



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Проектирование информационной системы «Приложение для планирования путешествий»»

Исполнитель Ковалев Илья Игоревич

Руководитель к.т.н., доцент Попов Николай Николаевич

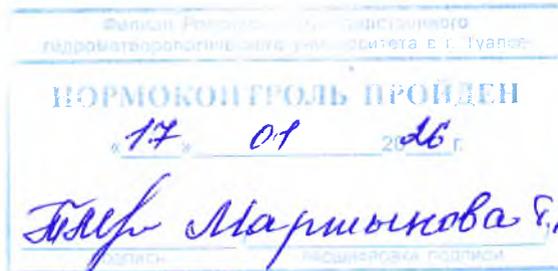
«К защите допускаю»

Руководитель кафедры _____

кандидат экономических наук

Майборода Евгений Викторович

«18» 01 2026 г.



Тюмень
2026

ОГЛАВЛЕНИЕ

Введение.....	3
1 Предпроектный анализ.....	6
1.1 Анализ предметной области	6
1.2 Анализ существующих приложений для путешествий	7
1.3 Анализ пользовательских сценариев и требований целевой аудитории.....	14
1.4 Описание бизнес-процессов, связанных с планированием и организацией путешествий	16
2 Методология проектирования приложения	20
2.1 Обоснование выбора технологий и инструментов для проектирования приложения.....	20
2.2 Описание архитектуры приложения.....	22
2.3 Разработка пользовательских сценариев (UserStories)	24
2.4 Проектирование интерфейса пользователя (UI/UX).....	26
2.5 UML диаграммы классов, компонентов и развёртывания	33
3 Реализация приложения для путешествий	38
3.1 Тестирование проектирования системы.....	38
3.2 Анализ затрат и оценка экономической эффективности внедрения приложения.....	41
Заключение	46
Список литературы	48
Приложение	52

Введение

В современном мире путешествия стали неотъемлемой частью жизни многих людей. Поездки позволяют отвлечься от повседневной рутины, познакомиться с новыми культурами, расширить кругозор и получить новые впечатления. При этом процесс подготовки к путешествию сегодня всё чаще осуществляется с использованием цифровых технологий. Пользователи самостоятельно планируют маршруты, выбирают способы передвижения, подбирают жильё и ищут информацию о достопримечательностях с помощью мобильных приложений и онлайн-сервисов [25, с.12]

Активное развитие мобильных технологий существенно упростило организацию поездок, однако одновременно выявило и ряд проблем. Несмотря на большое количество доступных сервисов, большинство из них решает лишь отдельные задачи: одни приложения ориентированы на бронирование билетов, другие на поиск жилья, третьи на навигацию или поиск достопримечательностей. В результате пользователю приходится использовать сразу несколько различных приложений, что усложняет процесс планирования и делает его менее удобным. В связи с этим возникает потребность в едином мобильном приложении, которое объединяло бы основные функции, необходимые для подготовки и сопровождения путешествия.

Актуальность данной выпускной квалификационной работы обусловлена необходимостью проектирования удобного и понятного мобильного приложения для планирования путешествий, ориентированного на реальные потребности пользователей. Такое приложение должно помогать не только на этапе подготовки поездки, но и в процессе самого путешествия, предоставляя структурированную и актуальную информацию в одном месте.

Объектом исследования является процесс планирования и организации путешествий — то, как люди готовят поездки, строят маршруты и собирают информацию о местах, которые хотят посетить. Предмет исследования —

проектирование мобильного приложения, которое помогает сделать этот процесс удобным и собранным в одном месте. В рамках работы анализируются основные сценарии использования приложения, его структура и логика взаимодействия пользователя с системой. Особое внимание уделяется созданию интерфейса и функциональных возможностей, которые обеспечивают удобство планирования и контроля поездки. Такой подход позволяет объединить реальные потребности пользователей и проектные решения в единой системе.

Цель выпускной квалификационной работы заключается в проектировании мобильного приложения для планирования путешествий, позволяющего пользователю эффективно организовывать поездки, создавать маршруты, добавлять точки интереса, получать рекомендации и управлять информацией о путешествии в удобной форме. Для достижения поставленной цели в рамках работы необходимо проанализировать существующие решения, определить требования целевой аудитории, разработать архитектуру информационной системы и создать прототип мобильного приложения.

Работа состоит из трёх основных разделов. В первом разделе проводится анализ предметной области, рассматриваются существующие сервисы и выявляются основные требования пользователей к мобильным приложениям для путешествий. Во втором разделе описываются объект и предмет исследования, анализируются бизнес-процессы и обосновывается выбор архитектурных и технологических решений. В третьем разделе рассматривается проектирование архитектуры приложения, пользовательских сценариев и интерфейсов, а также описывается разработка прототипа, его тестирование и оценка экономической эффективности.

Результаты выполненной работы позволяют сформировать целостный подход к проектированию мобильного приложения для планирования путешествий, соответствующего современным требованиям и ориентированного на удобство и практическую ценность для пользователей.

1 Предпроектный анализ

1.1 Анализ предметной области

Предметная область данного проекта связана с процессами планирования и организации путешествий с использованием цифровых сервисов. Основная идея приложения заключается в том, чтобы упростить для пользователя подготовку поездки и объединить все ключевые функции в одном удобном инструменте.

Современный путешественник ожидает, что приложение будет простым, понятным и функциональным. Пользователю важно быстро получить информацию о маршрутах, транспорте, местах проживания, достопримечательностях и погодных условиях. При этом всё чаще возникает потребность иметь доступ к сохранённым данным даже без постоянного подключения к интернету, например во время поездки за границей.

Пользователи приложения могут существенно различаться по своим целям и стилю путешествий. Одни планируют длительные маршруты с подробным расписанием и списком мест для посещения, другие ищут быстрые и комфортные решения для коротких поездок или командировок. Семейные туристы уделяют внимание безопасности и удобству, а деловые пользователи — скорости и минимальному количеству действий. Все эти сценарии необходимо учитывать при проектировании функционала и интерфейса приложения.

Для обеспечения полноценной работы системы требуется интеграция с внешними сервисами, такими как карты, сервисы бронирования, источники погодных данных и уведомлений. Это позволяет пользователю получать актуальную информацию и управлять своей поездкой в одном приложении, не переходя между разными сервисами.

Отдельное внимание в рамках анализа предметной области уделяется вопросам безопасности и удобства использования. Это включает защиту персональных данных, корректную работу с пользовательской информацией,

поддержку разных языков и валют, а также создание интуитивно понятного интерфейса.

Проведённый анализ предметной области позволяет определить ключевые функции будущего приложения, выявить основные пользовательские сценарии и заложить основу для дальнейшего проектирования информационной системы.

1.2 Анализ существующих приложений для путешествий

На сегодняшний день существует большое количество мобильных приложений, предназначенных для помощи в планировании путешествий. Они предоставляют пользователям возможность искать билеты и жильё, изучать достопримечательности, строить маршруты и получать рекомендации от других путешественников.

Для формирования требований к разрабатываемому приложению было проведено изучение популярных сервисов, таких как Tripster, TripAdvisor и Яндекс. Путешествия. Эти приложения широко используются и отражают основные тенденции рынка туристических цифровых решений.

Анализ показал, что пользователи высоко ценят удобный интерфейс, быстрый доступ к информации и возможность получать уведомления о важных изменениях, связанных с поездкой. При этом большинство существующих сервисов ориентированы на выполнение отдельных задач, например поиск жилья или отзывов, что вынуждает пользователя использовать несколько приложений одновременно.

Также было отмечено, что не все приложения обеспечивают удобную работу в офлайн-режиме, а интерфейсы некоторых сервисов перегружены лишними элементами. Это снижает комфорт использования, особенно во время путешествий, когда важно быстро и без лишних действий получить нужную информацию.

Результаты анализа существующих решений позволили определить

ключевые функции, которые должны быть реализованы в проектируемом приложении: создание маршрутов, сохранение данных о поездке, визуализация точек на карте, персонализация маршрута и удобная навигация. Кроме того, выявлены направления для улучшения, связанные с упрощением интерфейса и объединением функционала в одном приложении.

Полученные выводы стали основой для дальнейшего проектирования архитектуры и пользовательских сценариев разрабатываемой информационной системы.

Мобильное приложение «Tripster» представляет собой сервис, ориентированный преимущественно на подбор экскурсий и мероприятий, проводимых местными гидами. Его основная идея заключается в том, чтобы предоставлять пользователям уникальные авторские программы, которые невозможно найти в стандартных туристических каталогах. На рисунке 1 представлен интерфейс приложения «Tripster».

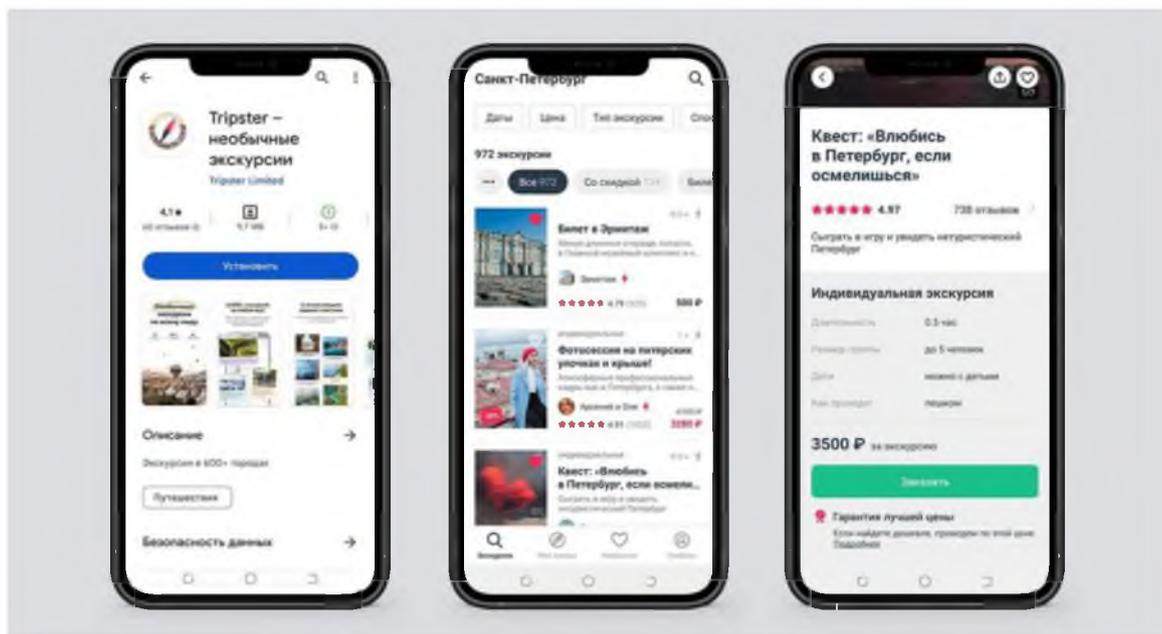


Рисунок 1 - интерфейс приложения «Tripster»[5, с.12]

Приложение предлагает детальные описания экскурсий, возможность связаться с гидом, уточнить детали маршрута или договориться о персональных

условиях проведения мероприятия. «Tripster» отличается высоким уровнем персонализации именно экскурсионного контента, однако при этом не предоставляет полноценных инструментов для планирования всего путешествия. В приложении отсутствуют функции построения маршрутов, интеграции с картами, системами бронирования отелей или транспортных услуг. Это делает сервис достаточно узкоспециализированным и ориентированным на решение одной конкретной задачи — подбор экскурсионной программы, а не комплексную подготовку поездки.

Приложение «TripAdvisor» как показано на рисунке 2 напротив, является многофункциональным международным сервисом, охватывающим практически все элементы путешествия. Его ключевая особенность заключается в наличии огромной базы отзывов пользователей, которая позволяет туристам ориентироваться в качестве различных объектов: от отелей и ресторанов до достопримечательностей и экскурсионных программ.

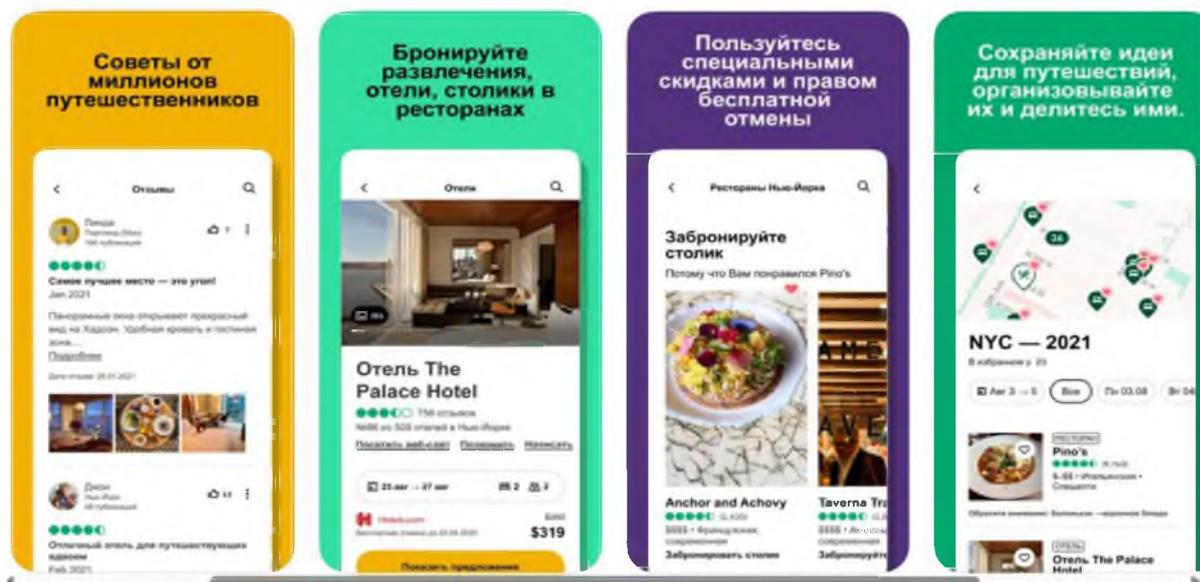


Рисунок 2 - интерфейс мобильного приложения «TripAdvisor»[7, с.8]

В приложении реализована возможность планирования поездки, создания маршрутов, сохранения интересных мест и просмотра объектов на карте. Также доступны инструменты бронирования гостиниц, туров и некоторых сервисов.

Тем не менее, из-за большого количества информации приложение иногда оказывается перегруженным и требует от пользователя значительных усилий для поиска нужных данных. Маршрутные функции есть, но они не всегда интуитивны, а часть операций сильно зависит от стабильного интернет-соединения.

Несмотря на широкую функциональность, «TripAdvisor» не предоставляет удобного пошагового планировщика маршрута, который учитывал бы индивидуальные параметры путешественника.

Яндекс.Путешествия ориентированы в первую очередь на поиск авиабилетов, гостиниц и туров, предлагая пользователю инструмент для сравнения цен и подбора оптимальных вариантов. Приложение тесно интегрировано с другими сервисами Яндекса, что делает его удобным для пользователей, знакомых с экосистемой компании.

Благодаря интеграции с сервисом Яндекс. Карты (рисунок 3) приложение Яндекс. Путешествия предоставляет пользователю удобный доступ к информации о местоположении объектов, транспорте и инфраструктуре.

Это значительно упрощает поиск билетов, жилья и ориентирование на местности. Пользователь может быстро посмотреть расположение отелей, вокзалов и аэропортов, а также оценить транспортную доступность выбранных объектов.

В то же время анализ функционала показал, что, несмотря на развитые возможности в сфере бронирования и сравнения предложений, приложение практически не ориентировано на полноценное планирование маршрута путешествия. В сервисе отсутствуют инструменты для создания детального плана поездки с несколькими этапами, а также возможности формировать маршрут с учётом интересов пользователя.

Система рекомендаций носит общий характер и не учитывает индивидуальные предпочтения туриста. Таким образом, приложение эффективно решает задачи подбора билетов и проживания, однако не предоставляет комплексного подхода к организации путешествия.

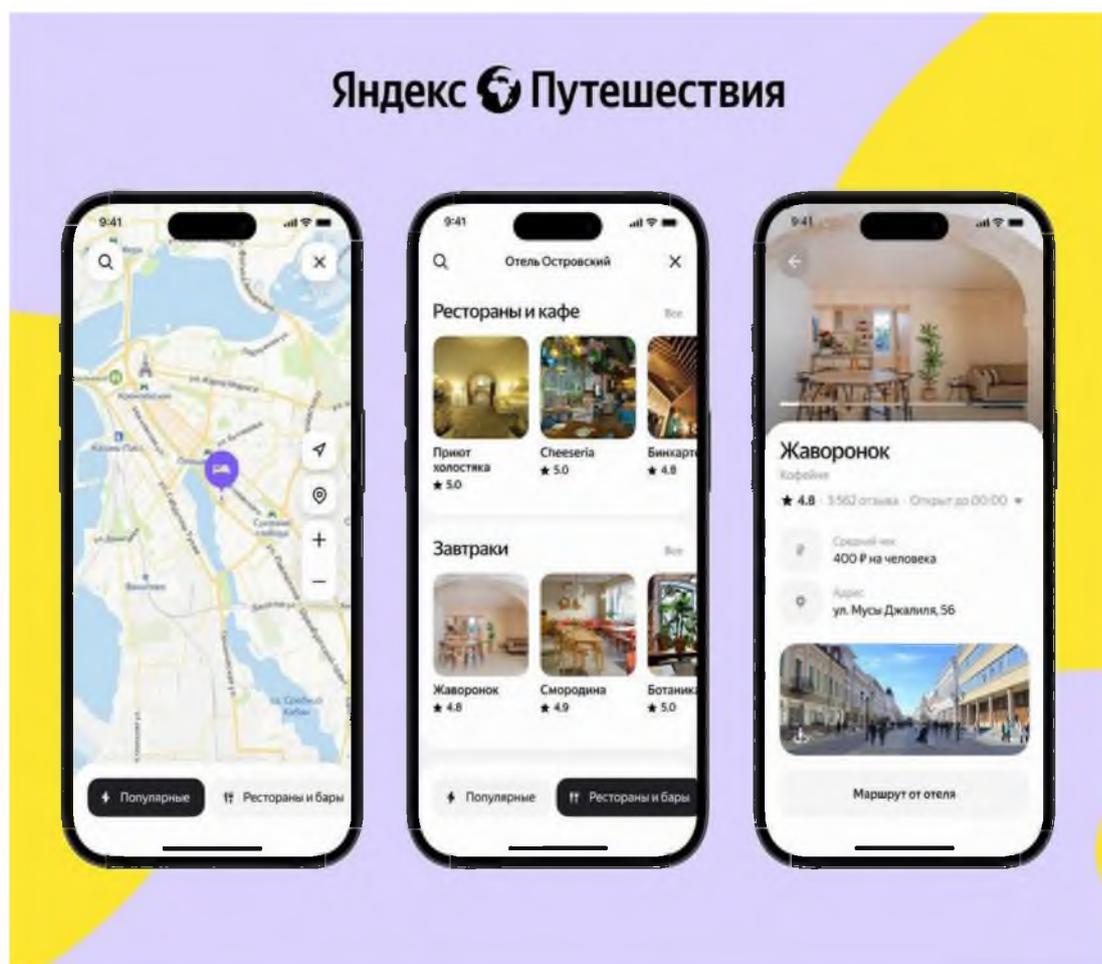


Рисунок 3 - интерфейс мобильного приложения Яндекс.Путешествия [10, с.15]

Сравнение трёх рассмотренных сервисов позволяет сделать вывод о том, что каждый из них ориентирован на решение отдельных задач туристического процесса. Tripster обеспечивает высокий уровень персонализации экскурсий и впечатлений, однако не поддерживает планирование путешествия в целом. TripAdvisor предлагает обширную базу данных, отзывы и разнообразные функции, но из-за перегруженного интерфейса работа с приложением может быть затруднена. Яндекс. Путешествия, в свою очередь, удобно решает задачи поиска билетов и проживания, но практически не предоставляет пользователю инструментов для самостоятельного формирования маршрута.

Таким образом, ни одно из проанализированных приложений не предлагает комплексного решения, которое объединяло бы планирование маршрутов, навигацию, рекомендации, учёт пользовательских предпочтений, работу в офлайн-режиме и единый набор инструментов для организации

путешествия от начала до конца. Это подтверждает необходимость разработки нового мобильного приложения, которое будет учитывать сильные стороны существующих сервисов и при этом предоставлять пользователю удобный, структурированный и целостный инструмент для планирования поездок.

В результате анализа были выделены основные характеристики, по которым оценивались рассматриваемые приложения:

- возможность планирования маршрутов;
- интеграция с картографическими сервисами;
- поддержка бронирования туристических услуг;
- наличие навигационных функций;
- возможность работы в офлайн-режиме;
- удобство и понятность пользовательского интерфейса;
- основные недостатки и ограничения функционала.

Полученные выводы используются в дальнейшем при формировании требований к проектируемому мобильному приложению и разработке его архитектуры.

Благодаря интеграции с сервисом Яндекс.Карты приложение предоставляет пользователю информацию о расположении объектов, доступном транспорте и городской инфраструктуре, что делает процесс поиска более удобным и наглядным.

Пользователь может быстро ориентироваться на местности и находить необходимые объекты без перехода в сторонние сервисы.

При этом основной функционал Яндекс.Путешествий сосредоточен преимущественно на подборе билетов и вариантов проживания.

Возможности детального планирования поездки в приложении ограничены: отсутствуют инструменты для создания сложных маршрутов, состоящих из нескольких точек, а также нет гибкой системы рекомендаций, которая учитывала бы индивидуальные предпочтения туриста. Пользователь не может заранее спланировать последовательность посещения мест или адаптировать маршрут под свой стиль путешествия.

Таблица 1 - Сравнение по критериям

Критерий сравнения	Tripster	TripAdvisor	Яндекс.Путешествия
Планирование маршрутов	Отсутствует	Частично реализовано, но ограничено	Ограниченное
Интеграция с картами	Нет	Есть встроенная карта	Полная интеграция с Яндекс.Картами
Бронирование услуг	Да (экскурсии)	Да (отели, туры, экскурсии)	Да (билеты, отели, туры)
Навигация	Нет	Ограничено доступна	Есть через Яндекс.Карты
Оффлайн-режим	Нет	Частично доступна	Практически отсутствует
Удобство интерфейса	Простой понятный интерфейс	Удобный интерфейс	Функциональный удобный интерфейс
Основные недостатки	Нет комплексного планирования	Перегруженность данными	Нет полноценного маршрутизатора

Таким образом, приложение эффективно решает задачи поиска и бронирования, но не охватывает процесс планирования путешествия целиком. Пользователю по-прежнему приходится использовать дополнительные сервисы для построения маршрутов и организации поездки.

Сравнение рассмотренных сервисов показывает, что каждый из них ориентирован на решение отдельных задач туриста. Tripster делает акцент на

подборе экскурсий и индивидуальных впечатлений, но не предоставляет инструментов для комплексного планирования поездки. TripAdvisor обладает обширной базой данных и большим количеством отзывов, однако из-за перегруженности интерфейса может быть неудобен для повседневного использования. Яндекс.Путешествия, в свою очередь, удобно использовать для поиска билетов и жилья, но функционал маршрутизации в нём развит слабо.

В результате ни одно из рассмотренных приложений не предлагает универсального решения, которое объединяло бы планирование маршрута, навигацию, рекомендации, анализ предпочтений пользователя, офлайн-доступ и удобное управление поездкой в одном интерфейсе. Это подтверждает необходимость разработки нового мобильного приложения, которое объединит лучшие стороны существующих решений и предложит пользователю единый, логичный и удобный инструмент для организации путешествий.

1.3 Анализ пользовательских сценариев и требований целевой аудитории

Целевая аудитория проектируемого мобильного приложения — это пользователи, которые предпочитают самостоятельно планировать свои поездки, не прибегая к помощи туристических агентств. К ним относятся молодые люди, семейные пары, начинающие путешественники, а также более опытные туристы, которым важно иметь под рукой удобный инструмент для организации поездки.

В последние годы самостоятельный туризм стал особенно популярным. Пользователи всё чаще выбирают цифровые сервисы, которые позволяют им самостоятельно контролировать маршрут, бюджет и время путешествия. При этом большинство туристов сталкиваются с одними и теми же задачами: выбор направления, поиск интересных мест, составление маршрута, изучение информации о достопримечательностях, ориентирование на местности и корректировка планов уже в процессе поездки.

Один из ключевых пользовательских сценариев начинается с выбора

направления. На этом этапе человек ищет идеи для путешествия, изучает описания мест, смотрит фотографии и отзывы. Пользователю важно получать рекомендации и готовые подборки, которые помогают быстрее принять решение. Приложение должно упрощать этот процесс и предлагать понятную и наглядную информацию[8, с.23]

Следующий этап - формирование маршрута. Пользователь выбирает места, которые хочет посетить, и продумывает порядок их посещения. Здесь важна возможность увидеть маршрут на карте, оценить расстояния между точками и примерное время в пути. Многие существующие сервисы не позволяют полноценно автоматизировать этот процесс, поэтому пользователю приходится вручную составлять маршрут в нескольких приложениях одновременно.

Отдельный пользовательский сценарий связан с получением информации о достопримечательностях и интересных местах. Пользователь ожидает увидеть краткое описание объекта, фотографии, режим работы, стоимость посещения и отзывы других путешественников. Такая информация помогает принять решение о включении объекта в маршрут. Большое значение имеет возможность доступа к этим данным без подключения к интернету, так как в поездках не всегда доступна стабильная связь.

Во время самого путешествия пользователю необходим удобный доступ к маршруту и функция навигации. Приложение должно отображать текущую позицию на карте, помогать ориентироваться и подсказывать путь между точками. При этом маршрут должен быть гибким — пользователь может в любой момент изменить планы, добавить новые точки или исключить уже запланированные.

Дополнительным требованием являются уведомления и подсказки. Пользователь ожидает напоминания о времени посещения объектов, информацию о погодных условиях и другие полезные уведомления, которые помогают сделать поездку более комфортной и организованной.

Не менее важны требования к интерфейсу приложения. Пользователи

ценят простоту и понятность. Интерфейс должен быть логичным, не перегруженным элементами и понятным даже для людей без технического опыта. Основными требованиями являются удобная навигация, быстрый доступ к основным функциям и минимальное количество лишних действий.

Таким образом, анализ пользовательских сценариев показывает, что современным путешественникам необходимо комплексное мобильное приложение, которое сопровождает их на всех этапах поездки — от выбора направления до навигации на местности. Проектируемое приложение должно учитывать реальные потребности пользователей и предлагать удобный, гибкий и понятный инструмент для самостоятельной организации путешествий.

1.4 Описание бизнес-процессов, связанных с планированием и организацией путешествий

Процесс организации путешествия состоит из нескольких последовательных действий, которые пользователь выполняет каждый раз при подготовке поездки. Эти действия можно рассматривать как взаимосвязанные бизнес-процессы, формирующие единый цикл планирования и сопровождения путешествия [4, с.30]

Как правило, всё начинается с выбора направления. Пользователь определяет, куда именно он хочет поехать, изучает различные варианты, просматривает фотографии, читает краткие описания городов и стран, сравнивает впечатления других путешественников. На этом этапе особенно важно предоставить понятную, наглядную и не перегруженную информацией подачу, которая помогает быстрее принять решение.

После выбора направления пользователь переходит к поиску интересных мест для посещения. Это могут быть достопримечательности, музеи, природные объекты, прогулочные зоны и другие локации. Для принятия решения человеку необходимо видеть базовую информацию: фотографии, краткое описание, режим работы, примерную стоимость посещения и

расположение на карте. Отзывы других пользователей также играют важную роль, так как позволяют заранее оценить ценность конкретного места. [7, с.47]

Следующим и одним из самых сложных этапов является составление маршрута. Пользователю необходимо определить порядок посещения выбранных точек, оценить расстояния между ними и понять, сколько времени займёт перемещение. На практике этот процесс часто требует использования нескольких разных сервисов и ручных расчётов. Качественное мобильное приложение должно упрощать этот этап, отображая все точки на карте, помогая выстраивать логичную последовательность маршрута и предоставляя подсказки по времени и способам передвижения.

После формирования маршрута пользователь приступает к организации логистики. Сюда относится выбор транспорта, поиск билетов и подбор вариантов проживания. Даже если приложение не занимается прямым бронированием, оно должно помогать пользователю ориентироваться в этом процессе и обеспечивать быстрый переход к надёжным внешним сервисам.

Во время самой поездки особое значение приобретает сопровождение маршрута. Пользователю важно иметь доступ к карте, описаниям запланированных мест и текущему маршруту. Возможность оперативно вносить изменения, добавлять новые точки или корректировать план по ходу путешествия делает приложение более гибким и полезным. Функции офлайн-доступа также являются важным элементом, так как стабильное интернет-соединение доступно не всегда.

После завершения поездки пользователь может быть заинтересован в сохранении маршрута, просмотре информации о пройденном пути или отметке понравившихся мест. Такие функции повышают ценность приложения и мотивируют пользователя возвращаться к нему при планировании следующих путешествий.

Таким образом, процесс планирования и организации путешествия включает в себя выбор направления, поиск информации, формирование маршрута, организацию логистики и сопровождение поездки.

Диаграмма пользовательских сценариев (UseCaseDiagram) отражает основные варианты взаимодействия пользователя с мобильным приложением. В роли основного актора выступает пользователь, который применяет приложение для планирования, организации и сопровождения путешествий.



Рисунок 4 - Диаграмма пользовательских сценариев (UseCase)

Диаграмма на рисунке 4 описывает ключевые действия, доступные пользователю: регистрацию и авторизацию в системе, создание нового путешествия, просмотр списка сохранённых маршрутов, редактирование плана поездки и добавление точек маршрута.

Также пользователь может просматривать маршрут на интерактивной карте, включая режим работы с офлайн-картами, что повышает удобство использования приложения во время путешествий.

Представленная IDEF0-диаграмма отражает основной бизнес-процесс планирования путешествия в мобильном приложении «Express».

Входными данными являются предпочтения пользователя, даты и бюджет поездки. Управление процессом осуществляется требованиями системы и правилами работы приложения.

В качестве механизмов используются мобильное приложение, серверная часть и база данных.

Результатом выполнения процесса является сформированный маршрут и план путешествия.

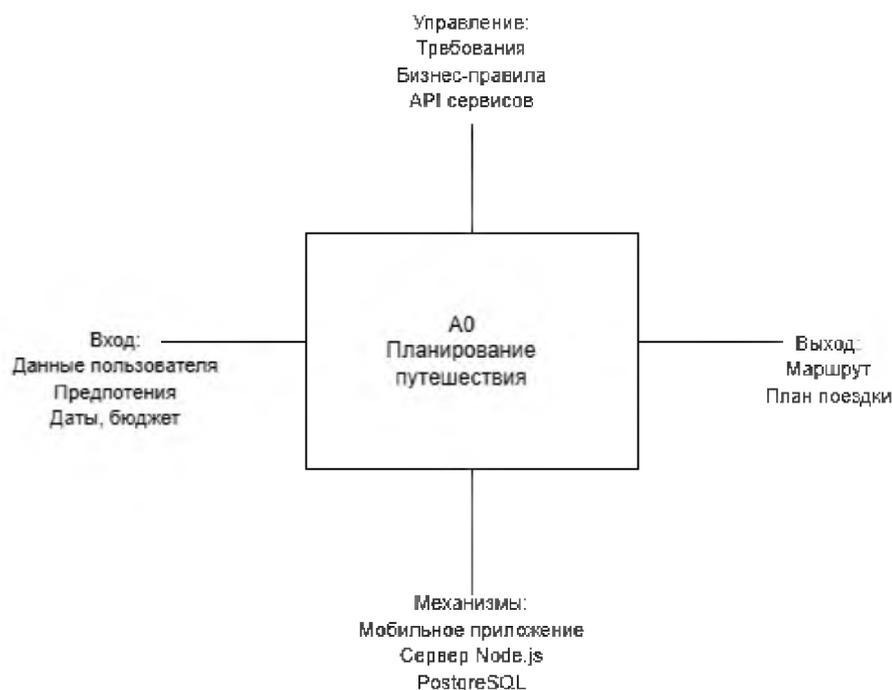


Рисунок 5 - Диаграмма IDEF0-Планирование путешествия

Как представлено на рисунке 5 показан процесс планирования путешествия. В центре находится блок A0, который как раз обозначает сам процесс планирования. Слева идут входные данные — это всё, что нам даёт пользователь: его предпочтения, даты поездки и бюджет. Снизу показаны механизмы, которые помогают системе работать, то есть мобильное приложение и сервер с базой данных. Сверху находятся управляющие элементы: правила, требования и API, которые следят, чтобы всё происходило правильно. Справа видим результат — готовый маршрут и план поездки, который получает пользователь. Диаграмма наглядно показывает, как данные пользователя проходят через приложение и сервер и превращаются в удобный план поездки. В целом, это помогает понять, как вся система взаимодействует с пользователем и как строится путешествие.

2 Методология проектирования приложения

2.1 Обоснование выбора технологий и инструментов для проектирования приложения

Разработка мобильного приложения требует выбора таких технологий, которые будут одновременно надёжными, удобными в использовании и подходящими для реализации функционала будущей системы. Поскольку проектируемое приложение должно работать быстро, корректно обрабатывать данные, обеспечивать навигацию и поддержку маршрутов, выбор инструментов разработки играет ключевую роль.

Для реализации клиентской части приложения был выбран фреймворк ReactNative. Причина выбора ReactNative заключается в том, что он является одним из самых популярных инструментов для создания удобных и быстрых пользовательских интерфейсов. Его главный плюс — работа с компонентами. Это значит, что приложение можно строить как набор небольших независимых частей, которые легко обновлять, изменять и расширять [8, с.12]

Серверная часть приложения разрабатывается с использованием платформы Node.js и фреймворка Express.js. Эта технология выбрана благодаря простоте интеграции с фронтендом, высокой производительности и широким возможностям для работы с API. Node.js позволяет обрабатывать запросы в реальном времени и хорошо подходит для приложений, которые должны быстро отвечать на запросы пользователей, например при загрузке туристических объектов, расчёте маршрута или сохранении данных профиля. Express.js, в свою очередь, упрощает создание серверных маршрутов и логики, обеспечивая надёжную основу для взаимодействия клиента и сервера [9, с.45]

Для хранения данных используется система управления базами данных PostgreSQL. Этот инструмент выбран благодаря стабильности, высокой скорости обработки запросов и гибким возможностям по работе со структурированными данными. Приложение работает с объектами, такими как точки маршрутов, категории мест, профили пользователей, сохранённые

поездки. Для такой информации необходима надёжная реляционная база данных. PostgreSQL подходит идеально, потому что поддерживает сложные запросы, связи между таблицами и позволяет безопасно хранить большие объёмы данных.

Особое внимание уделяется подключению внешних API, которые позволяют использовать готовые данные о локациях, погоде, картах и туристических объектах. Интеграции с API нужны для того, чтобы приложение не зависело только от собственной базы данных, а могло получать актуальную информацию из внешних источников. Например, данные о погоде, актуальных событиях, работе музеев или городском транспорте могут регулярно обновляться, а использование API облегчает этот процесс.

В процессе проектирования будут применяться инструменты моделирования, такие как UML-диаграммы. Они позволяют наглядно представить структуру приложения, связи между его компонентами и логику работы основных функций. Построение диаграмм помогает лучше понять архитектуру системы, определить заранее возможные ошибки и сделать разработку более организованной и последовательной. UML является стандартным инструментом в проектировании информационных систем и подходит для учебных проектов любого уровня сложности.

Таким образом, выбор технологий — React.js, Node.js, Express.js и PostgreSQL — основан на практичности, популярности, удобстве разработки и возможностях масштабирования. Эти инструменты позволяют создать современное, быстрое и удобное приложение, которое сможет обрабатывать маршруты, хранить данные, подключаться к внешним сервисам и работать стабильно даже при больших объёмах информации.

Проектирование мобильного приложения — один из ключевых этапов разработки информационной системы. От того, насколько продуманной будет архитектура, структура данных, логика интерфейса и взаимодействие компонентов, зависит стабильность работы приложения, удобство использования и возможность дальнейшего развития.

В данном разделе рассматриваются методы и подходы, применённые при создании мобильного приложения для планирования путешествий. Для разработки использовались принципы объектно-ориентированного анализа, диаграммы UML, клиент–серверная архитектура и современные практики проектирования пользовательского интерфейса.

Главная задача методологии проектирования состоит в том, чтобы сформировать целостное представление о системе: какие объекты в ней существуют, как они связаны, где хранятся данные, как взаимодействуют клиентская и серверная части, какие функции должен получить пользователь и как он будет взаимодействовать с ними через интерфейс.

На основе анализа требований и изучения существующих аналогов была выбрана архитектура, которая обеспечивает баланс между производительностью, удобством разработки и возможностью масштабирования. Далее подробно описаны архитектурные решения, модель данных, диаграммы классов, а также логика взаимодействия компонентов.

2.2 Описание архитектуры приложения

Архитектура проекта была построена с учётом необходимости обеспечивать надёжное хранение данных, стабильную работу интерфейса и возможность расширять функциональность в будущем [11, с.27] Основой для приложения стала классическая клиент–серверная архитектура, где:

- клиентская часть запускается на смартфоне пользователя,
- серверная часть работает в облаке и обрабатывает запросы,
- база данных хранит информацию о маршрутах, пользователях и объектах путешествия.

Приложение спроектировано на ReactNative, что обеспечивает кроссплатформенность - один и тот же код работает на Android и iOS [12, с.14]. Клиентская часть выполняет следующие задачи:

- отображение экранов (регистрация, маршруты, детали путешествия);

- обработка действий пользователя
- отправка запросов на сервер (RESTAPI);
- получение данных от сервера и обновление интерфейса;
- локальное хранение части данных (кэширование).

Сервер реализован с помощью Node.js и Express.js, что позволяет работать с большим количеством запросов и легко масштабировать приложение. Сервер выполняет такие функции:

- регистрация и аутентификация пользователей;
- управление маршрутами (создание, редактирование, удаление);
- работа с точками путешествия;
- передача данных клиенту в формате JSON;
- валидация и проверка корректности данных;
- взаимодействие с базой данных PostgreSQL.

Node.js обеспечивает высокую производительность и позволяет использовать JavaScript на обеих частях приложения (клиент и сервер), что упрощает поддержку проекта.

Для хранения данных используется PostgreSQL — надёжная реляционная СУБД, которая поддерживает:

- Чёткую структуру данных (таблицы, связи);
- транзакции;
- индексы и ускорения поиска;
- масштабируемость;

В базе данных хранятся:

- пользователи приложения;
- маршруты;
- настройки путешествий

Модель данных проектировалась с использованием принципов нормализации, чтобы избежать дублирования и обеспечить эффективную работу запросов.

2.3 Разработка пользовательских сценариев (UserStories)

Разработка пользовательских сценариев является важной частью проектирования мобильного приложения, так как позволяет заранее определить, как пользователь будет взаимодействовать с системой. Такой подход облегчает подбор функций, помогает определить приоритеты разработки и формирует общую логику поведения приложения. User Stories представляют собой короткие описания действий пользователя, отражающие его цели и ожидаемый результат. Благодаря этому разработчик может создавать интерфейс и функционал, максимально приближенные к реальным потребностям аудитории. Эти сценарии помогают понять, какие шаги будут для пользователя естественными, а где могут возникнуть сложности. Они позволяют заранее продумать удобство навигации и расположение кнопок. Пользовательские сценарии помогают сэкономить время на исправления после реализации, потому что многие проблемы удастся увидеть ещё на этапе проектирования. Сценарии также облегчают тестирование — можно проверить, что все действия работают так, как задумано. В итоге приложение получается более понятным, удобным и приятным для использования. [13, с.9]

Таблица 2 - Пользовательские сценарии (UserStories) с описанием цели

№	Пользовательские сценарии (User Story)	Описание цели
1	Как пользователь, я хочу зарегистрироваться чтобы иметь личный доступ к приложению	Создать собственный профиль и хранить данные в защищённом виде
2	Как пользователь, я хочу войти в свой аккаунт чтобы видеть свои маршруты	Получить доступ к сохранённым маршрутам

Продолжение таблицы 2

3	Как пользователь я хочу создавать новый маршрут чтобы планировать свои путешествия	Начать построение маршрута с нуля
4	Как пользователь я хочу видеть маршрут на карте, чтобы понимать последовательность движения	Получить визуальное представление о маршруте
5	Как пользователя хочу сохранять маршрут, чтобы иметь возможность вернуться к нему позже	Сохранить итоговое планирование и вернуться к нему позже

Представленные пользовательские сценарии отражают реальные задачи, которые выполняет человек при планировании путешествия. Каждый сценарий включает не только действие, но и конкретную цель, которая объясняет, зачем пользователю нужна та или иная функция. Такой подход делает проектирование приложения более осознанным и помогает разработчикам понимать практическую ценность каждой части функционала.

Регистрация и вход в систему позволяют пользователю создать индивидуальный профиль и обеспечивают доступ к его персональным маршрутам. Это важно, так как без личного аккаунта невозможно сохранить данные и продолжить работу с любого устройства. Создание маршрута — основная задача приложения, потому что пользователь запускает систему именно ради планирования путешествия.

Добавление точек на карту служит формированию структуры маршрута, позволяя выделить ключевые места, которые пользователь собирается посетить. Отображение маршрута на карте даёт визуальное понимание пути и помогает оценить расстояния. Редактирование точек обеспечивает гибкость работы, так

как планы путешествия могут изменяться. Функция сохранения завершает цикл взаимодействия, позволяя пользователю хранить результаты своей работы и возвращаться к ним позже.

Каждая UserStory выполняет важную роль: она описывает действие пользователя, объясняет, зачем оно ему нужно, и формирует основу для разработки интуитивного и удобного интерфейса. [13, с.10]

2.4 Проектирование интерфейса пользователя (UI/UX)

На этапе проектирования пользовательского интерфейса была разработана визуальная структура мобильного приложения для планирования путешествий. Основной целью данного этапа являлось создание наглядных и понятных экранов, обеспечивающих удобство взаимодействия пользователя с приложением.

Проектирование интерфейса выполнялось с использованием инструмента Figma, который позволяет создавать интерактивные макеты экранов и наглядно отображать логику пользовательского взаимодействия. Разработанные макеты использовались как основа для дальнейшей реализации клиентской части приложения [24, с.15]

При разработке интерфейса особое внимание уделялось простоте навигации, логичному расположению элементов управления и минимизации действий, необходимых для выполнения основных пользовательских сценариев

Основу интерфейса составляет главная навигационная панель, через которую пользователь может переходить между основными разделами: список маршрутов, карта, профиль и создание нового путешествия.

Такие элементы расположены в нижней панели, что делает их доступными для большого пальца и повышает удобство использования на смартфонах.

На экране маршрута пользователь видит список своих путешествий, а также кнопку для создания нового маршрута.

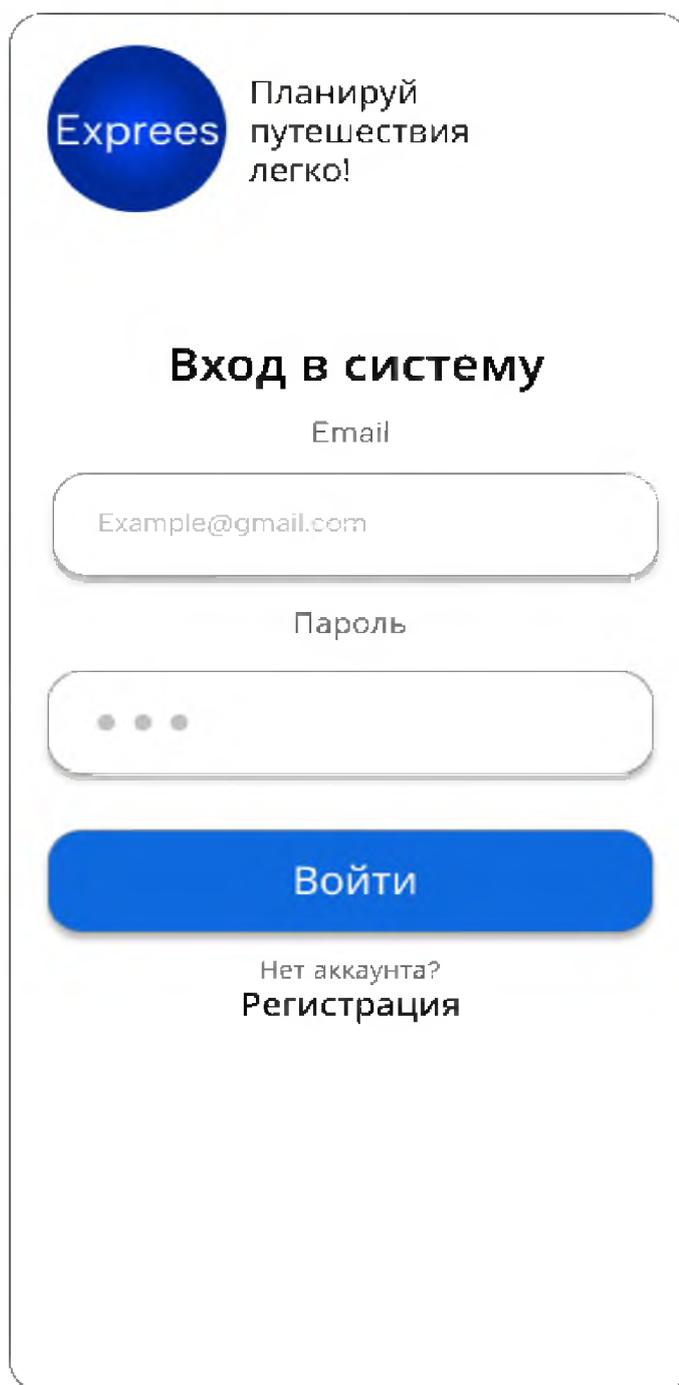


Рисунок 6 - Экран приложения «Вход в систему»

Экран входа в систему (рисунок 6) предназначен для авторизации пользователя в приложении.

На данном экране пользователь вводит свои учётные данные для получения доступа к персональному функционалу приложения.

Интерфейс экрана выполнен в минималистичном стиле и содержит только необходимые элементы, что делает процесс входа быстрым и понятным.

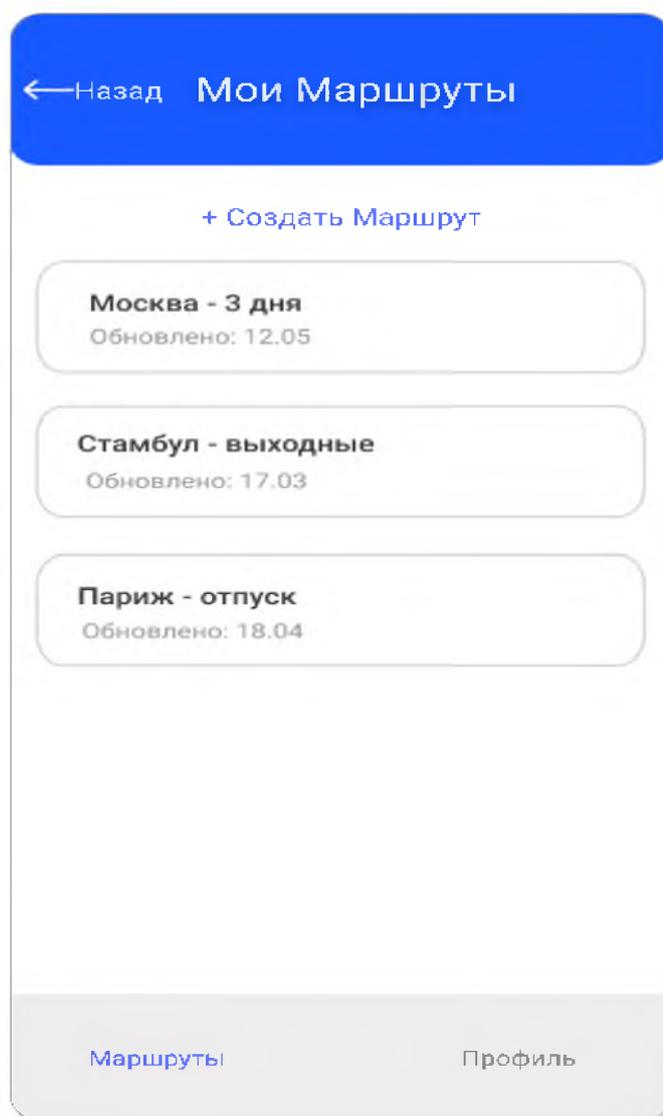


Рисунок 7 - Экран приложения «Мои маршруты»

На экране «Мои маршруты» как представлено на рисунке 7, предназначен для отображения списка всех маршрутов, созданных пользователем.

Каждый маршрут представлен в виде отдельного элемента списка с основной информацией. С данного экрана пользователь может перейти к просмотру выбранного маршрута или создать новый маршрут, что делает данный экран центральным элементом навигации приложения.

Экран «Создание маршрута» (Рисунок 8) организован максимально просто: в верхней части отображается название маршрута, а ниже находятся элементы для добавления точек. При выборе точки появляется карта, на которой пользователь может указать конкретную локацию.

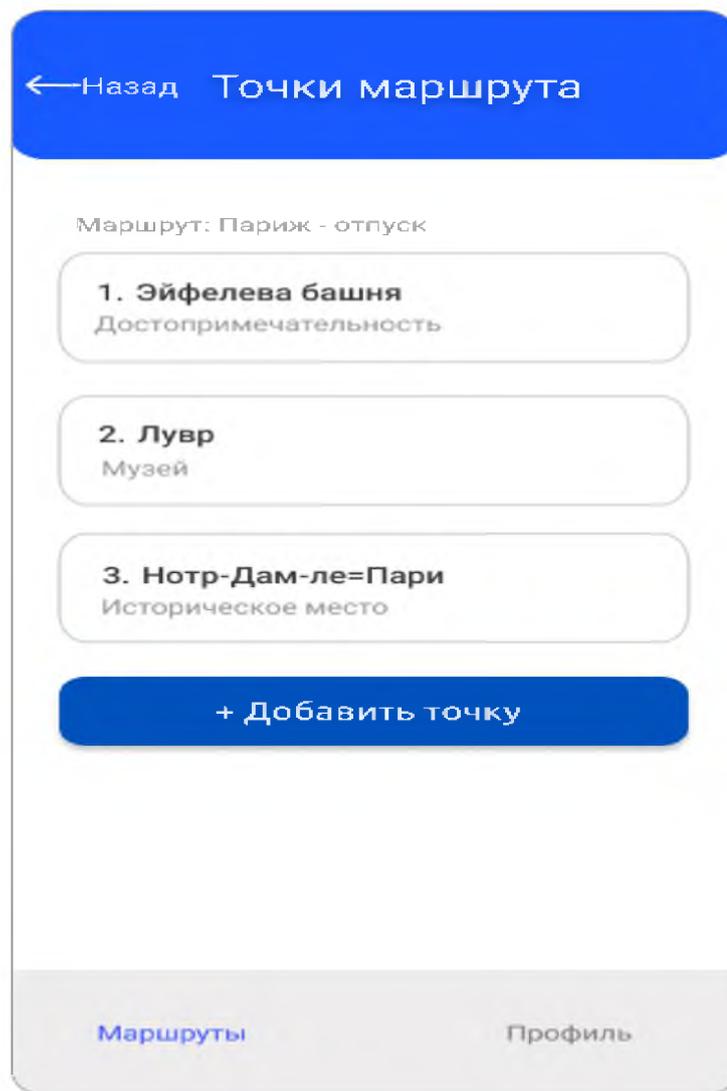


Рисунок 8 - Экран приложения «Создание маршрута»

После добавления точек маршрут отображается в виде линий между отмеченными позициями. Такой визуальный подход помогает сразу оценить расстояние и общую структуру будущей поездки.

Экран «Точки маршрута» как показано на рисунке 9 - предназначен для отображения полного списка мест, которые пользователь планирует посетить в рамках выбранного путешествия. Все точки маршрута представлены в виде отдельных элементов списка, что обеспечивает наглядность и удобство восприятия информации.

На данном экране пользователь может добавлять новые точки маршрута, удалять уже существующие, а также изменять их порядок в соответствии с личными предпочтениями или изменениями плана поездки.

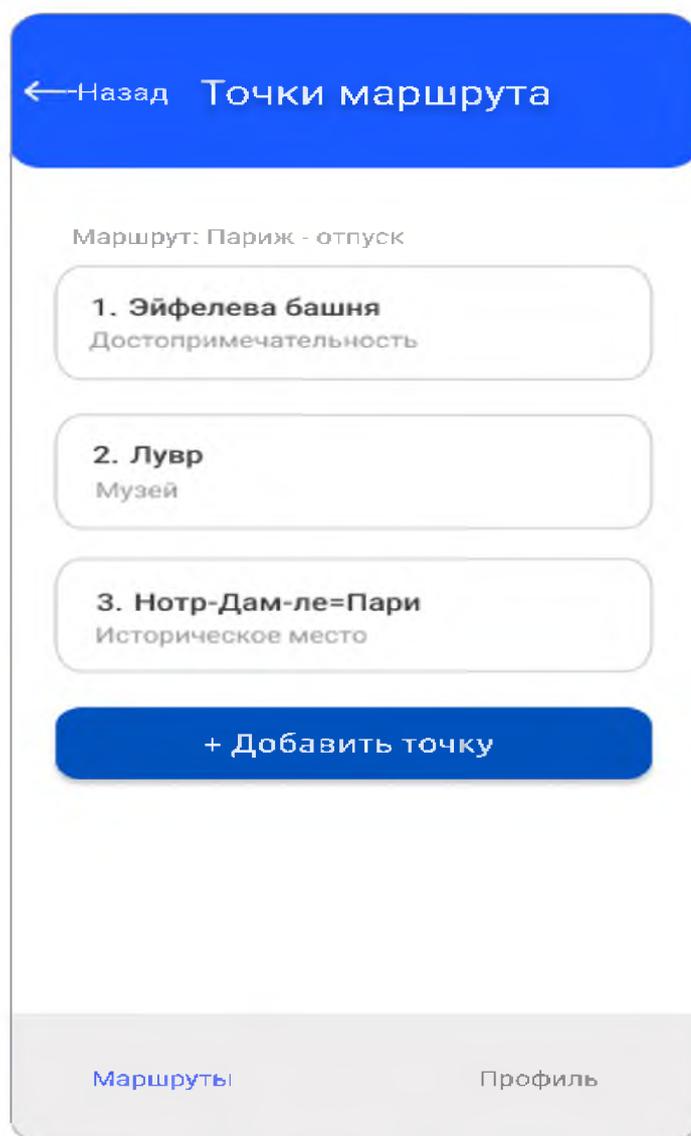


Рисунок 9 - Экран приложения «Точки маршрута»

Возможность управления последовательностью точек позволяет адаптировать маршрут под реальные условия путешествия.

Экран точек маршрута играет ключевую роль в процессе планирования поездки, так как позволяет пользователю гибко настраивать структуру маршрута, контролировать порядок посещения мест и оперативно вносить изменения, повышая удобство и эффективность использования приложения.

Экран «Редактирование маршрута» как показано на рисунке - 10 пользователь может добавлять или удалять точки, корректировать порядок посещения мест и вносить изменения в планы, если условия поездки изменились.

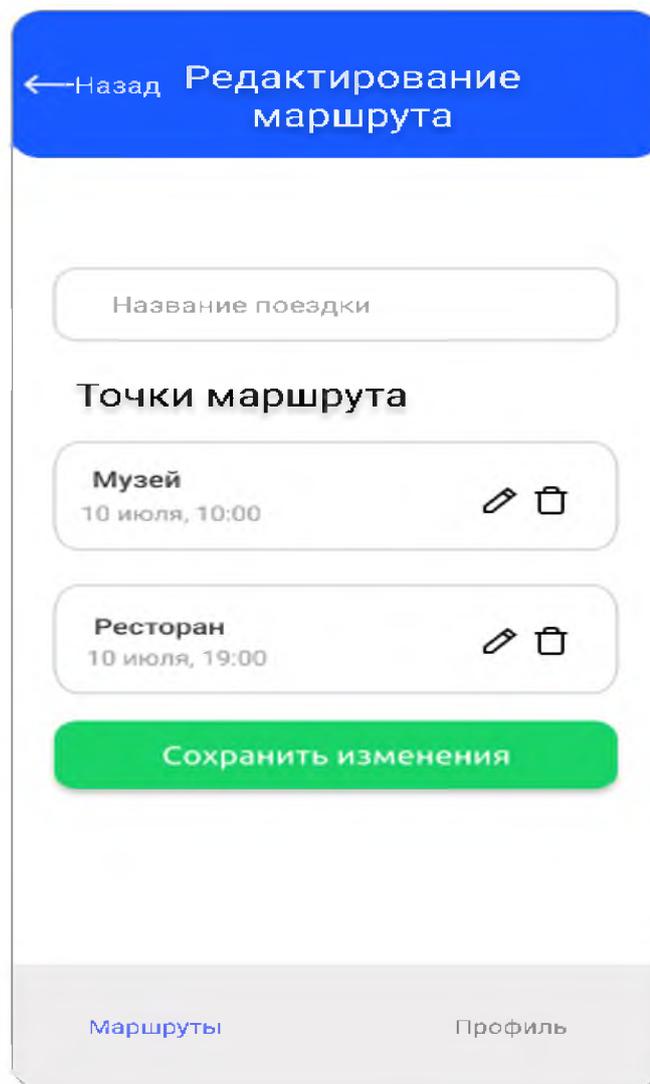


Рисунок 10 - Экран приложения «Редактирование точек»

Экран «Заметки и отзывы» предназначен для отображения оценок и комментариев других пользователей о местах, входящих в маршрут путешествия. На данном экране (Рисунок 11) пользователь может ознакомиться с мнениями людей, которые уже посещали указанные точки, что позволяет получить представление об их качестве, удобстве и актуальности.

Отзывы помогают пользователю принять более взвешенное решение при добавлении или редактировании точек маршрута, а также скорректировать план поездки с учётом реального пользовательского опыта. Наличие системы оценок позволяет быстро сравнивать места между собой и выбирать наиболее подходящие варианты.

Использование экрана отзывов повышает информативность приложения, улучшает качество планирования маршрутов и способствует формированию доверия к системе за счёт открытости и обмена пользовательским опытом.

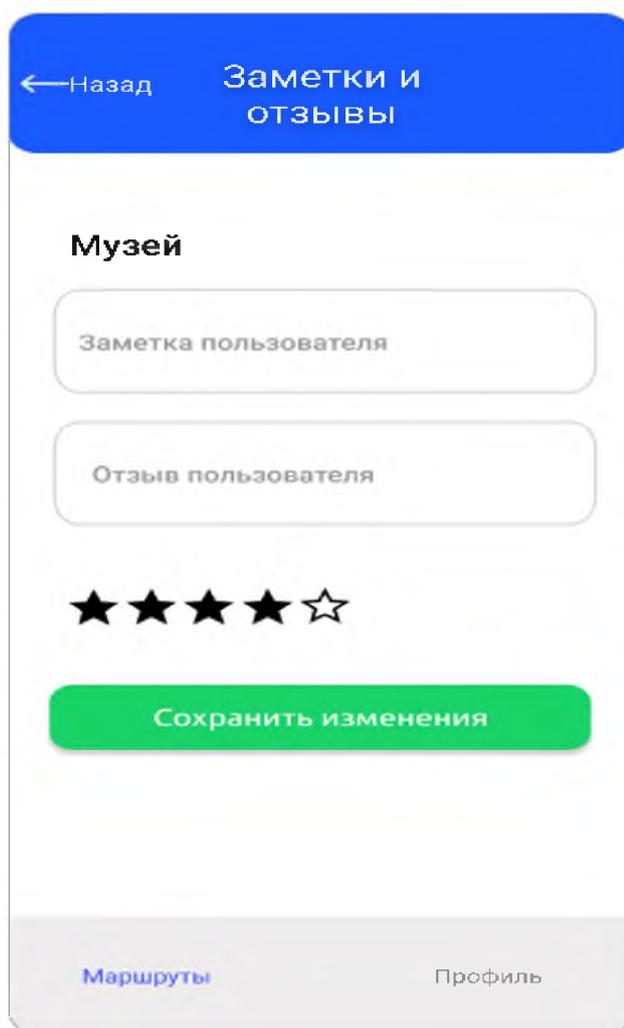


Рисунок11 - Экран приложения«Заметки и отзывы»

Проектирование пользовательского интерфейса включало продумывание логики взаимодействия между основными экранами приложения и последовательности действий пользователя. В основе интерфейса лежит понятный и последовательный пользовательский сценарий, который начинается с авторизации и далее ведёт пользователя к созданию и управлению маршрутами.

После входа в систему пользователь попадает на экран «Мои маршруты», где отображается список всех ранее созданных поездок. Данный экран выступает в качестве центральной точки навигации и обеспечивает переход к

созданию нового маршрута или просмотру уже существующего.

При создании маршрута пользователь последовательно заполняет основные параметры поездки, после чего переходит к экрану точек маршрута. На данном экране осуществляется управление содержимым маршрута: добавление новых точек, изменение их порядка и удаление ненужных элементов.

Переходы между экранами реализованы таким образом, чтобы пользователь всегда находился в рамках логичного сценария работы с приложением: авторизация → просмотр маршрутов → создание маршрута → добавление и управление точками маршрута. Данная последовательность снижает вероятность ошибок и упрощает использование приложения даже для пользователей без предварительного опыта.

2.5 UML диаграммы классов, компонентов и развёртывания

Для формализации структуры данных была построена диаграмма классов, отображающая сущности и связи между ними [17, с.45]

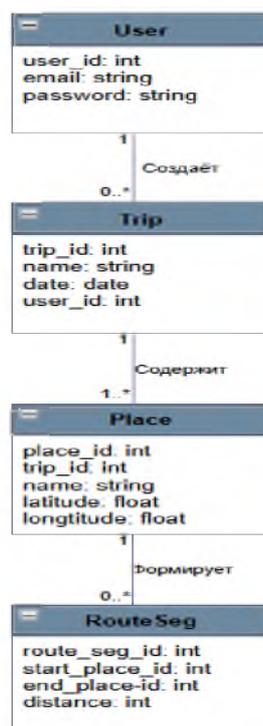


Рисунок 12 - Диаграмма классов

Эта диаграмма является основой для формирования структуры таблиц в PostgreSQL и разработки API методов сервера.

Диаграмма классов (рисунок 12) показывает логическую структуру данных, используемых в приложении. Она отражает основные объекты системы и связи между ними.

User (Пользователь) — хранит данные о пользователе и является владельцем маршрутов.

Один пользователь может создавать любое количество путешествий.

Trip (Маршрут) — это отдельное путешествие, в котором пользователь отмечает места, которые он планирует посетить. Маршрут включает название, дату создания и принадлежность к конкретному пользователю.

Place (Точка путешествия) — конкретная локация маршрута. В каждой точке есть координаты (широта и долгота), название и принадлежность к маршруту. Маршрут может состоять из одной или множества точек.

RouteSeg (Сегмент пути) — отрезок между двумя точками маршрута. Он содержит идентификаторы исходной и конечной точек, а также расстояние между ними.

Такие сегменты позволяют выстраивать маршрут на карте и рассчитывать общую протяжённость путешествия.

Диаграмма наглядно показывает вложенность данных: пользователь → маршрут → точки → сегменты пути.

Эта структура обеспечивает гибкость, простоту масштабирования и удобную работу с PostgreSQL.

Диаграмма компонентов как показано на (рисунке 13) отражает внутреннюю структуру приложения и показывает, какие программные части образуют систему и как именно они взаимодействуют между собой. В приложении для планирования путешествий выделяются три основных компонента, каждый из которых отвечает за свою область работы.

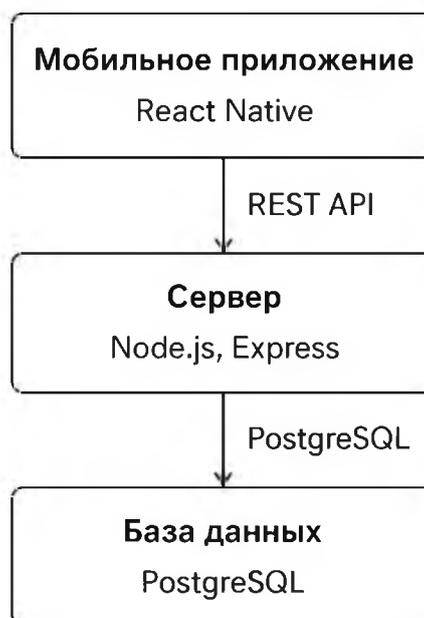


Рисунок 13 — UML диаграмма компонентов

Первый компонент — это мобильное приложение, разработанное на базе ReactNative. Оно запускается на смартфоне пользователя и обеспечивает весь пользовательский интерфейс: отображение маршрутов, точек на карте, создание нового путешествия, просмотр подробностей и работу с геолокацией. Вторым компонентом является серверная часть приложения, созданная на Node.js и Express.js. Сервер получает запросы, сформированные мобильным клиентом, и выполняет всю серверную обработку данных. Именно здесь проходят проверка данных, авторизация пользователей, создание и редактирование маршрутов, формирование информации о точках и расчёт переходов между ними. Сервер соединяет интерфейс пользователя с базой данных и обеспечивает корректную обработку логики приложения. Он выступает «центром управления», через который проходит весь обмен данными.

Третий компонент представлен базой данных PostgreSQL, в которой хранятся сведения о пользователях и их маршрутах. В базе содержится структурированная информация: названия маршрутов, точки путешествия с координатами, расстояния и прочие необходимые данные. Сервер передаёт запросы в эту базу, получает результаты и отправляет их обратно в мобильное

приложение. Такая организация компонентов позволяет разделить ответственность между частями системы, что повышает надёжность, упрощает поддержку и обеспечивает возможность масштабирования приложения в будущем.

Диаграмма компонентов отражает структуру приложения сверху: клиент отправляет запросы серверу, сервер выполняет обработку и работает с базой данных. Такой подход обеспечивает гибкость, модульность и удобство расширения системы [19, с.72]

Далее была сделана диаграмма развёртывания (Рисунок 14). На схеме показаны основные части системы: мобильное устройство, сервер приложения, база данных и внешние сервисы. Мобильное приложение работает на телефоне пользователя и позволяет удобно планировать поездку прямо с экрана устройства. Сервер в облаке принимает все запросы, обрабатывает их и соединяет между собой разные части системы. База данных хранит всю информацию о пользователях, маршрутах, поездках и бронированиях, чтобы её можно было быстро получить. Внешние сервисы дают актуальные данные о погоде, отелях и авиа рейсах.

Основные элементы развёртывания:

- Мобильное устройство - клиент, где установлено приложение.
- Сервер приложения (облако) - обрабатывает запросы, управляет бизнес-логикой.
- База данных - хранит информацию о пользователях, поездках и бронированиях.
- Внешние сервисы - предоставляют данные о погоде, отелях и авиа рейсах.

Когда пользователь создаёт маршрут или обновляет информацию, мобильное приложение отправляет запрос на сервер, а сервер проверяет данные и при необходимости обращается к базе данных. Всё это происходит быстро, так что пользователь видит результат почти сразу. Такой подход делает работу приложения удобной и надёжной.



Рисунок 14 - Диаграмма развёртывания

Пользователь может спокойно планировать поездку, видеть актуальную информацию и вносить изменения в маршруты без лишних сложностей. Также распределение функций между клиентом, сервером и внешними сервисами облегчает поддержку приложения и позволяет легко добавлять новые функции в будущем. Система построена так, что она остаётся гибкой и может расти вместе с потребностями пользователей. Всё это вместе делает приложение удобным, функциональным и современным инструментом для планирования путешествий [19, с.88]

3 Реализация приложения для путешествий

3.1 Тестирование проектирования системы

Разработка прототипа мобильного приложения «Express» стала основным этапом, на котором теоретические решения и проектные схемы были преобразованы в работающий программный продукт. Прототип включал базовые функции, необходимые для проверки концепции и оценки удобства интерфейса.

Основная цель создания прототипа заключалась в том, чтобы показать, как будет функционировать приложение, протестировать пользовательские сценарии, убедиться в удобстве навигации и предусмотреть возможные улучшения ещё до написания окончательной версии продукта [25, с.120]

Для разработки интерфейса использовался фреймворк ReactNative, так как он позволяет быстро создавать интерактивные элементы и легко управлять состоянием приложения. На этом этапе были реализованы:

- Каркас главных экранов (маршруты, профиль, карта, список точек).
- Простейшая навигация между экранами.
- Визуальная структура, основанная на макетах, выполненных ранее.

Интерфейс был разработан в светлой теме, соответствующей выбранной цветовой палитре (синий, серый, чёрный, белый, зелёный). На этапе прототипирования акцент делался на функциональности и расположении элементов, а не на финальных визуальных деталях.

Серверная часть была создана на платформе Node.js с использованием фреймворка Express.js. На этапе прототипа были реализованы базовые API-методы:

- регистрация и авторизация пользователя,
- создание маршрута,
- добавления точек путешествия,
- получение списка всех маршрутов,
- просмотр данных одной поездки.

API работал с тестовой структурой базы данных и позволял проверить взаимодействие между клиентом и сервером. Такой подход помог заранее выявить возможные ошибки в логике обмена данными [26, с.88]

Для тестирования сервера была создана минимальная версия базы данных:

Таблицы:

- users - хранение данных пользователя;
- routes - маршруты;
- points - точки маршрута, связанные с конкретным путешествием;
- sessions - для авторизации.

База данных была упрощена, но полностью соответствовала структуре, разработанной в UML и ER-диаграммах на предыдущих этапах.

Тестирование прототипа

После разработки клиентской и серверной частей было проведено первичное тестирование. Оно включало:

Функциональное тестирование:

- регистрация нового пользователя;
- вход в приложение;
- создание нового маршрута;
- добавление точек;
- переход между экранами;
- отображение карты и точек маршрута.

Интерфейсное тестирование.

- удобство расположения кнопок,
- читаемость текста,
- визуальная логика экранов,
- корректность переходов между разделами.

Прототип позволил убедиться, что разработанная архитектура корректно работает, интерфейс удобен и понятен пользователю, а также что приложение способно выполнять основные функции по планированию путешествий. Благодаря раннему тестированию удалось вовремя выявить недочёты и

оптимизировать дальнейшую разработку.

Диаграмма взаимодействие клиентской части, сервера, базы данных (рисунок 15) - отражает работу основных компонентов приложения на стадии прототипирования и демонстрирует, как между ними происходит обмен данными. Взаимодействие клиентской части, сервера и базы данных.

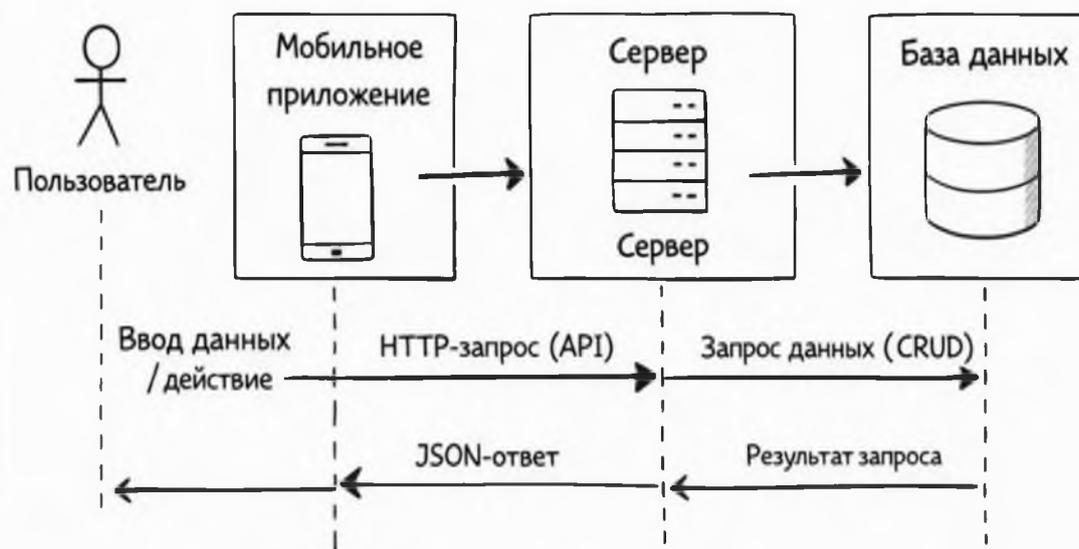


Рисунок 15 - Диаграмма взаимодействие клиентской части, сервера, базы данных

Клиентская часть, разработанная на ReactNative, отправляет запросы к серверу для выполнения ключевых операций, таких как получение списка маршрутов, создание новых точек путешествий и авторизация пользователя.

Серверный модуль, реализованный на Node.js с использованием Express.js, принимает эти запросы, выполняет необходимую бизнес-логику и управляет обработкой данных. Все результаты вычислений и изменения сохраняются в базе данных PostgreSQL, где централизованно хранятся сведения о пользователях, маршрутах и точках путешествий.

Данная диаграмма показывает, что приложение построено по принципу клиент–серверной архитектуры: фронтенд отвечает за взаимодействие с пользователем, сервер — за обработку данных, а база данных — за их надёжное хранение. Такое разделение обеспечивает гибкость разработки, удобство

масштабирования и чёткое разграничение ответственности между слоями системы [27, с.102]

3.2 Анализ затрат и оценка экономической эффективности внедрения приложения

Разработка мобильного приложения «Express» требует предварительной оценки затрат, связанных как с производственным процессом создания программного обеспечения, так и с дальнейшим сопровождением проекта. Проведение экономического анализа позволяет определить обоснованность разработки, рассчитать объём необходимых ресурсов и оценить потенциальные преимущества внедрения приложения в реальную эксплуатацию.

Основу затрат при создании программного продукта составляют трудозатраты специалистов. Даже если приложение создаётся как учебный проект, при расчётах используются усреднённые показатели рыночных зарплат, поскольку они позволяют объективно оценить стоимость разработки [19, с.128]

Для создания мобильного приложения требуется участие нескольких ключевых ролей: frontend-разработчика, backend-разработчика, специалиста по базам данных, дизайнера интерфейсов и тестировщика. В коммерческих условиях каждый из этих специалистов выполняет свои функции, однако в учебном проекте их обязанности часто совмещает один человек.

Если рассматривать приложение «Express» как пример разработки среднего уровня сложности, можно выделить следующие основные этапы работ:

- проектирование архитектуры
- разработка интерфейсов
- программирование клиентской и серверной части
- настройка базы данных
- тестирование и подготовка к развёртыванию

Время выполнения каждого этапа имеет прямое влияние на финансовые затраты. В среднем на разработку прототипа подобного приложения уходит от

250 до 350 часов работы, что при условной стоимости часа специалиста даёт ориентировочную стоимость разработки. Этот показатель может варьироваться в зависимости от опыта разработчика и сложности функционала.

Помимо трудозатрат значительная часть расходов связана с программно-техническими ресурсами. Чтобы обеспечить работу серверной части и базы данных, требуется аренда виртуального сервера или использование облачных сервисов. На начальном этапе достаточно минимальной конфигурации VPS, которая обеспечивает обработку запросов и хранение информации. Средняя стоимость аренды такого сервера находится в пределах фиксированной ежемесячной оплаты. Этот расход является постоянным и должен учитываться при расчёте стоимости поддержки приложения. Также следует учитывать расходы на доменное имя, обслуживание SSL-сертификата и возможные затраты на систему резервного копирования.

Тестирование приложения также является важной статьёй затрат. При создании прототипа используются инструменты, позволяющие проверять корректность работы интерфейса, API-методов и взаимодействия с базой данных. Функциональное, пользовательское и интеграционное тестирование требует участия специалистов или временных затрат разработчика. В коммерческой разработке тестирование может занимать до 30% общего времени проекта, однако в учебном проекте этот показатель снижается. Тем не менее, условная стоимость этих работ учитывается в оценке, поскольку качественное тестирование является обязательным элементом создания приложения.

Оценка экономической эффективности разработки приложения «Express» основывается на потенциальных выгодах, которые может принести внедрение продукта на рынок.

Современные пользователи активно используют сервисы для планирования путешествий, поскольку такие инструменты позволяют быстрее структурировать поездку, избегать ошибок и экономить время. Приложение «Express» ориентировано на эту потребность и может предложить

пользователям удобный механизм создания маршрутов, сохранения точек интереса и получения персонализированной информации.

С экономической точки зрения приложение может стать основой для построения бизнес-модели, включающей различные виды монетизации. Возможны следующие направления: размещение ненавязчивой рекламы, внедрение премиум-функций, партнерские программы с туристическими компаниями или агентствами, а также создание каталога платных экскурсий и сервисов. Потенциальный доход от этих составляющих способен компенсировать расходы на разработку и обеспечить прибыль при достаточном числе активных пользователей. Партнерская модель монетизации является одной из наиболее перспективных, так как обеспечивает доход без необходимости прямой продажи контента пользователю.

Экономическая эффективность проекта возрастает благодаря использованию бесплатных инструментов разработки. ReactNative, Node.js, Express.js и PostgreSQL предоставляются по лицензиям с открытым исходным кодом, что существенно снижает стоимость создания приложения. Основные затраты переводятся на оплату труда специалистов и аренду инфраструктуры. Это делает проект «Express» экономически привлекательным и устойчивым в перспективе.

В целом анализ затрат и экономической эффективности показывает, что разработка мобильного приложения «Express» является оправданной как с функциональной, так и с финансовой точки зрения.

Таблица 3 - Оценка затрат на разработку мобильного приложения

№	Статья затрат	Описание	Расчёт	Сумма (руб.)
1	Трудовозатраты Frtontend- разработки	Создание интерфейсов навигации, компонентов React	120 часов 800 руб./час	96 000

Продолжение таблицы 3

2	Трудозатраты Backend-разработки	Разработка API, логики, маршрутов	100 часов 900 руб./час	90 000
3	Проектирование базы данных	Модели, связи, оптимизация запросов	40 часов 850 руб./час	34 000
4	UI/UX дизайн	Создание макетов экранов	30 часов 1000 руб./час	30 000
5	Тестирование	Написание тест-кейсов, функциональное и интеграционное тестирование	50 часов 700 руб./час	35 000
6	Доменное имя	Регистрация домена для API	Фиксированная цена	1000

Итого затраты: 291 600 руб.

Таким образом на представленная таблица - 2 отражает ориентировочную структуру затрат на разработку мобильного приложения «Express». В расчётах использованы средние значения стоимости часа работы специалистов на рынке разработки программных продуктов.

Наибольшую часть бюджета занимает оплата труда разработчиков, что является типичной ситуацией для IT-проектов. К инфраструктурным расходам относятся аренда облачного сервера, домен и техническое обслуживание системы. Использование бесплатных технологий (ReactNative, Node.js, PostgreSQL, SSL-сертификат) позволяет существенно снизить общую стоимость проекта и повысить его экономическую эффективность.

На основе выполненных расчётов можно заключить, что приложение

«Express» обладает хорошими экономическими перспективами. При относительно низких затратах на разработку и инфраструктуру проект способен окупиться менее чем за год, а рентабельность в первые 12 месяцев может составить около 21%. Увеличение числа пользователей, внедрение расширенных платных функций или подключение новых партнерских сервисов может существенно повысить доходность. Это делает проект жизнеспособным и перспективным для дальнейшего развития и масштабирования [20, с.135]

Заключение

В ходе выполнения выпускной квалификационной работы была проведена комплексная работа, направленная на проектирование и разработку мобильного приложения для планирования путешествий «Express». Целью исследования являлось создание прототипа удобного, функционального и современного инструмента, который способен обеспечить пользователей возможностью формировать маршруты, сохранять точки интереса и получать структурированную информацию для подготовки поездок. В ходе работы поставленные цели были достигнуты, а задачи выполнены в полном объёме.

На первом этапе был проведён предпроектный анализ, включающий изучение предметной области и исследование существующих решений на рынке туристических приложений. Анализ показал, что современные сервисы обладают широким набором функций, однако часто страдают перегруженностью интерфейса, сложностью восприятия и отсутствием инструментов для персонального планирования маршрутов. Это подтвердило актуальность разработки нового приложения, ориентированного именно на удобство и простоту использования.

Анализ целевой аудитории позволил сформировать набор ключевых пользовательских сценариев и требований. Было установлено, что пользователи нуждаются в понятном интерфейсе, минимальном количестве действий для создания маршрута, возможности работы с интерактивной картой, а также доступе к сохранённым путешествиям. Эти выводы легли в основу последующей разработки интерфейса и архитектуры приложения.

В процессе проектирования была разработана логическая архитектура приложения, включающая клиентскую часть на ReactNative, серверную часть на Node.js и Express.js, а также реляционную базу данных PostgreSQL. Были созданы UML-диаграммы классов, компонентов и развёртывания, а также пользовательские сценарии (userstories), которые помогли формализовать структуру будущего функционала. На основе собранных требований

разработаны макеты пользовательского интерфейса, включающие главные экраны приложения, такие как экран входа, список маршрутов, создание нового маршрута и просмотр данных о путешествии.

В рамках реализации прототипа создана базовая функциональная версия приложения, отражающая основные механики работы: отображение маршрутов, создание записей, связь с сервером и базой данных. Были выполнены тестирование прототипа и анализ его корректности. В процессе тестирования были выявлены недочёты, которые впоследствии устранены. Это позволило получить стабильную и работоспособную версию приложения, пригодную для дальнейшего расширения.

Отдельное внимание было уделено экономическому анализу проекта. Проведённый расчёт затрат показал, что разработка приложения требует умеренных инвестиций благодаря использованию технологий с открытым исходным кодом. Прогноз экономической эффективности подтвердил, что приложение имеет потенциал для окупаемости в течение первого года эксплуатации. Возможные модели монетизации, такие как реклама, партнерские программы или премиум-функции, позволяют сделать проект финансово устойчивым и перспективным для дальнейшего развития.

В итоге выполненная работа демонстрирует, что разработка мобильного приложения для планирования путешествий является актуальной, технически реализуемой и экономически целесообразной. Приложение «Express» обладает потенциалом дальнейшего роста, масштабирования и интеграции с внешними туристическими сервисами. Разработанный прототип может служить основой для последующей коммерческой разработки, расширения функционала и выхода на рынок мобильных туристических решений.

Список литературы

1. Адаптивный дизайн сайта — вёрстка адаптивного сайта [Электронный ресурс]. – URL: <https://www.rush-analytics.ru/blog/adaptivnyj-dizajn-sajta-cto-eto-takoe> (дата обращения: 25.12.2024).
2. Бабенко, А.А., Ушакова, С.Н. Разработка web-приложения «кинотеатра» с использованием JavaScript, PHP и MySQL // Мирская наука. – 2018. – №12(21) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/razrabotka-web-prilozheniya-kinoteatra-s-ispolzovaniem-javascript-php-i-mysql>(дата обращения: 09.12.2025).
3. Герасимова, А.И. Актуальность прототипирования сайта // Форум молодых учёных. – 2018. – №5-1(21) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/aktualnost-prototipirovaniya-sayta>(дата обращения: 10.12.2025).
4. Лазченко, В.Р. Введение в тестирование программного обеспечения // Форум молодых учёных. – 2019. – №1-2(29) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/vvedenie-v-testirovanie-programmnogo-obespecheniya> (дата обращения: 10.12.2025).
5. Мостипака, А.Е. Тестирование программного обеспечения // Наука и образование сегодня. – 2020. – №12(59) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/testirovanie-programmnogo-obespecheniya> (дата обращения: 10.12.2024).
6. Несмеянов, П.П. Облачное тестирование в сравнении с традиционным тестированием ПО // Международный журнал гуманитарных и естественных наук. – 2023. – №6-3(81) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/oblachnoe-testirovanie-v-sravnenii-s-traditsionnym-testirovaniem-programmnogo-obespecheniya> (дата обращения: 11.12.2025).
7. Секлетова, Н.Н., Кондратьев, А.И. Методы тестирования программного обеспечения // Экономика и социум. – 2018. – №10(53) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/metody>

testirovaniya-programmnogo-obespecheniya(дата обращения: 10.12.2025).

8. Сиразетдинов, Р.Р., Белоус, Д.В. Архитектура информационных систем // Техника средств связи. – 2020. – №3(151) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/arhitektura-informatsionnyh-sistem> (дата обращения: 10.12.2025).

9. Акопян, А.Р., Аракелян, А.М., Воронцова, Ю.В., Крысов, В.В. Оценка подходов к регулированию цифрового кинопроката // Вестник университета. – 2021. – №4 [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/otsenka-podhodov-k-regulirovaniyu-tsifrovogo-kinoprokata> (дата обращения: 08.01.2025).

10. Акопян, А.Р., Аракелян, А.М., Воронцова, Ю.В., Крысов, В.В. Проблемы цифровой трансформации кинопроката // E-Management. – 2021. – №1 [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/problemy-tsifrovoy-transformatsii-kinoprokata>(дата обращения: 08.01.2025).

11. Березина, А.Д. Развитие рынка онлайн-видеосервисов в экосистеме цифровой экономики // Экономика и бизнес: теория и практика. – 2019. – №10-1 [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/razvitie-rynka-onlayn-videoservisov-v-ekosisteme-tsifrovoy-ekonomiki>(дата обращения: 08.01.2025).

12. Бритвина, П.В. Онлайн-безопасность: защита данных и современные возможности // Вестник науки. – 2023. – №8(65) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/onlayn-bezopasnost-zaschita-dannyh-i-sovremennye-vozmozhnosti>(дата обращения: 14.12.2025).

13. Ельсуков, Д.А. Безопасность данных в сети Интернет // Экономика и социум. – 2021. – №11-1(90) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/bezopasnost-dannyh-v-seti-internet>(дата обращения: 21.12.2025).

14. Зеленкина, А.С., Павлюкевич, Е.А., Поначугин, А.В. VPN: использование, безопасность, риски // ИТНОУ. – 2022. – №1(19) [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/vpn-ispolzovanie-bezopasnost-riski>

(дата обращения: 16.12.2025).

15. Линь, Д. Основные тенденции в сфере защиты авторского права и персональных данных в сети Интернет // Административное право и практика администрирования. – 2020. – №1 [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/osnovnyye-tendentsii-v-sfere-zaschityi-avtorskogo-prava-i-personalnyh-dannyh-v-seti-internet>(дата обращения: 21.12.2025).

16. Полный обзор инструментов адаптивного дизайна [Электронный ресурс]. – Habr. – URL:<https://habr.com/ru/articles/142120/> (дата обращения: 25.12.2025).

17. Разработка сайтов: тренды и современные технологии [Электронный ресурс]. – Citprofi. – URL: <https://citprofi.ru/blog/razrabotka-saytov-v-2023-godu-trendy-i-noveyshie-tekhnologii>(дата обращения: 25.12.2025).

18. Шишкин, Д.В. Мобильные приложения: понятие, виды и особенности разработки // Информационные технологии и системы. – 2023.

19. Проектирование мобильных и веб-интерфейсов [Электронный ресурс]. – RDBX. – URL: <https://rdbx.ru>(дата обращения: 12.01.2026).

20. Этапы UX-проектирования мобильного приложения [Электронный ресурс]. – Cleverence. – URL: <https://www.cleverence.ru/articles/ux> (дата обращения: 12.01.2026).

21. Комплексное руководство по дизайну мобильных приложений [Электронный ресурс]. – UX PUB. – URL: <https://uxpub.ru> (дата обращения: 12.01.2026).

22. Основы UX/UI проектирования мобильных приложений [Электронный ресурс]. – Яндекс Практикум. – URL:<https://practicum.yandex.ru> (дата обращения: 12.01.2026).

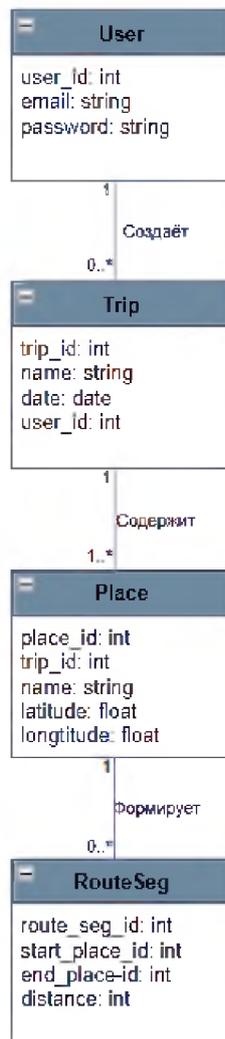
23. Проектирование и UX/UI-дизайн мобильных приложений [Электронный ресурс]. – IT-компания Aiston. – URL: <https://aiston.ru>(дата обращения: 12.01.2026).

24. Захаров, И.Н. Сервис-ориентированная архитектура информационных систем. – М.: ДМК-Пресс, 2022. – 256 с.

25. Шишкин, Д.В. Мобильные приложения в туризме: современные тренды // Туризм и технологии. – 2024.
26. Кузнецова, Е.А. Мобильные технологии и их применение в IT-сервисах. – Новосибирск: Сибирское издательство, 2024. – 320 с.
27. Смирнова, Н.И. Основы проектирования человеко-компьютерного взаимодействия. – Казань: Казанский университет, 2021. – 210 с.
28. Беляев, В.А. Базы данных: проектирование, реализация, оптимизация. – Санкт-Петербург: Питер, 2022. – 304 с.
29. Захаров, И.Н. Web-сервисы и API: проектирование и взаимодействие. – М.: ДМК-Пресс, 2023. – 280 с.
30. Новиков, М.П. Проектирование клиент-серверных систем. – Екатеринбург: УрФУ, 2022. – 272 с.

Приложение 1

UML-диаграмма классов



Код диаграммы:

```
// Пользователь
```

```
classUser {
```

```
constructor(id, name, email) {
```

```
    this.id = id;
```

```
    this.name = name;
```

```
    this.email = email;
```

```
    this.trips = [];
```

```
    this.bookings = [];
```

Продолжение приложения 1

```
    this.notifications = [];  
  }  
}
```

// Поездка

```
class Trip {  
  constructor(id, title) {  
    this.id = id;  
    this.title = title;  
    this.destinations = [];  
  }  
}
```

// Точка маршрута

```
class Destination {  
  constructor(name, date) {  
    this.name = name;  
    this.date = date;  
  }  
}
```

// Бронирование

```
class Booking {  
  constructor(type, info) {  
    this.type = type; // flight / hotel / car  
    this.info = info;  
  }  
}
```

Продолжение приложения 1

// Уведомление

```
class Notification {  
  constructor(message) {  
    this.message = message;  
    this.read = false;  
  }  
}
```

// Сервис работы с пользователем

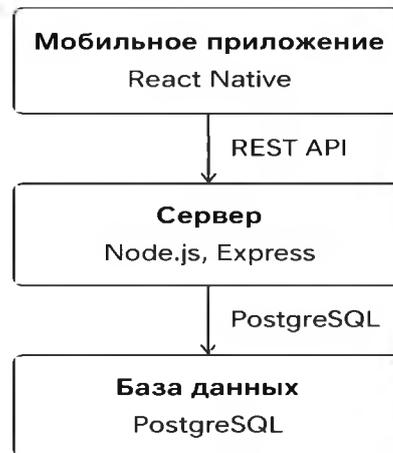
```
class UserService {  
  addTrip(user, trip) {  
    user.trips.push(trip);  
  }  
  
  addBooking(user, booking) {  
    user.bookings.push(booking);  
  }  
  
  notify(user, message) {  
    user.notifications.push(new Notification(message));  
  }  
}
```

// Внешний сервис карт

```
class MapService {  
  buildRoute(destinations) {  
    return `Маршрут через ${destinations.length} точек построен`;  
  }  
}
```

Приложение 2

Диаграмма компонентов



Код диаграммы компонентов:

```
@startuml
```

```
title Диаграмма компонентов мобильного приложения планирования  
путешествий
```

```
component "UI Component\n(Пользовательский интерфейс)" as UI  
component "Business Logic Component\n(Бизнес-логика)" as BL  
component "Data Access Component\n(Доступ к данным)" as DA
```

```
component "External APIs\n(Погода, Отели, Авиарейсы)" as API  
database "Database\n(Пользователи, поездки, бронирования)" as DB
```

```
UI --> BL : запросы пользователя
```

```
BL --> DA : запросы к данным
```

```
DA --> DB : CRUD-операции
```

```
DA --> API : REST API-запросы
```

```
@enduml
```

Приложение 3

Диаграмма развёртывания



Код диаграммы развёртывания:

// Мобильное устройство (клиент)

```
class MobileClient {
  constructor(platform) {
    this.platform = platform; // Android / iOS
  }

  sendRequest(apiServer) {
    return apiServer.processRequest();
  }
}
```

// Серверприложения (облако)

```
class ApplicationServer {
  constructor(database, externalServices) {
    this.database = database;
    this.externalServices = externalServices;
  }
}
```

Продолжение приложения 3

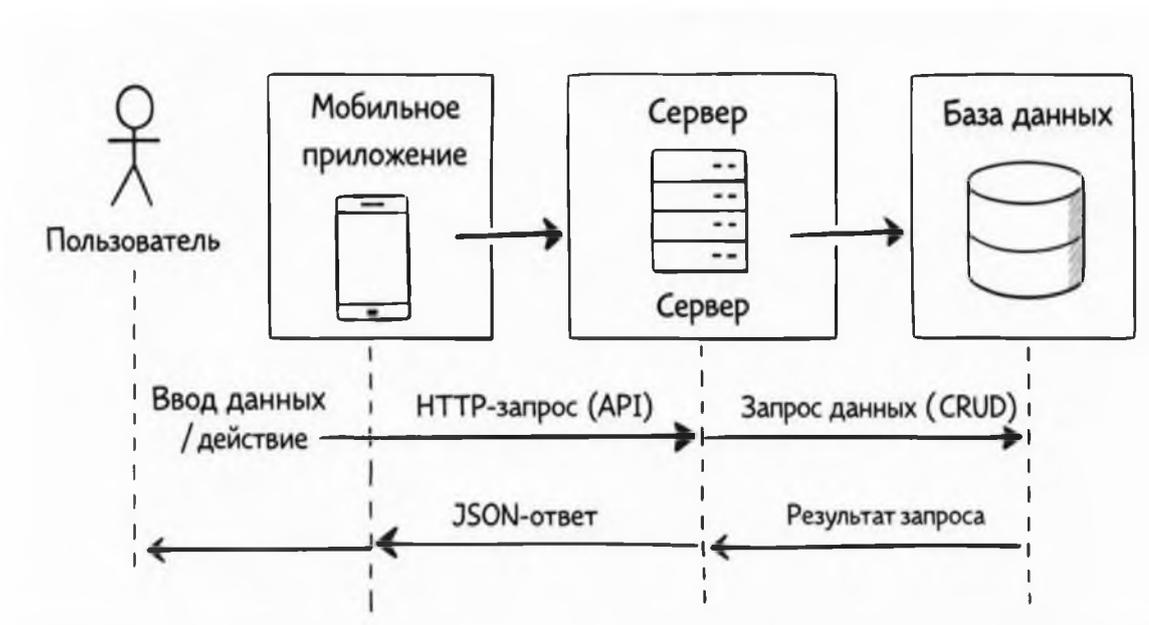
```
processRequest() {
    this.database.getData();
    this.externalServices.getExternalData();
return "Ответ клиенту сформирован";
}
}

// База данных
class Database {
    getData() {
        return "Данные пользователей, поездок и бронирований";
    }
}

// Внешние сервисы
class ExternalServices {
    getExternalData() {
return "Погода, отели, авиарейсы";
    }
}
mobileApp.sendRequest(server);
```

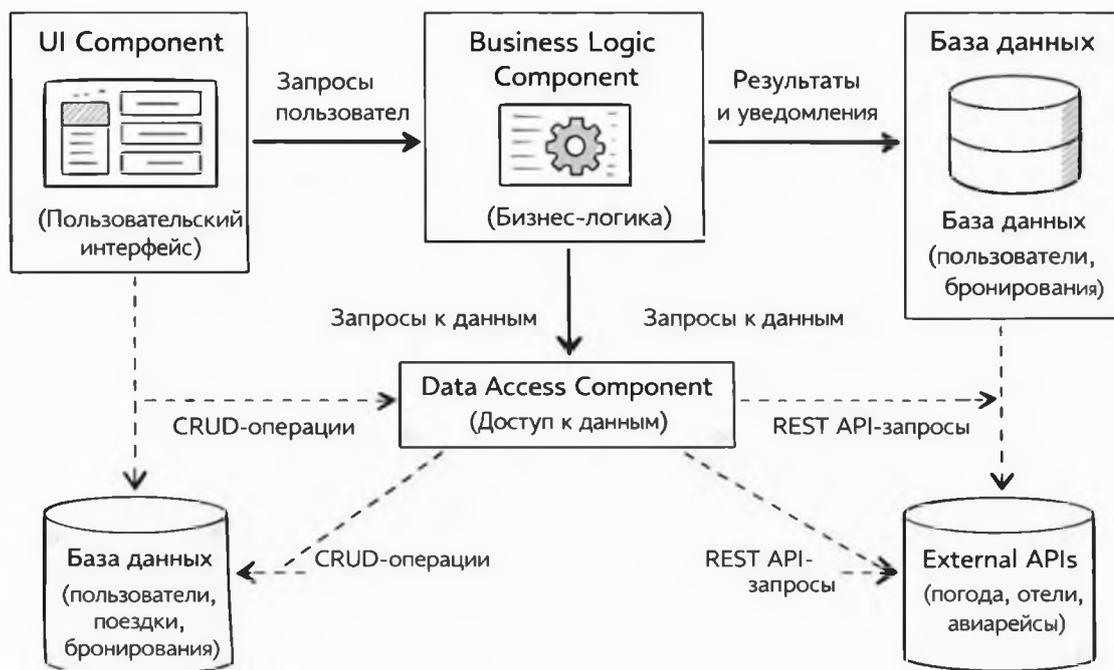
Приложение 4

Схема взаимодействия клиент–сервер–база данных



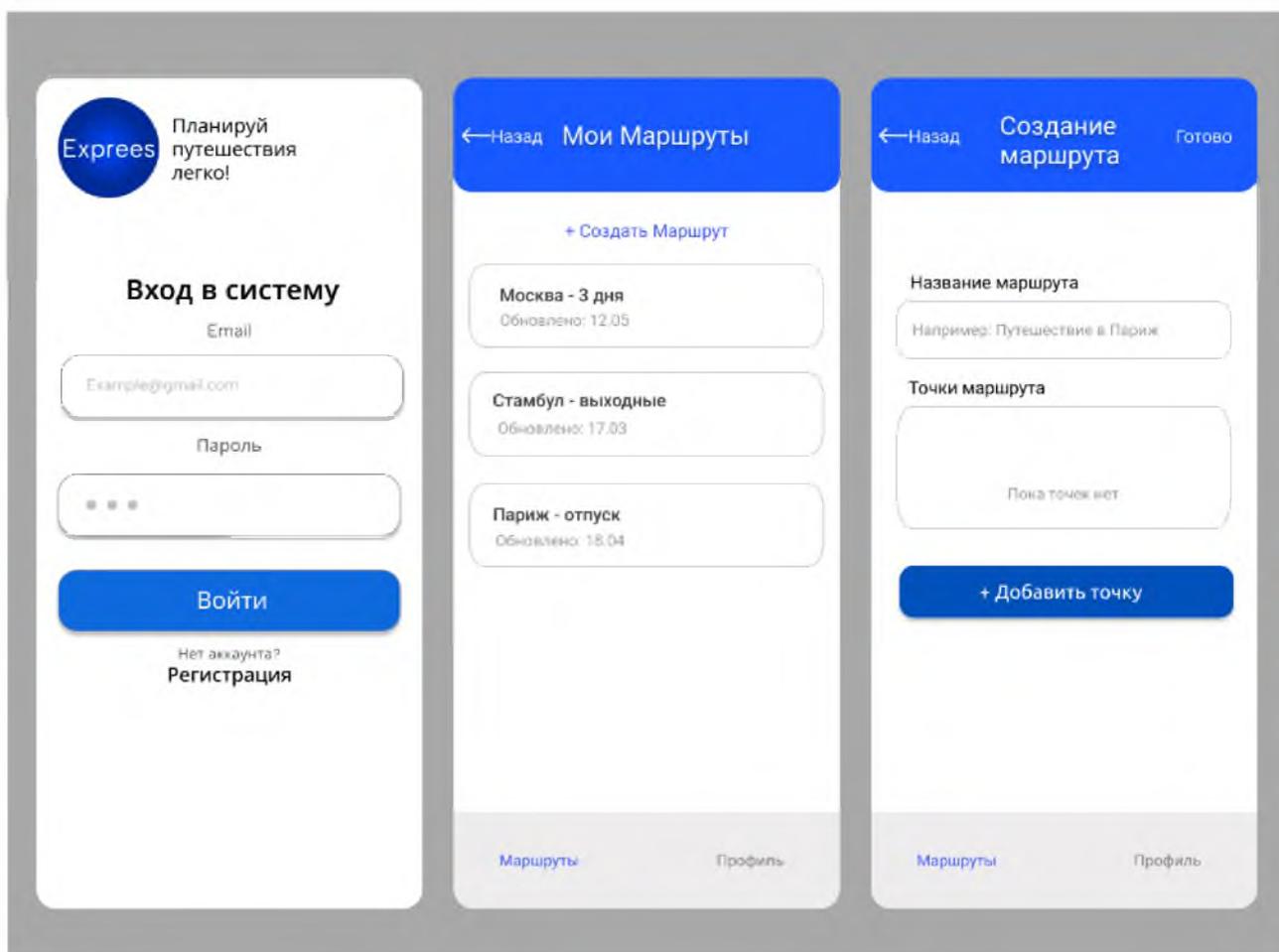
Приложение 5

Взаимодействие компонентов приложения

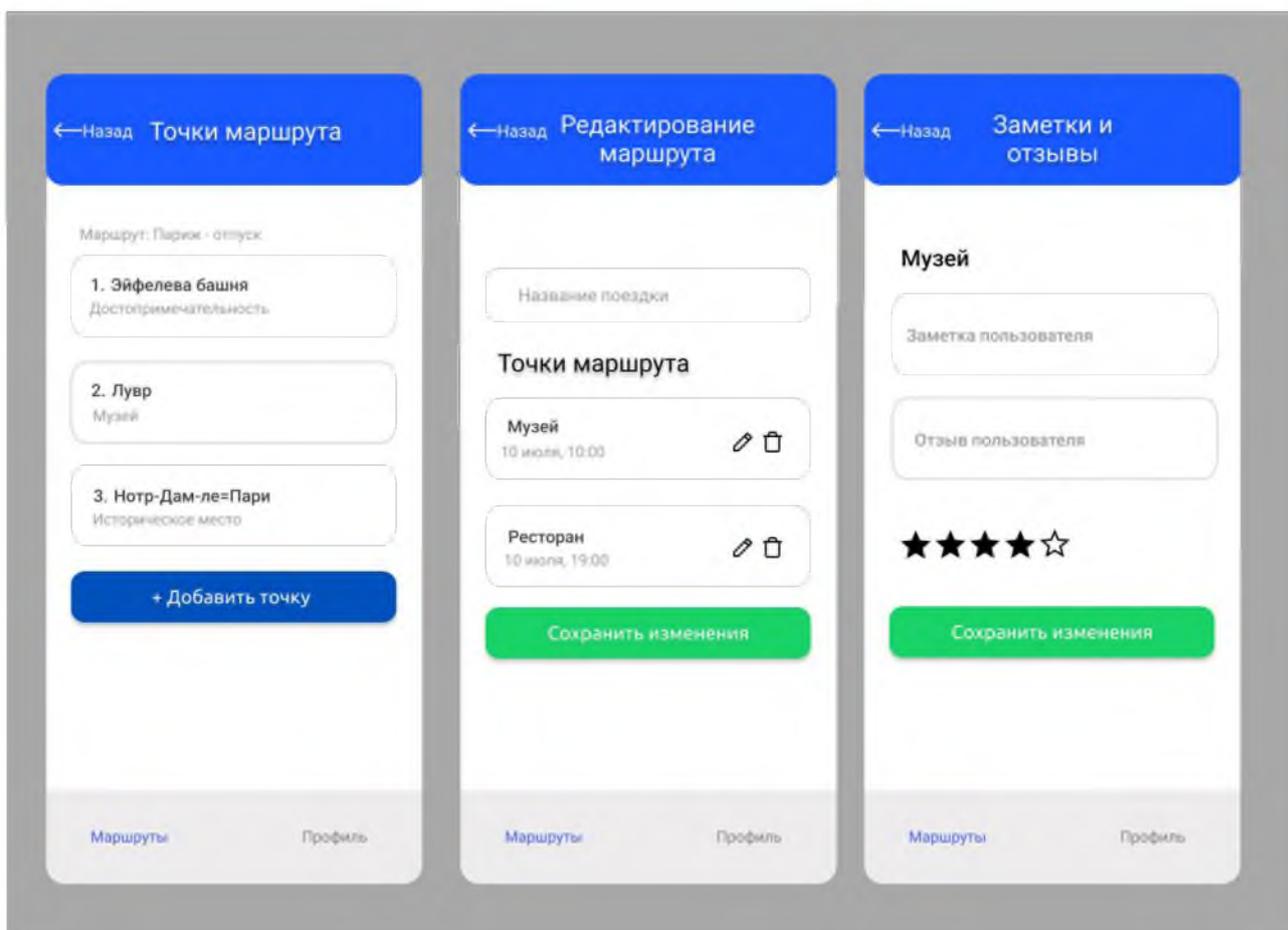


Приложение 6

Макеты экранов приложения



Продолжение приложения 6



Приложение 7

Таблица затрат

№	Статья затрат	Описание	Расчёт	Сумма (руб.)
1	Трудозатраты Frontend- разработки	Создание интерфейсов навигации, компонентов React	120 часов 800 руб./час	96 000
2	Трудозатраты Backend- разработки	Разработка API, логики, маршрутов	100 часов 900 руб./час	90 000
3	Проектирование базы данных	Модели, связи, оптимизация запросов	40 часов 850 руб./час	34 000
4	UI/UX дизайн	Создание макетов экранов	30 часов 1000 руб./час	30 000
5	Тестирование	Написание тест- кейсов, функциональное и интеграционное тестирование	50 часов 700 руб./час	35 000
6	Доменное имя	Регистрация домена для API	Фиксированн ая цена	1000
7	SSL-сертификат	Защита соединения HTTPS	Бесплатный (Lets encrypt)	0