система должна делать и нефункциональные, которые описывают, как она должна работать. К функциональным относятся регистрация пользователей разных типов (персона, гид, организация), работа с личным кабинетом, поиск и фильтрация мест, создание маршрутов и управление контентом. Каждая функция детализирована: например, поиск должен поддерживать фильтры по категориям, рейтингу и популярности, а личный кабинет — позволять редактировать профиль и сохранять избранные места.

Нефункциональные требования сосредоточены на качестве пользовательского интерфейса и удобстве взаимодействия. Интерфейс должен быть интуитивно понятным, с четкой навигацией, читаемыми шрифтами и минималистичным дизайном. Особое внимание уделено цветовой гамме (синие, голубые и белые тона), которая создает приятную визуальную атмосферу. Дополнительно подчеркивается важность контрастности элементов, чтобы текст и ключевые объекты выделялись на фоне. Эти требования направлены на обеспечение комфортного взаимодействия пользователей с платформой.

Для реализации требований разработаны макеты ключевых страниц: главной, регистрации, авторизации, поиска, личного кабинета и профиля организации. Каждая страница имеет четкую структуру, включая шапку, основную область и подвал, а также интерактивные элементы (кнопки, формы, карточки мест). Например, страница поиска содержит строку поиска, фильтры и сетку карточек с результатами, а личный кабинет — разделы для сохраненных мест и маршрутов. Такой подход обеспечивает единообразие дизайна и упрощает

навигацию, что соответствует как функциональным, так и нефункциональным требованиям.

Архитектурные стили разработки. В данном разделе проводится сравнительный анализ различных подходов к построению веб-приложений, обосновывая выбор оптимальной архитектуры для туристической платформы «Открой Саратов». Рассматриваются четыре ключевых архитектурных стиля: монолитная, микросервисная, клиент-серверная и MVC (Model-View-Controller). Каждый из них оценивается с точки зрения масштабируемости, производительности и удобства разработки, учитывая специфику проекта.

В качестве оптимального решения для проекта выбрана архитектура MVC, которая обеспечивает четкое разделение ответственности между компонентами. Модель отвечает за данные и бизнес-логику, Представление — за отображение информации, а Контроллер — за обработку пользовательских действий. Такой подход позволяет команде работать параллельно над разными частями системы, упрощает тестирование и обеспечивает гибкость для будущего расширения функциональности. Архитектура MVC идеально соответствует требованиям проекта, сочетая простоту разработки с возможностью масштабирования.

Технологический стек

Создание дизайна. В данном разделе подробно рассматривается процесс проектирования пользовательского интерфейса платформы «Открой Саратов». Основное внимание уделено выбору инструментов и методологии разработки дизайн-системы. В качестве ключевого инструмента выбрана платформа Figma, которая позволяет создавать интерактивные прототипы, обеспечивает удобную командную работу и поддерживает контроль версий. Этот выбор обусловлен богатым функционалом Figma, включающим векторные инструменты, компоненты, стили и возможности прототипирования.

Процесс проектирования начался с разработки wireframes — схематичных макетов, определяющих расположение основных элементов интерфейса. Особое внимание уделялось продумыванию пользовательских сценариев для разных типов пользователей (туристы, гиды, организации). На этом этапе были определены ключевые экраны приложения: главная страница, поиск, личный кабинет и страницы профилей, каждый из которых проектировался с учетом принципов юзабилити и визуальной иерархии.

Для обеспечения единообразия интерфейса была создана комплексная

дизайн-система, включающая цветовую палитру (основанную на синих и голубых тонах), типографику, набор UI-компонентов (кнопки, карточки, формы ввода) и правила их использования. Все элементы системы разрабатывались с учетом требований доступности и отзывчивости, чтобы интерфейс корректно отображался на различных устройствах. Особый акцент сделан на визуальной простоте и минимализме, что соответствует концепции удобного туристического сервиса.

Инструменты веб-верстки. В проекте «Открой Саратов» для реализации пользовательского интерфейса была выбрана связка React и TypeScript. React, как декларативная JavaScript-библиотека, обеспечила эффективное создание компонентного интерфейса с виртуальным DOM, что позволило добиться высокой производительности при работе с динамическим контентом. Компонентный подход React идеально подошел для построения сложного интерфейса с множеством интерактивных элементов.

TypeScript добавил в проект строгую типизацию, что значительно повысило надежность кодовой базы. Статическая проверка типов на этапе разработки помогла выявлять потенциальные ошибки до запуска приложения. Особенно полезной типизация оказалась при работе с:

- пропсами компонентов;
- состояниями приложения;
- структурами данных API;
- параметрами маршрутизации.

Совместное использование React и TypeScript создало мощный фундамент для разработки. TypeScript-типы для React-компонентов обеспечили:

1. Автодополнение и подсказки в IDE;
2. Защиту от ошибок при рефакторинге;
3. Четкую документацию интерфейсов компонентов;
4. Безопасную передачу данных между частями приложения.

Библиотеки для разработки. В данном разделе рассматриваются ключевые инструменты, дополняющие основной стек технологий и расширяющие функциональность платформы. Первой из них является TanStack Router (ранее React Location) — современная библиотека для маршрутизации в React-приложениях. Она предлагает продвинутые возможности, включая типизированные маршруты, параллельную загрузку данных и вложенную маршрутиза-

цию, что особенно важно для сложных SPA-приложений с множеством экранов.

Для работы с формами выбрана связка библиотек Zod и react-hook-form, обеспечивающая надежную валидацию данных. Zod позволяет декларативно описывать схемы данных с TypeScript-типами, а react-hook-form эффективно управляет состоянием форм, минимизируя количество повторных рендеров. Такое сочетание гарантирует как безопасность данных на клиентской стороне, так и удобство взаимодействия для пользователей при заполнении различных форм приложения.

Управление глобальным состоянием реализовано с помощью библиотеки Zustand — легковесного решения, альтернативного Redux. Zustand отличается простым API, основанным на хуках, и высокой производительностью за счет отсутствия избыточных ререндеров. В проекте она используется для хранения данных о сессиях пользователей, избранных местах и состоянии UI, обеспечивая предсказуемость работы с распределенными данными в разных компонентах приложения.

Завершает технологический стек Nginx — высокопроизводительный веб-сервер, выступающий в роли обратного прокси для API. Он решает несколько важных задач: обслуживание статических файлов фронтенда, балансировка нагрузки, маршрутизация запросов к API и поддержка HTTPS. Такая конфигурация обеспечивает безопасность, отказоустойчивость и высокую скорость работы платформы при любом количестве одновременных пользователей.

Вывод по первому разделу: проведенный анализ показал, что существующие решения не удовлетворяют потребностям пользователей в области локального туризма. Разработанные требования и выбранная архитектура MVC обеспечивают гибкость, масштабируемость и удобство разработки. Использование современных технологий (React, TypeScript) и инструментов (Figma) позволяет создать интуитивно понятный и функциональный интерфейс.

Разработка программной части

Второй раздел «Разработка программной части» посвящен реализации пользовательского интерфейса веб-приложения «Открой Саратов». В разделе подробно описывается процесс разработки ключевых страниц: регистрации, авторизации, поиска, личного кабинета пользователя, а также слоя работы с API и системы управления сессиями. Основное внимание уделено самостоятельной работе студента:

- Разработаны интерактивные формы регистрации и авторизации с валидацией данных с использованием Zod и react-hook-form;
- Реализована страница поиска с фильтрацией по категориям, рейтингу и популярности, а также автодополнением запросов;
- Создан личный кабинет пользователя с динамическим отображением информации в зависимости от типа профиля (персона, гид, организация);
- Разработан слой работы с API для взаимодействия с серверной частью, включая обработку ошибок и авторизацию;
- Реализована система управления сессиями пользователей с использованием Zustand для хранения состояний.

В данном разделе веб-приложения» детально описывается процесс создания клиентской части платформы «Открой Саратов». Разработка велась с использованием компонентного подхода React и TypeScript, что позволило создать модульную, легко поддерживаемую архитектуру интерфейса. Основное внимание уделено реализации ключевых экранов приложения, каждый из которых проектировался с учетом принципов юзабилити и пользовательского опыта.

Главная страница была разработана как визитная карточка платформы, сочетающая информативность и визуальную привлекательность. Она включает три основных секции: шапку с навигацией, центральный блок с логотипом и описанием сервиса, а также тематические карусели с анонсами мероприятий и популярными местами. Особое внимание уделено адаптивности дизайна, обеспечивающей корректное отображение на устройствах с разными разрешениями экрана.

Система аутентификации реализована через отдельный модуль, включающий страницы регистрации и входа. Формы разработаны с учетом разных типов пользователей (персона, гид, организация) и оснащены многоуровневой валидацией полей. Интерфейс регистрации использует пошаговый подход: сначала

выбор типа аккаунта, затем заполнение соответствующих данных, что упрощает процесс для пользователей и снижает количество ошибок при вводе информации.

Система создания маршрутов разработана как многофункциональный конструктор, позволяющий пользователям комбинировать различные места в тематические цепочки. Интерфейс включает возможность добавления описаний, фотографий и построения оптимального пути между точками маршрута. Готовые маршруты можно сохранять для личного использования или публиковать для других пользователей платформы.

Все элементы интерфейса объединены единой дизайн-системой, обеспечивающей визуальную согласованность. Особое внимание уделено микроанимациям и интерактивным элементам (кнопки, переключатели, формы), которые делают взаимодействие с платформой более плавным и приятным. В результате получился современный, отзывчивый интерфейс, полностью соответствующий поставленным функциональным и дизайнерским требованиям.

Слой работы с API. В данном разделе детально описывается архитектура взаимодействия клиентской части приложения с сервером. Основу составляет универсальный модуль `doRequest`, который инкапсулирует всю логику HTTP-запросов, включая обработку авторизации, ошибок и различных форматов данных. Этот модуль реализован как обертка над `Fetch API` и поддерживает основные методы (`GET`, `POST`, `DELETE`), что обеспечивает согласованность всех API-вызовов в проекте.

Ключевой особенностью реализации является строгая типизация запросов и ответов с использованием `TypeScript`. Для каждого API-метода определены соответствующие интерфейсы входных и выходных данных, что позволяет выявлять потенциальные проблемы на этапе разработки. Система типов интегрирована с библиотекой `Zod` для валидации данных, обеспечивая безопасность при работе с серверными ответами и предотвращая ошибки, связанные с несоответствием структур данных.

Модуль API включает специализированные обработчики для разных сценариев:

- Работа с JSON — автоматическая сериализация/десериализация данных;
- Загрузка файлов — использование `FormData` для передачи медиа-контента;

- Кэширование часто используемых данных — оптимизация производительности;
- Повтор запросов при сетевых ошибках — повышение отказоустойчивости.

Для удобства разработки API слой организован по feature-принципу, где каждый функциональный модуль приложения (аутентификация, места, маршруты) имеет собственный набор типизированных методов. Это обеспечивает четкое разделение ответственности и упрощает поддержку кода. Все методы возвращают промисы с типом `Result` из библиотеки `ts-results`, что делает обработку ошибок явной и предсказуемой.

Дополнительно реализована система искусственной задержки ответов (`API_DELAY`), которая активируется в `development`-режиме. Эта функция позволяет имитировать реальные сетевые условия и тестировать поведение интерфейса при медленном соединении. На `production`-сервере запросы перенаправляются через `Nginx`, который выступает как `reverse proxy` и обеспечивает балансировку нагрузки.

В заключении раздела подчеркивается, что разработанный API слой обеспечивает:

- Безопасность за счет типизации и валидации;
- Масштабируемость благодаря модульной структуре;
- Удобство разработки через единый интерфейс запросов;
- Надежность благодаря централизованной обработке ошибок;
- Производительность за счет оптимизированных запросов.

Такой подход позволил создать устойчивую систему взаимодействия с сервером, которая легко адаптируется под изменения бизнес-логики и требований к данным.

Выводы по второму разделу: в результате проведенной работы был создан функциональный и удобный пользовательский интерфейс, соответствующий поставленным требованиям. Применение современных технологий и библиотек позволило обеспечить высокую производительность, безопасность и масштабируемость приложения. Разработанный продукт готов к дальнейшему развитию и интеграции с серверной частью платформы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта была разработана платформа для туристического-агрегатора «Открой Саратов», которое объединяет функционал нахождения интересных мест и планирования досуга. Этот проект стал результатом участия в программе «Стартап как диплом», в рамках которой наша команда поставила перед собой цель — создать удобный, интуитивно понятный и масштабируемый ресурс для нахождения актуальной информации о локальных местах нашего города, где можно хорошо провести время.

Для достижения этой цели, были решены ключевые задачи. Такие как:

- был проведен анализ предметной области и существующих решений;
- была разработана техническая концепция (структуру) для интерфейса;
- были сформулированы требования к проекту;
- был разработан дизайн веб-приложения;
- был реализован анализ и выбор инструментов для разработки;
- был разработан программный продукт для пользовательского интерфейса.

В результате наш проект достиг всех поставленных целей и задач, представив современное и удобное веб-приложение, способное удовлетворить потребности широкой аудитории.