

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МОиПЭВМ

Направление подготовки 09.03.04 «Программная инженерия»

Направленность Программная инженерия

БАКАЛАВРСКАЯ РАБОТА

на тему

**Фронтенд-компонент информационной
системы для автоматизации бронирования и
учета услуг рыболовного хозяйства**

Студент

(подпись, дата)

Ермолаев Илья Олегович

(ФИО полностью)

Руководитель

(подпись, дата)

Козлов А. Ю.

(фамилия, инициалы)

Нормоконтролёр

(подпись, дата)

Такташкин Д.В.

(фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от 03.06.2026 № 12)

Заведующий кафедрой

(подпись)

Козлов А.Ю.

(фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК

(подпись)

Попова Н.А.

(фамилия, инициалы)

Пенза, 2026

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
А.Ю. Козлов
«26» декабря 2025 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Ермолаев Илья Олегович гр. 22ВП2 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы Фронтенд-компонент информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства

Тема утверждена приказом ПГУ №1583 от "30" декабря 2025г.

3. Руководитель работы заведующий кафедрой МОиПЭВМ, д. т. н. Козлов А. Ю.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

1) Назначение: разработка фронтенд-приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

2) Среда выполнения: Windows 10.

Языковые средства разработки: HTML, CSS, TypeScript, React, SQL, UML.

Инструментальные средства: Visual Studio Code, PostgreSQL, Visual Paradigm 18.0.

Технологии разработки: ООП.

3) Основные функции приложения:

а) авторизация и регистрация в системе;

б) управление бронированиями;

в) управление секторами;

г) управление тарифами;

д) управление товарами для аренды и продажи;

е) формирование отчётности;

5. Перечень подлежащих разработке вопросов _____

- 1) Анализ предметной области и постановка задачи;
- 2) Проектирование;
- 3) Реализация;
- 4) Тестирование программных средств;
- 5) Оформление пояснительной записки.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и требований к системе	10%	07.04.2026 – 13.04.2026	
2	Проектирование программных средств	20%	14.04.2026 – 25.04.2026	
3	Реализация программных средств	30%	27.04.2026 – 11.05.2026	
4	Тестирование программных средств	20%	12.05.2026 – 21.05.2026	
5	Оформление пояснительной записки	20%	22.05.2026 – 02.06.2026	

Дата выдачи задания «26» декабря 2025г.

Руководитель бакалаврской работы _____ 26.12.2025г.
(подпись, дата)

Задание к исполнению принял студент _____ 26.12.2025г.
(подпись, дата)

Работу к защите допустить
Декан ФВТ _____ Л.Р. Фионова
(подпись, дата)

Реферат

Пояснительная записка содержит 120 страниц, 64 рисунка, 32 таблицы, 21 использованный источник и 7 приложений.

ИНФОРМАЦИОННАЯ СИСТЕМА, ФРОНТЕНД-КОМПОНЕНТ, РЫБОЛОВНОЕ ХОЗЯЙСТВО, VISUAL STUDIO CODE, POSTGRESQL, VISUAL PARADIGM, HTML, CSS, TYPESCRIPT, REACT, UML.

Объект исследования: информационные системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

Цель выпускной квалификационной работы: разработать фронтенд-приложение информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

Технологии разработки: разработка всех средств фронтенд-приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства производилась в среде разработки Visual Studio Code. Разработка клиентской части проводилась на языках разметки HTML, CSS и языке программирования TypeScript с использованием фреймворка React. Разработка базы данных проводилась в СУБД PostgreSQL.

Среда выполнения: Windows 10, веб-браузер.

Результаты разработки: в ходе выполнения выпускной квалификационной работы было спроектировано и реализовано фронтенд-приложение информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства. Все необходимые требования к работе системы были соблюдены.

					ПГУ 09.03.04 – 08BP222.9 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Фронтенд-компонент информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства. Пояснительная записка	Лит.	Лист	Листов
Разраб.		Ермолаев И. О.					4	120
Провер.		Козлов А. Ю.				ГРУППА 22ВП2		
Реценз.								
Н. Контр.		Такташкин Д. В.						
Утверд.		Козлов А. Ю.						

Содержание

Введение	7
1 Постановка задачи и анализ требований к фронтенд-приложению информационной системе для автоматизации бронирования и учёта услуг рыболовного хозяйства	9
1.1 Постановка задачи на разработку приложения	9
1.2 Анализ программ-аналогов	11
1.2.1 Цели и задачи анализа	11
1.2.2 Анализ системы «Fishing-avalon»	12
1.2.3 Анализ системы «Фрегат»	13
1.2.4 Анализ системы «Клевое место»	14
1.2.5 Результат сравнения программ-аналогов	15
1.3 Анализ функциональных требований	17
1.4 Анализ нефункциональных требований	22
1.5 Планирование разработки и оценка бюджета	23
2 Проектирование фронтенд-приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства	27
2.1 Проектирование архитектуры системы	27
2.2 Проектирование структур данных	29
2.2.1 Концептуальная модель базы данных	29
2.2.2 Логическая модель базы данных	31
2.3 Проектирование программных средств	33
2.4 Проектирование пользовательского интерфейса	39
2.5 Выбор технологий и инструментальных средств разработки системы ..	41
2.5.1 Выбор технологий разработки приложения	41

2.5.2	Выбор инструментальных средств разработки приложения.....	43
3	Реализация фронтенд-приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.....	45
3.1	Реализация физической модели базы данных	45
3.2	Реализация серверной части информационной системы.....	46
3.3	Реализация клиентской части информационной системы.....	49
3.4	Реализация интерфейса информационной системы.....	50
4	Тестирование фронтенд-приложения информационной системы для автоматизации бронирования и учёту услуг рыболовного хозяйства	61
4.1	Функциональное тестирование приложения.....	61
4.2	Метрики кода приложения	63
4.2.1	Метрики размера программы	64
4.2.2	Метрики Холстеда	65
4.2.3	Объектная метрика WMC.....	67
	Заключение	68
	Список использованных источников	70
	Приложение А (справочное) Глоссарий	73
	Приложение Б (обязательное) Листинг приложения	75
	Приложение В (рекомендуемое) Спецификация сущностей базы данных	87
	Приложение Г (рекомендуемое) Диаграмма компонентов.....	95
	Приложение Д (рекомендуемое) Тест-кейсы функционального тестирования .	97
	Приложение Е (рекомендуемое) Результаты функционального тестирования	104
	Приложение Ж (рекомендуемое) Акт о внедрении.....	119

Введение

В современных условиях развитие цифровых технологий позволяет автоматизировать процессы управления в различных сферах деятельности, включая организацию досуга и отдыха. Рыболовные базы представляют собой популярный вид рекреационного бизнеса, эффективность которого напрямую зависит от качества обслуживания клиентов и оптимизации внутренних процессов.

В настоящее время владельцы рыболовных хозяйств сталкиваются с рядом трудностей, в том числе отсутствие возможности онлайн-бронирования, невозможность автоматического подсчёта итоговой суммы бронирования и отсутствие подробной информации о секторах в лице фотографий и характеристик.

В результате наличия подобных проблем возникает противоречие между спросом на осуществление отдыха и недостатком обеспечения автоматизации в сфере малого бизнеса. Именно поэтому возникает острая потребность в разработке комплексного и универсального решения, способного автоматизировать большинство операций и повысить качество пользовательского опыта.

Объектом исследования работы являются процессы бронирования услуг на рыболовной базе.

В качестве предмета исследования выступают программные средства для автоматизации бронирования и учёта услуг рыболовного хозяйства.

Ключевой целью выпускной квалификационной работы является разработка фронтенд-компонента информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- осуществить анализ предметной области и сформулировать ключевые функциональные и нефункциональные требования;

- провести планирование разработки и оценить стоимость бюджета на разработку программного продукта;
- спроектировать схемы концептуальной, логической и физической моделей баз данных;
- определить архитектуру приложения и используемую структуру данных;
- реализовать программный продукт с учётом полученных требований;
- сделать тестирование разработанной информационной системы и осуществить оценку программного кода при помощи метрик кода.

Выполнение каждого из этапов позволит реализовать многофункциональный и универсальный проект, который обеспечит высококачественное взаимодействие между клиентами и администраторами системы; а также повысит уровень масштабируемости и развертываемости на других платформах.

1 Постановка задачи и анализ требований к фронтенд-приложению информационной системе для автоматизации бронирования и учёта услуг рыболовного хозяйства

1.1 Постановка задачи на разработку приложения

В рамках разработки выпускной квалификационной работы ключевой рассматриваемой предметной областью является рыболовное хозяйство.

Рыболовное хозяйство представляет собой единый организационно-производственный комплекс, предназначенный для обеспечения услуг по реализации рыболовства, осуществления содержания различных водных биологических ресурсов; а также необходимого обслуживания всех посетителей и клиентов на территории хозяйства [1].

Главной целью деятельности данной системы является организация контролируемого рыболовства, обеспечение сохранности имеющихся в наличии рыбных ресурсов, ведение полноценного учёта всех оказываемых услуг и повышение качества обслуживания.

Главенствующим компонентом в рамках представленной системы является водоём, предназначенный для осуществления разведения и вылова рыбы. Сам водоём территориально подразделяется на сектора – обособленные участки, имеющие свои отличительные особенности: максимальное количество размещаемых гостей, глубина, сторона по отношению к водоёму, наличие беседки или жезлонга. Помимо самого водоёма, в состав рыболовного хозяйства также входит административная инфраструктура для ведения учёта и управления текущим состоянием пруда, платежная система для реализации транзакций аренды секторов и операций с каталогом товаров; а также персонал, непосредственно участвующий в функционировании хозяйства.

Основными действующими лицами внутри системы являются администратор рыболовного хозяйства и клиенты. Для администратора системы ключевое значение имеют операции по учёту и управлению инфраструктурой

рыболовного хозяйства, управлению бронированиями секторов со стороны пользователей, комплексному учёту имеющихся в наличии товаров; а также систематическому анализу по выручке за определенный промежуток времени, загрузке секторов и статистике по товарам. Для клиента же основной функционал связан с ознакомлением со всей информацией по секторам, бронированием выбранного сектора с автоматическим подсчётом итоговой суммы, а также реализацией заказа сопутствующих товаров и услуг.

В последнее время сфера рыболовного хозяйства сталкивается с рядом трудностей. Так, администраторам становится тяжело вести полноценный учёт и управление инфраструктурой системы из-за обилия имеющейся информации; наблюдается отсутствие высокотехнологических средств для обработки заявок на бронирование, что негативно сказывается на пользовательском опыте использования системы; а также необходимость в корректной работе подсистемы расчёта стоимости во избежание лишних наценок и платежей. Все вышеперечисленные факторы влияют на качество обслуживания и общую работу системы.

Применение современных информационных технологий позволяет решить данные проблемы. Так – автоматизация позволяет осуществить упрощение рутинных процессов в системе и реализацию своевременного обмена информацией между администраторами системы и клиентами. Внедрение подобных механизмов способно нормализовать обработку данных для внесения достоверных изменений; а также улучшить организацию комплексного процесса.

Целью разработки является создание клиентского приложения информационной системы по автоматизации и учёту услуг рыболовного хозяйства. Данный продукт должен включать в себя весь необходимый инструментарий для обеспечения работы как администратора системы, так и базового клиента. Среди ключевых задач – возможность регистрации и авторизации администраторов в системе; просмотр списка имеющихся секторов; наличие операции по созданию бронирования сектора со стороны клиента и

подтверждение текущей брони со стороны администратора; а также управление каталогом товаров и услуг, представленных в системе.

Исходя из того, что основной акцент делается на клиентской части информационной системы – необходимо, чтобы архитектура соответствовала тем условиям, которые обеспечат эффективную и высококачественную работу пользовательских сценариев и алгоритмов. К тому же – представленные критерии должны обеспечить поддержание основных функций разрабатываемого приложения: обработку всех запросов и управление доступом к различным ресурсам информационной системы. Клиентское приложение будет направлено на поддержку высокого уровня стабильности, надёжности и масштабируемости платформы для обеспечения вспомогательного функционала приложения и выполнения бизнес-логики проекта.

Таким образом, можно сказать, что автоматизация процессов учёта услуг рыболовного хозяйства – это актуальное и полезное решение, позволяющее повысить эффективность работы, качество обслуживания клиентов и достоверность управления инфраструктурой.

1.2 Анализ программ-аналогов

1.2.1 Цели и задачи анализа

В рамках разработки клиентского приложения информационной системы по автоматизации и учёту услуг рыболовного хозяйства необходимо проведение сравнительного анализа программ-аналогов подобного формата. Данная работы нужна для выявления преимуществ и недостатков каждого из существующих на рынке решений; а также определения базового функционала, необходимого для реализации программного продукта.

Рассмотрим ключевые задачи данного анализа:

- определение критериев оценки;
- выбор приложений для сравнения из потенциальных существующих аналогов;

- проведение исследования каждого из приложений на основании выбранных критериев;
- обработка полученных результатов.

Для проведения анализа программ-аналогов были выбраны 3 программных продукта, наиболее соответствующих к рассмотрению и проведению их комплексной оценки. Рассмотрим каждую из систем по отдельности.

1.2.2 Анализ системы «Fishing-avalon»

Первая система для анализа – сайт рыболовной базы Fishing-avalon, представляющий собой ресурс для осуществления отдыха и туризма на базе отдыха «Авалон» находящуюся в Астраханской области в зоне Нижней Волги [2].

Данный ресурс представляет набор возможностей для полноценного рыболовного отдыха: страницы с описанием имеющихся услуг и тарифов внутри сервиса; наличие опции бронирования участка на определённый срок; а также возможность покупки и продажи тематических товаров через рыболовный магазин.

Интерфейс главной страницы системы представлен на рисунке 1.

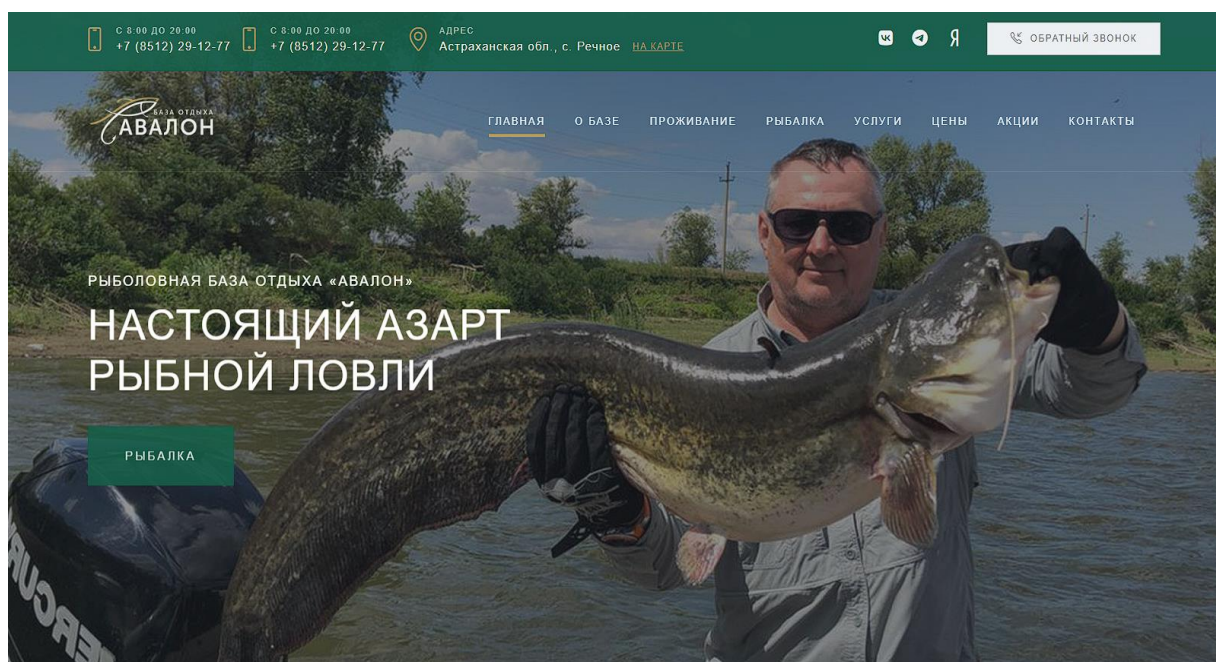


Рисунок 1 – Интерфейс главной страницы системы «Fishing-avalon»

Среди сильных сторон данного программного продукта можно выделить отличную визуальную и навигационную организацию клиентского интерфейса; благодаря чему новому пользователю намного проще перейти к нужной странице и осуществить необходимую операцию; наличие большого ассортимента различных услуг; а также адаптивность под различные платформы.

Однако, представленная система имеет и недостатки. Так, можно выделить отсутствие прозрачности отслеживания текущего статуса бронирования; а также сложность внедрения новых механизмов внутри имеющегося функционала.

1.2.3 Анализ системы «Фрегат»

Следующая система – это сайт рыболовной базы отдыха «Фрегат», располагающейся в Астраханской области [3]. Представленная система реализует осуществление отдыха у водоёма и управление рыболовными услугами. Она содержит в себе все необходимые разделы для клиента: просмотр текущей информации, ознакомление с прайс-листом, реализация бронирования и заказ сопутствующих услуг.

Интерфейс главной страницы системы представлен на рисунке 2.

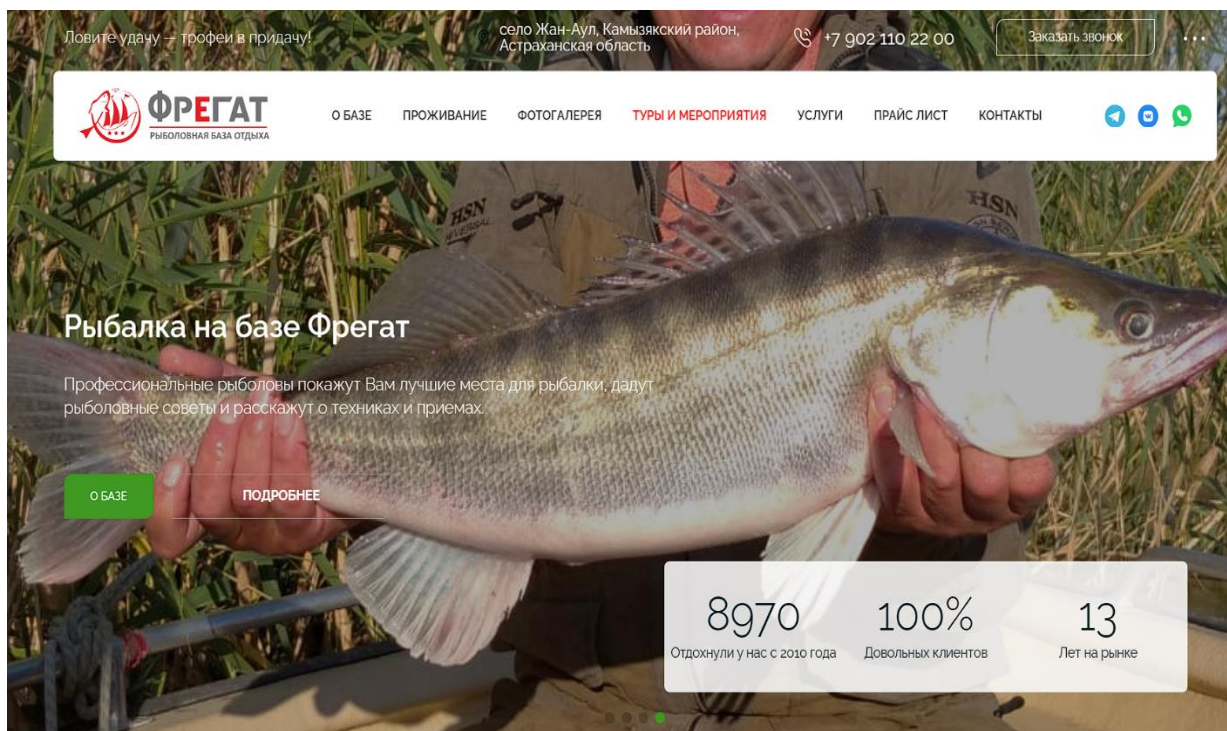


Рисунок 2 – Интерфейс главной страницы системы «Фрегат»

Так, среди преимуществ представленного ресурса стоит выделить наличие возможности сортировки и фильтрации имеющихся услуг и товаров; а также наличие отчётности и аналитики.

При этом, рассматриваемая система имеет и свои недостатки. Так, интерфейс ресурса является менее понятным и неинтуитивным, из-за чего новому пользователю сложно осуществить навигацию по разделам; а также менее выраженную визуальную читаемость.

1.2.4 Анализ системы «Клевое место»

Последняя система для рассмотрения – сайт рыболовной базы «Клевое место», включающей в себя как реализацию клиентского блока: бронирование и просмотр услуг и товаров; так и административный контур – управление имеющимися секторами и ведение аналитики [4].

Интерфейс главной страницы системы представлен на рисунке 3.

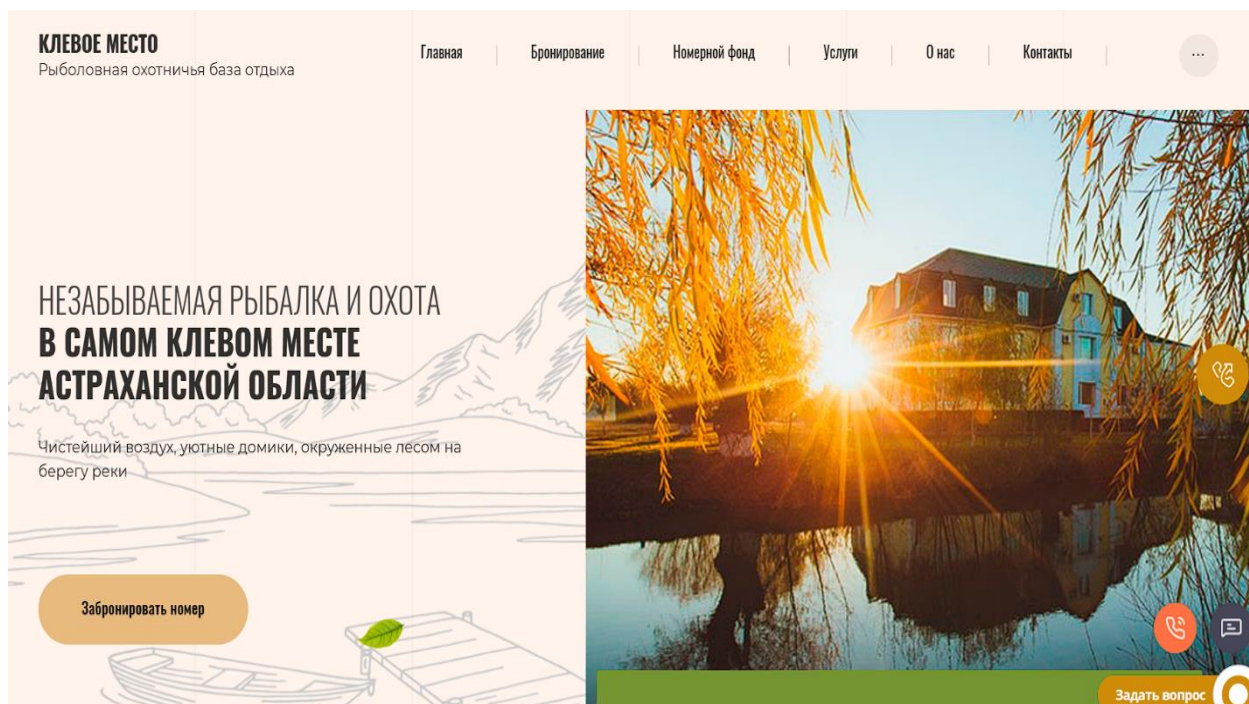


Рисунок 3 – Интерфейс главной страницы системы «Клевое место»

Среди преимуществ данного программного продукта выделяется высокий уровень интуитивности и простоты при реализации бронирования и отсутствие лишних шагов при осуществлении операции; а также высокоуровневая

организация масштабируемости, благодаря чему обработка страниц осуществляется быстро при нарастании числа клиентов и услуг.

Однако, система имеет и свои недостатки. Так, стоит выделить ограниченность в числе доступных тарифов, а также отсутствие дополнительной информации по имеющимся услугам и товарам.

1.2.5 Результат сравнения программ-аналогов

На основании подробного анализа каждой из рассмотренных систем; а также получении преимуществ и недостатков каждого программного продукта была составлен и определен ряд критериев, на основании которых осуществилась оценка существующих решений. Каждый критерий оценивался по следующей шкале:

- 0 баллов: представленный критерий полностью отсутствует;
- 1 балл: представленный критерий выполнен частично или со значительными ограничениями;
- 2 балла: представленный критерий реализован полноценно и качественно.

Таким образом, была получена сводка результатов сравнения программ-аналогов; которая представлена в таблице 1.

Таблица 1 – Сравнение систем-аналогов

Критерий	Fishing-avalon	Фрегат	Клевое место
Простота интерфейса	2	1	1
Удобство навигации	2	1	2
Визуальная читаемость	2	1	1

Продолжение таблицы 1

Критерий	Fishing-avalon	Фрегат	Клевое место
Адаптивность	2	1	1
Единообразие дизайна	2	1	1
Наличие дополнительных услуг	2	1	0
Масштабируемость	1	2	2
Наличие отчётности и аналитики	1	2	0
Прозрачность управления бронированиями	0	2	1
Итого	14	12	9

В итоге, проведя сравнительный анализ программ-аналогов, можно сделать вывод, что ни одно из существующий решений не обеспечивает одновременно высокий уровень простоты интерфейса и его навигации, прозрачности по управлению бронированиями, широкому административному контуру; а также высокому уровню адаптивности и масштабируемости. Поэтому, разрабатываемый программный продукт должен быть довольно гибким и универсальным; к тому же должен предусматривать расширение системы в будущем.

1.3 Анализ функциональных требований

Исходя из анализа предметной области и ключевых задач на разработку клиентского приложения информационной системы по автоматизации и учёту услуг рыболовного хозяйства были выявлены следующие функциональные требования рассматриваемого проекта:

- авторизация и регистрация в системе;
- просмотр списка всех секторов;
- возможность забронировать сектор;
- возможность управления товарами в рамках бронирования;
- просмотр бронирований;
- отмена брони;
- управление бронированиями;
- управление секторами;
- управление тарифами;
- управление товарами;
- анализ отчётности по работе системы.

Для наиболее полного отражения сформированных функциональных требований была осуществлена разработка диаграмм вариантов использования для каждой ключевой категории пользователей. Данная спецификация позволяет представить, какие действия способен выполнять пользователь и как осуществляет взаимодействие с программным обеспечением.

Диаграмма вариантов использования – это диаграмма, которая описывает, какой функционал разрабатываемого программного продукта доступен для каждой категории пользователей. Она служит для конкретного описания функциональности и поведения системы. Она представляет систему в виде актеров, которые взаимодействуют с системой через варианты использования [5].

Актер – внешняя по отношению к системе сущность, которая использует функциональные возможности системы для достижения определенных целей. В

качестве актера может выступать человек, внешняя сущность, класс или другая система.

Вариант использования – это выполняемое системой действие, которое приводит к наблюдаемым результатам со стороны актера. Так, каждый вариант использования описывает конкретный сценарий. В качестве сценария выступает последовательность шагов, выполняемых системой для получения необходимого результата.

Рассмотрим каждую из категорий пользователей по отдельности.

Для администратора основной функционал связан с авторизацией в системе; управлением имеющимся списком секторов; управлением бронированиями (подтверждение брони, отмена брони); управлением списка товаров как для аренды, так и для продажи; управлением тарифами и статистическим анализом имеющейся информации.

Диаграмма вариантов использования, описывающая функционал для администратора, представлена на рисунке 4.

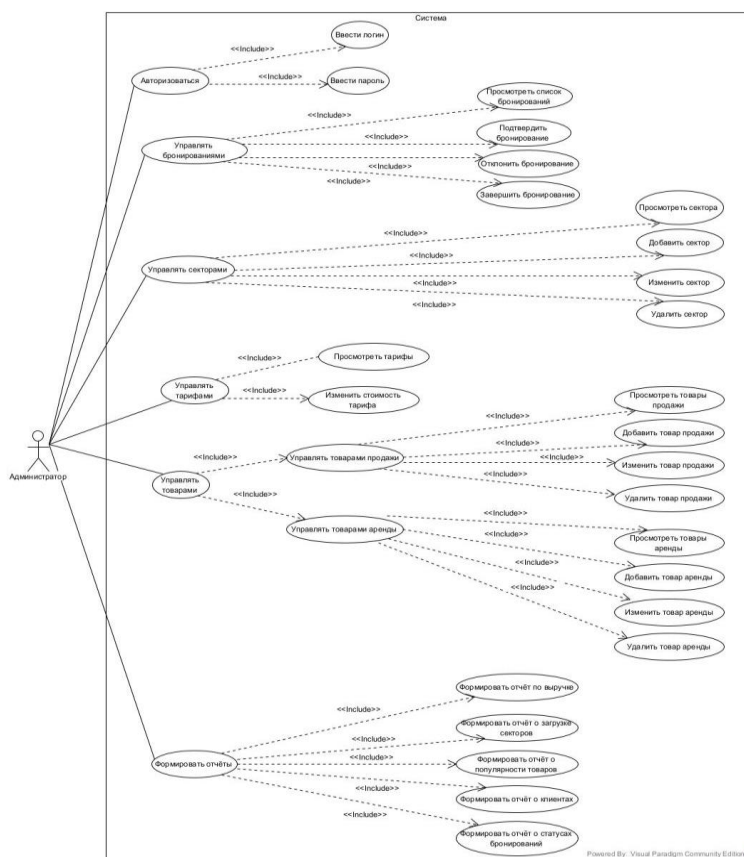


Рисунок 4 – Диаграмма вариантов использования для администратора

Клиент же имеет возможность ознакомиться со списком имеющихся секторов с применением фильтрации, перейти на конкретный сектор и детальнее узнать его информацию, посмотреть имеющиеся тарифы, осуществить бронирование желаемого сектора с автоматическим подсчётом итоговой суммы брони, посмотреть собственные брони; а также отменить собственную бронь.

Диаграмма вариантов использования, описывающая функционал для клиента, представлена на рисунке 5.

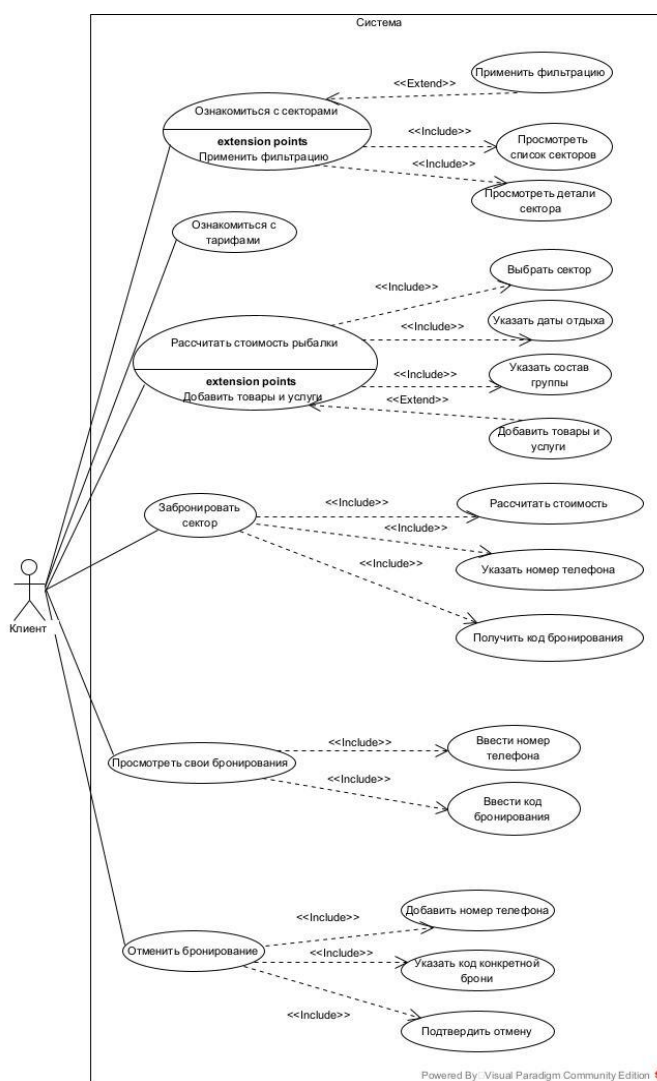


Рисунок 5 – Диаграмма вариантов использования для клиента

Ниже приведено описание спецификаций вариантов использования, связанных с основным функционалом представленной системы.

На рисунке 6 приведена спецификация варианта использования «Просмотреть сектора» для администратора.

Наименование	Просмотреть сектора
ID	10
Краткое описание	Администратор ознакомливается со всем списком имеющихся секторов.
Действующие лица	Администратор.
Предусловие	Администратор прошёл авторизацию в информационной системе.
Основной поток	1. Администратор переходит на главную страницу информационной системы. 2. Администратор переходит на вкладку «Сектора». 3. Администратор видит список всех имеющихся секторов.
Альтернативный поток	1а. Администратор не прошёл авторизацию и не может зайти на главную страницу информационной системы.
Постусловие	Администратор ознакомился со списком секторов.

Рисунок 6 – Спецификация варианта использования «Просмотреть сектора»

На рисунке 7 приведена спецификация варианта использования «Подтвердить бронирование» для администратора.

Наименование	Подтвердить бронирование
ID	7
Краткое описание	Администратор подтверждает конкретную бронь.
Действующие лица	Администратор.
Предусловие	Администратор прошёл авторизацию в информационной системе.
Основной поток	1. Администратор переходит на главную страницу информационной системы. 2. Администратор переходит на вкладку «Бронирования». 3. Администратор видит список всех имеющихся броней. 4. Администратор проверяет условия и подтверждает конкретную бронь.
Альтернативный поток	1а. Администратор не прошёл авторизацию и не может зайти на главную страницу информационной системы. 4а. Бронь не соответствует условиям, и администратор отклоняет её.
Постусловие	Администратор подтвердил указанную бронь.

Рисунок 7 – Спецификация варианта использования «Подтвердить бронирование»

На рисунке 8 приведена спецификация варианта использования «Рассчитать стоимость бронирования» для клиента.

Наименование	Рассчитать стоимость бронирования.
ID	5
Краткое описание	Система позволяет клиенту автоматически посчитать стоимость бронирования на основании данных.
Действующие лица	Клиент, система.
Предусловие	Клиент вошёл в систему.
Основной поток	<ol style="list-style-type: none"> 1. Клиент переходит на страницу секторов. 2. Клиент выбирает конкретный сектор. 3. Клиент указывает необходимые данные для бронирования: номер телефона, даты заезда и выезда, количество людей. 4. Клиент указывает при желании товары и услуги. 5. Система выводит итоговую сумму бронирования.
Альтернативный поток	3а. Клиент указывает неверные данные для брони или пропускает их.
Постусловие	Клиент получил сумму бронирования сектора.

Рисунок 8 – Спецификация варианта использования «Рассчитать стоимость бронирования»

На рисунке 9 приведена спецификация варианта использования «Отменить бронирование» для клиента.

Наименование	Отменить бронирование
ID	17
Краткое описание	Клиент отменяет собственную бронь на указанный сектор.
Действующие лица	Клиент, система.
Предусловие	Клиент ранее осуществил бронирование указанного сектора.
Основной поток	<ol style="list-style-type: none"> 1. Клиент заходит в раздел: «Мои бронирования». 2. Клиент вводит номер телефона и код бронирования для выбора конкретной брони. 3. Клиент указывает причину отмены брони. 4. Система подтверждает отмену бронирования.
Альтернативный поток	2а. Клиент ввёл неверные данные для выбора брони. 3а. Клиент не указал причину для отмены брони.
Постусловие	Клиент отменил собственную бронь.

Рисунок 9 – Спецификация варианта использования «Отменить бронирование»

Таким образом, полученные сформулированные функциональные требования и описанные варианты использования в полной мере отражают необходимую работу информационной системы по автоматизации и учёту услуг рыболовного хозяйства.

1.4 Анализ нефункциональных требований

Нефункциональные требования описывают, как должна работать система; а также определяют качество и характеристики работы системы, не связанные с определёнными функциями [6].

Данные требования являются не менее важными, чем функциональные. Эти требования влияют на качество, надежность и удобство работы с системой. В их число входит не только производительность, но и безопасность, удобство использования, масштабируемость и другие характеристики системы.

Для клиентского приложения информационной системы по автоматизации и учёту услуг рыболовного хозяйства можно выделить следующие нефункциональные требования: безопасность, производительность, удобство использования, адаптивность, привлекательность и расширяемость.

Рассмотрим каждую из характеристик информационной системы и её реализацию.

Безопасность приложения осуществляется благодаря внедрению многофакторной аутентификации пользователей. Для входа в систему используется уникальный логин – электронная почта; а также сложный пароль, включающий в себя минимум 8 символов и использование сложных конструкций, состоящих из цифр, букв и дефиса.

Производительность системы должна обеспечивать быструю реакцию на любые действия со стороны пользователей; включая ознакомление с секторами и реализацию бронирования. Для этого необходимо предусмотреть возможность масштабирования при увеличении нагрузки на ресурс.

Для удобства использования нужно, чтобы разрабатываемое приложение обладало понятным и наглядным интерфейсом, чтобы любой пользователь без

проблем мог выполнить необходимую операцию. Также, навигация по системе должна быть интуитивно понятной, чтобы любые пользователи мгновенно осуществляли поиск нужной информации и функций.

В рамках адаптивности ключевую роль играет возможность применения и переноса разрабатываемого продукта на мобильные платформы без потери контекста. Для этого в разработке применяется CSS-фреймворк Tailwind CSS, позволяющий без проблем адаптировать дизайн под любые задачи.

Для обеспечения высокой привлекательности необходимо рациональное расположение элементов и соответствие принципам эргономики, что способствует интенсивному и комфортному взаимодействию с системой.

Наконец, для продукта важно и наличие расширяемости. Так, архитектура системы должна позволять легко добавлять новые функции и возможности без модификации уже имеющихся и реализованных модулей, и к тому же быть совместимой с разнообразными технологическими решениями.

Таким образом, нефункциональные требования отражают архитектурный стиль и качество реализации системы, обеспечивая ее производительность, безопасность, адаптивность, привлекательность и удобство дальнейшего сопровождения.

1.5 Планирование разработки и оценка бюджета

Планирование разработки и оценка бюджета являются критически важными задачами в рамках разработки клиентского приложения информационной системы по автоматизации и учёту услуг рыболовного хозяйства; поскольку позволяют осуществить анализ сроков разработки программного продукта, а также всевозможных затрат в ходе выполнения ключевых задач.

Так, для наиболее комплексного управления программным продуктом была выбрана программа Microsoft Office Project 2007, позволяющая интуитивно и грамотно сопровождать разрабатываемое ПО на этапах планирования и своевременно вносить необходимые изменения [7].

В первую очередь, для наглядного представления сроков выполнения всех работ и последовательности выполнения этапов была разработана диаграмма Ганта в среде Microsoft Project 2007. Диаграмма Ганта представляет собой график работ по проекту, состоящий из двух ключевых осей: горизонтальной – сроков, и вертикальной – непосредственного списка задач реализуемого программного обеспечения рыболовного хозяйства.

Спроектированная диаграмма Ганта для клиентского приложения информационной системы по автоматизации и учёту услуг рыболовного хозяйства представлена на рисунке 10.

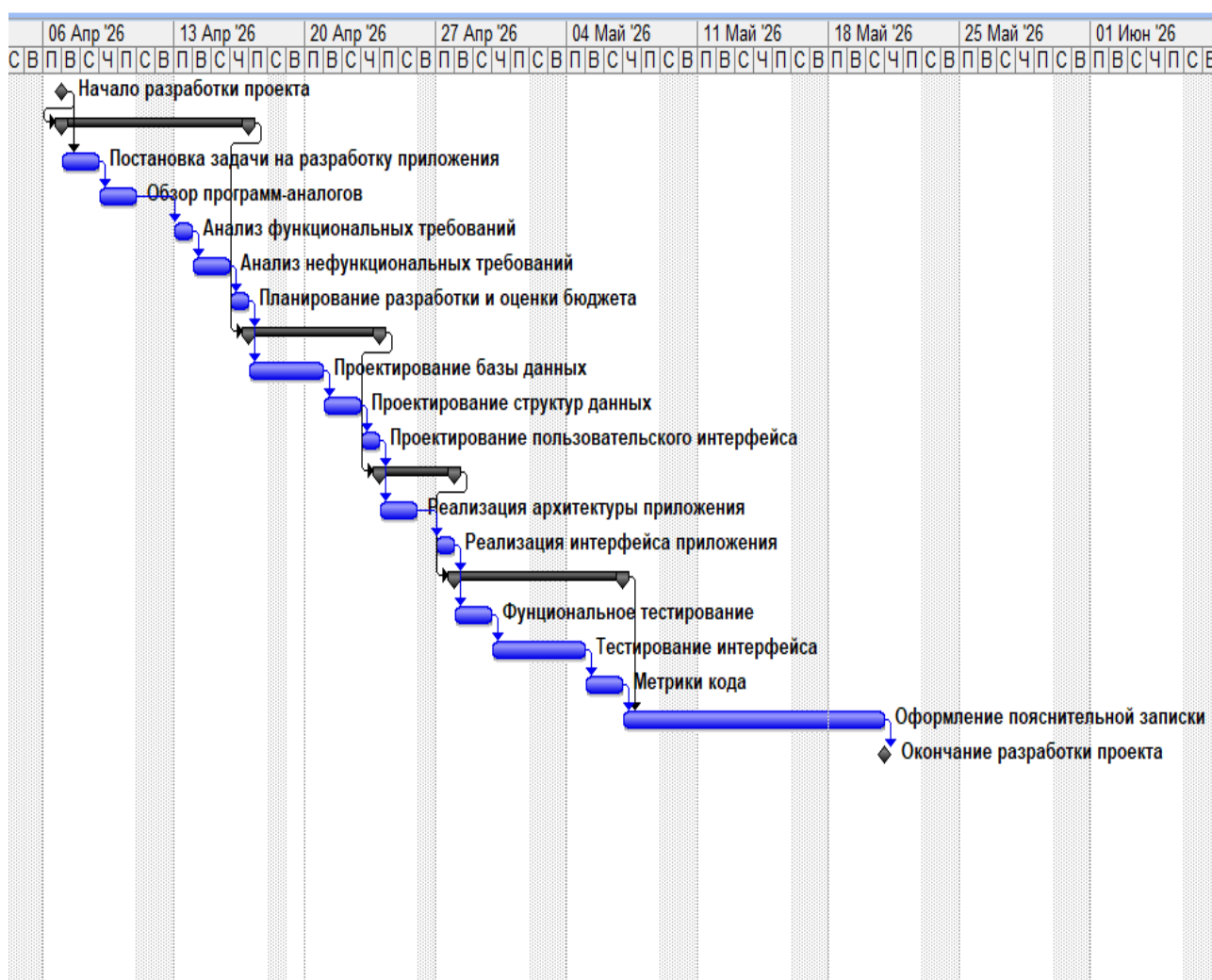


Рисунок 10 – Диаграмма Ганта

Также, отметим весь список реализуемых задач при разработке программного продукта, представленный на рисунке 11.

	Название задачи	Длительность	Начало	Окончание	Предшественники
1	Начало разработки проекта	0 дней	Вт 07.04.26	Вт 07.04.26	
2	Анализ предметной области и постановка задачи	8 дней	Вт 07.04.26	Чт 16.04.26	1
3	Постановка задачи на разработку приложения	2 дней	Вт 07.04.26	Ср 08.04.26	1
4	Обзор программ-аналогов	2 дней	Чт 09.04.26	Пт 10.04.26	3
5	Анализ функциональных требований	1 день	Пн 13.04.26	Пн 13.04.26	4
6	Анализ нефункциональных требований	2 дней	Вт 14.04.26	Ср 15.04.26	5
7	Планирование разработки и оценки бюджета	1 день	Чт 16.04.26	Чт 16.04.26	6
8	Проектирование приложения	6 дней	Пт 17.04.26	Чт 23.04.26	7;2
9	Проектирование базы данных	3 дней	Пт 17.04.26	Пн 20.04.26	7
10	Проектирование структур данных	2 дней	Вт 21.04.26	Ср 22.04.26	9
11	Проектирование пользовательского интерфейса	1 день	Чт 23.04.26	Чт 23.04.26	10
12	Программная реализация приложения	3 дней	Пт 24.04.26	Пн 27.04.26	11;8
13	Реализация архитектуры приложения	2 дней	Пт 24.04.26	Сб 25.04.26	11
14	Реализация интерфейса приложения	1 день	Пн 27.04.26	Пн 27.04.26	13
15	Отладка и тестирование программного обеспечения	7 дней	Вт 28.04.26	Ср 06.05.26	14;12
16	Функциональное тестирование	2 дней	Вт 28.04.26	Ср 29.04.26	14
17	Тестирование интерфейса	3 дней	Чт 30.04.26	Пн 04.05.26	16
18	Метрики кода	2 дней	Вт 05.05.26	Ср 06.05.26	17
19	Оформление пояснительной записки	10 дней	Чт 07.05.26	Ср 20.05.26	15;18
20	Окончание разработки проекта	0 дней	Ср 20.05.26	Ср 20.05.26	19

Рисунок 11 – Список поставленных задач

Как можно заметить – в качестве начала разработки была принята дата: 7 апреля 2026 года. Общая длительность разработки проекта составляет 42 дня.

Также; рассмотрим трудозатраты на разработку программного продукта. В качестве основного ресурса был выбран программист; его ставка составляет 200 рублей за час.

Исходя из того, что ресурс-программист участвует во всех поставленных задачах на проектирование и разработку фронтенд-приложения информационной системы по автоматизации бронирования и учёта услуг рыболовного хозяйства – то общие затраты на разработку проекта составят 89600 рублей.

Выводы о затратах на проект и его общей длительности представлены на рисунке 12.

<u>Даты</u>			
Начало:	Вт 07.04.26	Окончание:	Ср 20.05.26
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	НД	Факт. окончание:	НД
Отклон. начала:	0 дней	Отклон. окончания:	0 дней
<u>Длительность</u>			
Текущий план:	34 дней	Оставшаяся:	34 дней
Базовый план:	0 дней?	Фактическая:	0 дней
Отклонение:	34 дней	Процент завершения:	0%
<u>Трудозатраты</u>			
Текущий план:	448 ч	Оставшиеся:	448 ч
Базовый план:	0 ч	Фактические:	0 ч
Отклонение:	448 ч	Процент завершения:	0%
<u>Затраты</u>			
Текущий план:	89 600,00 Р	Оставшиеся:	89 600,00 Р
Базовый план:	0,00 Р	Фактические:	0,00 Р
Отклонение:	89 600,00 Р		

Рисунок 12 – Отчёт о затратах и длительности проекта

Таким образом, можно сказать, что время работы над проектом составляет 448 часов. С учётом данной продолжительности получаем итоговую сумму трудозатрат – 89.600 рублей.

2 Проектирование фронтенд-приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства

2.1 Проектирование архитектуры системы

В рамках разработки фронтенд-приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства необходимо выбрать наиболее корректное и универсальное архитектурное решение, которое позволит удобно и гибко осуществлять развёртывание создаваемого программного продукта.

Так, для представленного продукта была выбрана модель многоуровневой архитектуры, обладающей рядом преимуществ перед другими вариантами реализации приложения.

Среди основных преимуществ можно выделить обеспечение высокого уровня масштабируемости при значительной нагрузке на систему; повышение надежности в работе системы и возможность дальнейшего сопровождения в эксплуатации программного кода.

Многослойная архитектура представляет собой последовательную структуру из взаимосвязанных друг с другом уровней, каждый из которых отвечает за собственную бизнес-логику и выполнение определенных процессов.

Топология данного архитектурного решения представлена на рисунке 13.

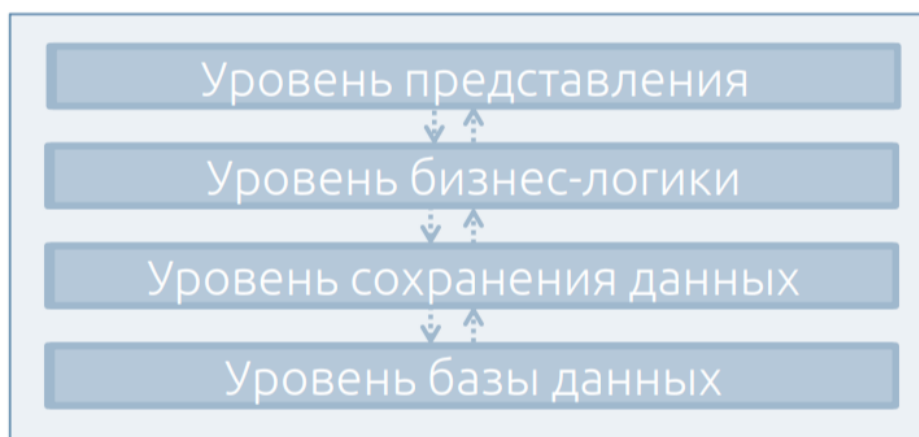


Рисунок 13 – Топология многоуровневой архитектуры

Данная топология включает в себя четыре ключевых уровня работы:

- уровень базы данных;
- уровень сохранения данных;
- уровень бизнес-логики;
- уровень представления.

Рассмотрим каждый из слоёв системы по отдельности.

Уровень базы данных отвечает за хранение ключевых доменных сущностей представленной предметной области, которые используются в большинстве бизнес-процессов программного средства. В данном случае – среди доменных объектов можно выделить сущности имеющих тарифов, товаров для аренды и продажи, броней, рыболовных секторов, клиентов и других моделей данных.

Уровень сохранения данных предназначен для обеспечения взаимодействия программного продукта с реализуемой базой данных. В качестве ключевых операций данного слоя выделяются действия для получения доступа к данным; а также их своевременной обработке, в том числе добавление, изменение или удаление.

Уровень бизнес-логики используется для реализации совокупности бизнес-операций представленной информационной системы и их корректной логики выполнения. Так, среди подобных процессов можно выделить осуществление подсчёта итоговой суммы бронирования сектора, формирование аналитических отчётов для статистики и систематическое управление всеми товарами, в том числе для аренды и продажи.

Уровень представления в свою очередь отвечает за осуществление взаимодействия программного продукта с пользователями путём клиентского приложения. Для реализации представленного слоя используются контроллеры, которые служат основным инструментом в выполнении процессов, в том числе просмотра имеющих секторов, а также подсчёта итоговой стоимости

бронирования, оформления заявки на бронирование и выполнения управлением категориями товаров.

В итоге, можно сказать, что в результате использования представленного архитектурного стиля обеспечивается высокий уровень масштабируемости, поддерживаемости и надежности программного продукта, что положительно сказывается на эксплуатации информационной системы со стороны различных групп пользователей в будущем.

2.2 Проектирование структур данных

2.2.1 Концептуальная модель базы данных

В первую очередь – при осуществлении проектирования информационной системы по автоматизации бронирования и учёту услуг рыболовного хозяйства необходимый акцент нужно сделать на представление модели базы данных, используемой в рамках приложения для хранения и обработки различной информации. Так, в качестве первоначального вида реализуемой базы данных используется формат концептуальной модели.

Концептуальная модель базы данных – это абстрактное описание фрагмента предметной области, которое получается в результате выделения ключевых объектов, указания их атрибутов и возможных связей между данными объектами [8]. Для традиционных баз данных концептуальная схема разрабатывается на начальном этапе проектирования информационной системы; что впоследствии служит основой для её физической реализации и позволяет оставаться ей неизменной в процессе накопления и обработки данных. Это позволяет сделать необходимые шаги в разработке последующих форм моделей баз данных: логической модели и физической модели.

Так, для реализации концептуальной модели был выделен список ключевых сущностей предметной области, используемых в рамках программного продукта. Описание всех сущностей концептуальной схемы базы данных представлено в таблице 2.

Таблица 2 – Описание сущностей концептуальной модели

Определение сущности	Идентификатор сущности	Описание сущности
Клиент	Client	Содержит сведения о клиенте рыболовного хозяйства: фамилию, имя, отчество и номер телефона
Администратор	Admin	Содержит учетные данные администратора системы, необходимые для авторизации и выполнения административных действий
Бронирование	Booking	Содержит информацию о бронировании сектора клиентом, включая даты отдыха, данные клиента, статус бронирования, стоимость услуг и выбранный сектор.
Тариф	Tariff	Содержит информацию о тарифе, его названии, описании и стоимости. Используется при расчете итоговой стоимости бронирования
Сектор	Sector	Предназначена для хранения информации о рыболовных секторах, их характеристиках, вместимости, глубине, наличии беседки, фотографиях и доступности для бронирования.
Сторона водоёма	WaterSide	Содержит сведения о стороне водоёма или части водоёма, на которой расположен рыболовный сектор.
Товар продажи	SaleProduct	Хранит информацию о товарах, доступных для покупки клиентами во время отдыха.
Товар аренды	RentProduct	Содержит данные о товарах и оборудовании, доступных клиентам в аренду на период отдыха.
Уведомление	Notification	Содержит уведомление, связанное с изменением состояния бронирования

В результате выделения списка сущностей была создана графическая модель концептуальной модели базы данных, которая представлена на рисунке 14.

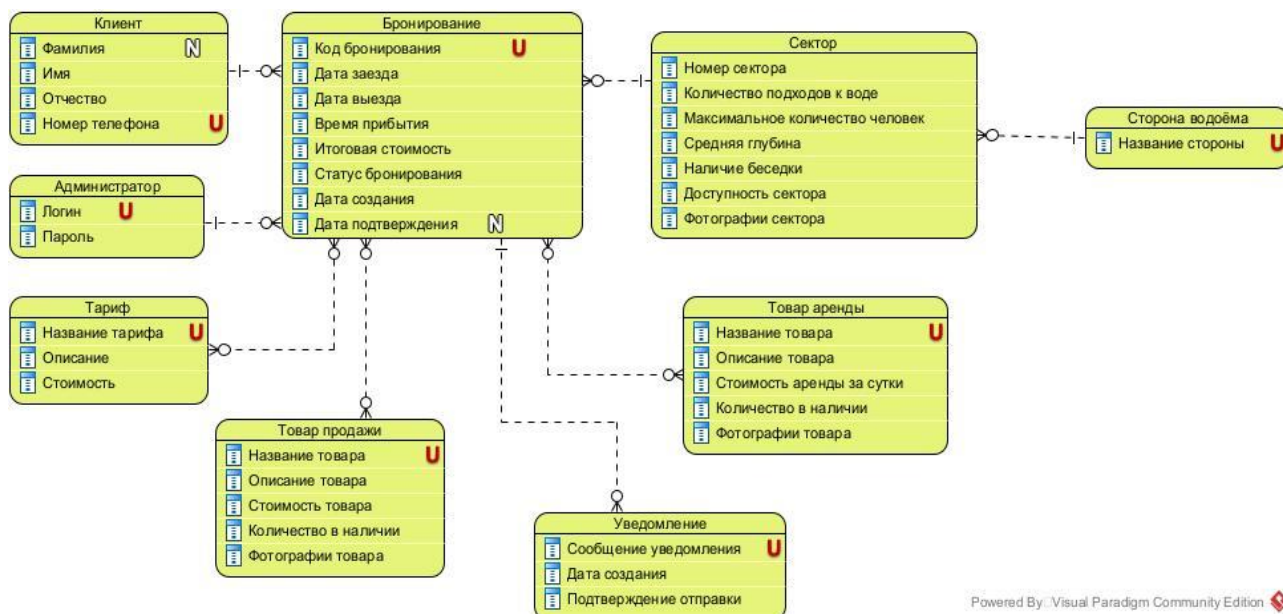


Рисунок 14 – Концептуальная модель базы данных

Таким образом, в результате тщательного анализа предметной области был определен список сущностей и построена концептуальная модель базы данных информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

2.2.2 Логическая модель базы данных

Логическая схема данных представляет собой более детальное описание объектов предметной области, их атрибутов и взаимосвязей между ними в том объеме, в каком они подлежат непосредственному хранению в базе данных системы [9]. Она строится на основе концептуальной модели данных; и при этом учитывает все виды ограничений целостности; благодаря чему проводится нормализация первоначальной структуры с учётом имеющихся первичных и внешних ключей.

Исходя из построенной концептуальной модели предметной области, были выделены следующие сущности:

- клиент;
- администратор;
- сектор;
- товар для продажи;
- товар для аренды;
- бронирование;
- тариф;
- сторона водоёма;
- уведомление.

К тому же, исходя из правил нормализации, были выделены дополнительные сущности для разрешения связи «многие ко многим». Среди таких таблиц можно выделить:

- позиция бронирования;
- позиция товара для продажи;
- позиция товара для аренды.

В результате, была построена логическая модель базы данных, которая приведена на рисунке 15.

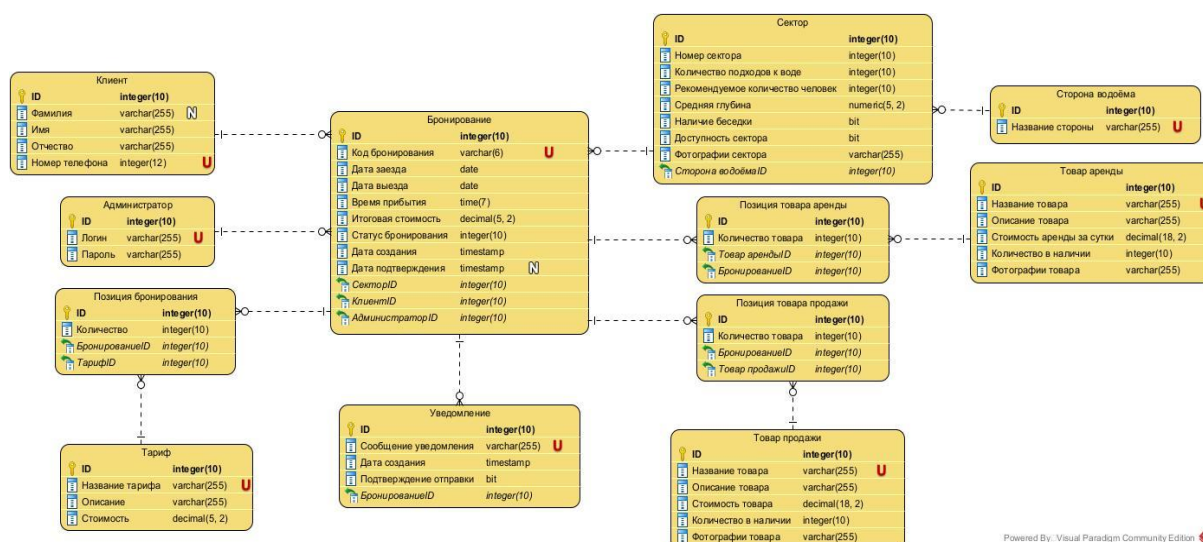


Рисунок 15 – Логическая модель базы данных

Спецификация атрибутов всех сущностей логической схемы базы данных представлена в приложении В.

2.3 Проектирование программных средств

Исходя из выбранного архитектурного стиля для разработки информационной системы, а также смоделированной логической схемы базы данных – в результате было осуществлено проектирование программных средств для текущего приложения. В ходе данного этапа были определены ключевые составные части для серверной части, а также их взаимозависимости.

Рассмотрим более детально состав серверной части проекта.

Серверная часть текущей информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства была реализована с применением универсального фреймворка ASP.NET Core, используемого в большинстве веб-приложений, который позволяет выполнять значительный ассортимент операций, в том числе обработку различных HTTP-запросов от клиентов, подключение маршрутизации и контроль работы с базой данных [10].

Учитывая контекст использования многоуровневой архитектуры в рамках информационной системы – стоит выделить следующую классификацию слоёв архитектурного решения:

- слой контроллеров;
- слой бизнес-логики;
- слой интерфейсов;
- слой доступа к данным;
- слой моделей.

Более детально рассмотрим каждый уровень системы.

Слой контроллеров служит механизмом для обработки всех видов HTTP-запросов, вызова конкретных методов имеющихся сервисов и возвращении ответа в клиентское приложение.

Слой бизнес-логики необходим для определения ключевой логики работы программного проекта.

Слой интерфейсов выполняет задачу назначения контрактов между сервисами и репозиториями.

Слой доступа к данным применяется для осуществления взаимодействия с базой данных.

Слой моделей используется при описывании объектов предметной области.

В рамках разработки было сформулировано следующее описание контроллеров программного проекта, которое приведено в таблице 3.

Таблица 3 – Спецификация контроллеров

Определение контроллера	Назначение контроллера
AdminAuthController	Позволяет обрабатывать запросы, связанные с авторизацией администратора и получением JWT-токена
AdminBookingsController	Предназначен для административного управления бронированиями: просмотра, подтверждения, отклонения и завершения заявок
AdminRentProductsController	Обеспечивает управление товарами аренды со стороны администратора
AdminReportsController	Обрабатывает запросы администратора на формирование отчетов по деятельности рыболовного хозяйства
AdminSaleProductsController	Обеспечивает управление товарами продажи со стороны администратора
AdminSectorsController	Предназначен для управления рыболовными секторами: добавления, изменения и удаления данных
AdminTarifsController	Обеспечивает управление тарифами, используемыми при расчете стоимости отдыха
BookingsController	Обрабатывает клиентские операции, связанные с расчетом стоимости, созданием, просмотром и отменой бронирования
ProductsController	Предоставляет клиенту возможность просматривать товары продажи и товары аренды
SectorsController	Предоставляет клиенту возможность просматривать список секторов, применять фильтрацию и получать подробную информацию о секторе
TariffsController	Позволяет клиенту просматривать действующие тарифы

Далее, каждый из имеющихся контроллеров осуществляет взаимодействие с соответствующим сервисом. Сервисы позволяют выполнять совокупность операций и контроль проверок.

Спецификация для текущих сервисов приведена в таблице 4.

Таблица 4 – Спецификация сервисов

Определение сервиса	Назначение сервиса
AdminAuthService	Реализует логику авторизации администратора и формирования JWT-токена
BookingService	Реализует создание бронирования, просмотр данных брони, отмену бронирования и изменение статуса заявок
PricingService	Выполняет расчет стоимости отдыха на основе выбранного сектора, дат, состава группы, тарифов и дополнительных товаров
RentProductService	Реализует бизнес-логику работы с товарами аренды
TariffService	Обеспечивает получение и изменение данных о тарифах
SaleProductService	Реализует бизнес-логику с товарами продажи
ReportService	Формирует отчеты по выручке, загрузке секторов, популярности товаров, клиентам и статусам бронирований
SectorService	Обрабатывает получение, фильтрацию, создание, изменение и удаление данных о рыболовных секторах

Каждый из реализованных сервисов имеет собственный интерфейс, который позволяет назначить совокупность методов для осуществления взаимодействия с остальными компонентами.

Спецификация для интерфейсов сервисов представлена в таблице 5.

Таблица 5 – Спецификация интерфейсов сервисов

Определение интерфейса сервиса	Назначение интерфейса
IadminAuthService	Определяет методы авторизации администратора
IbookingService	Определяет методы создания, просмотра, отмены и изменения статусов бронирований
IpricingService	Определяет методы расчета стоимости бронирований
IrentProductService	Определяет методы работы с товарами аренды
ItariffService	Определяет методы получения и изменения тарифов

Продолжение таблицы 5

Определение интерфейса сервиса	Назначение интерфейса
IreportService	Определяет методы формирования аналитических отчетов
IsaleProductService	Определяет методы работы с товарами продажи
IsectorService	Определяет методы работы с рыболовными секторами

Предназначение репозиторий заключается в обеспечении доступа к имеющимся данным. Причем, каждый из репозиторий отвечает за собственные методы для реализации операций по чтению, записи или изменению конкретных данных.

Спецификация репозиторий приведена в таблице 6.

Таблица 6 – Спецификация репозиторий

Определение репозитория	Назначение репозитория
AdminRepository	Обеспечивает доступ к учетным данным администратора
BookingRepository	Выполняет операции с бронированиями клиентов
NotificationRepository	Выполняет операции с уведомлениями, связанными с бронированиями
RentProductRepository	Выполняет операции с товарами аренды
SaleProductRepository	Выполняет операции с товарами продажи
SectorRepository	Выполняет операции с данными рыболовных секторов
TariffRepository	Обеспечивает доступ к данным тарифов
WaterSideRepository	Обеспечивает получение данных о сторонах водоёма

Аналогично с сервисами, для каждого репозитория также реализован и собственный интерфейс, который позволяет назначить набор операций для работы с информацией.

Спецификация интерфейсов репозиторий представлена в таблице 7.

Таблица 7 – Спецификация интерфейсов репозиторий

Определение интерфейса	Назначение интерфейса
IadminRepository	Определяет методы доступа к учетным данным администратора
IbookingRepository	Определяет методы работы с бронированиями
InotificationRepository	Определяет методы работы с уведомлениями
IrentProductRepository	Определяет методы работы с товарами аренды

Продолжение таблицы 7

Определение интерфейса	Назначение интерфейса
IsaleProductRepository	Определяет методы работы с товарами продажи
IsectorRepository	Определяет методы работы с данными рыболовных секторов
ItariffRepository	Определяет методы доступа к тарифам
IwaterSideRepository	Определяет методы получения данных о сторонах водоёма

Помимо этого, были выделены соответствующие доменные сущности. С помощью них можно определить название всех таблиц, параметры для полей; а также виды связей между характерными сущностями.

Спецификация доменных сущностей для проекта представлена в таблице 8.

Таблица 8 – Спецификация доменных сущностей

Определение сущности	Назначение сущности
Admin	Содержит данные администратора системы
Client	Содержит сведения о клиенте рыболовного хозяйства
Booking	Содержит информацию о бронировании секторе клиентом
BookingPosition	Содержит тарифную позицию бронирования, включая ссылку на бронирование, тариф и количество
Sector	Содержит характеристики рыболовного сектора
SaleProduct	Содержит данные о товаре, доступном для продажи
BookingRentProduct	Связывает бронирование с товарами аренды и хранит количество выбранного товара
BookingSaleProduct	Связывает бронирование с товарами продажи и хранит количество выбранного товара
Notification	Содержит уведомление, связанное с изменением состояния бронирования
RentProduct	Содержит данные о товаре, доступном для аренды
Tariff	Содержит информацию о тарифе, участвующем в расчете стоимости
WaterSide	Содержит данные о стороне водоёма, к которой относится сектор

К существующим доменным сущностям были составлены характерные конфигурационные классы, определяющие параметры для каждого объекта.

Спецификация всех конфигурационных классов для доменных сущностей приведена в таблице 9.

Таблица 9 – Спецификация конфигурационных классов

Определение класса	Назначение класса
AdminConfiguration	Определяет настройки таблицы администраторов
ClientConfiguration	Определяет настройки таблицы клиентов
BookingConfiguration	Определяет настройки таблицы бронирований
BookingPositionConfiguration	Определяет настройки таблицы тарифных позиций бронирования
RentProductConfiguration	Определяет настройки таблицы товаров аренды
SaleProductConfiguration	Определяет настройки таблицы товаров продажи
BookingRentProductConfiguration	Определяет настройки позиций товаров аренды в бронировании
BookingSaleProductConfiguration	Определяет настройки позиций товаров продажи в бронировании
NotificationConfiguration	Определяет настройки таблицы уведомлений
SectorConfiguration	Определяет настройки таблицы рыболовных секторов
TariffConfiguration	Определяет настройки таблицы тарифов
WaterSideConfiguration	Определяет настройки таблицы сторон водоёма

Для осуществления своевременного взаимодействия между клиентской и серверной частью применяется DTO-модели. При помощи них возможно раграничение возвращаемой информации. Все имеющиеся модели специализируются на собственных объектах: бронирования, сектора, тарифы и товары для аренды и продажи.

В итоге, можно сделать вывод, что в результате определения ключевых программных средств были сформированы компоненты для серверной части программного продукта. Благодаря разделению на многоуровневую структуру, состоящую из контроллеров, сервисов, интерфейсов, репозиториев и доменных сущностей, осуществлено повышение развертываемости, адаптивности и масштабируемости информационной системы в дальнейших целях.

2.4 Проектирование пользовательского интерфейса

Для разрабатываемого приложения информационной системы по автоматизации бронирования и учёту услуг рыболовного хозяйства были сформированы ключевые требования для создаваемого пользовательского интерфейса.

Так, интерфейс представленного программного продукта будет реализован с применением языков разметки HTML и Tailwind CSS; а также использования языка программирования TypeScript вместе с фреймворком React. Такое сочетание инструментария позволит обеспечить высокий уровень адаптируемости и масштабируемости в рамках разработки; а также применить кроссплатформенность при переносе на альтернативные устройства [11].

При реализации интерфейса необходимо придерживаться принципов удобства использования, читаемости и единообразия дизайна; причём как для администраторов системы, так и для базовых клиентов.

Таким образом, были сформирован требуемый список страниц для каждой категории пользователей.

Для администратора системы применяются следующие страницы:

- окно регистрации: позволяет пройти аутентификацию и авторизацию администраторам системы, используя логин и пароль;
- окно управления секторами: позволяет контролировать список имеющихся секторов, добавляя или удаляя любой из секторов;
- окно просмотра сектора: позволяет перейти на выбранный сектор, при необходимости возможно изменение параметров для текущего сектора;
- окно управления бронированиями: позволяет открыть список имеющихся броней, которые администратор либо подтверждает в случае успешного выполнения всех условий, либо же отклоняет текущую бронь;
- окно управления тарифами: позволяет ознакомиться с имеющимися тарифами, которые используются в итоговом подсчете стоимости бронирования; при желании возможно также изменение уже существующего тарифа;

– окно управления товарами на аренду и продажу: позволяет ознакомиться с текущим каталогом списка товаров на аренду или продажу, которые используются в конечном подсчёте суммы бронирования сектора; при необходимости возможно добавление или удаление товара, а также его изменение;

– окно аналитики: позволяет получить текущую статистику системы по группе параметров: выручке за определённый период, популярности секторов или загрузке секторов.

Также, для базового клиента был сформирован следующий список страниц:

– окно главной страницы: позволяет ознакомиться с основной информацией о рыболовном хозяйстве, узнать его правила и структуру;

– окно списка секторов: позволяет перейти к каталогу имеющихся секторов для отдыха на выбор;

– окно осуществления бронирования: позволяет перейти к конкретному сектору и подсчитать сумму бронирования, которая зависит от введённым клиентом параметров: даты заезда и выезда, количества людей, наличия беседки или добавления желаемых товаров;

– окно просмотра броней: позволяет узнать список собственных броней и ознакомиться с их статусом в ходе продвижения; при обновлении статус брони меняется на соответствующую;

– окно тарифов: позволяет посмотреть имеющиеся тарифы для разных категорий людей и узнать их стоимость, а также посуточную стоимость аренды беседки и налог за бронирование сектора;

– окно товаров: позволяет изучить каталог товаров на аренду и продажу, который клиент может добавить в собственную бронь на выбор.

В итоге, каждая страница разрабатываемого интерфейса взаимодействует с серверной частью при помощи протокола REST API. Каждый из запросов отправляется с соответствующим ему HTTP-видом (POST, PUT, DELETE).

Также, стоит учесть, что в ходе ввода некорректных или недопустимых параметров способно отображение соответствующего текстового сообщения о смене данных; к тому же и при успешном выполнении какого-либо бизнес-процесса система уведомит пользователя о корректном исполнении операции.

Таким образом, стоит отметить, что перемещение и ключевые объекты для взаимодействия необходимо выполнить корректно и последовательно во избежание ошибок и неточностей.

2.5 Выбор технологий и инструментальных средств разработки системы

2.5.1 Выбор технологий разработки приложения

В ходе разработки приложения по автоматизации бронирования и учёту услуг рыболовного хозяйства необходимо учесть список используемых технологий разработки, которые позволят гибко и интуитивно осуществлять развертывание программного продукта и поддерживать его на протяжении долгого периода времени.

Так, в качестве ключевой парадигмы для разработки текущего ресурса был выбран принцип объектно-ориентированного программирования (ООП). ООП отвечает за представление совокупности имеющейся предметной области в виде понятный класса и объекта. Класс – это модель объекта, содержащая в себе поля и методы, которые позволяет сформировать структуру объекта и необходимую работу с ним. Объект же является реализацией определённого класса, содержащей в себе совокупность информации [12].

Так, реализация ООП позволяет применить разделение структуры программного кода на обособленные классы, причём каждый из таких классов выполняет собственную задачу. Благодаря такому принципу осуществляется высокий уровень масштабируемости и адаптируемости системы под различные платформы. Кроме этого, одним из отличительных свойств данной парадигмы является свойство инкапсуляции – изоляции данных от использования в чужих

целях. В итоге, все вышеперечисленные преимущества несомненно позволяют сделать выбор на ООП.

В качестве первичных языков разметки были выбраны языки HTML и CSS; а также для CSS был применен фреймворк TailwindCSS. TailwindCSS представляет собой CSS-фреймворк с открытым исходным кодом, который ориентируется на утилитарном подходе – работой с небольшими классами стилей для комбинации их в разметке страницы и качественной стилизации элементов [13]. При помощи такого подхода обеспечивается высокий уровень скорости, единообразия и адаптивности разрабатываемого интерфейса.

Для реализации функционального обеспечения приложения выбор пал на язык программирования TypeScript. TypeScript представляет собой усовершенствованный вариант JavaScript, поддерживающий принципы объектно-ориентированного программирования, статическую типизацию данных; а также наличие автодополнение кода и рефакторинга [14].

Для данного языка программирования был выбран универсальный и популярный фреймворк React. React является декларативной библиотекой на основе JavaScript, который используется для разработки различных пользовательских интерфейсов [15]. React позволяет работать с отдельными и независимыми компонентами. В качестве компонентов здесь выступают различные формы, кнопки, поля или экраны. В совокупности их использование позволяет интегрировать их для различных решений и составлять комбинированные блоки.

В качестве основной СУБД для хранения базы данных акцент был сделан на PostgreSQL. PostgreSQL является представителем объектно-реляционных СУБД с открытым исходным кодом, в которой возможна реализация ключевых операций для хранения, модификации и извлечения данных [16]. В качестве основных преимуществ стоит выделить поддержку различных типов данных, обеспечение высокого уровня надежности и безопасности данных; а также кроссплатформенность на другие платформы.

В качестве протокола для организации взаимодействия между серверной и клиентской частью используется протокол REST API. REST API позволяет формировать ответы на соответствующие HTTP-запросы в виде JSON-файлов. JSON же, в свою очередь, обладает возможностью передачи структурированных данных и обеспечения высокого уровня целостности данных.

Также, в ходе разработки был выбран тип многоуровневой архитектуры. Благодаря такому формату разделения обеспечивается разграниченность бизнес-логики, контроллеров и доступа к данным в отдельных слоях, что уменьшает их влияние друг на друга.

2.5.2 Выбор инструментальных средств разработки приложения

Помимо технологий разработки, важный акцент нужно сделать на выборе инструментальных средств разработки программных средств, которые обеспечат комфортную настройку, ведение и тестирование программного продукта на всех этапах его эксплуатации.

Так, в качестве ключевой инструментальной среды разработки была выбрана среда Visual Studio Code. Visual Studio Code является универсальным редактором кода от Microsoft с открытым исходным кодом, который позволяет разрабатывать широкий ассортимент проектов: веб-приложений, сайтов, программ и других продуктов [17]. Среди преимуществ данной среды стоит выделить наличие встроенного терминала, автоматическое дополнение кода в ходе написания приложения; а также высокий уровень масштабируемости, из-за чего возможно подключение большого количества плагинов в лице языков программирования, библиотек, фреймворков, компиляторов и других средств разработки.

Для осуществления управления работы базы данных был выбран инструмент pgAdmin. Он способен предоставить универсальный графический интерфейс, позволяющий отслеживать выполнение транзакций, изменение данных в отношениях и контроль целостности связей.

В качестве средства разработки и проектирования диаграмм выбор пал на инструмент Visual Paradigm. Visual Paradigm представляет собой инструмент для проектирования и моделирования разного рода диаграмм на языке UML. Так, в ходе выполнения выпускной квалификационной работы были смоделированы и спроектированы в среде Visual Paradigm как модели базы данных: концептуальная, логическая и физическая; так и диаграмма вариантов использования с диаграммой компонентов. К тому же, данная среда поддерживает обширную документацию с подробным описанием спецификации каждой из диаграмм и её решения.

В дополнение, для разработки экономической составляющей программного продукта был выбран инструментальный Microsoft Office Project 2007. Данная среда позволяет детально настроить сроки выполнения каждого из этапов проекта, список необходимых ресурсов, назначение каждого из ресурсов на определённый этап работы и получение итоговой сводной статистики по затратам и нагрузке на работу.

3 Реализация фронтенд-приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства

3.1 Реализация физической модели базы данных

В конечном счёте, после получения формы логической схемы базы данных, необходимо преобразовать её в формат непосредственной реализации внутри СУБД с учётом всех зависимостей и ограничений целостности. Для этого формируется физическая модель базы данных.

Физическая схема базы данных – это вид представления структуры базы данных, который позволяет описать данные фрагмента предметной области средствами определённой СУБД. Такая модель учитывает технические средства выбранной СУБД и позволяет сформировать результат на физическом уровне [18].

В результате сформированных моделей концептуальной и логической схемы базы данных; было реализована разработка физической модели базы данных. Полученная физическая схема базы данных представлена на рисунке 16.

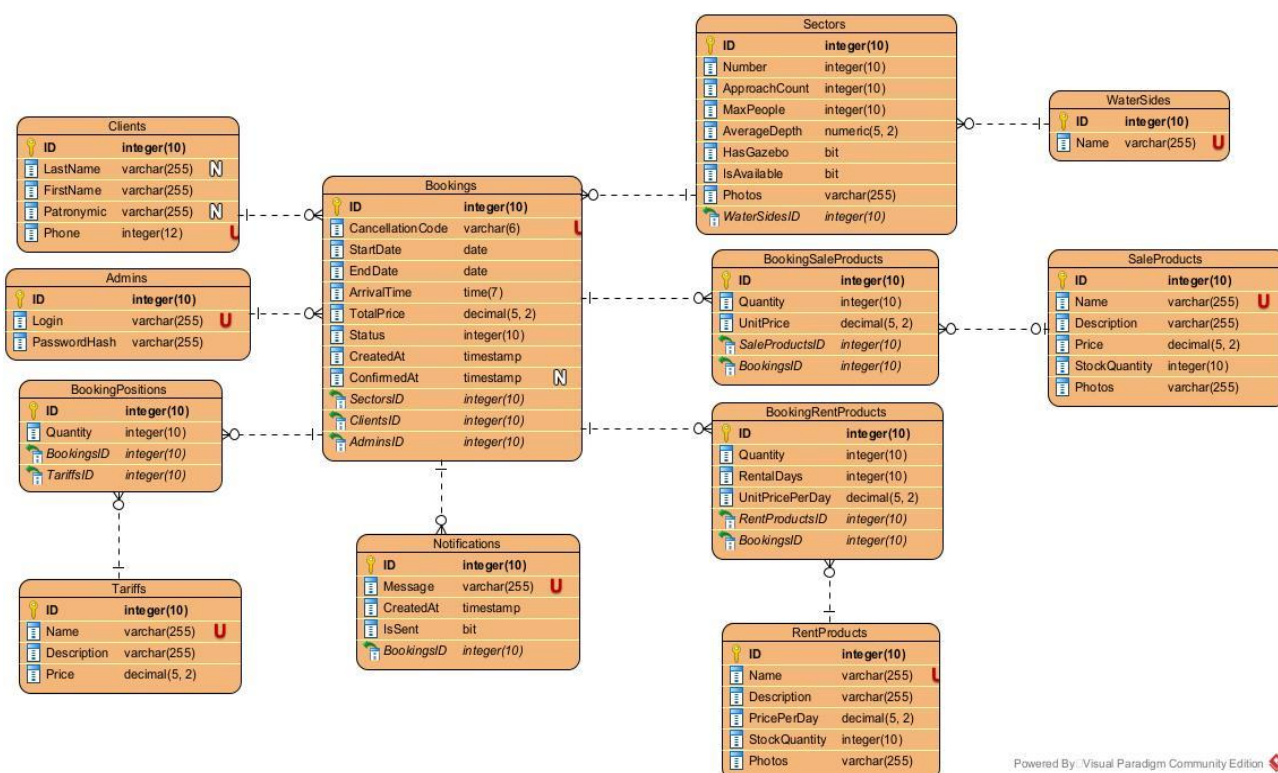


Рисунок 16 – Физическая модель базы данных

В итоге, с использованием среды PostgreSQL осуществлено развертывание физической модели базы данных информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

Спецификация атрибутов сущностей базы данных приведена в приложении В.

Текущий формат обеспечивает полноценные функции хранения, обработки и централизации имеющимися данными, а также позволяет управлять ограничениями целостности для корректного выполнения всех бизнес-операций реализуемого программного продукта.

3.2 Реализация серверной части информационной системы

Рассмотрим более детально реализацию серверной части информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

Так, серверная часть представленного приложения реализована в среде Visual Studio Code на языке программирования C# с учётом использования фреймворка ASP.NET Core.

Представленная система разработана по принципу многоуровневой архитектуры, включающей в себя следующие слои реализации:

- fishingPondBookingSystem.API;
- fishingPondBookingSystem.Application;
- fishingPondBookingSystem.Domain;
- fishingPondBookingSystem.Infrastructure.

Каждый из проектов, в свою очередь, включает в себя собственную совокупность операций и процессов.

Проект fishingPondBookingSystem.API отвечает за вход в информационную систему, формирование маршрутизации и своевременном подключении сервисов.

Проект fishingPondBookingSystem.Application включает в себя главную бизнес-логику проекта; он отвечает за содержание сервисов и соответствующих интерфейсов как сервисов, так и репозиториев. В контексте предметной области

можно выделить расчёт конечной стоимости бронирования, формирование отчётности и управление секторами.

Проект fishingPondBookingSystem.Domain ориентируется на определение доменных сущностей и их формирование. В качестве ключевых сущностей предметной области выделяются: сектор, бронирование, товар для аренды или продажи, тариф и администратор.

Проект fishingPondBookingSystem.Infrastructure отвечает за осуществление доступа к базе данных. Он содержит весь контекст базы данных, сущностей и реализуемые миграции.

Для обеспечения своевременного взаимодействия между клиентской и серверной частью применяется протокол REST API. Согласно структуре, клиентская составляющая отправляет конкретный HTTP-запрос на сервер, тот в свою очередь направляет контроллер для получения запроса и передачи данных в соответствующий сервис. Сервис же выполняет необходимую операцию; в некоторых случаях обращаясь к репозиториям для взаимодействия с базой данных.

Ключевые основные модули серверной части представлены в таблице 10.

Таблица 10 – Основные модули серверной части

Название модуля	Назначение модуля
Модуль авторизации администратора	Обеспечивает вход администратора в систему и выдачу JWT-токена
Модуль клиентов	Обеспечивает хранение и получение данных клиентов, связанных с бронированиями
Модуль бронирований	Отвечает за расчет стоимости, создание, просмотр, отмену и изменение статуса бронирования
Модуль секторов	Обеспечивает получение, фильтрацию, добавление, изменение и удаление рыболовных секторов
Модуль товаров продажи	Реализует просмотр и административное управление товарами, доступными для покупки
Модуль товаров аренды	Реализует просмотр и административное управление товарами, доступными для аренды
Модуль тарифов	Обеспечивает просмотр и изменение тарифов, используемых при расчете стоимости отдыха
Модуль отчетов	Формирует аналитические отчеты по деятельности рыболовного хозяйства

Для обработки всех запросов были разработаны контроллеры. Причём, контроллеры отличаются по своему функциональному контуру: клиентскому или административному.

Клиентские контроллеры предназначены для реализации просмотра имеющихся секторов, осуществления бронирования и просмотра статуса брони.

Методы API клиентского контура приведены в таблице 11.

Таблица 11 – Методы API клиентского контура

URL запроса	HTTP-метод	Описание метода
/sectors	GET	Получение списка рыболовных секторов с возможностью фильтрации
/sectors/{id}	GET	Получение подробной информации о выбранном секторе
/tariffs	GET	Получение списка действующих тарифов
/products/sale	GET	Получение списка товаров продажи
/products/rent	GET	Получение списка товаров аренды
/bookings	GET	Получение информации о бронированиях
/bookings/{id}	GET	Получение подробной информации о конкретном бронировании
/bookings	POST	Создание бронирования
/bookings/{id}	DELETE	Отмена бронирования клиентом
/bookings/calculate	POST	Расчет стоимости отдыха на основе выбранного сектора, дат, состава группы и дополнительных товаров

Административные контроллеры отвечают за управление секторами, осуществление контроля за бронированиями; ведение учёта за тарифами и товарами, а также ведение статистики.

Методы API административного контура приведены в таблице 12.

Таблица 12 – Методы API административного контура

URL запроса	HTTP-метод	Описание метода
/admin/login	POST	Авторизация администратора и получение JWT-токена
/admin/bookings	GET	Получение списка бронирований для администратора
/admin/bookings/{id}/status	PUT	Изменение статуса бронирования администратором
/admin/products/rent	GET	Получение списка товаров аренды для администратора
/admin/products/rent	POST	Добавление нового товара аренды
/admin/products/rent/{id}	PUT	Изменение данных товаров аренды
/admin/products/rent/{id}	DELETE	Удаление товара аренды

Продолжение таблицы 12

URL запроса	HTTP-метод	Описание метода
/admin/products/sale	GET	Получение списка товаров продажи для администратора
/admin/products/sale	POST	Добавление нового товара продажи
/admin/products/sale/{id}	PUT	Изменение данных товара продажи
/admin/products/sale/{id}	DELETE	Удаление товара продажи
/admin/sectors	GET	Получение списка рыболовных секторов для администратора
/admin/sectors	POST	Добавление нового рыболовного сектора
/admin/sectors/{id}	PUT	Изменение данных рыболовного сектора
/admin/sectors/{id}	DELETE	Удаление рыболовного сектора
/admin/tariffs	GET	Получение списка тарифов для администратора
/admin/tariffs/{id}	PUT	Изменение данных тарифа

В конечном счете – такой подход к систематизации позволяет значительно снизить зависимость контроллеров друг от друга и повысить масштабируемость в ходе модификации и совершенствования программного кода.

3.3 Реализация клиентской части информационной системы

В ходе разработки приложения информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства особое внимание было уделено разработке клиентской части, отвечающей за отображение качественного и интуитивного интерфейса.

Для наиболее наглядного представления взаимодействия между разработанными компонентами была построена диаграмма компонентов, которая представлена на рисунке Г.1 в приложении Г.

Спецификация и описание всех используемых компонентов в рамках клиентского приложения приведено в таблице 13.

Таблица 13 – Спецификация компонентов клиентской части

Название компонента	Назначение компонента
AdminPanel.tsx	Главная страница администратора
BookingForm.tsx	Страница оформления бронирования
BookingList.tsx	Страница просмотра текущих броней администратором

Продолжение таблицы 13

Название компонента	Назначение компонента
ClientHome.tsx	Главная страница клиента
ClientNotifications.tsx	Страница просмотра уведомлений
ClientProductPage.tsx	Страница просмотра товаров
ClientTariffPage.tsx	Страница просмотра тарифов
LoginForm.tsx	Страница входа для администратора
ProductTable.tsx	Страница управления товарами
SectorTable.tsx	Страница управления секторами
StatsPage.tsx	Страница управления аналитикой
TariffTable.tsx	Страница управления тарифами
UserBooking.tsx	Страница просмотра текущих броней клиентом
App.tsx	Структура главной страницы
App.css	Стилизация главной страницы

Для реализации клиентской части применялись языки разметки HTML и CSS вместе с фреймворком Tailwind CSS; а также язык программирования TypeScript в совокупности с фреймворком React.

Для взаимодействия с серверной частью применялись HTTP-запросы вместе с протоколом Rest API. Фрагмент листинга клиентской части информационной системы приведен в приложении Б. Полный же код разработанного программного продукта приведен в репозитории на GitHub [19].

3.4 Реализация интерфейса информационной системы

Согласно сформированным требованиям к разработке интерфейса информационной системы для автоматизации бронирования и учёта услуг

рыболовного хозяйства был получен следующий список необходимых страниц, которые будут спроектированы и разработаны:

- главная страница сайта;
- страница ознакомления со списком секторов;
- страница оформления брони;
- страница ознакомления с тарифами;
- страница ознакомления с товарами;
- страница ознакомления с собственными бронированиями;
- страница входа администратора в панель;
- страница управления бронированиями;
- страница управления секторами;
- страница управления товарами;
- страница управления тарифами;
- страница ведения аналитической статистикой.

Рассмотрим каждую из страниц в контексте её содержания и взаимодействия с другими страницами.

Первоначально пользователь попадает на главную страницу информационной системы. Интерфейс главной страницы представлен на рисунке 17.

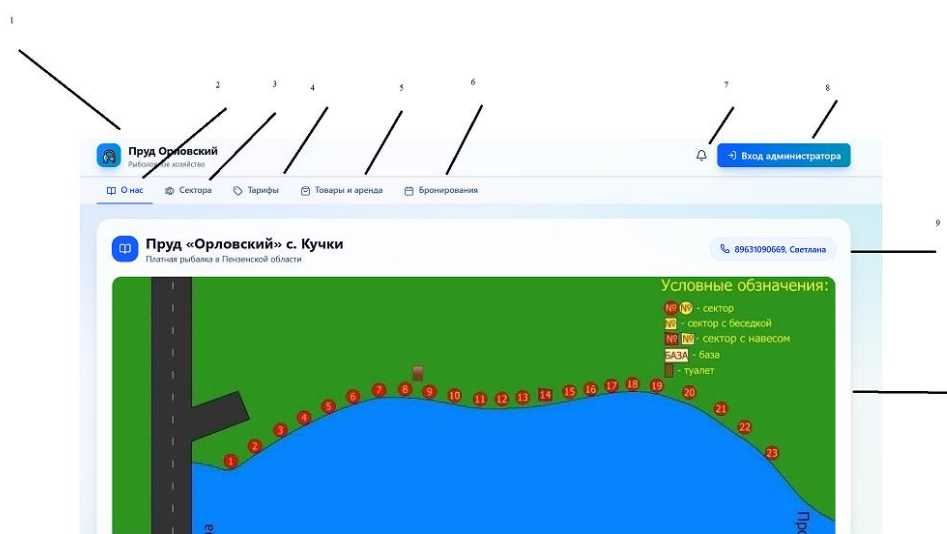


Рисунок 17 – Главная страница системы

Здесь, пользователь видит заголовок «Пруд Орловский» (1); вкладки с переходами на другие страницы, включая «О нас» (2), «Сектора» (3), «Тарифы» (4), «Товары» (5) и «Бронирования» (6). Помимо этого, пользователь может посмотреть уведомления (7) или увидеть кнопку – Вход администратора (8). Главным компонентом на странице является структура пруда (9), показывающая все элементы на карте. Также, можно позвонить владельцу пруда и узнать все подробности (10).

Далее, пользователь может перейти на страницу просмотра секторов. Интерфейс страницы просмотра секторов представлен на рисунке 18.

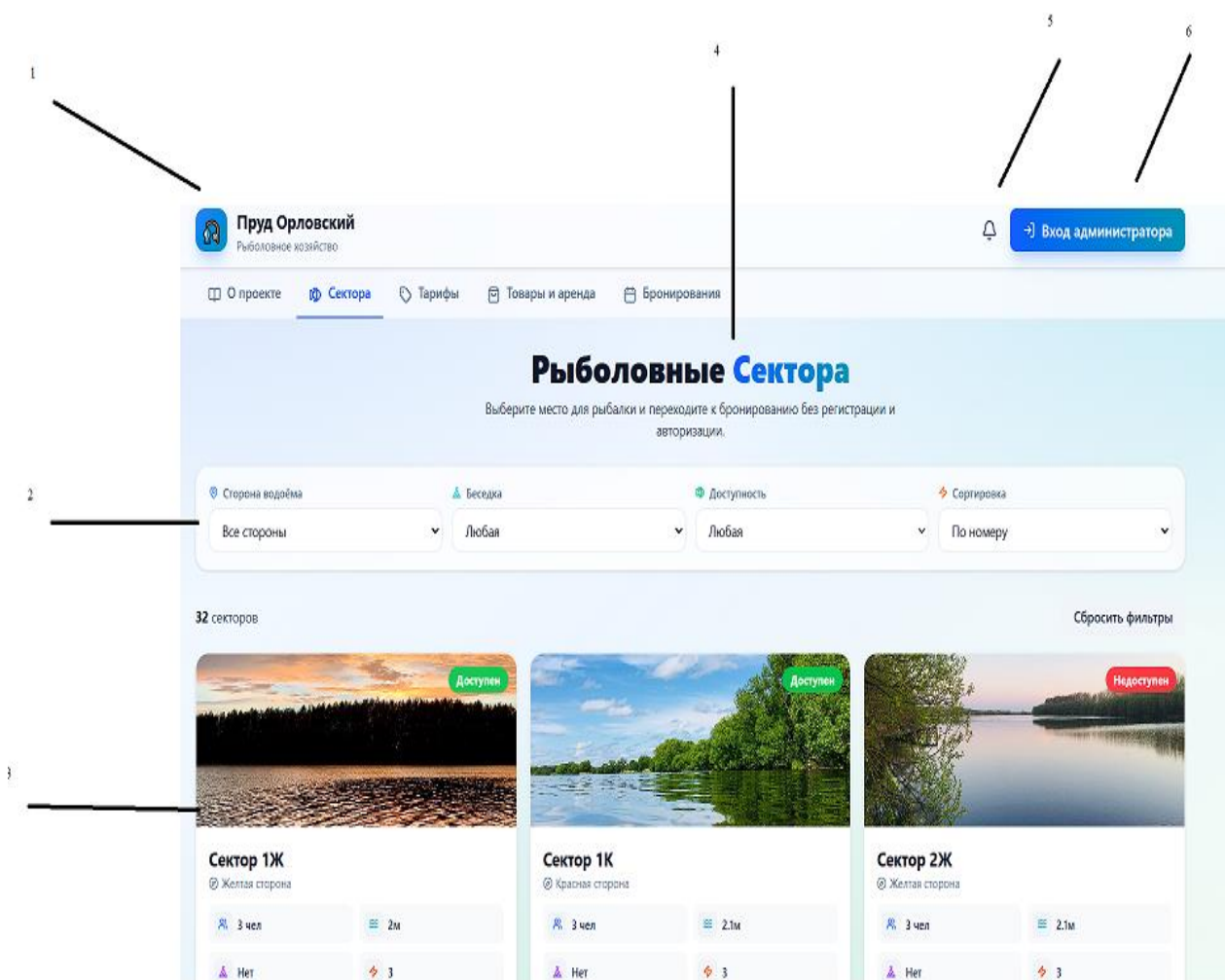


Рисунок 18 – Страница просмотра секторов

Здесь пользователь также видит название пруда (1). При желании пользователь может осуществить сортировку и фильтрацию секторов (2), чтобы

поменять список карточек секторов (3). Имеется центральный заголовок перед карточками секторов (4). Помимо этого, можно посмотреть уведомления (5) и увидеть кнопку для входа администратора (6).

Отсюда клиент выбирает сектор и осуществляет бронь. Интерфейс реализации брони представлен на рисунке 19.

Рисунок 19 – Страница оформления брони

Здесь присутствует фотография выбранного сектора (1) и основная информация о секторе (2). Пользователь вводит ФИО (3), контактные данные (4), время прибытия (5), даты заезда и выезда (6), количество людей для каждого тарифа (7) и добавляет желаемые товары для покупки и аренды (8); после чего нажимает на кнопку – Подтвердить бронь (9).

Также, клиент может ознакомиться со списком тарифов в информационной системе. Интерфейс страницы тарифов представлен на рисунке 20.

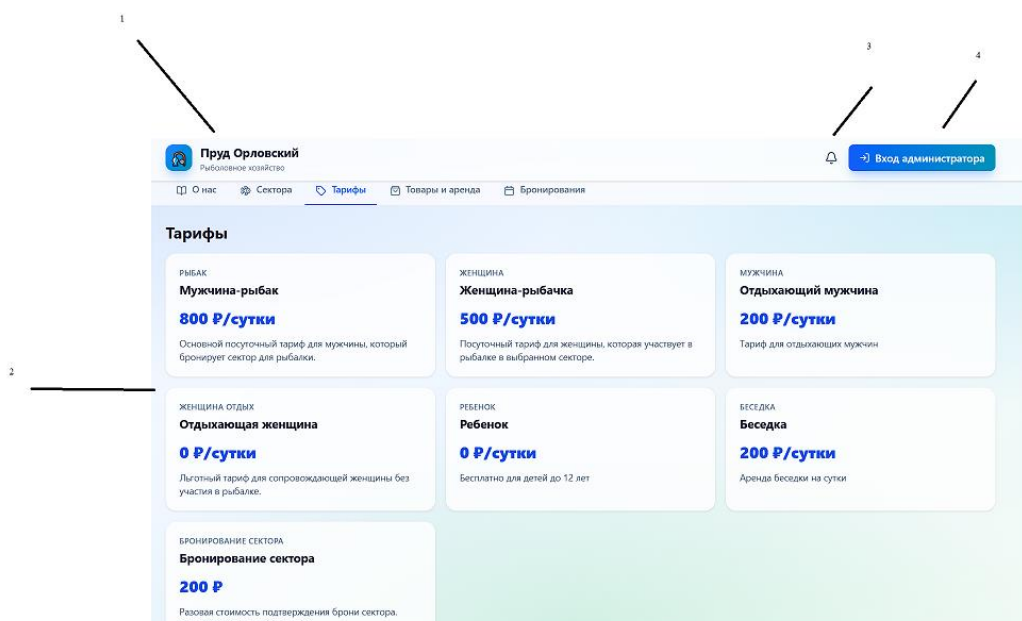


Рисунок 20 – Страница просмотра тарифов

Тут также имеется название пруда (1); а также карточки со всеми тарифами (2). Кроме этого – можно увидеть кнопку входа в административный контур (3).

Также клиент способен посмотреть и каталог товаров. Интерфейс страницы товаров приведен на рисунке 21.

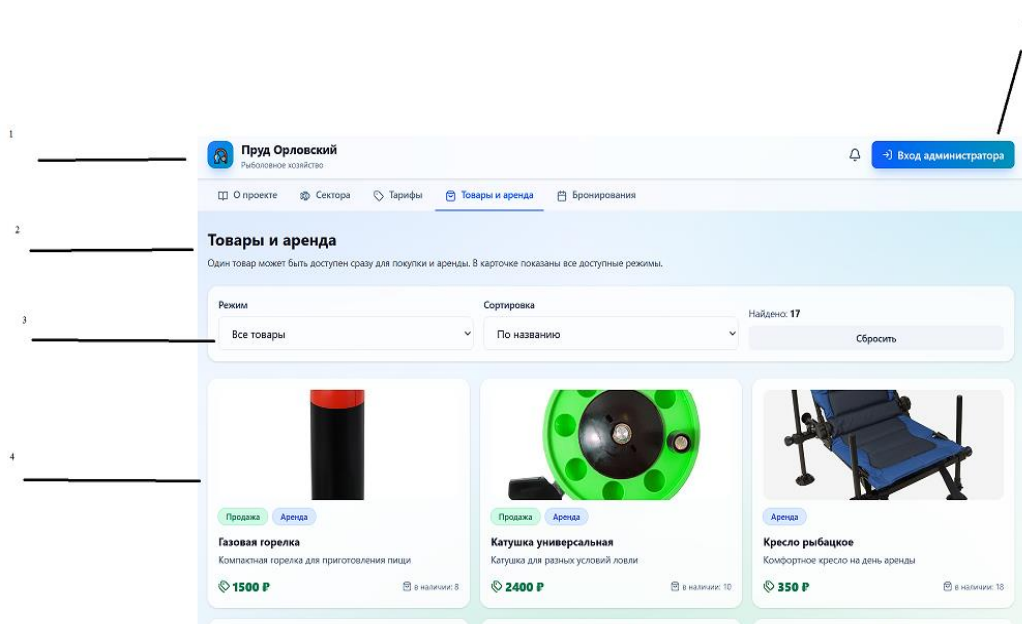


Рисунок 21 – Страница просмотра товаров

Здесь же имеется заголовок системы (1) и заголовок текущей страницы (2). Есть возможность сортировки и фильтрации товаров по типу покупки или аренды, цене, оставшемуся количеству на складе или же возможностью сбросить все существующие фильтры (3), что влияет на список отображаемых товаров (4). В дополнение к этому имеется и кнопка перехода в административный контур (5).

В конечно счёте, клиент может ознакомиться со списком собственных броней и узнать их статус. Интерфейс страницы просмотра бронирований представлен на рисунке 22.

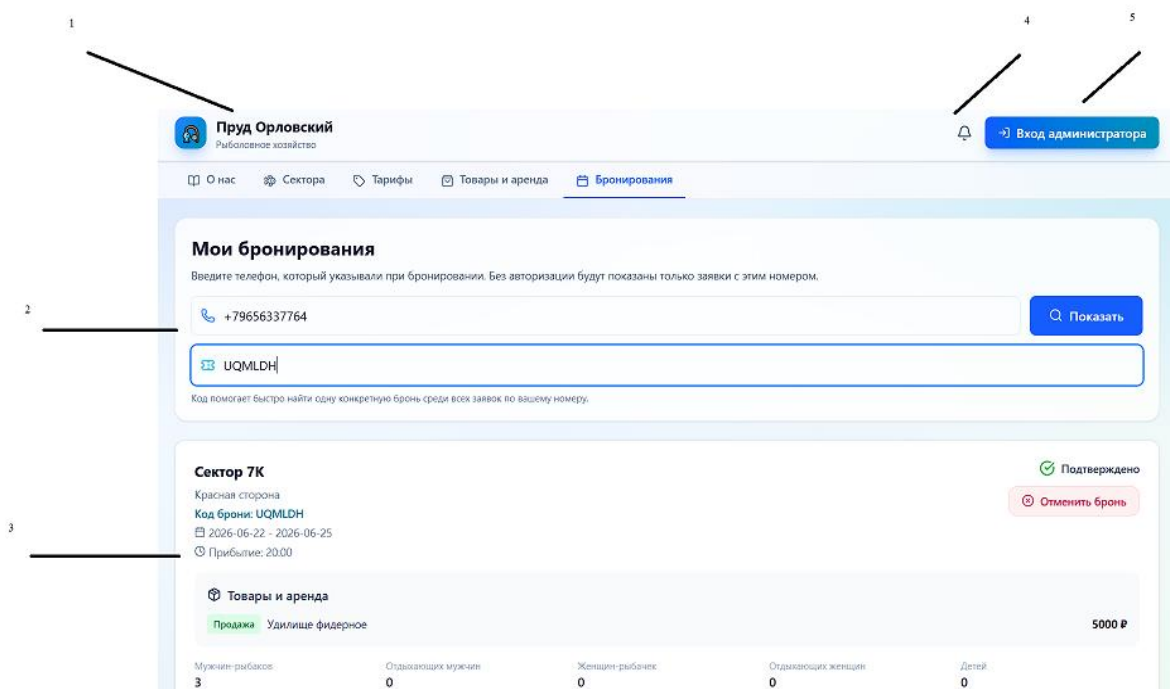


Рисунок 22 – Страница просмотра броней

Как и на других страницах – здесь присутствует заголовок системы (1) и заголовок текущей страницы, где можно ввести номер для сопоставления с бронированиями (2). В результате – отображается список броней для клиента (3). К тому же – имеется кнопка перехода на регистрацию администратора (4) и непосредственная кнопка показа списка броней (5).

О статусе всех текущих броней клиент может узнать по окну уведомлений. Интерфейс страницы уведомлений приведен на рисунке 23.

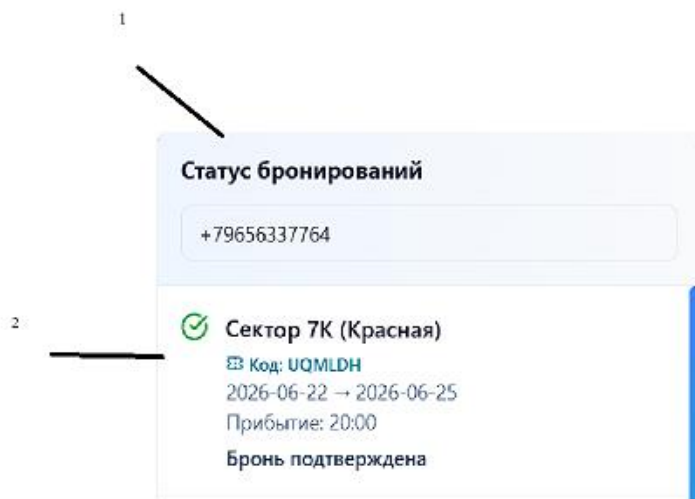


Рисунок 23 – Страница получения уведомлений

Данная страница довольно минималистична. Тут присутствует заголовок страницы с номером (1) и список соответствующих броней (2).

С главной страницы можно войти в административный контур. Интерфейс для входа администратора представлен на рисунке 24.

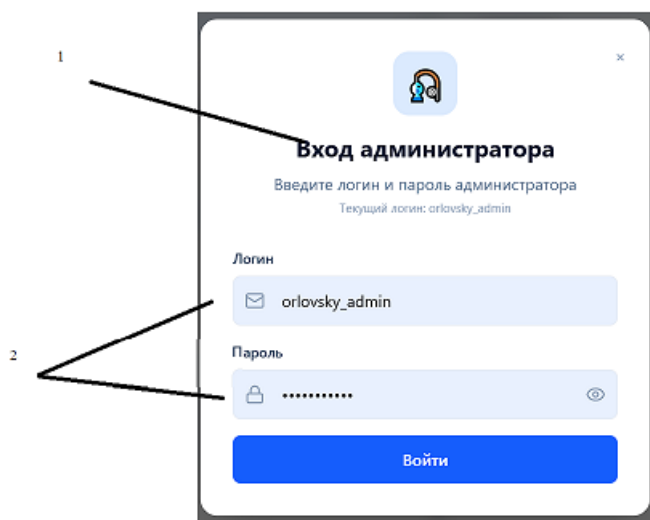


Рисунок 24 – Страница входа администратора в аккаунт

Здесь также довольно просто. Имеется заголовок окна (1) и поля для ввода логина и пароля для успешного входа (2).

Затем, администратор попадает на список всех броней в системе. Интерфейс страницы управления бронированиями представлен на рисунке 25.

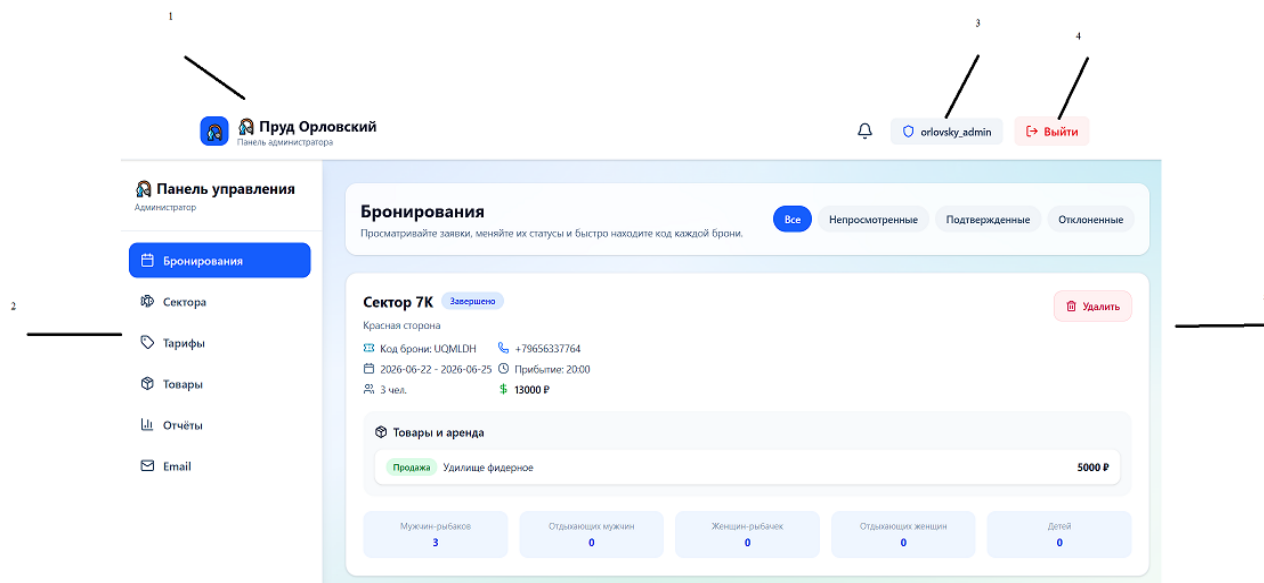


Рисунок 25 – Страница управления бронированиями

Тут имеется вкладка со всеми основными разделами (1) и заголовок страницы (2). Также, присутствует текущее имя администратора (3) и возможность выхода из системы (4). Основную часть страницы занимает каталог броней (5).

При навигации по панели можно перейти к управлению секторами. Интерфейс страницы управления секторами приведен на рисунке 26.

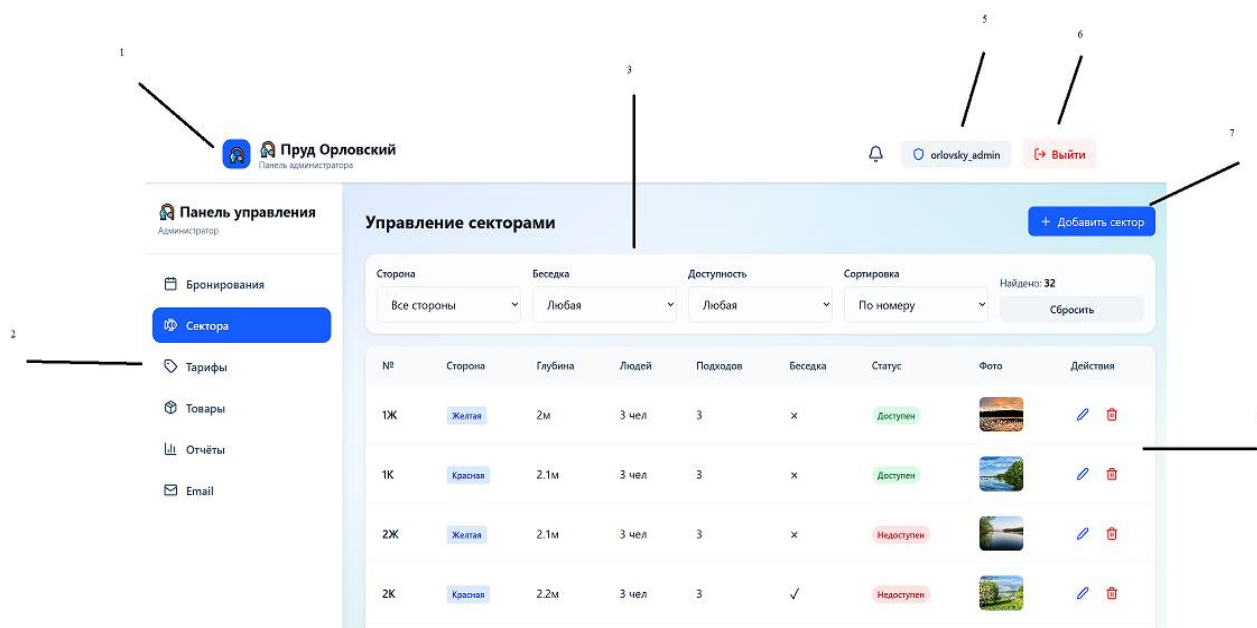


Рисунок 26 – Страница управления секторами

На данной странице присутствует вкладка с панелью основных страниц (1) и главный заголовок страницы (2). Имеется возможность реализовать сортировку и фильтрацию данных по доступности, наличию беседки, максимальной глубине или количеству подходов (3), влияющую на список отображаемых секторов (4). К тому же, имеется возможность посмотреть имя (5), выйти из аккаунта (6) или же добавить новый сектор в информационную систему (7).

Администратор способен перейти на вкладку управления тарифами. Интерфейс страницы управления тарифами представлен на рисунке 27.

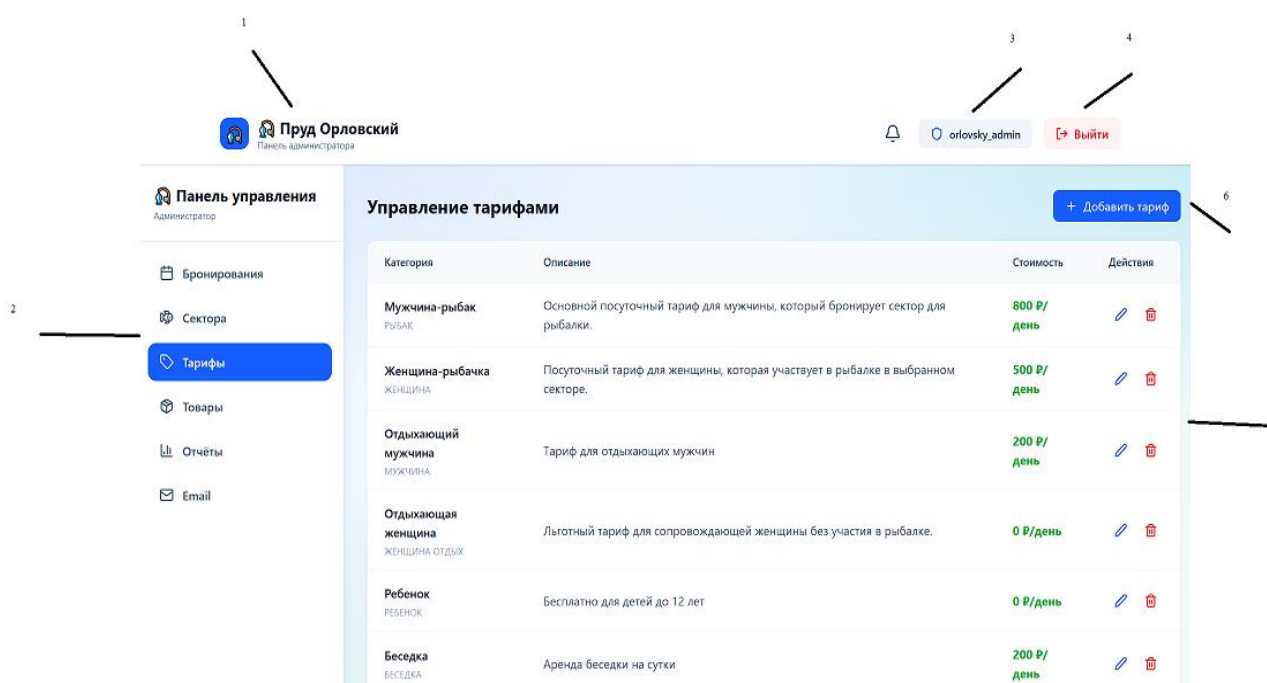


Рисунок 27 – Страница управления тарифами

Центральным звеном является вкладка с переходами между страницами (1). Имеется возможность посмотреть свой логин (2) или выйти из аккаунта (3). Во главе интерфейса расположен список тарифов, которыми можно воспользоваться в ходе бронирования (4). При желании администратор может изменить имеющийся тариф (5).

Переходя между вкладками, можно направиться на страницу управления товарами. Интерфейс страницы управления товарами приведен на рисунке 28.

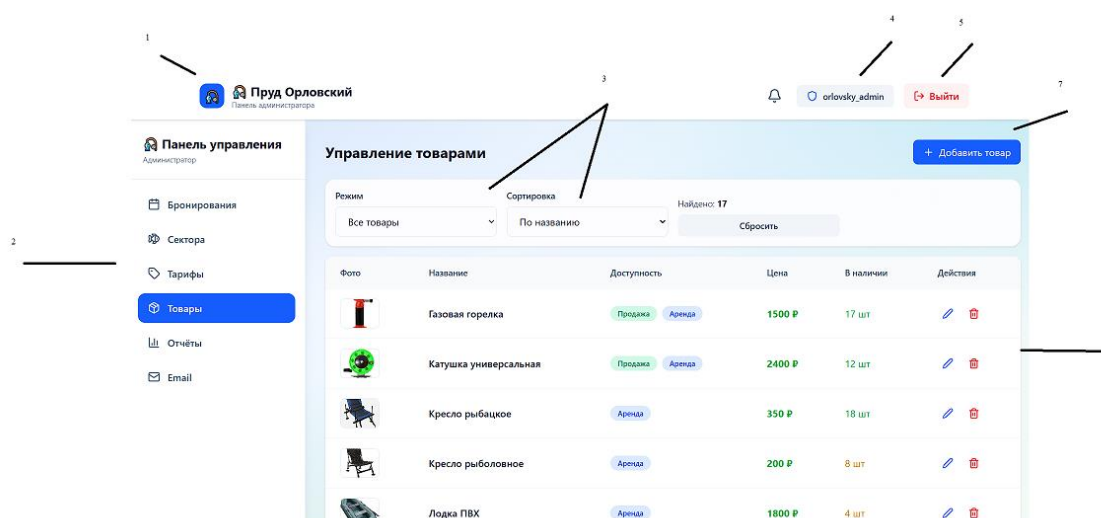


Рисунок 28 – Страница управления товарами

На текущей странице представлена вкладка с основными административными страницами (1) и заголовок текущей страницы (2); есть возможность отфильтровать и сортировать товары по типу продажи или аренды, цене использования или остаточному количеству на складе (3), что влияет на отображаемый список товаров (4). К тому же – можно посмотреть текущий логин (5), выйти из своего аккаунта (6) или же реализовать добавление нового товара (7).

В заключение, в качестве статистики можно перейти в раздел аналитики. Интерфейс страницы аналитики представлен на рисунке 29.

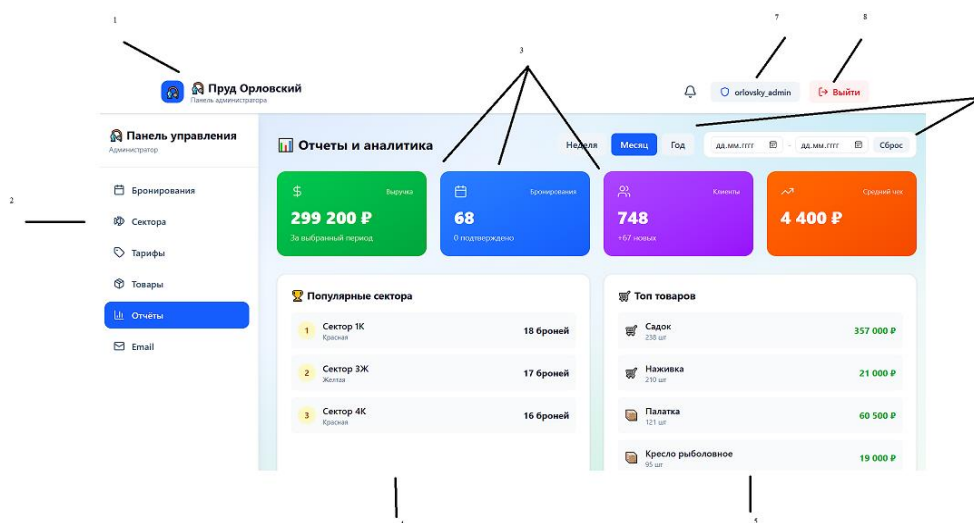


Рисунок 29 – Страница управления аналитикой

Здесь присутствует панель с переходом между вкладками (1) и центральный заголовок страницы (2). Главные компоненты – это показатели по выручке, среднему чеку, количеству броней и гостей (3). Также, есть возможность посмотреть статистику секторов по популярности (4) или рейтинг товаров (5). Все результаты можно конкретизировать за определенный промежуток времени (6). Помимо этого, можно посмотреть текущий логин (7) или выйти из данного аккаунта (8).

В итоге, подводя вывод, можно сказать, что полученный список страниц соответствует поставленным задачам на разработку приложения по автоматизации бронирования и учёту услуг рыоловного хозяйства и позволяет выполнить сформулированные функциональные и нефункциональные требования к программному продукту.

4 Тестирование фронтенд-приложения информационной системы для автоматизации бронирования и учёту услуг рыболовного хозяйства

4.1 Функциональное тестирование приложения

Функциональное тестирование проводилось методом чёрного ящика с применением техники функциональных диаграмм (cause-effect graph) и таблиц решений [20]. Объектом тестирования является сценарий создания бронирования сектора. Данный метод позволяет установить однозначное соответствие между ними и ожидаемыми результатами.

В рамках анализа сценария были выделены 5 причин и 6 возможных результатов (следствий). Они отображены на таблицах 14 и 15.

Таблица 14 – Состав причин

ID	Формулировка
П1	Корректный период (дата заезда < дата выезда)
П2	Сектор свободен на выбранный период
П3	Имя заполнено
П4	Контактные данные заполнены
П5	Количество людей не превышает допустимое

Таблица 15 – Состав следствий

ID	Формулировка
C1	Сообщение: «Некорректный период бронирования»
C2	Сообщение: «Сектор занят на выбранный период»
C3	Сообщение: «Не заполнено имя клиента»
C4	Сообщение: «Не заполнены контактные данные клиента»
C5	Сообщение: «Превышено допустимое количество людей»
C6	Бронирование создано успешно (статус «Ожидает подтверждения»)

Как можно увидеть, в представленном списке несоблюдение каждой причины ведёт в её соответствующее следствие. Причины идут за другом; и при несоблюдении всех причин все они автоматически отправляются в успешное следствие – С6; информирующее о том, что бронирование создано успешно и ожидает подтверждения.

Функциональная диаграмма изображена на рисунке 30. Она включает в себя причины и следствия, которые были озвучены выше.

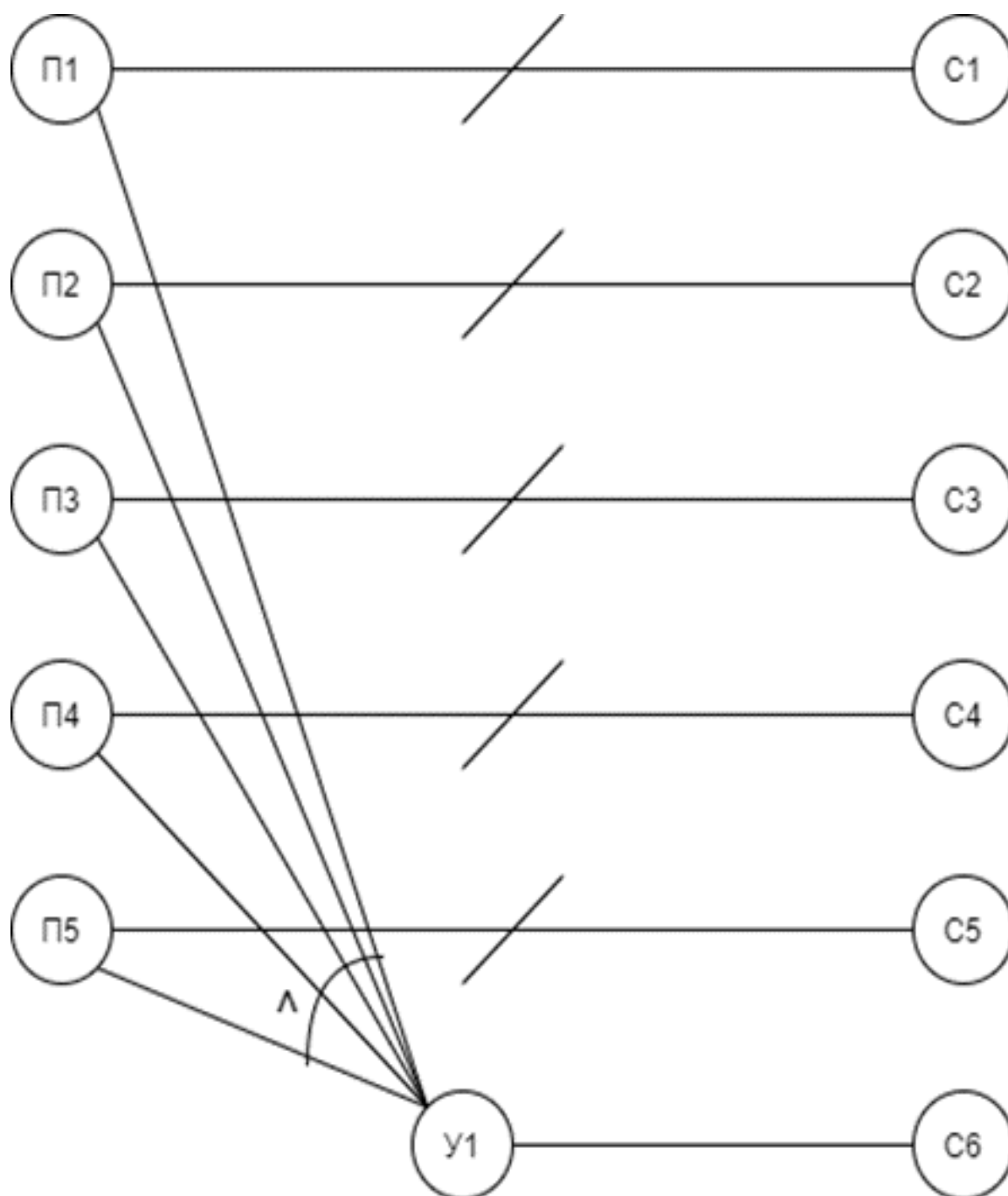


Рисунок 30 – Функциональная диаграмма

Как можно заметить – У1 наступает только при выполнении каждой из причин, образующих данную функцию. При неверном соблюдении каждой из причин она ведёт в её следствие, информирующее о соответствующей ошибке.

Полный комбинаторный перебор 5 причин даёт $2^5 = 32$ комбинации. Рассмотрим полный перебор сочетания всех вариантов причин и результирующих следствий.

Для подробного описания каждого тест-кейса была составлена соответствующая таблица тест-кейсов с входными и выходными результатами, а также фактическими результатами. Она представлена в Приложении Д.

Таким образом, функциональное тестирование выявило 5 причин и 6 следствий, созданы соответствующие тест-кейсы. Функция по созданию бронирования сектора при тестировании показала корректные результаты.

4.2 Метрики кода приложения

Статический анализ метрик кода был выполнен для метода, реализующего сценарий корректного отображения названий секторов при их просмотре. Метод расположен в файле `src/pond-master/SectorTable.tsx`.

Функция по нормализации названий секторов приведена в листинге 1.

Листинг 1 – Функция нормализации названия сектора

```
// Функция для нормализации названия сектора
function normalizeSectorNumber(side, number) {
  const rawNumber = (number ?? '').trim();
  const normalizedSide = normalizeSectorSide(side);

  // Проверка на пустое значение
  if (!rawNumber) {
    return '';
  }

  const preparedNumber = rawNumber.replace(/^#/ , '').trim();
  const formatted = preparedNumber.match(/^([A-ZА-Я]*) (\d+) \s* ([КК
ЖЖЖЖ])?$/i);
  if (!formatted) {
    return preparedNumber;
  }
  const numericPart = formatted[2];
  const suffix = formatted[3]?.toUpperCase();
```

```

// Проверка на К
if (suffix === 'К') {
    return `${numericPart}К`;
}
// Проверка на Ж
if (suffix === 'Ж') {
    return `${numericPart}Ж`;
}
if (normalizedSide === 'Красная') {
    return `${numericPart}К`;
}
if (normalizedSide === 'Желтая') {
    return `${numericPart}Ж`;
}

return numericPart;
}

```

Данная функция является основной для формирования корректного отображения названия сектора. Главный принцип заточен на получение стороны водоёма и номера сектора. Затем, номер очищается от использования решётки, добавляется корректный суффикс в зависимости от стороны и возвращается форматированный результат.

4.2.1 Метрики размера программы

Для начала был реализован подсчёт метрики размера кода при помощи процента комментариев.

Подсчёт данной метрики приведён в таблице 16.

Таблица 16 – Метрика размера программы

Метрика	Количество
Общее число строк кода (без пустых)	27
Число строк-комментариев	4

Процент комментариев программы составляет 15%.

Строки-комментарии объясняются необходимостью контроля переменных и их значений.

4.2.2 Метрики Холстеда

Метрики Холстеда оценивают сложность программы через словарь и частоту использования операторов и операндов. Все операторы и операнды приведены в таблицах 17 и 18 соответственно. Кроме того, сводные результаты приведены в таблице 19.

Таблица 17 – Словарь операторов

Операторы	Число вхождений
Function	1
Const	6
=	6
??	1
If	6
!	2
====	4
Return	7
n1 = 8	N1 = 33

Таблица 18 – Словарь операндов

Операнды	Число вхождений
normalizeSectorNumber	1
side	2
number	2
rawNumber	3
‘,’	2

Продолжение таблицы 18

Операнды	Число вхождений
normalizeSectorNumber	1
normalizeSectorSide	1
preparedNumber	3
/^#/	1
formatted	4
/^([A-ZА-Я]*)(\d+)\s*([ККЖЖжж])?\$/i	1
numericPart	6
2	1
suffix	3
3	1
К	3
Ж	3
Красная	1
Желтая	1
n2 = 19	N2 = 42

Таблица 19 – Сводные результаты расчётов

Метрика	Обозначение	Значение
Число уникальных операторов	n1	8
Число уникальных операндов	n2	19
Словарь программы	n	27
Суммарное число операторов	N1	33
Суммарное число операндов	N2	42
Длина программы	N	75
Объём программы	V	356.62
Сложность программы	D	8.84
Интеллектуальные усилия	E	3152.52

Таким образом, значение объема $V = 356.62$ находится в диапазоне 20–1000, что соответствует умеренной сложности функции.

4.2.3 Объектная метрика WMC

Метрика WMC (Weighted Methods per Class) определяет суммарную сложность всех методов класса [21].

WMC компонента BookingList = 42, вычислена как сумма цикломатических сложностей всех методов.

Значение умеренно высокое. Самый весомый вклад дает updateStatus () с CC = 9. Это объясняется тем; что внутри есть поиск нужной брони, ранний выход при ее отсутствии и обновление записи. getStatusColor и getStatusText дают по CC = 4.

Таким образом, статический анализ метода нормализации названия секторов показал умеренную сложность: объём по Холстеду $V = 356.62$, а цикломатическая сложность $CC = 42$. Показатели соответствуют уровню, приемлемому для сопровождения и тестирования.

Заключение

В ходе выполнения выпускной квалификационной работы была достигнута цель в проектировании и реализации фронтенд-компонента информационной системы для автоматизации бронирования и учёта услуг рыболовного хозяйства.

На этапе анализа предметной области было реализовано изучение рассматриваемой тематики, выбраны программы-аналоги для реализации сравнительного анализа, определены ключевые функциональные и нефункциональные требования; а также выполнены планирование разработки и оценка бюджета для разработки поставленного программного продукта.

В ходе планирования разработки проекта и оценки бюджета были получены показатели, что общая продолжительность разработки приложения составляет 448 часов; а общие затраты на разработку – 89.600 рублей.

Для реализации данного проекта в качестве архитектурного решения выбрана многоуровневая архитектура, обладающая рядом значительных преимуществ, таких как обеспечение модульности, масштабируемости и повышенный показатель удобства сопровождения системы.

Для проектирования базы данных было принято решение об использовании концептуального моделирования, включающего в себя стадии построения концептуальной, логической и физической схемы базы данных.

На стадии построения графической составляющей информационной системы был использован инструментарий Visual Paradigm, который позволил структурировать различные состояния и взаимосвязь компонентов в виде диаграмм.

В результате разработки программных средств были использованы языки разметки HTML и CSS с использованием фреймворка Tailwind CSS, а также язык программирования TypeScript с применением фреймворка React. Ключевой инструментальной средой выступала Visual Studio Code.

На этапе проведения тестирования программного продукта были реализованы различные формы тестов, в том числе функциональное тестирование программных средств с построением функциональной диаграммы и осуществление тестирования разработанного интерфейса.

В качестве анализа метрик кода был выбран инструментарий среды Visual Studio Code. В ходе проведения оценки были получены выводы, что разработанный программный код имеет умеренную цикломатическую сложность, приемлемую для проведения качественного сопровождения программного продукта.

Также, стоит отметить, что представленное программное обеспечение было внедрено в деятельность индивидуального предпринимателя Гришиной Светланы Николаевны, являющейся владельцем рыболовного хозяйства «Пруд Орловский». Акт о внедрении программного продукта представлен в приложении Ж.

В итоге, можно сделать вывод, что поставленные цели для выпускной квалификационной работы были достигнуты. Полученная разработанная система готова к внедрению на реальной рыболовной базе и способна быть масштабируемой на других объектах рекреационного хозяйства, в том числе для кемпингов и спортивных комплексов.

Список использованных источников

1. Жигин, А. В. Рыбоводно-рыболовное рекреационное хозяйство / А. В. Жигин, П. В. Терентьев. – М. : ВНИРО, 2015. – 216 с.
2. Рыболовная база отдыха «Авалон» // fishing-avalon.ru [Электронный ресурс]. URL: <https://fishing-avalon.ru/?ysclid=mpc8qv1lmb72516449> (дата обращения: 10.05.2026).
3. База отдыха «Фрегат» // fregat30.ru [Электронный ресурс]. URL: <https://fregat30.ru/?ysclid=mpc911a57646252602> (дата обращения: 10.05.2026).
4. Рыболовно-охотничья база отдыха «Клевое место» // klevoemesto.top [Электронный ресурс]. URL: <https://klevoemesto.top/?ysclid=mpc951zcj266559337> (дата обращения: 10.05.2026).
5. Использование диаграммы вариантов использования UML при проектировании программного обеспечения // habr.com [Электронный ресурс]. URL: <https://habr.com/ru/articles/566218/> (дата обращения: 12.05.2026).
6. Нефункциональные требования: как не пустить систему ко дну // habr.com [Электронный ресурс]. URL: <https://habr.com/ru/companies/simbirsoft/articles/688428/> (дата обращения: 12.05.2026).
7. Использование MS Project для управления проектами по разработке ПО // habr.com [Электронный ресурс]. URL: <https://habr.com/ru/articles/151593/> (дата обращения: 12.05.2026).
8. Концептуальная модель базы данных // studfile.net [Электронный ресурс]. URL: <https://studfile.net/preview/9959196/page:2/> (дата обращения: 12.05.2026).
9. Логическое моделирование баз данных: принципы и лучшие практики // sky.pro [Электронный ресурс]. URL: <https://sky.pro/wiki/sql/logicheskaya-model-bazy-dannyh-chto-eto-i-kak-eyo-sozdat/?ysclid=mpww2olj5m873156625> (дата обращения: 12.05.2026).

10. Введение в ASP.NET Core // metanit.com [Электронный ресурс]. URL: <https://metanit.com/sharp/aspnet6/1.1.php?ysclid=mpww7d9ubg361429869> (дата обращения: 13.05.2026).
11. Основы кроссплатформенной верстки // sky.pro [Электронный ресурс]. URL: <https://sky.pro/wiki/html/osnovy-krossplatforming-verstki/> (дата обращения: 15.05.2026).
12. Основы ООП: классы, объекты, атрибуты, методы – базовые концепции // sky.pro [Электронный ресурс]. URL: <https://sky.pro/wiki/python/osnovnye-ponyatiya-oop-obuekty-klassy-atributy-i-metody/?ysclid=mpwwekndlv254312212> (дата обращения: 15.05.2026).
13. Обзор синтаксиса Tailwind CSS // habr.com [Электронный ресурс]. URL: <https://habr.com/ru/companies/otus/articles/796101/> (дата обращения: 17.05.2026).
14. Введение в TypeScript // metanit.com [Электронный ресурс]. URL: <https://metanit.com/web/typescript/1.1.php?ysclid=mpwxod0d8a547795876> (дата обращения: 17.05.2026).
15. Что такое React // metanit.com [Электронный ресурс]. URL: <https://metanit.com/web/react/1.1.php?ysclid=mpwxkikie2486984960> (дата обращения: 17.05.2026).
16. Что такое PostgreSQL? // metanit.com [Электронный ресурс]. URL: <https://metanit.com/sql/postgresql/1.1.php?ysclid=mq50pp5i6w611210682> (дата обращения: 17.05.2026).
17. Visual Studio Code: универсальный инструмент для современной разработки // sky.pro [Электронный ресурс]. URL: <https://sky.pro/wiki/profession/visual-studio-code-opisanie-i-funkcii/?ysclid=mpwxwr77rx65163735> (дата обращения: 20.05.2026).
18. Лекция 9: Физические модели баз данных // www.opennet.ru [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/1001/297/lecture/7415?ysclid=mq50rjzge2862202639> (дата обращения: 22.05.2026).

19. Репозиторий Pond_App: клиентское приложение // GITHUB [Электронный ресурс]. URL: https://github.com/Ilya24062004/Pond_App (дата обращения: 24.05.2026).

20. Что такое Black Box Testing? // habr.com [Электронный ресурс]. URL: <https://habr.com/ru/companies/ruvds/articles/700858/> (дата обращения: 24.05.2026).

21. Программный код и его метрики // habr.com [Электронный ресурс]. URL: <https://habr.com/ru/companies/intel/articles/106082/> (дата обращения: 24.05.2026).

Приложение А
(справочное)
Глоссарий

База данных – это организованная и структурированная в соответствии с набором правил совокупность данных определённой предметной области.

СУБД – это комплекс специализированного программного обеспечения, направленного на осуществление управления, хранения и модификации базы данных.

Фреймворк – набор готового стандартизированного инструментария, позволяющего ускорить разработку программного обеспечения.

React – декларативная библиотека на основе JavaScript, которая используется для разработки различных пользовательских интерфейсов.

Tailwind CSS – CSS-фреймворк с открытым исходным кодом, который ориентируется на работе с небольшими классами стилей для комбинации их в разметке страницы и качественной стилизации элементов.

Информационная система – это взаимосвязанная совокупность средств и методов для осуществления поиска, хранения и обработки имеющейся информации в контексте рассматриваемой предметной области.

PostgreSQL – объектно-реляционная СУБД с открытым исходным кодом, направленная для реализацию ключевых операций для хранения, модификации и извлечения данных.

UML – унифицированный язык моделирования, предназначенный для проектирования и построения графических схем и диаграмм, описывающих контекст поведения и структуру программной системы.

Приложение Б
(обязательное)
Листинг приложения

Листинг Б.1 – ProductTable.tsx

```
import { useEffect, useMemo, useRef, useState } from 'react';
import { Check, Edit2, Image, Plus, Trash2, X } from 'lucide-react';
import { supabase } from '../services/supabaseClient';

interface ProductItem {
  id?: string;
  name: string;
  type: 'продажа' | 'аренда';
  is_for_sale?: boolean;
  is_for_rent?: boolean;
  price: number;
  stock: number;
  description: string;
  image_url: string;
}

type ProductModeFilter = 'all' | 'sale' | 'rent' | 'both';
type ProductSort = 'name-asc' | 'price-asc' | 'price-desc' | 'stock-asc' | 'stock-desc';

const emptyProduct: ProductItem = {
  name: '',
  type: 'продажа',
  is_for_sale: true,
  is_for_rent: false,
  price: 0,
  stock: 100,
  description: '',
  image_url: '',
};

const getProductMode = (product: ProductItem) => {
  const sale = product.is_for_sale ?? product.type === 'продажа';
  const rent = product.is_for_rent ?? product.type === 'аренда';

  if (sale && rent) return 'both';
  if (sale) return 'sale';
  if (rent) return 'rent';
  return 'none';
};

export default function ProductTable() {
  const [products, setProducts] = useState<ProductItem[]>([]);
  const [editing, setEditing] = useState<ProductItem | null>(null);
  const [isAdd, setIsAdd] = useState(false);
  const [uploading, setUploading] = useState(false);
  const [modeFilter, setModeFilter] =
    useState<ProductModeFilter>('all');
```

```

    const [productSort, setProductSort] =
useState<ProductSort>('name-asc');
    const fileInputRef = useRef<HTMLInputElement>(null);

    const visibleProducts = useMemo(() => {
        const filtered = products.filter((product) => {
            const mode = getProductMode(product);

            switch (modeFilter) {
                case 'sale':
                    return mode === 'sale' || mode === 'both';
                case 'rent':
                    return mode === 'rent' || mode === 'both';
                case 'both':
                    return mode === 'both';
                default:
                    return true;
            }
        });

        return [...filtered].sort((a, b) => {
            switch (productSort) {
                case 'price-asc':
                    return a.price - b.price;
                case 'price-desc':
                    return b.price - a.price;
                case 'stock-asc':
                    return a.stock - b.stock;
                case 'stock-desc':
                    return b.stock - a.stock;
                default:
                    return a.name.localeCompare(b.name, 'ru');
            }
        });
    }, [modeFilter, productSort, products]);

    useEffect(() => {
        void fetchProducts();
    }, []);

    const fetchProducts = async () => {
        const { data, error } = await
supabase.from('products').select('*').order('name');

        if (error) {
            console.error('Error fetching products:', error);
            return;
        }

        setProducts((data ?? []) as ProductItem[]);
    };

```

```

    const handlePhotoUpload = async (file: File, bucket = 'product-
photos'): Promise<string | null> => {
        setUploading(true);

        try {
            if (file.size > 5 * 1024 * 1024) {
                alert('Файл слишком большой. Максимальный размер: 5 МБ.');
```

return null;

```
            }

            if (!file.type.startsWith('image/')) {
                alert('Можно загружать только изображения: JPEG, PNG или
WebP.');
```

return null;

```
            }

            const fileExt = file.name.split('.').pop();
            const fileName = `product-${Date.now()}-${
Math.random().toString(36).slice(2)}.${fileExt}`;

            const { error: uploadError } = await
supabase.storage.from(bucket).upload(fileName, file, {
                cacheControl: '3600',
                contentType: file.type,
                upsert: false,
            });

            if (uploadError) throw uploadError;

            const { data } =
supabase.storage.from(bucket).getPublicUrl(fileName);
            return data.publicUrl;
        } catch (error) {
            console.error('Error uploading product photo:', error);
            alert(`Ошибка загрузки фото: ${error instanceof Error ?
error.message : 'неизвестная ошибка'}`);
            return null;
        } finally {
            setUploading(false);
        }
    };

    const saveProduct = async (product: ProductItem) => {
        if (!product.name.trim()) {
            alert('Укажите название товара.');
```

return;

```
        }

        const isForSale = Boolean(product.is_for_sale);
        const isForRent = Boolean(product.is_for_rent);

        if (!isForSale && !isForRent) {
```

```

    alert('Выберите хотя бы один режим: продажа или аренда.');
```

```

    return;
}

const payload = {
  name: product.name.trim(),
  type: isForSale ? 'продажа' : 'аренда',
  is_for_sale: isForSale,
  is_for_rent: isForRent,
  price: Number(product.price) || 0,
  stock: Number(product.stock) || 0,
  description: product.description ?? '',
  image_url: product.image_url ?? '',
  updated_at: new Date().toISOString(),
};

const result = product.id
  ? await supabase.from('products').update(payload).eq('id',
product.id).select()
  : await
supabase.from('products').insert([payload]).select();

if (result.error) {
  console.error('Save product error:', result.error);
  alert(`Ошибка сохранения товара: ${result.error.message}`);
  return;
}

setEditing(null);
setIsAdd(false);
await fetchProducts();
alert('Товар сохранён.');
```

```

};

const deleteProduct = async (product: ProductItem) => {
  if (!product.id) return;
  if (!confirm(`Удалить товар "${product.name}"?`)) return;

  const { error } = await
supabase.from('products').delete().eq('id', product.id);

  if (error) {
    console.error('Delete product error:', error);
    alert(`Ошибка удаления товара: ${error.message}`);
    return;
  }

  await fetchProducts();
};

if (!editing) {
  return (

```

```

    <div>
      <div className="mb-6 flex items-center justify-between
gap-4">
        <h2 className="text-2xl font-bold">Управление
товарами</h2>
        <button
          onClick={() => {
            setEditing(emptyProduct);
            setIsAdd(true);
          }}
          className="flex items-center gap-2 rounded-lg bg-blue-
600 px-4 py-2 text-white hover:bg-blue-700"
        >
          <Plus size={18} /> Добавить товар
        </button>
      </div>

      <div className="mb-4 grid gap-4 rounded-2xl bg-white/90 p-
4 shadow sm:grid-cols-2 xl:grid-cols-4">
        <label className="block">
          <span className="mb-2 block text-sm font-medium text-
slate-700">Режим</span>
          <select className="input-field" value={modeFilter}
onChange={(event) => setModeFilter(event.target.value as
ProductModeFilter)}>
            <option value="all">Все товары</option>
            <option value="sale">Продажа</option>
            <option value="rent">Аренда</option>
            <option value="both">Продажа и аренда</option>
          </select>
        </label>

        <label className="block">
          <span className="mb-2 block text-sm font-medium text-
slate-700">Сортировка</span>
          <select className="input-field" value={productSort}
onChange={(event) => setProductSort(event.target.value as
ProductSort)}>
            <option value="name-asc">По названию</option>
            <option value="price-asc">Цена: сначала
дешевле</option>
            <option value="price-desc">Цена: сначала
дороже</option>
            <option value="stock-asc">Остаток: меньше
сначала</option>
            <option value="stock-desc">Остаток: больше
сначала</option>
          </select>
        </label>

```

// Фрагмент кода

Листинг Б.2 – TariffTable.tsx

```
import { useEffect, useState } from 'react';
import { Check, Edit2, X } from 'lucide-react';
import { supabase } from '../services/supabaseClient';
import { TARIFF_CATEGORY } from '../constants';
import { AdminNoticeToast, type AdminNoticeState } from
'../AdminFeedback';
import { DEFAULT_TARIFFS, getTariffLabelFromCategory,
normalizeTariffs, type TariffItem } from '../utils/tariffs';
import { keepNumericSelectionOnMouseUp, selectNumericValueOnFocus
} from '../utils/numericInput';

export default function TariffTable() {
  const [tariffs, setTariffs] = useState<TariffItem[]>([]);
  const [editing, setEditing] = useState<TariffItem | null>(null);
  const [notice, setNotice] = useState<AdminNoticeState |
null>(null);

  const showNotice = (title: string, message: string, tone:
'success' | 'error') => {
    setNotice({ title, message, tone });
    window.setTimeout(() => {
      setNotice((current) => (current?.title === title &&
current?.message === message ? null : current));
    }, 4000);
  };

  const setVisibleTariffs = (items: TariffItem[] | null |
undefined) => {
    setTariffs(normalizeTariffs((items as TariffItem[] | null) ??
DEFAULT_TARIFFS));
  };

  async function fetchTariffs() {
    const { data } = await supabase.from('tariffs').select('*');
    setVisibleTariffs((data as TariffItem[] | null) ??
DEFAULT_TARIFFS);
  }

  useEffect(() => {
    let active = true;
    let channel: ReturnType<typeof supabase.channel> | null =
null;

    const loadTariffs = async () => {
      const { data } = await supabase.from('tariffs').select('*');
      if (active) {
        setVisibleTariffs((data as TariffItem[] | null) ??
DEFAULT_TARIFFS);
      }
    };
  });
}
```

```

void loadTariffs();

const handleFocus = () => {
  void loadTariffs();
};

const handleVisibility = () => {
  if (!document.hidden) {
    void loadTariffs();
  }
};

const intervalId = window.setInterval(() => {
  void loadTariffs();
}, 30000);

try {
  channel = supabase
    .channel('admin-tariffs')
    .on('postgres_changes', { event: '*', schema: 'public',
table: 'tariffs' }, () => {
    void loadTariffs();
  })
    .subscribe();
} catch (error) {
  console.warn('Realtime updates for tariffs are
unavailable.', error);
}

window.addEventListener('focus', handleFocus);
document.addEventListener('visibilitychange',
handleVisibility);

return () => {
  active = false;
  window.clearInterval(intervalId);
  window.removeEventListener('focus', handleFocus);
  document.removeEventListener('visibilitychange',
handleVisibility);
  if (channel) {
    void supabase.removeChannel(channel);
  }
};
}, []);

const saveTariff = async (tariff: TariffItem) => {
  const payload = {
    ...tariff,
    label: getTariffLabelFromCategory(tariff.category),
    description: tariff.description?.trim() ?? '',
    price_per_day: Number(tariff.price_per_day) || 0,

```

```

    };

    if (!tariff.id) {
      showNotice('Не удалось сохранить тариф', 'Можно изменять
только существующие тарифы.', 'error');
      return;
    }

    const result = await
supabase.from('tariffs').update(payload).eq('id',
tariff.id).select();

    if (result.error) {
      showNotice('Не удалось сохранить тариф', `Причина:
${result.error.message}`, 'error');
      return;
    }

    const savedTariffs = (result.data as TariffItem[] | null) ??
null;
    if (savedTariffs && savedTariffs.length > 0) {
      setVisibleTariffs(
tariffs.map((item) =>
item.id === tariff.id
? {
    ...item,
    ...savedTariffs[0],
    label:
getTariffLabelFromCategory(savedTariffs[0].category),
  }
: item,
      ),
    );
  }

  setEditing(null);
  await fetchTariffs();
  showNotice('Тариф сохранён', `Категория «${payload.label}»
успешно обновлена.`, 'success');
};

if (!editing) {
  return (
    <div>
      <div className="mb-6 flex items-center justify-between">
        <h2 className="text-2xl font-bold">Управление
тарифами</h2>
      </div>

      <div className="overflow-hidden rounded-xl bg-white
shadow">
        <table className="w-full">

```

```

        <thead className="bg-slate-50">
            <tr>
                <th className="px-6 py-4 text-left text-sm font-
semibold text-slate-600">Категория</th>
                <th className="px-6 py-4 text-left text-sm font-
semibold text-slate-600">Описание</th>
                <th className="px-6 py-4 text-left text-sm font-
semibold text-slate-600">Стоимость</th>
                <th className="px-6 py-4 text-left text-sm font-
semibold text-slate-600">Действия</th>
            </tr>
        </thead>
        <tbody>
            {tariffs.map((tariff) => (
                <tr key={tariff.id ?? tariff.category}
className="border-t border-slate-100 hover:bg-slate-50">
                    <td className="px-6 py-4 font-medium">
                        <div>{tariff.label}</div>
                        <div className="mt-1 text-xs font-normal
uppercase tracking-wide text-slate-400">
                            {tariff.label.toUpperCase()}
                        </div>
                    </td>
                    <td className="px-6 py-4 text-slate-
600">{tariff.description || 'Описание не указано'}</td>
                    <td className="px-6 py-4 font-bold text-green-
600">
                        {tariff.price_per_day} ₸{tariff.category ===
TARIFF_CATEGORY.БРОНИРОВАНИЕ_СЕКТОРА ? '' : '/день'}
                    </td>
                    <td className="px-6 py-4">
                        <button
                            onClick={() => setEditing(tariff)}
                            className="rounded-lg p-2 text-blue-600
hover:bg-blue-50"
                        >
                            <Edit2 size={18} />
                        </button>
                    </td>
                </tr>
            ))}
        </tbody>
    </table>
</div>

    <AdminNoticeToast notice={notice} />
</div>
);
}

return (

```

```

    <div className="fixed inset-0 z-50 flex items-center justify-
center bg-black/60 backdrop-blur-sm">
    <div className="w-full max-w-lg rounded-2xl bg-white p-6">
      <div className="mb-6 flex items-center justify-between">
        <h3 className="text-xl font-bold">Редактировать
тариф</h3>
        <button onClick={() => setEditing(null)} className="p-2
text-slate-400 hover:text-slate-600">
          <X size={24} />
        </button>
      </div>

      <div className="space-y-4">
        <div>
          <label className="mb-2 block text-sm font-medium text-
slate-700">Тип категории</label>
          <div className="input-field bg-slate-50 text-slate-
600">{getTariffLabelFromCategory(editing.category)}</div>
        </div>

        <div>
          <label className="mb-2 block text-sm font-medium text-
slate-700">Название</label>
          <div className="input-field bg-slate-50 text-slate-
600">{getTariffLabelFromCategory(editing.category)}</div>
        </div>

        <div>
          <label className="mb-2 block text-sm font-medium text-
slate-700">
            {editing.category ===
TARIFF_CATEGORY.БРОНИРОВАНИЕ_СЕКТОРА ? 'Разовая стоимость *' :
'Стоимость за сутки *'}
          </label>
          <input
            type="number"
            className="input-field"
            value={editing.price_per_day}
            onFocus={selectNumericValueOnFocus}
            onMouseUp={keepNumericSelectionOnMouseUp}
            onChange={(event) => setEditing({ ...editing,
price_per_day: Number(event.target.value) })}
          />
        </div>

        <div>
          <label className="mb-2 block text-sm font-medium text-
slate-700">Описание</label>
          <textarea
            className="input-field"
            value={editing.description}

```

```

        onChange={(event) => setEditing({ ...editing,
description: event.target.value })}
        rows={3}
      />
    </div>

    <div className="flex gap-4 pt-4">
      <button onClick={() => void saveTariff(editing)}
className="btn-primary flex flex-1 items-center justify-center
gap-2">
        <Check size={18} /> Сохранить
      </button>
      <button
        onClick={() => setEditing(null)}
        className="flex-1 rounded-lg bg-slate-100 px-4 py-3
font-medium text-slate-700 hover:bg-slate-200"
      >
        Отмена
      </button>
    </div>
  </div>
</div>

  <AdminNoticeToast notice={notice} />
</div>
);
}

// Фрагмент кода

```

Приложение В
(рекомендуемое)
Спецификация сущностей базы данных

Таблица В.1 – Спецификация атрибутов сущности «Клиент»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор клиента	integer	10	от 1 до 2 147 483 467	нет	PK	1
Фамилия	Фамилия клиента	varchar	255	а – я А – Я	да		Иванов
Имя	Имя клиента	varchar	255	а – я А – Я	нет		Александр
Отчество	Отчество клиента	varchar	255	а – я А – Я	да		Сергеевич
Номер телефона	Номер телефона клиента	varchar	12	строка, формат +7XXXXXXXXXX	нет	U	+79996120242

Таблица В.2 – Спецификация атрибутов сущности «Администратор»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор администратора	integer	10	от 1 до 2 147 483 467	нет	PK	1
Логин	Логин администратора для входа в систему	varchar	255	а – z; А – Z; 0 – 9.	нет	U	admin
Пароль	Хешированный пароль администратора	varchar	255	а – z А – Z 0 – 9.	нет		maxibon_17!

Таблица В.3 – Спецификация атрибутов сущности «Тариф»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор тарифа	integer	10	от 1 до 2 147 483 467	нет	PK	1

Продолжение таблицы В.3

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
Описание	Описание тарифа	varchar	255	а – я; А – Я; 0 – 9.	нет		Стоимость суточной рыбалки для рыбака-мужчины
Стоимость	Стоимость тарифа	decimal	10,2	от 0 до 999999999,99	нет		800,00

Таблица В.4 – Спецификация атрибутов сущности «Сторона водоёма»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор стороны водоёма	integer	10	от 1 до 2 147 483 467	нет	PK	1
Название стороны	Название стороны водоёма	varchar	255	а – я А – Я 0 – 9,	нет	U	Жёлтая сторона

Таблица В.5 – Спецификация атрибутов сущности «Товар продажи»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор товара продажи	integer	10	от 1 до 2 147 483 467	нет	PK	1
Название товара	Название товара продажи	varchar	255	а – я А – Я	нет	U	Кукуруза
Описание товара	Описание товара продажи	varchar	255	а – я А – Я	нет		Насадка для рыбалки
Стоимость товара	Стоимость единицы товара продажи	varchar	10,2	от 0 до 999999999,99	нет		180,00
Количество в наличии	Количество товара продажи в наличии	varchar	10	от 0 до 2 147 483 467	нет	U	50
Фотографии товара	Ссылка на фотографию товара продажи	varchar	255	а – z А – Z 0 – 9.	нет		corn.jpg

Таблица В.6 – Спецификация атрибутов сущности «Товар аренды»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор товара аренды	integer	10	от 1 до 2 147 483 467	нет	PK	1
Название товара	Название товара аренды	varchar	255	а – я А – Я	нет	U	Удочка
Описание товара	Описание товара аренды	varchar	255	а – я А – Я	нет		Поплавочная удочка
Стоимость аренды за сутки	Стоимость аренды товара за сутки	varchar	10,2	от 0 до 999999999,99	нет		300,00
Количество в наличии	Количество товара аренды в наличии	varchar	10	от 0 до 2 147 483 467	нет	U	20
Фотографии товара	Ссылка на фотографии товара аренды	varchar	255	а – z А – Z 0 – 9.	нет		rod.jpg

Таблица В.7 – Спецификация атрибутов сущности «Сектор»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор сектора	integer	10	от 1 до 2 147 483 467	нет	PK	1
Номер сектора	Номер рыболовного сектора	integer	10	а – я; А – Я.	нет		17
Количество подходов к воде	Количество подходов к воде в секторе	integer	10	а – я; А – Я.	нет		2
Максимально разрешенное количество человек	Максимальное разрешенное количество отдыхающих на секторе	integer	10	от 1 до 2 147 483 467	нет		5

Продолжение таблицы В.7

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
Средняя глубина	Средняя глубина водоёма в зоне сектора, м	numeric	5,2	от 0 до 999,99	нет	U	2,50
Наличие беседки	Признак наличия беседки в секторе	bit	1	0 – нет, 1 – да	нет		1
Доступность сектора	Признак доступности сектора для бронирования	bit	1	0 – нет, 1 – да	нет		1
Фотографии сектора	Ссылка на фотографии сектора	varchar	255	a – z A – Z 0 – 9	нет		sector6.jpg
Сторона водоёмаID	Идентификатор стороны водоёма	integer	10	от 1 до 2 147 483 467	нет	FK	1

Таблица В.8 – Спецификация атрибутов сущности «Бронирование»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор бронирования	integer	10	от 1 до 2 147 483 467	нет	PK	1
Код бронирования	Уникальный код бронирования клиента	varchar	6	a – z A – Z 0 – 9	нет	U	A7K92B
Дата заезда	Дата начала бронирования	date		20.04.2026 – 31.12.2100	нет		10.08.2026
Дата выезда	Дата окончания бронирования	date		20.04.2026 – 31.12.2100	нет		13.08.2026
Время прибытия	Планируемое время прибытия клиента	time	7	00:00:00 – 23:59:59	нет	U	08:00:00

Продолжение таблицы В.8

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
Итоговая стоимость	Итоговая стоимость бронирования	decimal	10,2	от 0 до 999999999,99	нет		7400,00
Статус бронирования	Текущий статус бронирования	integer	10	0 – 9	нет		1
Дата создания	Дата и время создания бронирования	timestamp		20.04.2026 00:00 – 31.12.2100 23:59	нет		10.08.2026 12:30
Дата подтверждения	Дата и время подтверждения бронирования	timestamp		20.04.2026.00:00:00 – 31.12.2100.23:59:59	да		10.08.2026 13:00
СекторID	Идентификатор выбранного сектора	integer	10	от 1 до 2 147 483 467	нет	FK	6
КлиентID	Идентификатор клиента, оформившего бронирование	integer	10	от 1 до 2 147 483 467	нет	FK	1
АдминистраторID	Идентификатор администратора, изменившего статус бронирования	integer	10	от 1 до 2 147 483 467	нет	FK	1

Таблица В.9 – Спецификация атрибутов сущности «Позиция бронирования»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор позиции бронирования	integer	10	от 1 до 2 147 483 467	нет	PK	1

Продолжение таблицы В.9

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
Количество	Количество единиц выбранного тарифа	integer	10	от 1 до 2 147 483 467	нет		2
БронированиеID	Идентификатор связанного бронирования	integer	10	от 1 до 2 147 483 467	нет	FK	1
ТарифID	Идентификатор выбранного тарифа	integer	10	от 1 до 2 147 483 467	нет	FK	1

Таблица В.10 – Спецификация атрибутов сущности «Позиция товара продажи»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор позиции товара продажи	integer	10	от 1 до 2 147 483 467	нет	PK	1
Количество товара	Количество выбранного товара продажи	integer	10	от 1 до 2 147 483 467	нет		3
Стоимость на момент продажи	Стоимость товара продажи на момент оформления бронирования	decimal	10,2	от 0 до 99999999,99	нет		180,00
БронированиеID	Идентификатор связанного бронирования	integer	10	от 1 до 2 147 483 467	нет	FK	1
Товар продажиID	Идентификатор товара продажи	integer	255	от 1 до 2 147 483 467	нет	FK	1

Таблица В.11 – Спецификация атрибутов сущности «Позиция товара аренды»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор позиции товара аренды	integer	10	от 1 до 2 147 483 467	нет	PK	1

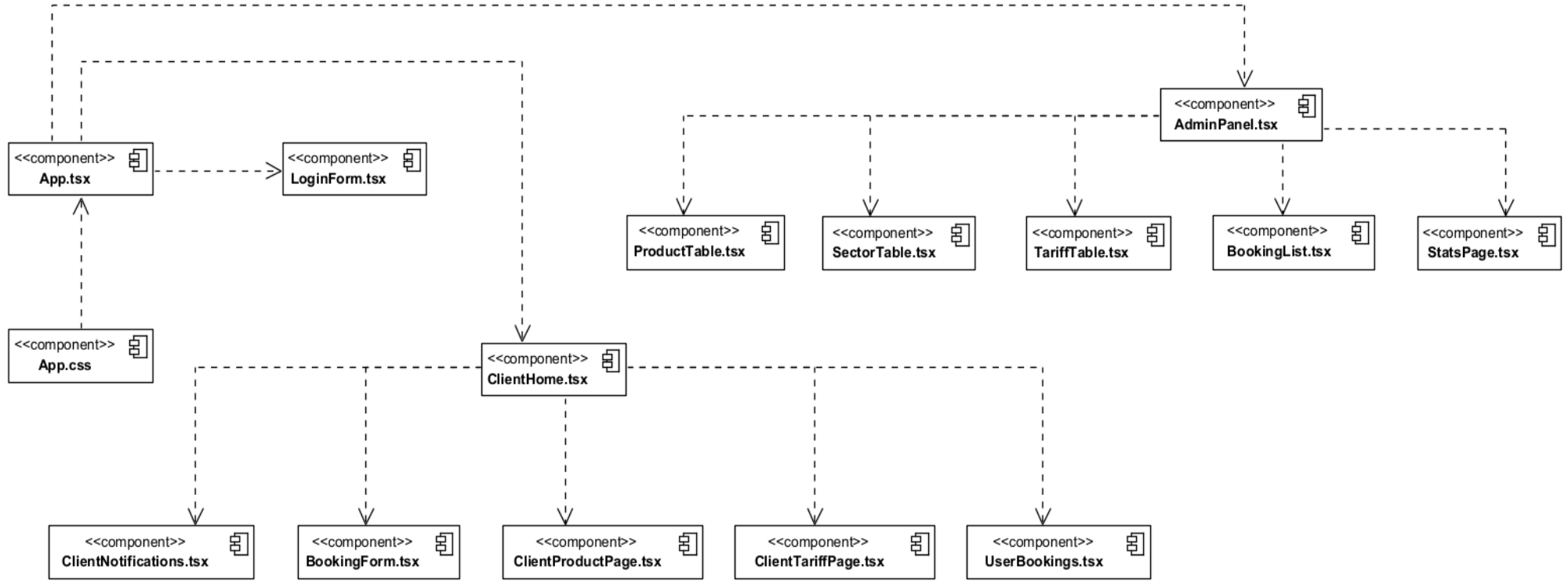
Продолжение таблицы В.11

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
Количество товара	Количество выбранного товара аренды	integer	10	от 1 до 2 147 483 467	нет		3
Количество дней аренды	Количество дней аренды товара	integer	10	от 1 до 2 147 483 467	нет		3
Стоимость на момент аренды	Стоимость товара аренды на момент оформления бронирования	decimal	10,2	от 0 до 99999999,99	нет		300,00
БронированиеID	Идентификатор связанного бронирования	integer	10	от 1 до 2 147 483 467	нет	FK	1
Товар арендыID	Идентификатор товара аренды	integer	255	от 1 до 2 147 483 467	нет	FK	1

Таблица В.12 – Спецификация атрибутов сущности «Уведомление»

Идентификатор столбца	Содержательное описание	Тип данных	Размерность	Область допустимых значений	Null	Роль	Пример
ID	Уникальный идентификатор уведомления	integer	10	от 1 до 2 147 483 467	нет	PK	1
Сообщение уведомления	Текст уведомления, связанного с бронированием	varchar	255	а – я; А – Я; 0 – 9.	нет	U	Бронирование подтверждено
Дата создания	Дата и время создания уведомления	timestamp		20.04.2026 00:00 – 31.12.2100 23:59	нет		01.08.2026 13:00
Подтверждение отправки	Признак подтверждения отправки уведомления	bit	1	0 – нет, 1 – да	нет		1
БронированиеID	Идентификатор связанного бронирования	integer	10	от 1 до 2 147 483 467	нет	FK	1

Приложение Г
(рекомендуемое)
Диаграмма компонентов



Приложение Д
(рекомендуемое)
Тест-кейсы функционального тестирования

Таблица Д – Тест-кейсы

№	Входные данные	Ожидаемый результат	Фактический результат
1	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15 (макс. 10) ФИО пусто. Телефон: пусто.	Некорректный период бронирования, сектор занят на выбранный период, превышено рекомендуемое количество людей, не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.1
2	Сектор: занят Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15 ФИО пусто. Телефон: пусто.	Сектор занят на выбранный период, превышено рекомендуемое количество людей, не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.2
3	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15. ФИО пусто. Телефон: пусто.	Некорректный период бронирования, превышено рекомендуемое количество людей, не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.3
4	Сектор: свободен Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15. ФИО пусто. Телефон: пусто.	Превышено рекомендуемое количество людей. не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.4
5	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8. ФИО пусто. Телефон: пусто.	Некорректный период бронирования, сектор занят на выбранный период, не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.5
6	Сектор: занят Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 8. ФИО пусто. Телефон: пусто.	Сектор занят на выбранный период, не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.6

Продолжение таблицы Д

№	Входные данные	Ожидаемый результат	Фактический результат
7	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8. ФИО пусто. Телефон: пусто.	Некорректный период бронирования, не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.7
8	Сектор: свободен Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 8. ФИО пусто. Телефон: пусто.	Не заполнено имя клиента, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.8
9	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15 ФИО: Иванов Иван Иванович, Тел.: +79181234567.	Некорректный период бронирования, сектор занят на выбранный период, превышено рекомендуемое количество людей.	Тест пройден. Результат приведен на рисунке Е.9
10	Сектор: занят Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15 ФИО+телефон заполнены.	Сектор занят на выбранный период, превышено рекомендуемое количество людей.	Тест пройден. Результат приведен на рисунке Е.10
11	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15 ФИО+телефон заполнены.	Некорректный период бронирования, превышено рекомендуемое количество людей.	Тест пройден. Результат приведен на рисунке Е.11
12	Сектор: свободен Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15 ФИО+телефон заполнены.	Превышено рекомендуемое количество людей.	Тест пройден. Результат приведен на рисунке Е.12

Продолжение таблицы Д

№	Входные данные	Ожидаемый результат	Фактический результат
13	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8 ФИО+телефон заполнены.	Некорректный период бронирования, сектор занят на выбранный период.	Тест пройден. Результат приведен на рисунке Е.13
14	Сектор: занят Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 8 ФИО+телефон заполнены.	Сектор занят на выбранный период.	Тест пройден. Результат приведен на рисунке Е.14
15	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8 ФИО+телефон заполнены.	Некорректный период бронирования.	Тест пройден. Результат приведен на рисунке Е.15
16	Сектор: свободен Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 8 Телефон: +79656337764. ФИО: пусто.	Не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.16
17	Сектор: свободен Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15. Телефон: +79656337764. ФИО: пусто.	Превышено рекомендуемое количество людей, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.17
18	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15 Телефон: пусто. ФИО: Иванов Иван Иванович.	Некорректный период бронирования, сектор занят на выбранный период, превышено рекомендуемое количество людей, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.18

Продолжение таблицы Д

№	Входные данные	Ожидаемый результат	Фактический результат
19	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15 ФИО/телефон: пусто.	Некорректный период бронирования, превышено рекомендуемое количество людей, не заполнены контактные данные клиента, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.19
20	Сектор: занят Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 8. Телефон: +79656337764. ФИО: пусто.	Сектор занят на выбранный период, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.20
21	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8 ФИО/телефон: пусто.	Некорректный период бронирования, не заполнены контактные данные клиента, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.21
22	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8 ФИО/телефон: пусто.	Некорректный период бронирования, сектор занят на выбранный период, не заполнены контактные данные клиента, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.22
23	Сектор: свободен Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15. Телефон: пусто. ФИО: Иванов Иван Иванович.	Превышено рекомендуемое количество людей, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.23
24	Сектор: занят Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15. Телефон: +79656337764. ФИО: пусто.	Сектор занят на выбранный период, превышено рекомендуемое количество людей, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.24

Продолжение таблицы Д

№	Входные данные	Ожидаемый результат	Фактический результат
25	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15. Телефон: +79656337764. ФИО: пусто.	Некорректный период бронирования, превышено рекомендуемое количество людей, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.25
26	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 15. Телефон: +79656337764. ФИО: пусто.	Некорректный период бронирования, сектор занят на выбранный период, превышено рекомендуемое количество людей, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.26
27	Сектор: свободен Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 8 ФИО/телефон: пусто.	Превышено рекомендуемое количество людей, не заполнены контактные данные клиента, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.27
28	Сектор: занят Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 8. Телефон: пусто. ФИО: Иванов Иван Иванович.	Сектор занят на выбранный период, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.28
29	Сектор: занят Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8. Телефон: +79656337764. ФИО: пусто.	Некорректный период бронирования, сектор занят на выбранный период, не заполнены контактные данные клиента.	Тест пройден. Результат приведен на рисунке Е.29
30	Сектор: свободен Заезд: 02.04.2026, Выезд: 01.04.2026 Людей: 8 ФИО+телефон пусто.	Некорректный период бронирования, не заполнены контактные данные клиента, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.30

Окончание таблицы Д

№	Входные данные	Ожидаемый результат	Фактический результат
31	Сектор: занят. Заезд: 05.04.2026, Выезд: 10.04.2026 Людей: 15. Телефон: +79656337764. ФИО: пусто.	Сектор занят на выбранный период, превышено рекомендуемое количество людей, не заполнено имя клиента.	Тест пройден. Результат приведен на рисунке Е.31
32	Сектор: свободен. Заезд: 05.04.2026, Выезд: 10.04.2026. Людей: 3. ФИО: Ермолаев И.О., Тел.: +79656337764.	С6 – бронирование создано успешно (статус «Ожидает подтверждения»).	Тест пройден. Результат приведен на рисунке Е.32

Приложение Е
(рекомендуемое)
Результаты функционального тестирования

КРАСНАЯ СТОРОНА
Сектор 1K

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Вместимость
до 3 чел

Глубина
2.1 м

Подлодов
3

Беседка
Нет

Текущая сумма

2 200 Р

В итог уже включён разовый тариф бронирования сектора.

Контактные данные

Имя *

Имя

Укажите имя

Фамилия

Фамилия

Отчество

Отчество

Телефон *

+7 () - - -

Укажите телефон

Время прибытия *

09:00

Период пребывания

Дата заезда *

25.06.2026

Дата выезда *

23.06.2026

Дата выезда не может быть раньше даты заезда

Сектор уже занят на эти даты.
Подтверждённая брони: 23 июня - 26 июня.

Гости и тарифы

Выбрано 4 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак
800 Р/день

Женщина-рыбачка
500 Р/день

Отдыхающий мужчина
200 Р/день

Итог к оплате

2 200 Р

Разовый тариф брони: 200 Р, аренда товаров уже учтена за 1 дн.

Заккрыть

Сектор занят

Рисунок Е.1 – Результат тест-кейса №1

КРАСНАЯ СТОРОНА
Сектор 1K

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Имя

Укажите имя

Фамилия

Фамилия

Отчество

Отчество

Телефон *

+79656337764

Время прибытия *

09:00

Период пребывания

Дата заезда *

27.06.2026

Дата выезда *

28.06.2026

Сектор свободен на выбранные даты.
Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 4 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак
800 Р/день

Женщина-рыбачка
500 Р/день

Отдыхающий мужчина
200 Р/день

Итог к оплате

2 200 Р

Разовый тариф брони: 200 Р, аренда товаров уже учтена за 1 дн.

Заккрыть

Подтвердить бронь

Рисунок Е.2 – Результат тест-кейса №2

КРАСНАЯ СТОРОНА

Сектор 1К

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Илья

Фамилия

Фамилия

Отчество

Отчество

Телефон *

+7 () - - - -

Время прибытия *

09:00

Укажите телефон

Период пребывания

Дата заезда *

27.06.2026

Дата выезда *

28.06.2026

✓ Сектор свободен на выбранные даты.

Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 4 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак

800 Р/день

-

0

+

Женщина-рыбачка

500 Р/день

-

4

+

Отдыхающий мужчина

200 Р/день

-

0

+

Вместимость

до 3 чел

Глубина

2.1 м

Подходов

3

Беседка

Нет

Текущая сумма

2 200 Р

В итог уже включён разовый тариф бронирования сектора.

Итог к оплате

2 200 Р

Разовый тариф брони: 200 Р, аренда товаров уже учтена за 1 дн.

Заккрыть

Подтвердить бронь

Рисунок Е.3 – Результат тест-кейса №3

ЖЕЛТАЯ СТОРОНА

Сектор 1Ж

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Илья

Фамилия

Фамилия

Отчество

Отчество

Телефон *

+79656337764

Время прибытия *

10:30

Период пребывания

Дата заезда *

06.06.2026

Дата выезда *

07.06.2026

⚠ Сектор уже занят на эти даты.

Подтверждённая брони: 6 июня - 7 июня.

Гости и тарифы

Выбрано 2 из 5 допустимых гостей для этого сектора.

Мужчина-рыбак

800 Р/день

-

0

+

Женщина-рыбачка

500 Р/день

-

0

+

Отдыхающий мужчина

200 Р/день

-

0

+

Вместимость

до 5 чел

Глубина

2.5 м

Подходов

4

Беседка

Есть

Текущая сумма

1 200 Р

В итог уже включён разовый тариф бронирования сектора.

Итог к оплате

1 200 Р

Разовый тариф брони: 200 Р, аренда товаров уже учтена за 1 дн.

Заккрыть

Сектор занят

Рисунок Е.4 – Результат тест-кейса №4

Гости и тарифы

Выбрано 4 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак 800 Р/день − 0 +	Женщина-рыбачка 500 Р/день − 4 +	Отдыхающий мужчина 200 Р/день − 0 +
Отдыхающая женщина 0 Р/день − 0 +	Ребёнок 0 Р/день − 0 +	

❗ Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человек, а допустимо не более 3.

Рисунок Е.5 – Результат тест-кейса №5

КРАСНАЯ СТОРОНА

Сектор 1K



Вместимость
до 3 чел

Глубина
2.1 м

Подходов
3

Беседка
Нет

Текущая сумма

2 200 Р

В итог уже включён разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя * Илья

Фамилия * Фамилия

Отчество * Отчество

Телефон * +79656337764

Время прибытия * 09:00

Период пребывания

Дата заезда * 29.06.2026

Дата выезда * 28.06.2026

Дата выезда не может быть раньше даты заезда

✓ Сектор свободен на выбранные даты.
Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 4 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак 800 Р/день − 0 +	Женщина-рыбачка 500 Р/день − 4 +	Отдыхающий мужчина 200 Р/день − 0 +
---	---	--

Итог к оплате

2 200 Р

Разовый тариф брони: 200 Р, аренда товаров уже учтена за 1 дн.

Заккрыть

Подтвердить бронь

Рисунок Е.6 – Результат тест-кейса №6

КРАСНАЯ СТОРОНА
Сектор 1К

Перейти к калькулятору

Вместимость
до 3 чел

Глубина
2.1 м

Подготово
3

Беседка
Нет

Текущая сумма

1 700 ₽

В итог уже включён разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Имя

Укажите имя

Фамилия

Фамилия

Отчество

Отчество

Телефон *

+7 () - - - -

Укажите телефон

Время прибытия *

09:00

Период пребывания

Дата заезда *

29.06.2026

Дата выезда *

30.06.2026

✓ Сектор свободен на выбранные даты.
Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 3 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак
800 ₽/день

Женщина-рыбачка
500 ₽/день

Отдыхающий мужчина
200 ₽/день

Итог к оплате

1 700 ₽

Разовый тариф брони: 200 ₽, аренда товаров уже учтена за 1 дн.

Заккрыть

Подтвердить бронь

Рисунок Е.7 – Результат тест-кейса №7

КРАСНАЯ СТОРОНА
Сектор 1К

Перейти к калькулятору

Вместимость
до 3 чел

Глубина
2.1 м

Подготово
3

Беседка
Нет

Текущая сумма

1 700 ₽

В итог уже включён разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Имя

Укажите имя

Фамилия

Фамилия

Отчество

Отчество

Телефон *

+7965637764

Время прибытия *

09:00

Период пребывания

Дата заезда *

29.06.2026

Дата выезда *

27.06.2026

Дата выезда не может быть раньше даты заезда

✓ Сектор свободен на выбранные даты.
Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 3 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак
800 ₽/день

Женщина-рыбачка
500 ₽/день

Отдыхающий мужчина
200 ₽/день

Итог к оплате

1 700 ₽

Разовый тариф брони: 200 ₽, аренда товаров уже учтена за 1 дн.

Заккрыть

Подтвердить бронь

Рисунок Е.8 – Результат тест-кейса №8

ЖЕЛТАЯ СТОРОНА

Сектор 1Ж

Перейти к калькулятору

Вместимость

до 5 чел

Глубина

2.5 м

Подводное

4

Беседка

Есть

Текущая сумма

6 200 Р

В итог уже включён разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Имя

Фамилия

Фамилия

Отчество

Отчество

Укажите имя

Телефон *

+79656337764

Время прибытия *

10:30

Период пребывания

Дата заезда *

08.06.2026

Дата выезда *

14.06.2026

Сектор уже занят на эти даты.
Подтвержденная брони: 9 июня - 11 июня.

Гости и тарифы

Выбрано 2 из 5 допустимых гостей для этого сектора.

Мужчина-рыбак

Женщина-рыбачка

Отдыхающий мужчина

Итого к оплате

6 200 Р

Разовый тариф брони: 200 Р, аренда товаров уже учтена за 6 дн.

Заккрыть

Сектор занят

Рисунок Е.9 – Результат тест-кейса №9

КРАСНАЯ СТОРОНА

Сектор 1К

Перейти к калькулятору

Вместимость

до 3 чел

Глубина

2.1 м

Подводное

3

Беседка

Нет

Текущая сумма

4 800 Р

В итог уже включён разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Имя

Фамилия

Фамилия

Отчество

Отчество

Укажите имя

Телефон *

+79656337764

Время прибытия *

09:00

Период пребывания

Дата заезда *

29.06.2026

Дата выезда *

01.07.2026

Сектор свободен на выбранные даты.
Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 4 из 3 допустимых гостей для этого сектора.

Мужчина-рыбак

800 Р/день

-

1

+

Женщина-рыбачка

500 Р/день

-

3

+

Отдыхающий мужчина

200 Р/день

-

0

+

Отдыхающая женщина

0 Р/день

-

0

+

Ребёнок

0 Р/день

-

0

+

Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человека, а допустимо не более 3.

Рисунок Е.10 – Результат тест-кейса №10

ЖЕЛТАЯ СТОРОНА
Сектор 1К

Перейти к калькулятору

Изображение

Вместимость
до 3 чел

Глубина
2.1 м

Полосы
3

Безопасность
Нет

Текущая сумма

1 700 Р

В этот уже включен разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя * Илья

Фамилия * Фамилия

Отчество * Отчество

Телефон * +7 () - - - - -

Время прибытия * 09:00

Укажите телефон

Период пребывания

Дата заезда * 29.06.2026

Дата выезда * 27.06.2026

Дата выезда не может быть раньше даты заезда

Сектор свободен на выбранные даты.

Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 3 из 3 доступных гостей для этого сектора.

Мужчина-рыбак 800 руб/сут

Женщина-рыбачка 500 руб/сут

Отдыхающий мужчина 200 руб/сут

Рисунок Е.11 – Результат тест-кейса №11

ЖЕЛТАЯ СТОРОНА
Сектор 1Ж

Перейти к калькулятору

Изображение

Вместимость
до 5 чел

Глубина
2.5 м

Полосы
4

Безопасность
Есть

Текущая сумма

3 200 Р

В этот уже включен разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя * Илья

Фамилия * Фамилия

Отчество * Отчество

Телефон * +7 () - - - - -

Время прибытия * 10:30

Укажите телефон

Период пребывания

Дата заезда * 06.06.2026

Дата выезда * 09.06.2026

Сектор уже занят на эти даты.

Подтвержденная брони с июня - 7 июля.

Гости и тарифы

Выбрано 2 из 3 доступных гостей для этого сектора.

Мужчина-рыбак

Женщина-рыбачка

Отдыхающий мужчина

Итого к оплате

3 200 Р

Разовый разовый тариф 200 Р, аренда таракана уже учтена на 3 дн.

Закрыть

Сектор занят

Рисунок Е.12 – Результат тест-кейса №12

ЖЕЛТАЯ СТОРОНА
Сектор 1К

Перейти к калькулятору

Изображение

Вместимость
до 3 чел

Глубина
2.1 м

Полосы
3

Безопасность
Нет

Текущая сумма

1 700 Р

В этот уже включен разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя * Илья

Фамилия * Фамилия

Отчество * Отчество

Телефон * +7 () - - - - -

Время прибытия * 09:00

Укажите телефон

Период пребывания

Дата заезда * 29.06.2026

Дата выезда * 30.06.2026

Сектор свободен на выбранные даты.

Можно переходить к подтверждению брони.

Гости и тарифы

Выбрано 4 из 5 доступных гостей для этого сектора.

Мужчина-рыбак 800 руб/сут

Женщина-рыбачка 500 руб/сут

Отдыхающий мужчина 200 руб/сут

Отдыхающая женщина 100 руб/сут

Ребенок 100 руб/сут

Вы привнесете своих гостей для этого сектора.

Сейчас выбрано 4 человека, а доступно не более 3.

Рисунок Е.13 – Результат тест-кейса №13

ЖЕЛТАЯ СТОРОНА

Сектор 1Ж

Видимость
до 5 ч/д

Глубина
2.5 м

Температура
4

Берега
Есть

Текущая сумма
1 200 Р

3 отъездов уже включен разовый тариф бронирования сектора.

Итого к оплате
1 200 Р

Акция: при бронировании 2000 Р, каждая поездка уже стоит на 1 руб.

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Контактные данные

Имя *
Илья

Фамилия
Фамилия

Отчество
Отчество

Телефон *
+79656337764

Время прибытия *
10:30

Период пребывания

Дата заезда *
07.06.2026

Дата выезда *
06.06.2026

Дата выезда не может быть раньше даты заезда

Сектор уже занят на эти даты.
Подтвержденная бронь: 6 июня - 7 июня.

Гости и тарифы

Выбрано 2 из 3 доступных гостей для этого сектора.

Мужчина-рыбак
800 Р/день

Женщина-рыбачка
500 Р/день

Отдыхающий мужчина
200 Р/день

Закрыть

Сектор занят

Рисунок Е.14 – Результат тест-кейса №14

КРАСНАЯ СТОРОНА

Сектор 1К

Видимость
до 3 ч/д

Глубина
2.1 м

Температура
3

Берега
Нет

Текущая сумма
1 700 Р

6 отъездов уже включен разовый тариф бронирования сектора.

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Контактные данные

Имя *
Илья

Фамилия
Фамилия

Отчество
Отчество

Телефон *
+79656337764

Время прибытия *
09:00

Период пребывания

Дата заезда *
29.06.2026

Дата выезда *
27.06.2026

Дата выезда не может быть раньше даты заезда

Сектор свободен на выбранные даты.
Можно переводить и подтверждать бронь.

Гости и тарифы

Выбрано 4 из 3 доступных гостей для этого сектора.

Мужчина-рыбак
800 Р/день

Женщина-рыбачка
500 Р/день

Отдыхающий мужчина
200 Р/день

Отдыхающая женщина
0 Р/день

Ребенок
0 Р/день

Рисунок Е.15 – Результат тест-кейса №15

Сектор уже занят на эти даты.
Подтвержденная бронь: 6 июня - 7 июня.

Гости и тарифы

Выбрано 11 из 5 доступных гостей для этого сектора.

Мужчина-рыбак
800 Р/день

Женщина-рыбачка
500 Р/день

Отдыхающий мужчина
200 Р/день

Отдыхающая женщина
0 Р/день

Ребенок
0 Р/день

Вы превысили лимит гостей для этого сектора.
Сейчас указано 11 человек, а доступно не более 5.

Рисунок Е.16 – Результат тест-кейса №16

ЖЕЛТАЯ СТОРОНА
Сектор 1К

Оформление брони | Калькулятор суммы

Контактные данные:
Имя * Имя Фамилия Отчество
Иванов Иван Иванович
Телефон * +7 (911) 123-4567
Время прибытия * 09:00

Период пребывания:
Дата заезда * 29.06.2026
Дата выезда * 30.06.2026

✓ Сектор свободен на выбранные даты.
Можно перенести в подтвержденную бронь.

Гости и тарифы
Выбрано 4 из 5 доступных гостей для этого сектора.

Мужчина рыбак 800 руб/сут	Женщина рыбака 500 руб/сут	Однородный мужчина 200 руб/сут
0	3	0
Однородная женщина 100 руб/сут	Ребенок 100 руб/сут	
1	0	

Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человека, а доступно не более 3.

Текущая сумма: **1 700 P**
В эту сумму включены расходы на проживание и питание.

Рисунок Е.17 – Результат тест-кейса №17

ЖЕЛТАЯ СТОРОНА
Сектор 1К

Оформление брони | Калькулятор суммы

Контактные данные:
Имя * Имя Фамилия Отчество
Иванов Иван Иванович
Телефон * +7 (911) 123-4567
Время прибытия * 10:30

Период пребывания:
Дата заезда * 09.06.2026
Дата выезда * 18.06.2026

✗ Сектор уже занят на эти даты.
Подтвержденная бронь: 9 июня - 11 июня.

Гости и тарифы
Выбрано 3 из 5 доступных гостей для этого сектора.

Мужчина рыбак 800 руб/сут	Женщина рыбака 500 руб/сут	Однородный мужчина 200 руб/сут
Однородная женщина 100 руб/сут	Ребенок 100 руб/сут	

Закрывать | Сектор занят

Рисунок Е.18 – Результат тест-кейса №18

ЖЕЛТАЯ СТОРОНА
Сектор 1К

Оформление брони | Калькулятор суммы

Контактные данные:
Имя * Имя Фамилия Отчество
Иванов Иван Иванович
Телефон * +7985637756
Время прибытия * 09:00

Период пребывания:
Дата заезда * 29.06.2026
Дата выезда * 28.06.2026

✓ Сектор свободен на выбранные даты.
Можно перенести в подтвержденную бронь.

Гости и тарифы
Выбрано 4 из 5 доступных гостей для этого сектора.

Мужчина рыбак 800 руб/сут	Женщина рыбака 500 руб/сут	Однородный мужчина 200 руб/сут
0	3	0
Однородная женщина 100 руб/сут	Ребенок 100 руб/сут	
1	0	

Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человека, а доступно не более 3.

Текущая сумма: **1 700 P**
В эту сумму включены расходы на проживание и питание.

Рисунок Е.19 – Результат тест-кейса №19

[illegible][illegible]

Рисунок Е.22 – Результат тест-кейса №22

КРАСИВАЯ СТОРОНА
Сектор 1K

Перейти к калькулятору

Оформление брони Калькулятор суммы

Контактные данные

Имя * Фамилия Отчество

Телефон * Укажите телефон Время прибытия *

Период пребывания

Дата заезда * Дата выезда *
Дата выезда не может быть раньше даты заезда

✓ Сектор свободен на выбранные даты.
Можно переходить к подтверждению брони.

Гости и тарифы
Выбрано 4 из 3 доступных гостей для этого сектора.

Мужчина-рыбак 800 ₽/день - 0 +	Женщина-рыбачка 500 ₽/день - 3 +	Отдыхающий мужчина 200 ₽/день - 0 +
Отдыхающая женщина 0 ₽/день - 1 +	Ребенок 0 ₽/день - 0 +	

❗ Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человека, а доступно не более 3.

Топикаты и апгрейды

Итого к оплате
1 700 Р
Расовый тариф: Бронь: 200 Р, аренда товаров: уже учтена на 1 дн.

Закрыть Подтвердить бронь

Рисунок Е.23 – Результат тест-кейса №23

КРАСИВАЯ СТОРОНА
Сектор 1K

Перейти к калькулятору

Оформление брони Калькулятор суммы

Контактные данные

Имя * Фамилия Отчество

Телефон * Укажите телефон Время прибытия *

Период пребывания

Дата заезда * Дата выезда *

❗ Сектор уже занят на эти даты.
Подтвержденная бронь: 23 июня - 26 июня.

Гости и тарифы
Выбрано 4 из 3 доступных гостей для этого сектора.

Мужчина-рыбак 800 ₽/день - 0 +	Женщина-рыбачка 500 ₽/день - 3 +	Отдыхающий мужчина 200 ₽/день - 0 +
Отдыхающая женщина 0 ₽/день - 1 +	Ребенок 0 ₽/день - 0 +	

❗ Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человека, а доступно не более 3.

Рисунок Е.24 – Результат тест-кейса №24

КРАСНАЯ СТОРОНА
Сектор 1К

Перейти к калькулятору

Вместимость
до 3 чел

Глубина
2.1 м

Подогрев
3

Беспровод
Нет

Текущая сумма
1 700 Р

В итог уже включён разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя *
Илья

Фамилия
Фамилия

Отчество
Отчество

Телефон *
+79656337764

Время прибытия *
09:00

Период пребывания

Дата заезда *
24.06.2026

Дата выезда *
23.06.2026

Дата выезда не может быть раньше даты заезда

Сектор уже занят на эти даты.
Подтверждённая бронь: 23 июня - 26 июня.

Гости и тарифы

Выбрано 4 из 3 доступных гостей для этого сектора.

Мужчина-рыбак
800 Р/день

0

Женщина-рыбачка
500 Р/день

3

Отдыхающий мужчина
200 Р/день

0

Отдыхающая женщина
0 Р/день

1

Ребёнок
0 Р/день

0

Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человека, а доступно не более 3.

Рисунок Е.25 – Результат тест-кейса №25

КРАСНАЯ СТОРОНА
Сектор 1К

Перейти к калькулятору

Вместимость
до 3 чел

Глубина
2.1 м

Подогрев
3

Беспровод
Нет

Текущая сумма
1 700 Р

В итог уже включён разовый тариф бронирования сектора.

Оформление брони

Калькулятор суммы

Контактные данные

Имя *
Илья

Фамилия
Фамилия

Отчество
Отчество

Телефон *
+79656337764

Время прибытия *
09:00

Период пребывания

Дата заезда *
24.06.2026

Дата выезда *
23.06.2026

Дата выезда не может быть раньше даты заезда

Сектор уже занят на эти даты.
Подтверждённая бронь: 23 июня - 26 июня.

Гости и тарифы

Выбрано 4 из 3 доступных гостей для этого сектора.

Мужчина-рыбак
800 Р/день

0

Женщина-рыбачка
500 Р/день

3

Отдыхающий мужчина
200 Р/день

0

Отдыхающая женщина
0 Р/день

1

Ребёнок
0 Р/день

0

Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человека, а доступно не более 3.

Рисунок Е.26 – Результат тест-кейса №26

КРАСНАЯ СТОРОНА

Сектор 1К

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Контактные данные

Имя *
Илья

Фамилия
Фамилия

Отчество
Отчество

Телефон *
+7 () - - - -

Время прибытия *
09:00

Период пребывания

Дата заезда *
24.06.2026

Дата выезда *
23.06.2026

Гости и тарифы

Мужчина-рыбак
800 ₽/день

Женщина-рыбачка
500 ₽/день

Отдыхающий мужчина
200 ₽/день

Отдыхающая женщина
0 ₽/день

Ребёнок
0 ₽/день

Вместимость
до 3 чел

Глубина
2.1 м

Платформа
3

Бетонка
Нет

Текущая сумма
1 700 ₽

Сектор уже занят на эти даты.
Подтверждённая броня: 23 июня - 26 июня.

Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человек, а доступно не более 3.

Рисунок Е.27 – Результат тест-кейса №27

КРАСНАЯ СТОРОНА

Сектор 1К

Перейти к калькулятору

Оформление брони

Калькулятор суммы

Контактные данные

Имя *
Илья

Фамилия
Фамилия

Отчество
Отчество

Телефон *
+7 () - - - -

Время прибытия *
09:00

Период пребывания

Дата заезда *
24.06.2026

Дата выезда *
26.06.2026

Гости и тарифы

Мужчина-рыбак
800 ₽/день

Женщина-рыбачка
500 ₽/день

Отдыхающий мужчина
200 ₽/день

Отдыхающая женщина
0 ₽/день

Ребёнок
0 ₽/день

Вместимость
до 3 чел

Глубина
2.1 м

Платформа
3

Бетонка
Нет

Текущая сумма
3 200 ₽

Сектор уже занят на эти даты.
Подтверждённая броня: 23 июня - 26 июня.

Вы превысили лимит гостей для этого сектора.
Сейчас указано 4 человек, а доступно не более 3.

Рисунок Е.28 – Результат тест-кейса №28

КРАСНАЯ СТОРОНА

Сектор 1К

Перейти в календарь

✕



Оформление брони

Калькулятор суммы

Контактные данные

Имя *

Фамилия

Отчество

Укажите имя

Укажите фамилию

Укажите отчество

Телефон *

Время прибытия *

Укажите телефон

09:00

Период пребывания

Дата заезда *

Дата выезда *

26.06.2026

24.06.2026

Дата выезда не может быть раньше даты заезда

Сектор уже занят на эти даты.

Подтверждена бронь: 23 июня - 26 июня.

Текущая сумма

1 700 ₽

В итог уже включены расширенный тариф бронирования сектора.

Гости и тарифы

Выбрано 3 из 3 доступных гостей для этого сектора.

Мужчина-рыбак

800 ₽/день

0

Женщина-рыбачка

500 ₽/день

3

Отдыхающий мужчина

200 ₽/день

0

Отдыхающая женщина

0 ₽/день

0

Ребёнок

0 ₽/день

0

Товары и аренда

Выберите товары для продажи или аренды. Аренда считается за каждый день пребывания.

Продажа

Рисунок Е.30 – Результат тест-кейса №30

КРАСНАЯ СТОРОНА
Сектор 1К

Перейти к калькулятору

Оформление брони Калькулятор суммы

Контактные данные

Имя * Фамилия Отчество

Укажите имя

Телефон * Время прибытия *

Укажите телефон

Период пребывания

Дата заезда * Дата выезда *

Дата выезда не может быть раньше даты заезда

✓ Сектор свободен на выбранные даты.
Можно переходить к подтверждению брони.

Гости и тарифы
Выбрано 3 из 3 доступных гостей для этого сектора.

<p>Мужчина-рыбак 800 Р/день</p> <p>— 0 +</p>	<p>Женщина-рыбачка 500 Р/день</p> <p>— 3 +</p>	<p>Отдыхающий мужчина 200 Р/день</p> <p>— 0 +</p>
<p>Отдыхающая женщина 0 Р/день</p> <p>— 0 +</p>	<p>Ребёнок 0 Р/день</p> <p>— 0 +</p>	

Текущая сумма
1 700 Р
В итог уже включён разовый тариф бронирования сектора.

Рисунок Е.31 – Результат тест-кейса №31

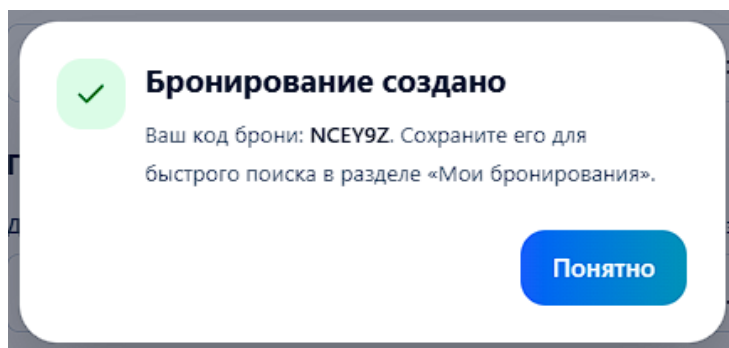


Рисунок Е.32 – Результат тест-кейса №32

Приложение Ж
(рекомендуемое)
Акт о внедрении

**ИНДИВИДУАЛЬНЫЙ ПРЕДПРИНИМАТЕЛЬ
ГРИШИНА СВЕТЛАНА НИКОЛАЕВНА
ИНН 583401990443
ОГРН 317583500021289**

440061, г. Пенза, ул. Луначарского, д. 7а, кв. 3
тел. +7 902 343-51-01, E-mail: grishinasvetlana1973@gmail.com
Р/с: 40802810429170003445 в ФИЛИАЛ "НИЖЕГОРОДСКИЙ" АО "АЛЬФА-БАНК"
г. Нижний Новгород БИК: 042202824 К/с: 30101810200000000824

21.05.2026г.

Акт о внедрении

Результаты теоретических и практических исследований, приведённые в выпускной квалификационной работе Ермолаева Ильи Олеговича на тему «Фронтенд-компонент информационной системы для автоматизации бронирования и учета услуг рыболовного хозяйства», внедрены в деятельность ИП Гришиной Светланы Николаевны, являющейся владельцем рыболовного хозяйства «Пруд "Орловский"», местоположение: на ручье без названия в 1,5 км на северо-запад от с. Кучки, Пензенский район Пензенской области.

Индивидуальный Предприниматель _____



Гришина С.Н./