

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(наименование профиля, специализации)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему: «Разработка веб-приложения для системы бронирования
номеров в гостинице»

Обучающийся

Н.С. Чекин

(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н., доцент, Д.Г. Токарев

(ученая степень, звание, И.О. Фамилия)

Тольятти 2025

Аннотация

Тема: Разработка веб-приложения для системы бронирования номеров в гостинице.

Ключевые слова: информационная система, веб приложение, бронирование номеров, программное обеспечение, гостиница.

Целью выпускной квалификационной работы является разработка веб-приложения для автоматизации системы бронирования номеров в гостинице, обеспечивающего удобство и эффективность управления процессами бронирования.

Предметом исследования является процесс автоматизации бронирования номеров в гостинице посредством разработки веб-приложения.

Объектом исследования является система бронирования номеров в гостинице как комплекс взаимосвязанных процессов и данных.

В работе проведен анализ деятельности гостиницы и выявлены основные бизнес-процессы, связанные с бронированием номеров, изучены лучшие практики автоматизации процессов бронирования номеров в гостиницах, разработаны модели информационной системы, выбраны оптимальные технологии для реализации веб-приложения, разработано программное обеспечение, реализующее функциональность системы бронирования, проведено тестирование разработанного программного обеспечения.

Выпускная квалификационная работа состоит из 44 страниц, 23 рисунков, 4 таблиц, 20 источников.

Содержание

Введение.....	4
1 Анализ объекта исследования.....	7
1.1 Описание существующей системы бронирования номеров.....	7
1.2 Анализ лучших практик автоматизации процессов бронирования номеров.....	10
1.3 Разработка требований	15
2 Проектирование программного обеспечения системы бронирования номеров.....	17
2.1 Логическое проектирование программного обеспечения системы бронирования номеров в гостинице.....	17
2.2 Проектирование базы данных веб приложения бронирования номеров в гостинице.....	26
3 Разработка программного обеспечения системы бронирования номеров...	29
3.1 Выбор технологий реализации программного обеспечения	29
3.2 Реализация информационной модели программного обеспечения.....	29
3.3 Разработка программного обеспечения.....	31
3.4 Тестирование программного обеспечения	36
Заключение	42
Список используемой литературы и используемых источников.....	43

Введение

В современном мире гостиничный бизнес является одной из наиболее динамично развивающихся отраслей. Конкуренция в этой сфере постоянно растет, и успех гостиницы во многом зависит от эффективности управления ресурсами и качественного обслуживания клиентов. Одним из ключевых аспектов обеспечения конкурентоспособности является автоматизация процессов бронирования, позволяющая оптимизировать загрузку номерного фонда, минимизировать ошибки и повысить лояльность клиентов. В связи с этим, разработка эффективной и удобной в использовании системы бронирования номеров является актуальной задачей, направленной на повышение эффективности работы гостиниц.

Актуальность выбранной темы обусловлена необходимостью оптимизации бизнес-процессов в гостиничном бизнесе посредством внедрения современных информационных технологий. Ручной ввод данных, сложность отслеживания доступности номеров, длительное время обработки запросов клиентов - все это приводит к снижению эффективности работы и потере потенциальных клиентов. Автоматизированная система бронирования позволяет избежать этих проблем, обеспечивая оперативный и точный учет информации о номерном фонде, бронированиях и клиентах.

Существует множество разработанных систем бронирования номеров, как проприетарных, так и с открытым исходным кодом. Однако, не все они в полной мере отвечают требованиям конкретной гостиницы. Некоторые системы могут быть слишком сложными и дорогими в использовании, другие - недостаточно функциональными или не адаптированными к особенностям бизнес-процессов. Поэтому, разработка собственной системы бронирования, учитывающей специфику конкретной гостиницы, может оказаться более эффективным решением.

Объектом исследования является система бронирования номеров в гостинице как комплекс взаимосвязанных процессов и данных.

Предметом исследования является процесс автоматизации бронирования номеров в гостинице посредством разработки веб-приложения.

Целью данной работы является разработка веб-приложения для автоматизации системы бронирования номеров в гостинице, обеспечивающего удобство и эффективность управления процессами бронирования.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ деятельности гостиницы и выявить основные бизнес-процессы, связанные с бронированием номеров;
- изучить лучшие практики автоматизации процессов бронирования номеров в гостиницах;
- разработать концептуальную и логическую модели разрабатываемой системы;
- выбрать оптимальные технологии для реализации веб-приложения;
- разработать программное обеспечение, реализующее функциональность системы бронирования;
- провести тестирование разработанного программного обеспечения.

В процессе исследования были использованы следующие методики: анализ литературы, моделирование бизнес-процессов, проектирование баз данных, объектно-ориентированное программирование, тестирование программного обеспечения.

Практическая значимость работы заключается в возможности внедрения разработанного веб-приложения в реальную гостиницу, что позволит повысить эффективность работы, снизить затраты и улучшить качество обслуживания клиентов.

В первом разделе проведен анализ деятельности гостиницы, выявлены основные бизнес-процессы, связанные с бронированием номеров, а также рассмотрены лучшие практики автоматизации данных процессов, существующие на данный момент.

Во втором разделе описаны этапы проектирования программного обеспечения, включая проектирование базы данных, пользовательского интерфейса и архитектуры системы. Разработана информационная модель проекта.

В третьем разделе представлены этапы реализации веб-приложения, описан выбор технологий реализации, реализация информационной модели, разработка программного обеспечения и его тестирование.

В заключении представлены основные выводы по результатам работы, подведены итоги, оценена степень достижения поставленных целей и задач, а также предложены направления для дальнейшего развития системы.

1 Анализ объекта исследования

1.1 Описание существующей системы бронирования номеров

Гостиница предоставляет временное проживание, питание и другие сопутствующие услуги для гостей, путешественников и деловых людей. Деятельность гостиницы состоит из множества взаимосвязанных процессов, направленных на обеспечение комфортного и качественного пребывания клиентов.

Организационная структура гостиницы показана на рисунке 1.



Рисунок 1 – Организационная структура гостиницы

Деятельность гостиницы состоит в предоставлении номерного фонда, обслуживании номеров, организации питания гостей и других связанных процессов.

Предоставление номерного включает предоставление различных типов номеров, оборудованных всем необходимым для комфортного проживания. Состояние номеров напрямую влияет на удовлетворенность гостей.

Обслуживание номерного фонда состоит в поддержании чистоты и порядка в номерах и общих зонах гостиницы и включает в себя ежедневную

уборку номеров, смену постельного белья и полотенец, вынос мусора, дезинфекцию, а также текущий ремонт и обслуживание оборудования.

Многие гостиницы предлагают питание своим гостям. Это может быть завтрак, включенный в стоимость проживания, обеды и ужины в ресторане или кафе при гостинице, обслуживание в номерах, а также организация банкетов и конференций.

Неотъемлемой составляющей гостиницы является круглосуточная стойка регистрации (ресепшн), сотрудники которой занимаются регистрацией и выселением гостей, информированием об услугах, приемом и обработкой входящих звонков, обработкой жалоб и предложений, организацией трансферов и заказом такси.

Служба безопасности гостиницы обеспечивает круглосуточную охрану, установку систем видеонаблюдения, пожарную безопасность, контроль доступа в номера.

Бронирование номеров – это один из ключевых и наиболее важных процессов в деятельности гостиницы. Это процесс резервирования определенного номера в гостинице на определенный период времени для конкретного клиента. Бронирование является первым этапом взаимодействия гостиницы с гостем и во многом определяет дальнейшее впечатление клиента.

Эффективная система бронирования позволяет гостинице максимально загрузить свой номерной фонд, избегая простоев и увеличивая прибыль.

Информация о бронированиях позволяет гостинице планировать необходимые ресурсы (персонал, продукты питания, расходные материалы) и обеспечивать своевременное предоставление услуг.

Система бронирования может использоваться для динамического ценообразования, позволяющего максимизировать доходность в зависимости от спроса.

Предварительное бронирование позволяет гостинице заранее узнать о потребностях гостей и подготовиться к их прибытию, что способствует повышению уровня сервиса.

Информация о бронированиях может использоваться для анализа клиентской базы и проведения маркетинговых кампаний.

Основные этапы процесса бронирования показаны на рисунке 2.



Рисунок 2 - Этапы Процесса Бронирования

Бронирование выполняется в следующем порядке. Клиент связывается с гостиницей (по телефону, через веб-сайт, по электронной почте, через онлайн-системы бронирования) и запрашивает бронирование номера на определенные даты. Сотрудник гостиницы (или система бронирования) проверяет наличие свободных номеров нужной категории на запрошенные даты. Клиенту предоставляется информация о доступных номерах, ценах и

условиях бронирования. Если клиент согласен с условиями, бронирование подтверждается. Информация о бронировании заносится в информационную систему гостиницы. Перед прибытием гостя бронирование проверяется, и номер подготавливается к заселению.

Веб-приложение во многом облегчит процедуру бронирования как со стороны клиента так и со стороны работников гостиницы.

1.2 Анализ лучших практик автоматизации процессов бронирования номеров

Во-первых, существует достаточное количество онлайн-сервисов бронирования отелей (например, Booking.com [1], Expedia [2], Hotels.com [3], Airbnb [4], Ostrovok [5]).

Во-вторых бронирование может проводиться с сайтов гостиниц, например : Изучите веб-приложения крупных гостиничных сетей (например, Marriott, Hilton, Hyatt). Они могут иметь особенности, связанные с управлением собственными ресурсами.

Также существуют отдельные нишевые решения, которые специализируются на определенном типе жилья (например, бутик-отели, апартаменты, хостелы) и ориентированы на определенную аудиторию.

Онлайн сервис бронирования Booking.com предлагает широкий спектр фильтров поиска, включая цену, удобства, рейтинг, тип жилья и тематику. Их система рекомендаций основана на истории просмотров и бронирований пользователя, что повышает вероятность нахождения подходящего варианта. Они используют A/B-тестирование для постоянного улучшения пользовательского интерфейса и процесса бронирования.

Онлайн сервис бронирования Airbnb в большей степени сконцентрирован на визуальном контенте, используя качественные фотографии и видео для демонстрации жилья. Сервис имеет мобильное

приложение, которое обладает интуитивно понятным интерфейсом и позволяет легко искать и бронировать жилье с мобильного телефона.

Сайт гостиничной сети Marriott [6] интегрирован с программой лояльности Marriott Bonvoy, предлагая участникам эксклюзивные цены и привилегии. Они предоставляют подробную информацию о каждом отеле, включая виртуальные туры и интерактивные карты.

Примером нишевых сервисов бронирования могут быть хостелы Hostelworld [7]. Они ориентированы на бюджетных путешественников, особенно молодежь и бэкпекеров. Предлагают широкий выбор хостелов по всему миру, с возможностью бронирования койко-мест в общих номерах или отдельных комнат. Акцент делается на социальной составляющей, с информацией о мероприятиях, организуемых хостелами, и возможностью общения с другими путешественниками. Система рейтингов и отзывов помогает выбрать хостел с подходящей атмосферой и удобствами. Часто предлагаются скидки и специальные предложения для длительного проживания.

Эко-отели и глэмпинг (Glamping Hub [8], Escobnb [9]) сосредоточены на устойчивом туризме и экологически чистых вариантах размещения. Предлагают широкий выбор эко-отелей, глэмпингов, фермерских домов и других уникальных мест, где можно насладиться природой и отдохнуть от городской суеты. Фильтры поиска включают экологические сертификаты, использование возобновляемых источников энергии, органическое питание и другие критерии устойчивости. Система рекомендаций ориентирована на тех, кто заботится об окружающей среде и ищет аутентичный опыт.

На основе анализа приведенных выше источников выделены ключевые аспекты автоматизации процессов бронирования.

Таблица 1 - Результаты анализа лучших практик автоматизации процессов бронирования номеров

Сервис/Платформа	Широта охвата	Персонализация	UX/UI	Визуальный контент	Мобильность	Специфичность	Уникальные особенности	Экологичность
Booking.com	Очень широкая	История просмотров/бронирований, рекомендации	Постоянное улучшение, широкий набор фильтров	Среднее	Есть, удобное приложение	Общая	А/В-тестирование для оптимизации, глобальный охват	Низкая
Airbnb	Широкая	Отзывы пользователей, система рекомендаций	Интуитивно понятный, акцент на визуализации	Высокое, акцент на фото/видео	Есть, удобное приложение	Альтернативное жилье (апартаменты, дома)	Уникальный опыт проживания, локальные хозяева	Средняя
Marriott (Сайт сети)	Средняя	Программа лояльности Marriott Bonvoy	Подробная информация об отелях, виртуальные туры, интерактивные карты	Высокое	Есть	Гостиничная сеть	Привилегии для участников программы лояльности	Низкая
Hostelworld/Hostels.com	Средняя	Отзывы и рейтинги, социальная составляющая	Специализированный интерфейс для хостелов	Среднее	Есть	Хостелы, бюджетное размещение	Социальная составляющая, общение с другими путешественниками, мероприятия, скидки на длительное проживание	Низкая
Glamping Hub/Ecobnb	Узкая	Фильтры по экологическим сертификатам, использованию возобновляемых источников энергии, органическому питанию	Специализированный интерфейс для эко-отелей и глэмпингов	Высокое	Есть	Эко-отели и глэмпинг	Устойчивый туризм, экологически чистые варианты размещения, уникальный опыт на природе	Очень высокая

Описание критериев сравнения.

Широта охвата - насколько широкий выбор жилья предлагает сервис.

Персонализация – насколько хорошо сервис учитывает предпочтения пользователя.

Удобство использования (UX/UI) – насколько интуитивно понятен и удобен интерфейс.

Визуальный контент – качество и важность визуального представления жилья.

Мобильность – удобство использования на мобильных устройствах.

Специфичность – ориентированность на определенный тип жилья или аудиторию.

Уникальные особенности – отличительные черты, выделяющие сервис среди других.

Экологичность – учет экологических факторов при выборе жилья.

Разработка собственного веб-приложения для бронирования является стратегическим решением, которое нуждается в предварительном анализе и обосновании относительно альтернативных вариантов, к которым относятся централизованные сервисы бронирования, бронирование по телефону, система бронирования встроенная в сайт гостиницы.

Централизованные сервисы бронирования (OTAs - Online Travel Agencies, например, Booking.com, Expedia) имеют ряд существенных недостатков. OTAs взимают значительные комиссии (15-30% и выше) за каждое бронирование, что существенно снижает прибыльность. OTAs владеют данными клиентов, а гостиница имеет ограниченный доступ к этой информации, что затрудняет создание клиентской базы и проведение целевых маркетинговых кампаний. OTAs часто диктуют правила представления гостиницы на своей платформе, что может ограничить возможности продвижения уникальных особенностей и преимуществ своего бренда. OTAs, стремясь к унификации, часто не позволяют гостиницам предлагать уникальные пакеты услуг и гибкие условия бронирования.

видимость гостиницы на OTAs зависит от их алгоритмов, которые могут меняться.

Недостатки системы бронирования по телефону очевидны. Это высокие трудозатраты, ограниченная доступность, неудобство для клиентов, сложность сбора и анализа данных, что ограничивает возможности для улучшения сервиса.

Недостатки встроенной системы бронирования состоят в следующем. Встроенные системы часто имеют ограниченный набор функций и возможностей кастомизации. Интеграция со сторонними системами (например, платежными шлюзами) может быть сложной и дорогостоящей. Функциональность системы бронирования ограничена возможностями платформы, на которой построен сайт гостиницы. Встроенные системы могут быть более уязвимы для хакерских атак. Сложно масштабировать систему бронирования интегрированную в сайт по мере роста гостиницы.

Результаты анализа альтернативных вариантов (таблица 2) свидетельствуют о целесообразности разработки собственного веб-приложения для бронирования.

Таблица 2 – Анализ альтернативных вариантов

Альтернатива	Недостатки	Преимущества собственного веб-приложения
Централизованные OTA	Высокие комиссии, потеря клиентской базы, контроль бренда	Снижение расходов, прямой доступ к клиентам, усиление бренда, гибкость
Бронирование по телефону	Трудозатраты, ограниченная доступность, неудобство для клиентов	Автоматизация, удобство, сбор данных
Встроенная на сайт	Ограниченная функциональность, зависимость от сайта, безопасность	Полный контроль, интеграция, безопасность, масштабируемость

При правильной реализации, собственное веб-приложение может стать ключевым конкурентным преимуществом гостиницы и обеспечить ее долгосрочный успех.

1.3 Разработка требований

Функциональные требования.

Бронирование номеров клиентами. Система должна позволять клиентам находить, просматривать информацию о номерах, бронировать, оплачивать и отменять бронирования.

Управление профилем клиента. Система должна позволять клиентам регистрироваться, просматривать и редактировать информацию о своем профиле, а также просматривать историю бронирований.

Управление номерным фондом. Система должна позволять менеджерам добавлять, редактировать и удалять информацию о номерах, а также управлять их ценами и доступностью.

Управление бронированиями (административная часть). Система должна позволять менеджерам просматривать, редактировать и подтверждать бронирования.

Отчетность. Система должна предоставлять менеджерам возможность формировать различные отчеты о загрузке отеля, доходах и других ключевых показателях.

Нефункциональные требования.

Производительность и надежность. Система должна быть быстрой, доступной и стабильной, обеспечивая бесперебойную работу для всех пользователей.

Безопасность данных. Система должна обеспечивать защиту персональных данных пользователей и информации о бронированиях от несанкционированного доступа.

Удобство использования (Usability). Система должна быть интуитивно понятной и удобной для использования как клиентами, так и менеджерами на различных устройствах.

Масштабируемость. Система должна быть способна выдерживать растущую нагрузку и количество пользователей.

Выводы по разделу. В первом разделе описана деятельность гостиницы, показана ее организационная структура. Описан процесс бронирования. Выполнен анализ лучших практик и альтернативных вариантов бронирования номеров в гостинице. Сформулированы требования к информационной системе. В целом проведенное обследование предметной области и обоснование проектного решения служит основой для проектирования информационной системы бронирования номеров в гостинице.

2 Проектирование программного обеспечения системы бронирования номеров

2.1 Логическое проектирование программного обеспечения системы бронирования номеров в гостинице

Начальным этапом проектирования систем является описание взаимодействия системы с окружающей средой и ее функциональности. Это выполнено на основе разработанных требований с использованием диаграммы вариантов использования (рисунок 3).



Рисунок 3 – Диаграмма прецедентов системы бронирования номеров в гостинице

Прецедент это именованная совокупность различных сценариев доставляющих актору значимый для него результат взаимодействия с системой.

Дальнейшее проектирование логики системы состоит в детализации прецедентов (более подробном описании сценариев) различными диаграммами UML.

Рассмотрим сценарии для прецедента «Поиск и бронирование номера».

Цель - предоставить клиенту возможность найти подходящий номер в гостинице и забронировать его на желаемые даты.

Действующее лицо – Клиент.

Предварительные условия:

- веб-приложение доступно и работает;
- список номеров и их доступность актуальны в системе;
- клиент либо зарегистрирован, либо может искать номер как гость.

Основные сценарии:

Клиент начинает поиск – переходит на страницу поиска номеров, указывает желаемые даты заезда и выезда, указывает количество гостей, нажимает кнопку «Поиск».

Система отображает результаты – список доступных номеров, соответствующих критериям поиска, для каждого номера отображается основная информация (фото, тип номера, цена).

Клиент просматривает детали номера – выбирает один из номеров и переходит на страницу деталей номера, на странице деталей номера отображается подробная информация о номере.

Клиент оформляет бронирование. Для оформления бронирования требуется регистрация, при которой клиент указывает свои контактные данные (имя, фамилия, email, телефон). Клиент соглашается с условиями бронирования и правилами отмены.

Система подтверждает бронирование – отображает страницу подтверждения бронирования с деталями бронирования (номер брони, даты, номер, стоимость, контактные данные).

Бронирование успешно создано – информация о бронировании сохраняется в системе, статус номера меняется на «забронирован» на указанные даты.

Альтернативные сценарии:

Нет доступных номеров – если система не находит доступных номеров, соответствующих критериям поиска, она отображает сообщение об отсутствии результатов.

Клиент отказывается от бронирования – на любом этапе процесса бронирования клиент может отказаться от бронирования и вернуться на предыдущую страницу.

Проблемы с подключением к сети - во время процесса поиска или бронирования может возникнуть проблема с подключением к сети. Система должна обрабатывать эти ошибки и предоставлять пользователю соответствующие сообщения.

Исключительные сценарии:

Системная ошибка – во время процесса бронирования может произойти системная ошибка (например, ошибка базы данных). Система должна зафиксировать ошибку в логах и отобразить пользователю информативное сообщение об ошибке.

Несанкционированный доступ – система должна быть защищена от несанкционированного доступа к данным бронирования.

Постусловия:

- информация о бронировании сохранена в системе;
- статус номера обновлен;

Дополнительные условия. Процесс поиска и бронирования должен быть максимально простым и понятным для пользователя. Обеспечение безопасности данных клиента. Быстрая загрузка страниц и обработка запросов. Корректное отображение на различных устройствах (компьютеры, планшеты, смартфоны).

Укрупненный вариант поведения системы при реализации прецедента «Поиск и бронирование номера» показан на диаграмме активности (рисунок 4).

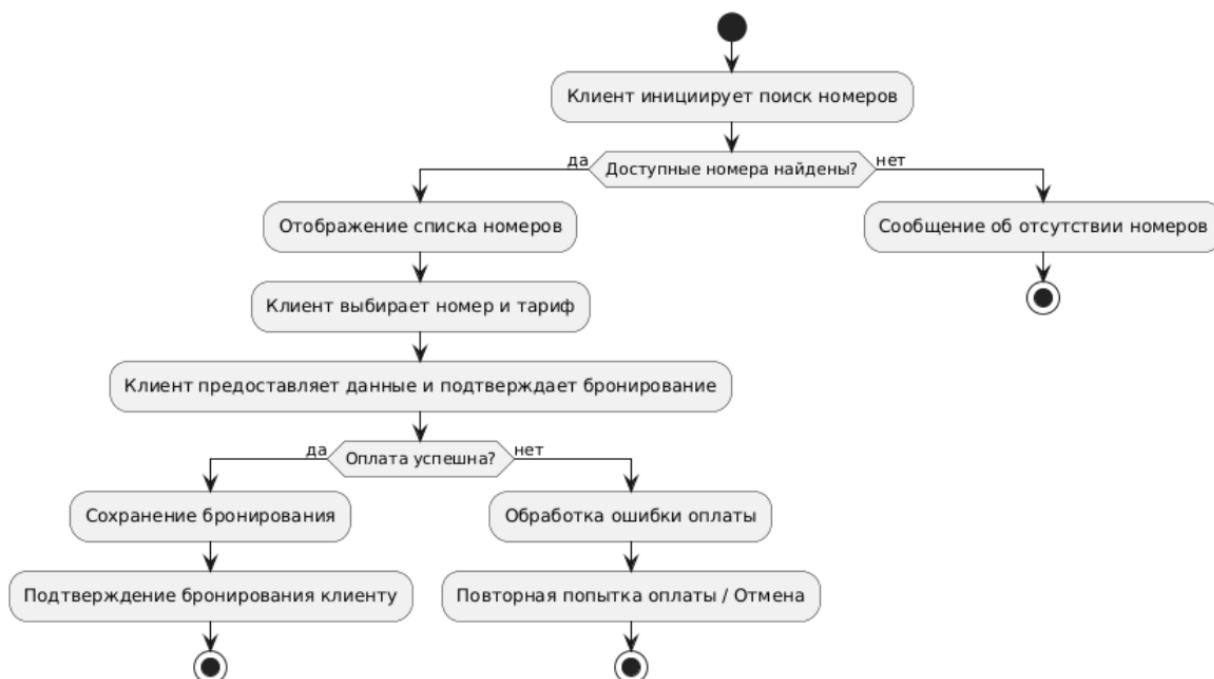


Рисунок 4 – Диаграмма активности реализации прецедента «Поиск и бронирование номера»

Диаграмма активности визуализирует процесс бронирования на высоком уровне абстракции и описывает условия возникновения и реализации различных сценариев.

Более детальное представление каждого из сценариев выполнено с использованием диаграмм последовательности.

На рисунке 5 показана диаграмма последовательности сценария «Нет доступных номеров»

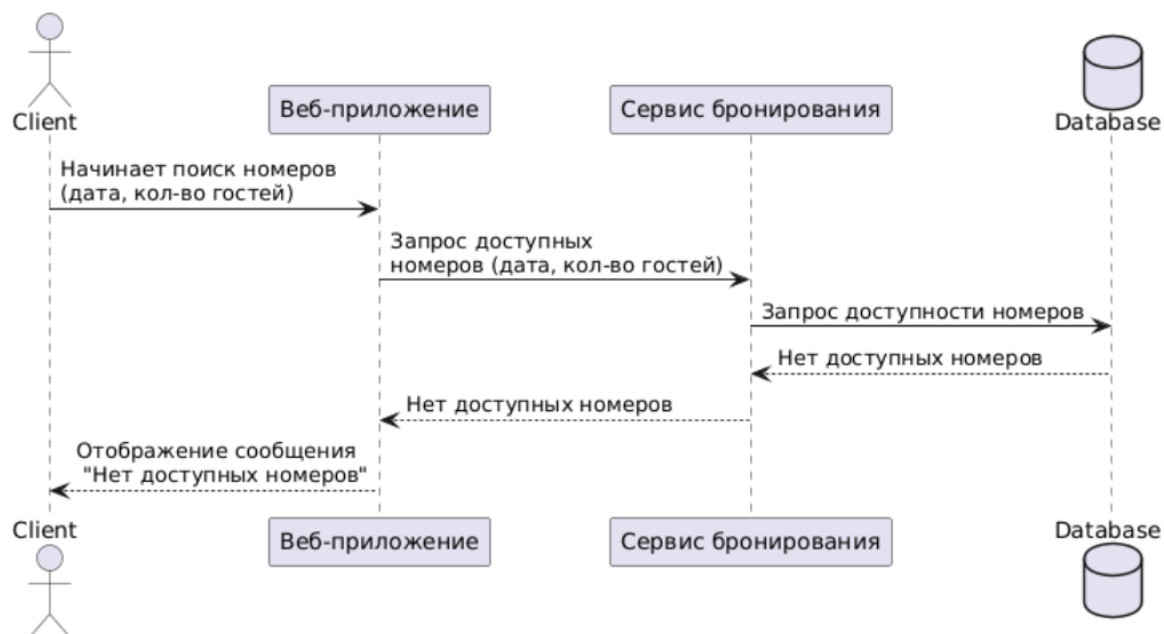


Рисунок 5 - Диаграмма последовательности сценария «Нет доступных номеров»

На рисунке 6 показана диаграмма последовательности сценария «Успешное бронирование».

На рисунке 7 показана диаграмма последовательности сценария «Ошибка оплаты».

На рисунке 8 показана диаграмма последовательности сценария «Системная ошибка во время сохранения бронирования»

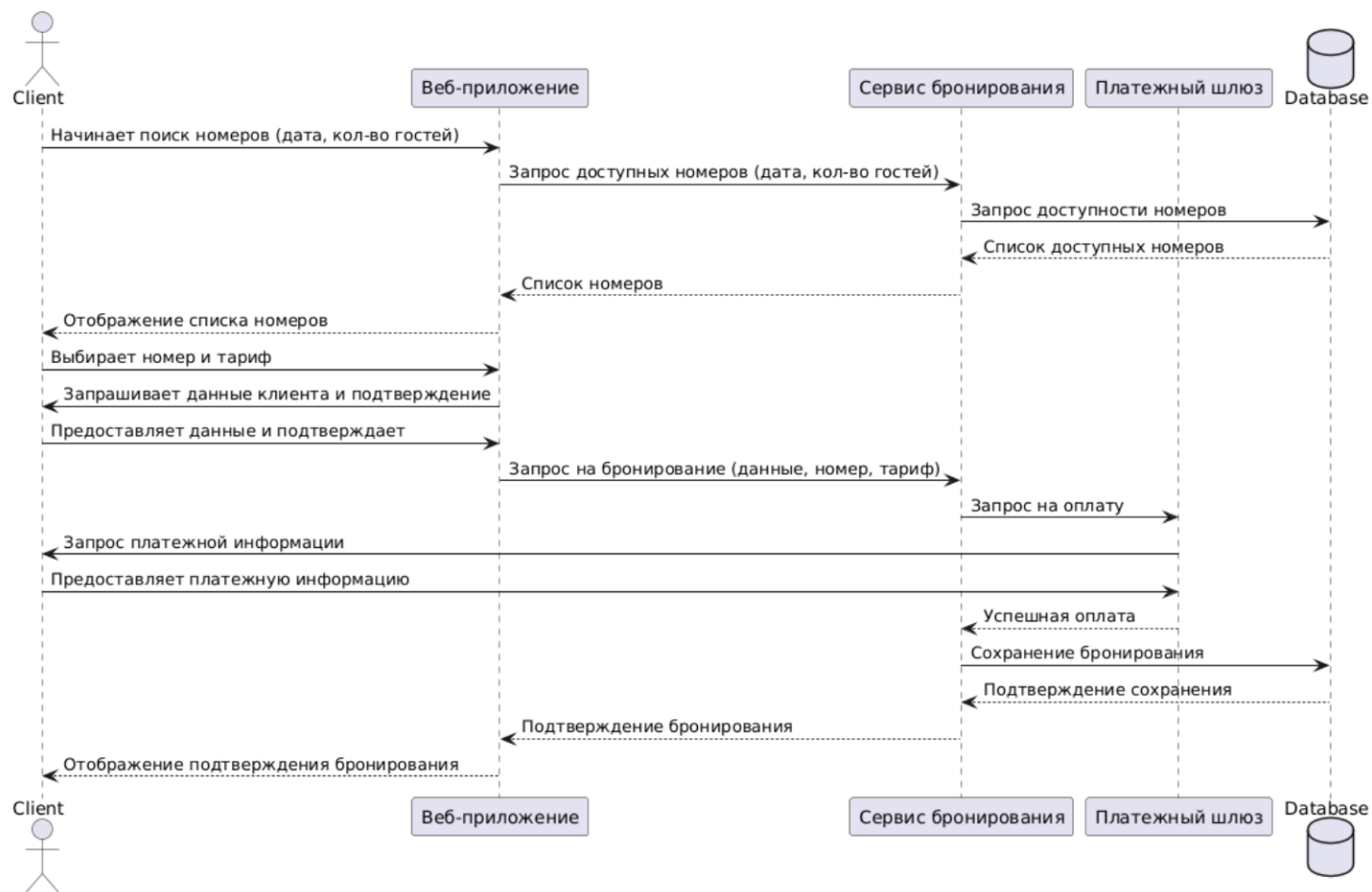


Рисунок 6 - Диаграмма последовательности сценария «Успешное бронирование»

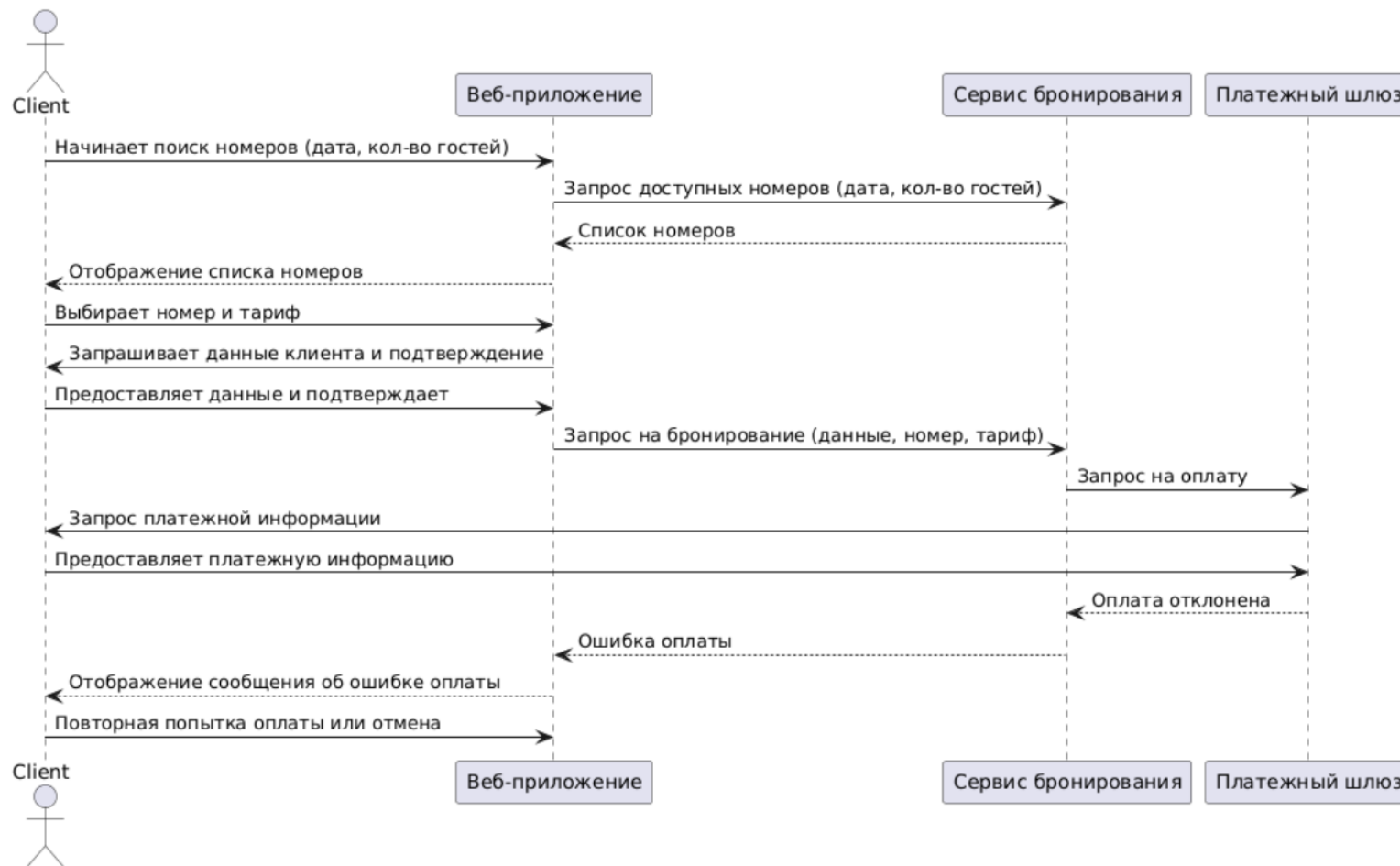


Рисунок 7 - Диаграмма последовательности сценария «Ошибка оплаты»

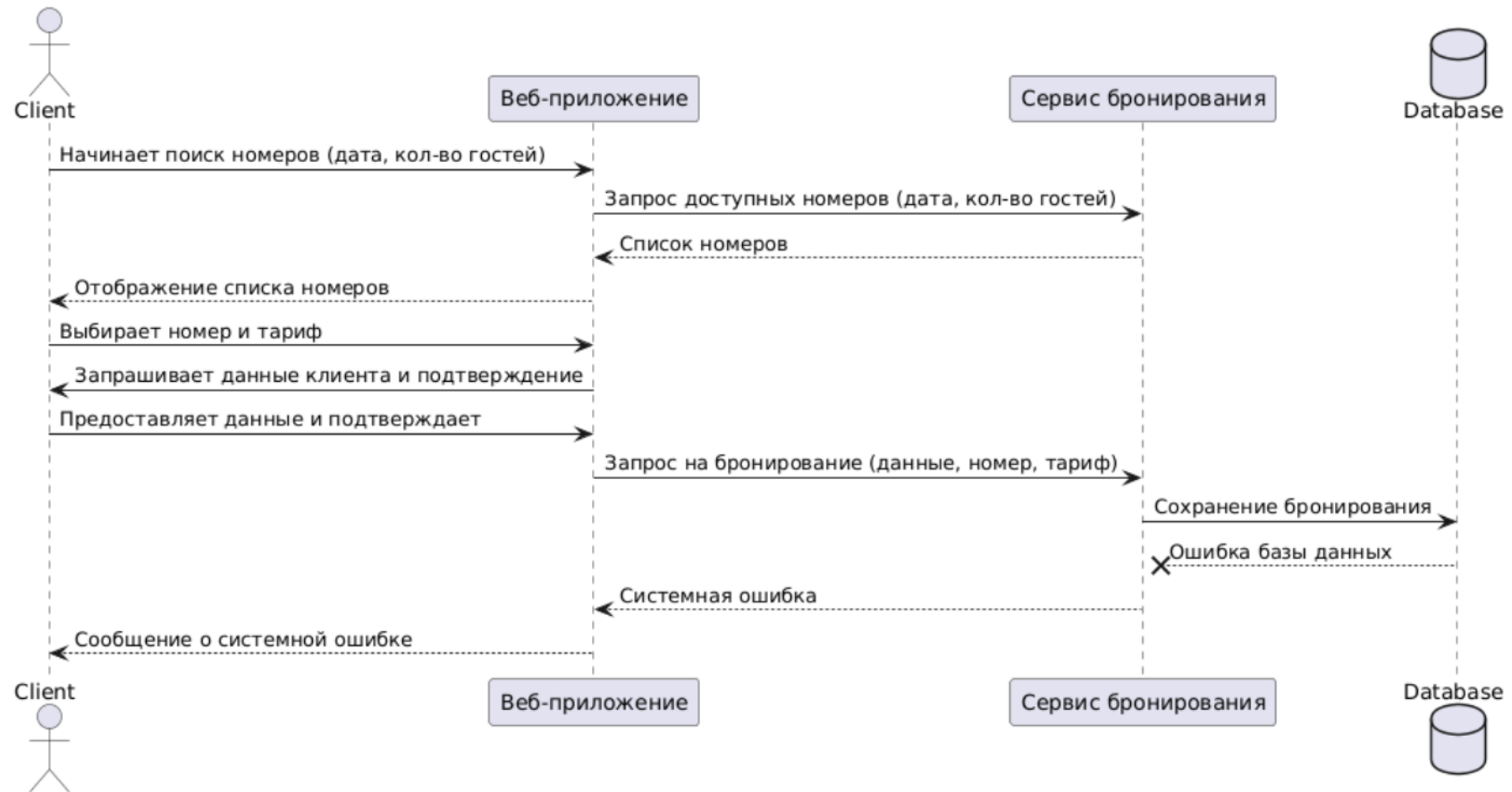


Рисунок 8 - Диаграмма последовательности сценария «Системная ошибка во время сохранения бронирования»

На рисунке 9 показана диаграмма последовательности сценария «Ошибка сети».



Рисунок 9 - Диаграмма последовательности сценария «Ошибка сети»

Эти диаграммы последовательности дают более детальное представление о том, как система работает в различных сценариях и могут быть использованы при разработке и тестировании.

Диаграмма состояний брони показана на рисунке 10.

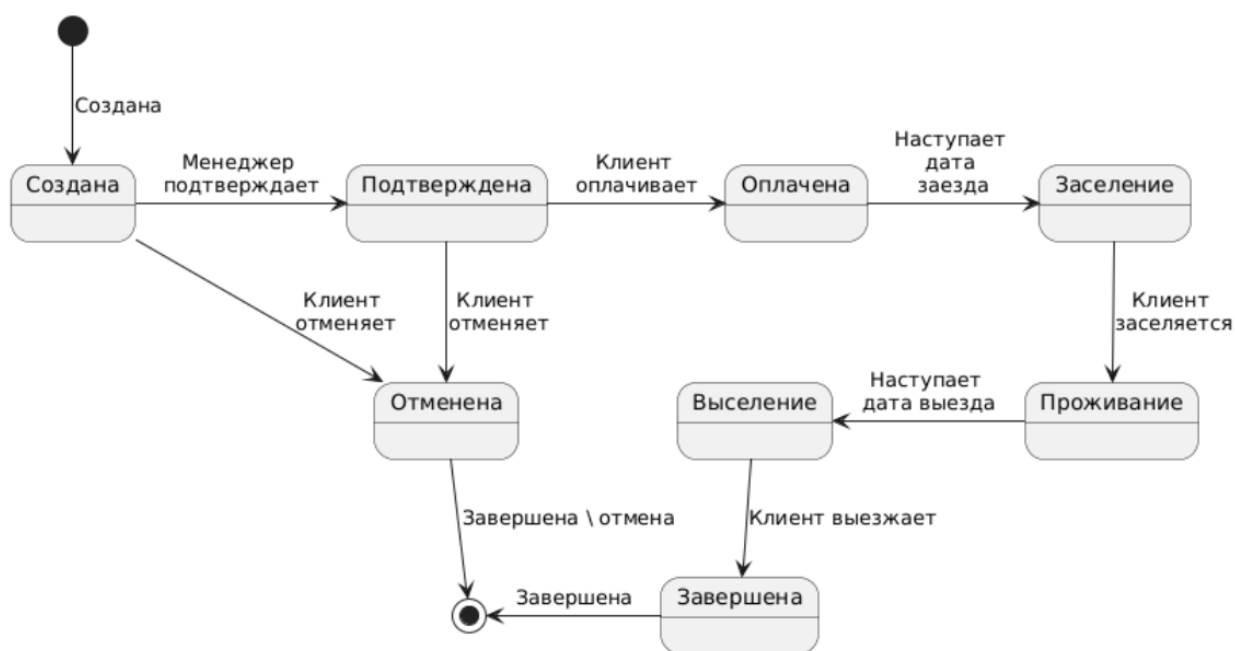


Рисунок 10 - Диаграмма состояний брони

Диаграмма состояний показывает, как бронь переходит из одного состояния в другое в зависимости от различных событий. Эта диаграмма полезна для проектирования логики обработки бронирований в системе.

Диаграмма классов показана на рисунке 11.

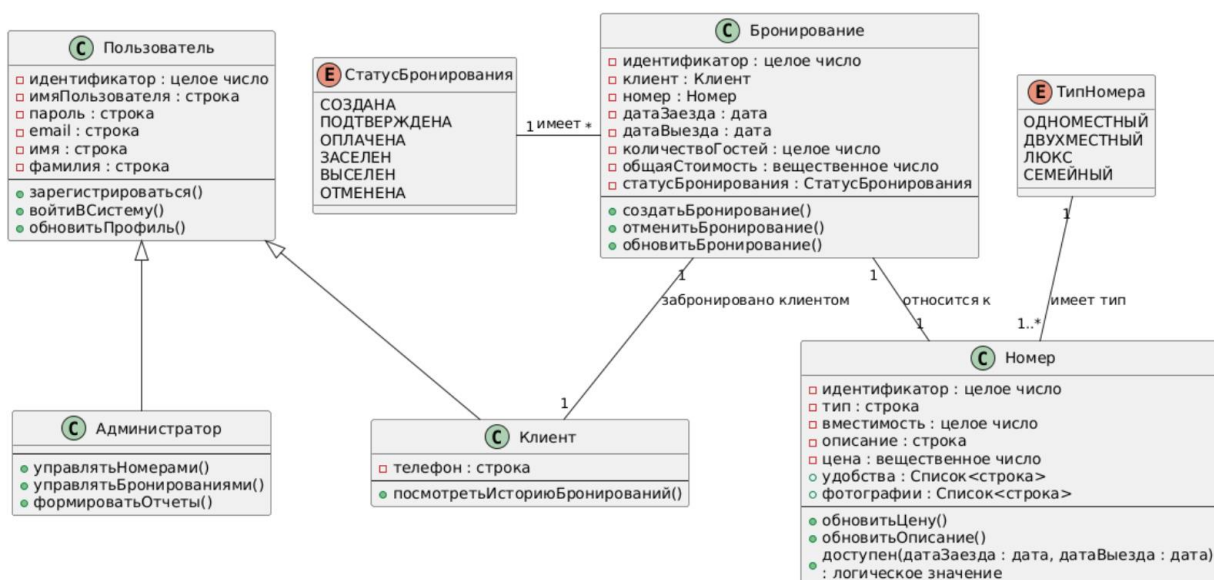


Рисунок 11 – Диаграмма классов

Диаграмма классов описывает статическое состояние проектируемой системы и служит основой для проектирования базы данных.

2.2 Проектирование базы данных веб приложения бронирования номеров в гостинице

База данных веб-приложения для бронирования номеров в гостинице состоит из семи таблиц (рисунок 12).

Таблица Пользователь хранит информацию о пользователях системы, включая как клиентов, так и администраторов. Она содержит уникальный идентификатор пользователя, логин пользователя для входа в систему,

зашифрованный пароль, адрес электронной почты, имя и фамилию, роль пользователя в системе (клиент или админ).

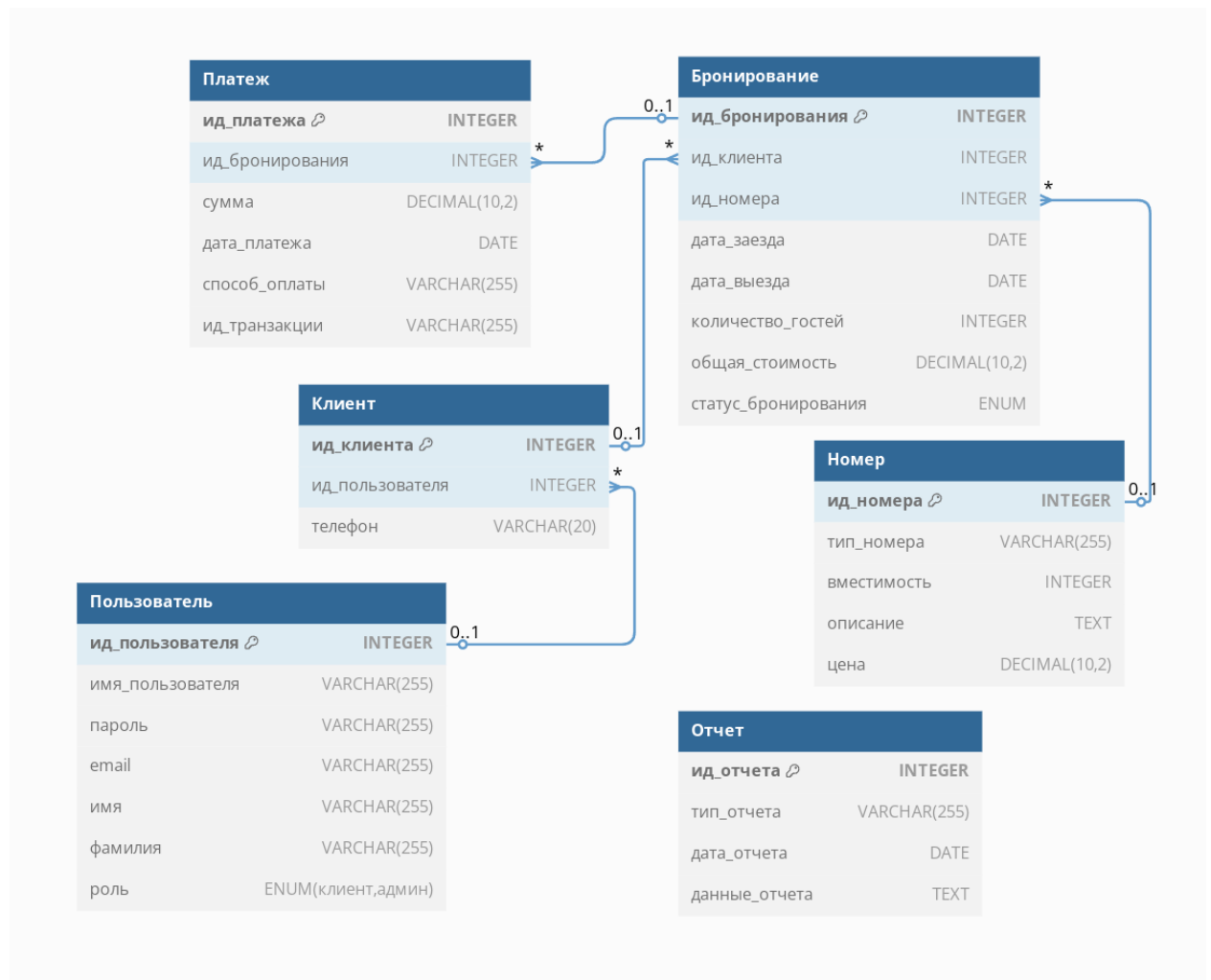


Рисунок 12 – Модель базы данных

Таблица Клиент хранит дополнительную информацию о клиентах, связанную с их учетной записью и содержит уникальный идентификатор клиента, идентификатор пользователя, связанного с этим клиентом, телефон.

Таблица Номер хранит информацию о доступных номерах в гостинице и содержит идентификатор, тип номера, количество людей, которое может вместить номер, описание номера, цена номера за сутки.

Таблица Бронирование хранит информацию о бронированиях номеров и содержит идентификатор бронирования, идентификатор клиента, сделавшего бронирование, идентификатор забронированного номера, даты заезда и выезда, количество гостей, статус бронирования.

Таблица Платеж хранит информацию о платежах, связанных с бронированиями и содержит идентификатор платежа, идентификатор бронирования, к которому относится платеж, сумма платежа, идентификатор транзакции.

Таблица Отчет хранит информацию об отчетах, создаваемых в системе и содержит идентификатор отчета, тип отчета, дату и данные отчета.

В базе данных для установления связей между таблицами используются внешние ключи. Этим обеспечивается ссылочная целостность данных.

Каждая таблица имеет уникальный первичный ключ, который обеспечивает уникальную идентификацию каждой записи в таблице.

Использование перечислений (ENUM) для полей роль в таблице Пользователь и статус_бронирования в таблице Бронирование позволяет ограничить значения этих полей предопределенным списком, что обеспечивает консистентность данных.

Спроектированная структура базы данных предоставляет основу для разработки веб-приложения бронирования номеров, обеспечивая хранение и управление информацией о пользователях, номерах, бронированиях и платежах.

Выводы по разделу.

Во втором разделе выполнено логическое проектирование веб приложения для бронирования номеров в гостинице. Проведено проектирование базы данных веб приложения. Полученные результаты позволяют перейти к разработке веб приложения.

3 Разработка программного обеспечения системы бронирования номеров

3.1 Выбор технологий реализации программного обеспечения

В качестве языка программирования выбран Python [10].

Python обладает обширной экосистемой библиотек и фреймворков, которые значительно упрощают разработку. Библиотеки для работы с базами данных (SQLAlchemy [11]) и фреймворки для веб приложений (Flask [12]) значительно сокращают время разработки.

В качестве IDE выбрано Visual Studio Code [13]/

В качестве СУБД выбрана SQLite [14].

SQLite – это надежная и стабильная система с высокой степенью защиты данных. Она используется во многих популярных приложениях, что подтверждает ее качество и надежность.

3.2 Реализация информационной модели программного обеспечения

Переход от логической к физической модели связан с выбором конкретной системы управления базой данных. В данном случае в качестве СУБД выбран SQLite.

Программная реализация физической модели в рамках фреймворка Flask выполнена следующим кодом.

```
# --- Модели данных ---  
class Пользователь(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    имяПользователя = db.Column(db.String(80), unique=True,  
nullable=False)
```

пароль = db.Column(db.String(120), nullable=False) # В реальном приложении хешировать!

email = db.Column(db.String(120), unique=True, nullable=False)

имя = db.Column(db.String(80))

фамилия = db.Column(db.String(80))

тип = db.Column(db.String(20), default='клиент') # 'клиент' или 'администратор'

def __repr__(self):

return f'<Пользователь {self.имяПользователя}>'

class Номер(db.Model):

id = db.Column(db.Integer, primary_key=True)

тип = db.Column(db.String(50), nullable=False) # Можно использовать enum как строку

вместимость = db.Column(db.Integer, nullable=False)

описание = db.Column(db.Text)

цена = db.Column(db.Float, nullable=False)

удобства = db.Column(db.Text) # Список удобств как строка, разделенная запятыми

фотографии = db.Column(db.Text) # Список фотографий как строка, разделенная запятыми

def __repr__(self):

return f'<Номер {self.тип} ({self.вместимость} чел.) - {self.цена}>'

class Бронирование(db.Model):

id = db.Column(db.Integer, primary_key=True)

клиент_id = db.Column(db.Integer, db.ForeignKey('пользователь.id'), nullable=False)

номер_id = db.Column(db.Integer, db.ForeignKey('номер.id'), nullable=False)

датаЗаезда = db.Column(db.Date, nullable=False)

датаВыезда = db.Column(db.Date, nullable=False)

```

    количествоГостей = db.Column(db.Integer, nullable=False)
    общаяСтоимость = db.Column(db.Float, nullable=False)
    статусБронирования = db.Column(db.String(20), default='СОЗДАНА')
# Можно использовать enum как строку

    клиент = db.relationship('Пользователь',
backref=db.backref('бронирования', lazy=True))

    номер = db.relationship('Номер', backref=db.backref('бронирования',
lazy=True))

    def __repr__(self):
        return f'<Бронирование {self.id} - {self.датаЗаезда} - {self.датаВыезда}>'

class Отчет(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    типОтчета = db.Column(db.String(50), nullable=False)
    датаОтчета = db.Column(db.Date, nullable=False)
    данныеОтчета = db.Column(db.Text)
    def __repr__(self):
        return f'<Отчет {self.типОтчета} - {self.датаОтчета}>'

```

Разработанная модель данных позволяет эффективно организовать бронирование номеров в гостинице.

3.3 Разработка программного обеспечения

Веб-приложение, разработанное с использованием фреймворка Flask, состоит из следующих компонентов:

Приложение Flask является ядром и включает в себя файлы app.py и routes.py. app.py содержит основную логику запуска приложения, инициализирует необходимые расширения и импортирует другие модули. routes.py определяет маршруты приложения, обрабатывает HTTP-запросы,

вызывает функции для обработки запросов, взаимодействует с моделями данных и формами, а также отвечает за рендеринг HTML-шаблонов.

Компонент Модели данных представлен файлом `models.py` и определяет структуру данных приложения, используемую для взаимодействия с базой данных. Он определяет классы, представляющие таблицы в базе данных, их атрибуты и взаимосвязи между ними. Использует внешнюю библиотеку `SQLAlchemy` для создания и управления базой данных.

Компонент Формы содержащийся в файле `forms.py`, отвечает за обработку пользовательского ввода при выполнении таких действий как регистрация, вход, бронирование. Использует библиотеку `WTForms` [15] для создания форм, валидации введенных данных и обработки ошибок.

Компонент Шаблоны (HTML) используется для создания динамического пользовательского интерфейса. Шаблоны содержат структуру HTML-страниц и логику для отображения данных, полученных от приложения. Для рендеринга шаблонов используется библиотека `Jinja2` [16].

Компонент Статические файлы содержит файлы стилей `CSS`, используемые для визуального оформления страниц.

База данных представлена файлом `database.db` и хранит данные приложения.

Приложение использует внешние библиотеки `Flask`, `SQLAlchemy`, `WTForms` и `Jinja2`.

Порядок взаимодействия компонентов при работе приложения показан на диаграмме компонентов (рисунок 13).

Для первоначального запуска и тестирования выполнено локальное развертывание в `Visual Studio Code (VS Code)`.

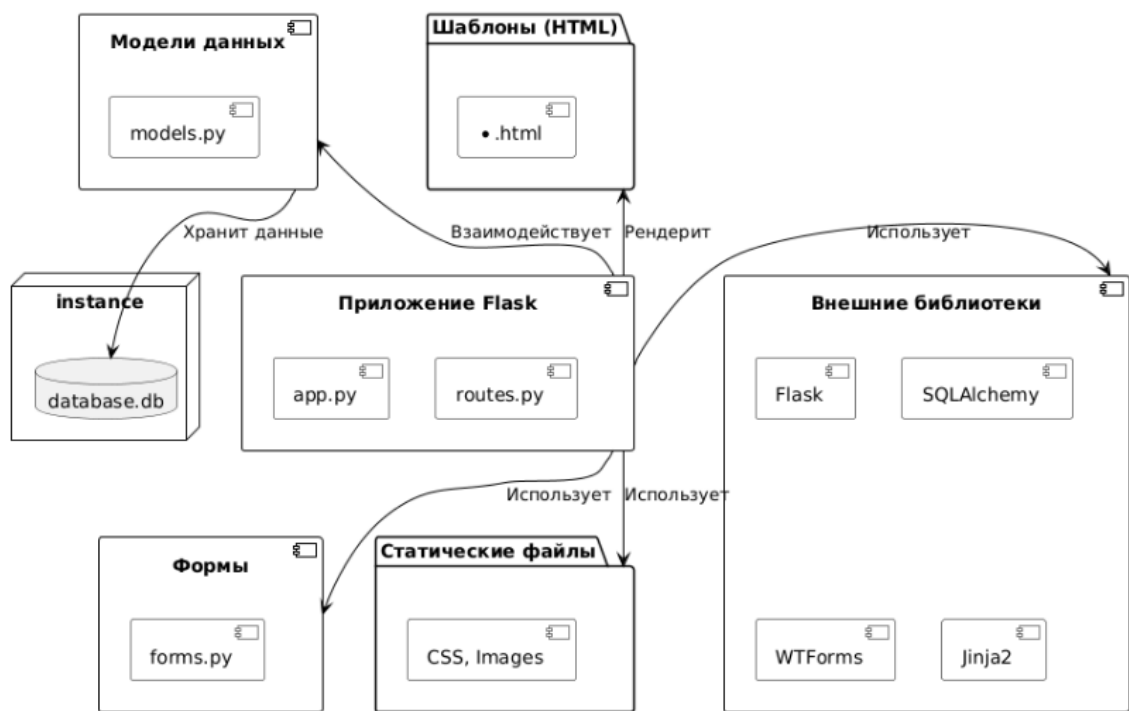


Рисунок 13 - Диаграмма компонентов приложения

Диаграмма локального развертывания показана на рисунке 14

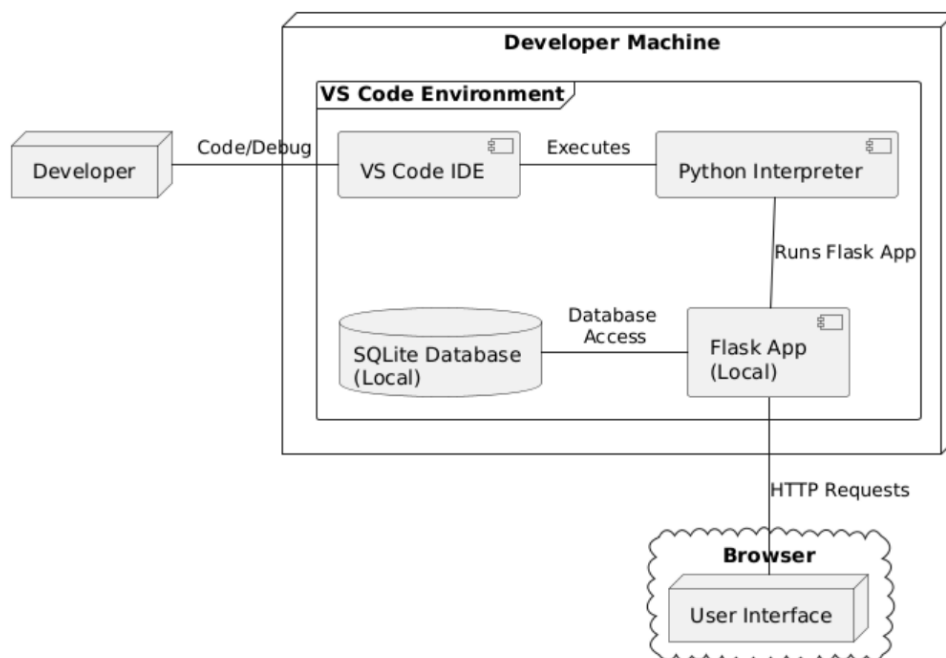


Рисунок 14 – Диаграмма локального развертывания приложения

Результате запуска и перехода в браузере по локальному адресу появляется страница с доступными для просмотра номерами гостиницы (рисунок 15), видимая любому посетителю сайта без регистрации.

На этой странице отображается список номеров, статус номера, отражающий есть ли бронирования на данный номер.

Без регистрации пользователь может посмотреть список бронирований номера перейдя по ссылке Подробности (рисунок 16). Если посетитель сайта решит, что ему целесообразно забронировать номер и нужные ему даты бронирования не пересекаются с уже имеющимися бронями, то он должен зарегистрироваться с указанием необходимой для бронирования информации и создать заявку на бронирование. Статус бронирования Заявка означает, что бронирование пока не рассмотрено администратором.

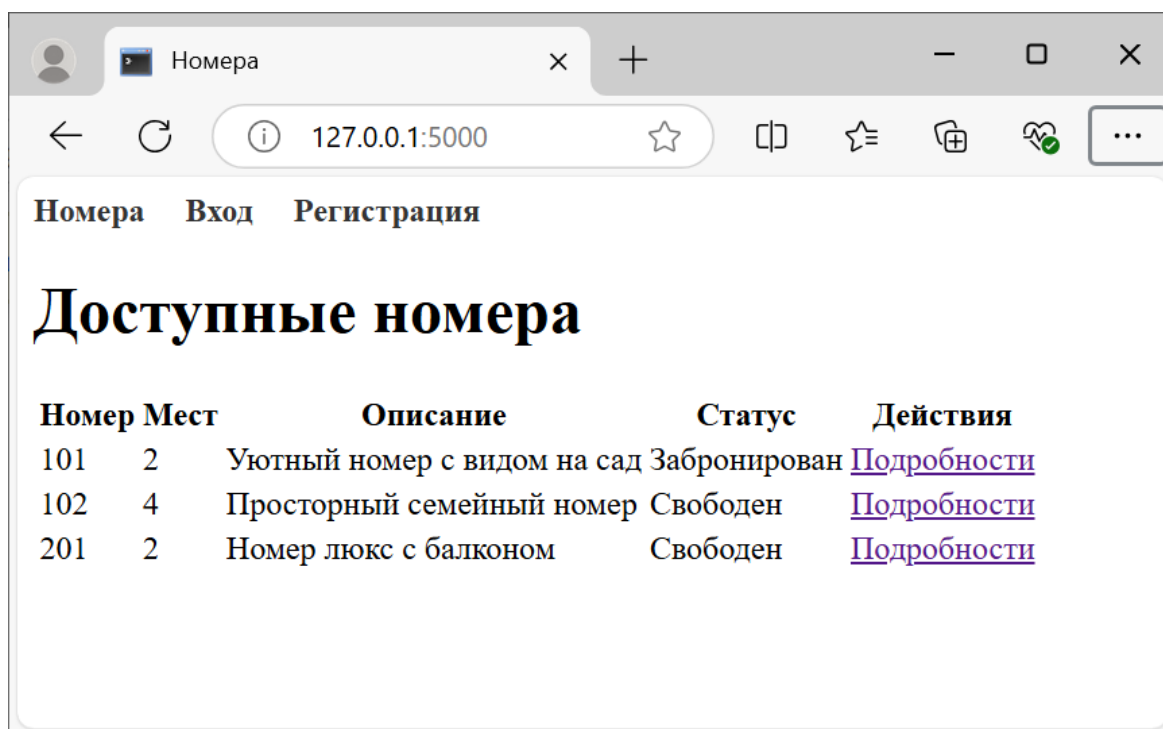


Рисунок 15 – Стартовая страница приложения для незарегистрированного пользователя

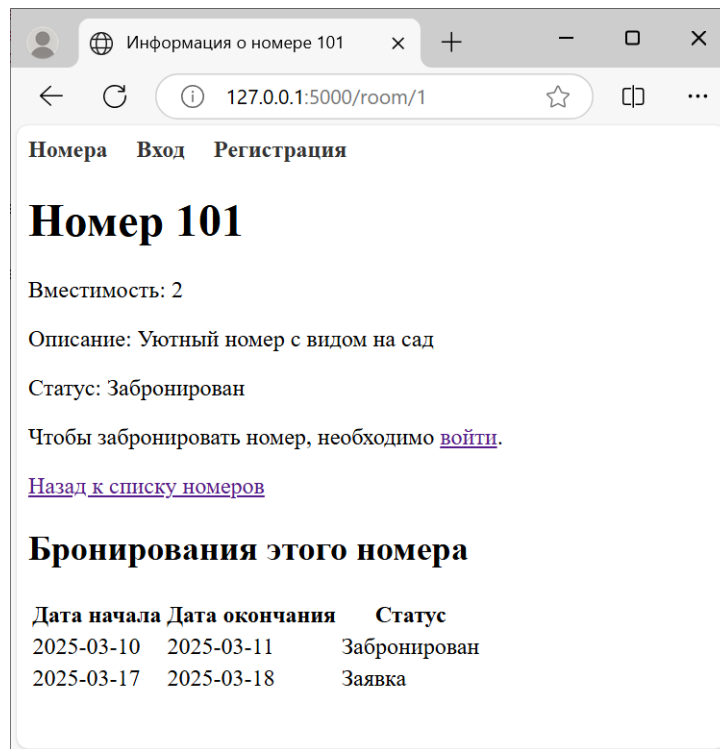


Рисунок 16 – Окно с бронированиями номера

В целом разработанное веб приложение работоспособно. Для его размещения в продакшене разработана диаграмма развертывания, показанная на рисунке 17.

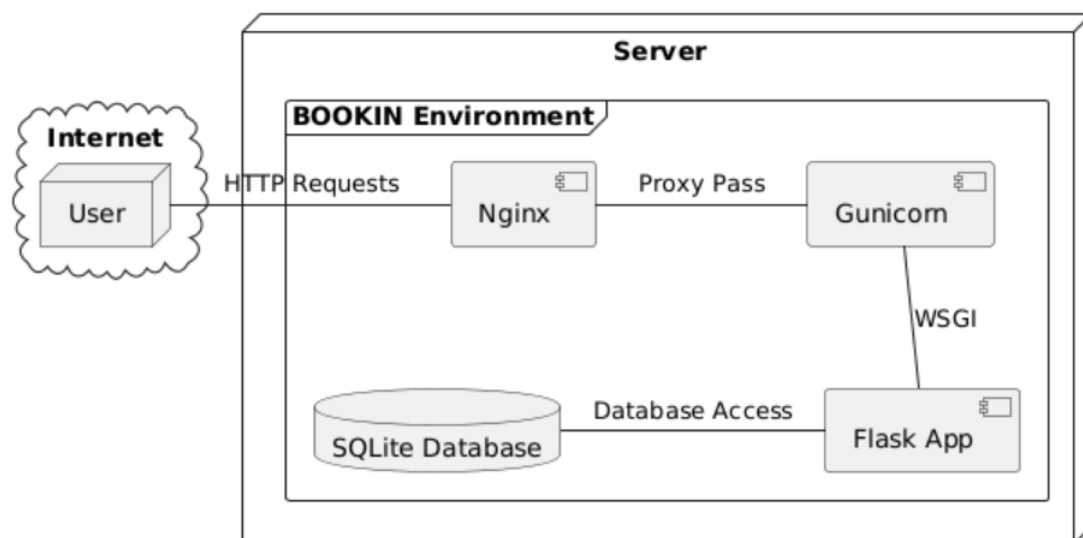


Рисунок 17 – Диаграмма развертывания приложения в продакшене

Сценарий развертывания включает.

Один виртуальный сервер, например, на AWS EC2, Google Compute Engine, Azure VM под управлением операционной системы Ubuntu Server [17].

Web Server: Nginx [18] в качестве обратного прокси-сервера.

Сервер приложений Gunicorn WSGI server [19].

Базу данных SQLite.

3.4 Тестирование программного обеспечения

Выполнено локальное развертывание веб-приложения и проведено ручное тестирование [20] его функциональности согласно тест-плану, показанному в таблице 3.

Действующие лица: Посетитель – неавторизованный или незарегистрированный пользователь, Клиент – авторизованный пользователь, Администратор – авторизованный пользователь с правами управления бронированием.

Предварительно создан администратор с логином и паролем admin admin и клиент с логином и паролем поль22 22. База данных заполнена тестовыми данными по трем номерам.

План тестирования выполнен в полном объеме без сбоев и ошибок.

Пример реализации тестового сценария бронирования номера зарегистрированным пользователем.

Таблица 3 – План тестирования веб-приложения бронирования номеров

Содержание теста	Действующие лица	Ожидаемый результат
Просмотр бронирований номера	Посетитель	По номеру со статусом Забронирован отображается список бронирований со статусами и датами.
Бронирование номера	Посетитель	Переход на страницу регистрации
Бронирование номера	Клиент	Переход на страницу бронирования, После успешного бронирования отображение бронирования на странице Мои бронирования со статусом Заявка.
Просмотр своих бронирований	Клиент	Отображение списка бронирований клиента с указанием статуса и дат
Регистрация	Посетитель, Клиент	Переход на форму регистрации. После успешной регистрации переход на форму авторизации. После успешной авторизации возможность перехода к странице Мои бронирования.
Авторизация с вводом неправильного логина и пароля	Посетитель	Сообщение о неудачной авторизации, переход на страницу регистрации
Авторизация с вводом логина и пароля Клиента	Посетитель	Возможность перехода к странице Мои бронирования
Авторизация с вводом логина и пароля Администратора	Посетитель	Возможность перехода к странице Панель администратора
Изменение статуса бронирования	Администратор	Отображение брони на странице Панель администратора с возможностью фильтрации по статусу. Возможность просмотреть информацию о пользователе для каждой брони Возможность изменить статус брони. Возможность удалить бронь при принятии заявки на отмену
Отмена брони	Клиент, Администратор, Посетитель	Отмена брони со страницы Мои брони переводит брони в статус Заявка на отмену. Одобрение Администратором отмены брони удаляет бронь, после чего бронь не отображается в списке Мои брони для Клиента и не отображается в датах бронирования номера для Посетителя

Стартовая страница показана на рисунке 18.

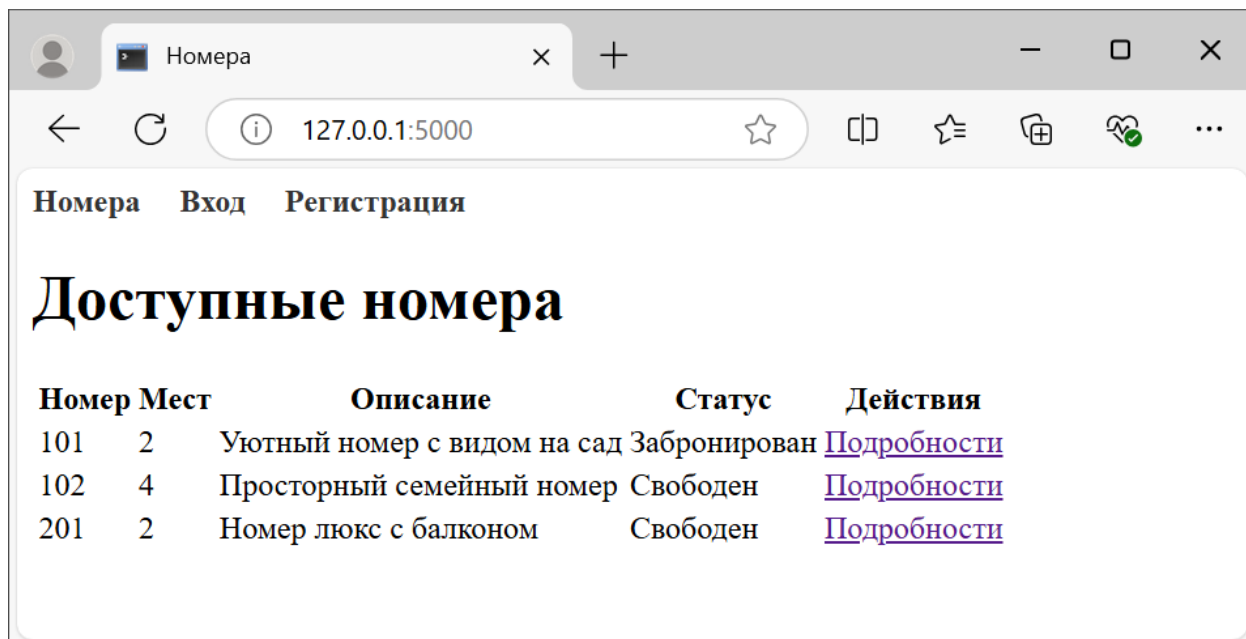


Рисунок 18 – Стартовая страница тестового сценария бронирования номера

Вход по логину и паролю показан на рисунке 19.

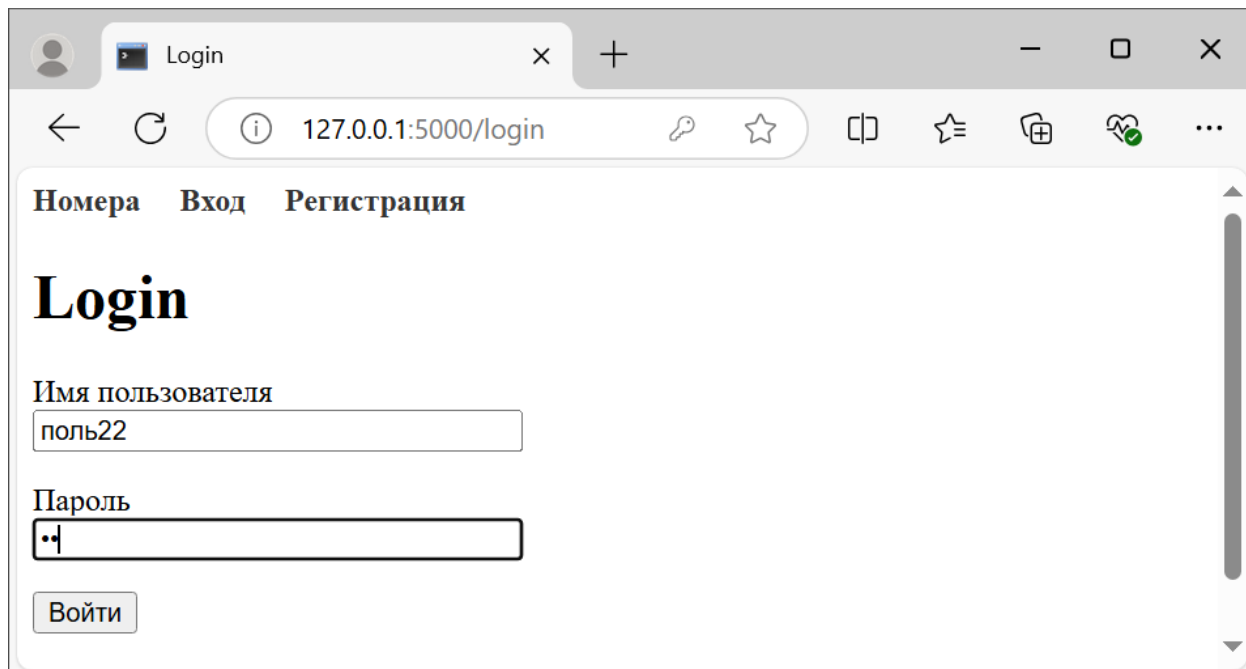


Рисунок 19 – Страница входа

После успешного входа появляется стартовая страница с пунктом меню Мои брони и списком номеров с возможностью забронировать (рисунок 20).

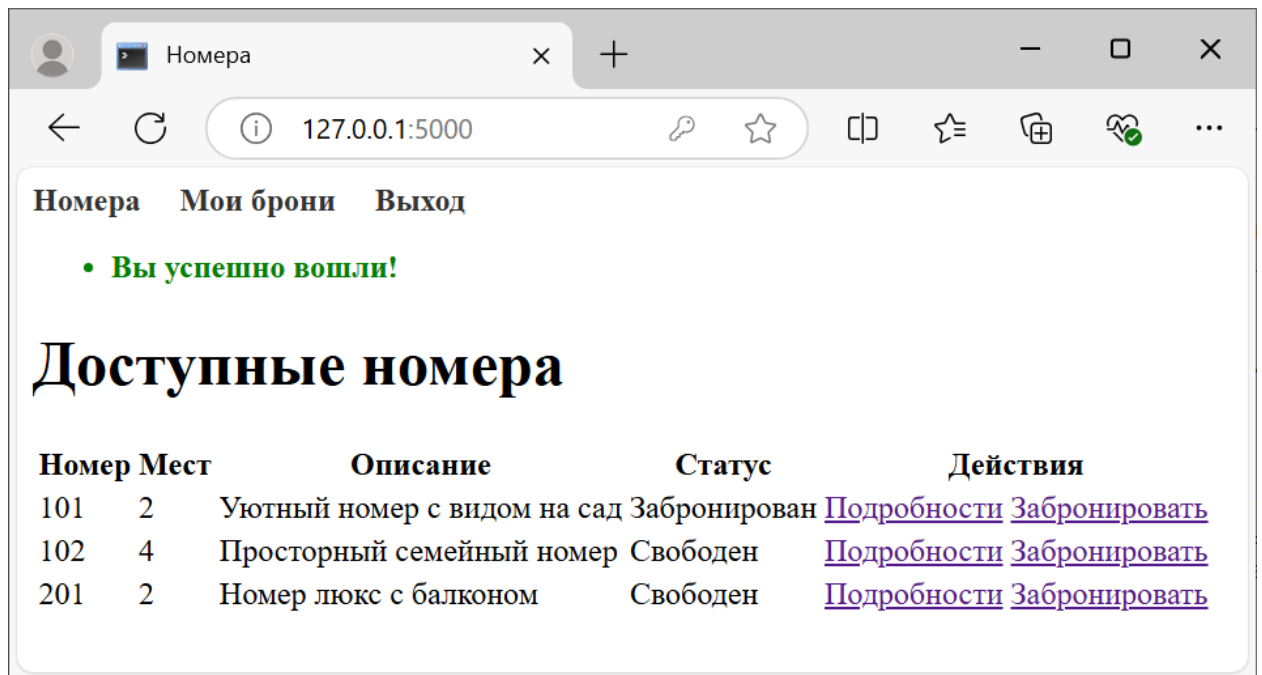


Рисунок 20 – Стартовая страница для Клиента

При переходе по ссылке Подробности номера 201 появляется страница подробного описания номера (рисунок 21).

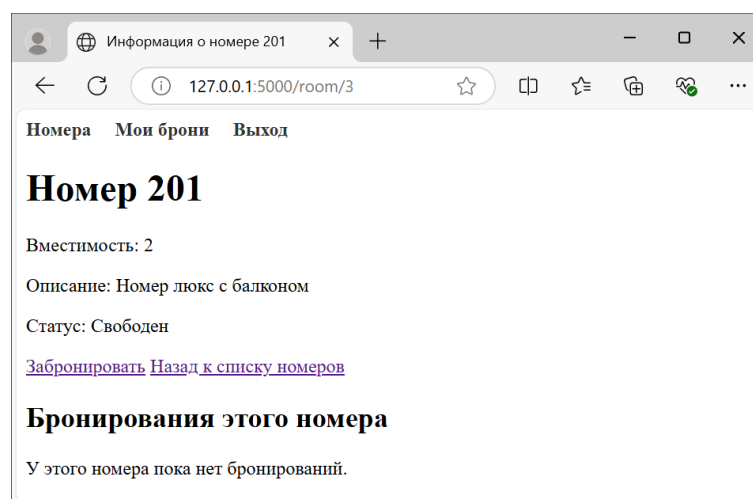


Рисунок 21 – Страница подробного описания номера 201

После перехода по ссылке **Забронировать** появляется окно бронирования номера (рисунок 22).

Номера Мои брони Выход

Бронирование номера 201

Дата начала
06.03.2025

Дата окончания
08.03.2025

Забронировать

[Назад к информации о номере](#)

Рисунок 22 – Окно бронирования номера

После нажатия кнопки **Забронировать** выполняется переход к странице **Мои бронирования** где отображается новая бронь со статусом **Заявка**.

Номера Мои брони Выход

• Номер успешно забронирован!

Мои бронирования

Номер	Дата начала	Дата окончания	Статус	Действия
101	2025-03-10	2025-03-11	Забронирован	Отменить
101	2025-03-17	2025-03-18	Заявка	Отменить
201	2025-03-06	2025-03-08	Заявка	Отменить

Рисунок 23 – Новая бронь в списке бронирований Клиента

Результаты тестирования приведены в таблице 4

Таблица 4 – Результаты тестирования

Действие	Результат
Просмотр бронирований номера	Успешно. Сбоев и ошибок не выявлено
Бронирование номера Посетителем	Успешно. Сбоев и ошибок не выявлено
Бронирование номера Клиентом	Успешно. Сбоев и ошибок не выявлено
Просмотр своих бронирований Клиентом	Успешно. Сбоев и ошибок не выявлено
Регистрация	Успешно. Сбоев и ошибок не выявлено
Авторизация с вводом неправильного логина и пароля	Успешно. Сбоев и ошибок не выявлено
Авторизация с вводом логина и пароля Клиента	Успешно. Сбоев и ошибок не выявлено
Авторизация с вводом логина и пароля Администратора	Успешно. Сбоев и ошибок не выявлено
Изменение статуса бронирования	Успешно. Сбоев и ошибок не выявлено
Отмена брони	Успешно. Сбоев и ошибок не выявлено

Выводы по разделу.

В результате выбора технологий реализации программного обеспечения выбраны язык программирования, фреймворк, база данных и среда разработки для реализации веб приложения бронирования номеров в гостинице.

Выполнена реализация модели данных СУБД SQLite.

Проведена разработка программного обеспечения с разработкой архитектуры приложения отображенной на диаграмме компонентов, требований к локальному развертыванию и требований к развертыванию в продакшене, проиллюстрированными диаграммами развертывания.

Выполнено локальное развёртывание приложения и проведено тестирование в ручном режиме. В результате тестирования приложения ошибок и сбоев не выявлено. Тестирование прошло успешно.

Заключение

В работе выполнена разработка веб-приложения для системы бронирования номеров в гостинице.

Описана деятельность гостиницы, показана ее организационная структура. Описан процесс бронирования. Выполнен анализ лучших практик и альтернативных вариантов бронирования номеров в гостинице. Сформулированы требования к информационной системе.

Выполнено логическое проектирование веб приложения для бронирования номеров в гостинице. Проведено проектирование базы данных веб приложения.

В результате выбора технологий реализации программного обеспечения выбраны язык программирования, фреймворк, база данных и среда разработки для реализации веб приложения бронирования номеров в гостинице.

Выполнена реализация модели данных СУБД SQLite.

Проведена разработка программного обеспечения с разработкой архитектуры приложения отображенной на диаграмме компонентов, требований к локальному развертыванию и требований к развертыванию в продакшене, проиллюстрированными диаграммами развертывания.

Выполнено локальное развёртывание приложения и проведено тестирование в ручном режиме. В результате тестирования приложения ошибок и сбоев не выявлено. Тестирование прошло успешно.

В целом разработанное веб приложение для бронирования номеров в гостинице соответствует требованиям к ПО, обладает необходимой функциональностью и проверено в использовании посредством тестирования.

Таким образом все задачи выпускной квалификационной работы выполнены, цель выпускной квалификационной работы достигнута.

Список используемой литературы и используемых источников

1. Booking.com [Электронный ресурс]. URL: <https://www.booking.com/> (дата обращения: 06.03.2025).
2. Expedia Group platform [Электронный ресурс]. URL: <https://www.expediagroup.com/home/> (дата обращения: 06.03.2025).
3. Hotels.com [Электронный ресурс]. URL: <https://www.hotels.com/> (дата обращения: 06.03.2025).
4. Airbnb [Электронный ресурс]. URL: <https://www.airbnb.ru/> (дата обращения: 06.03.2025).
5. Ostrovok [Электронный ресурс]. URL: <https://ostrovok.ru/> (дата обращения: 06.03.2025).
6. Marriott Bonvoy [Электронный ресурс]. URL: <https://www.marriott.com/ru/default.mi> (дата обращения: 06.03.2025).
7. Hostels Worldwide - Online Bookings, Ratings and Reviews [Электронный ресурс]. URL: <https://www.hostelworld.com/ru/> (дата обращения: 06.03.2025).
8. Discover glamping, cabins, tiny homes, tree houses [Электронный ресурс]. <https://glampinghub.com/> (дата обращения: 06.03.2025).
9. Ecobnb [Электронный ресурс]. URL: <https://ecobnb.com/> (дата обращения: 06.03.2025).
10. Full Grammar specification [Электронный ресурс]. URL: <https://docs.python.org/3/reference/grammar.html> (дата обращения: 06.03.2025).
11. SQLAlchemy 2.0 Documentation [Электронный ресурс]. URL: <https://docs.sqlalchemy.org/en/20/> (дата обращения: 06.03.2025).
12. Flask Documentation [Электронный ресурс]. URL: <https://kimpea.github.io/technical-doc/> (дата обращения: 06.03.2025).
13. Visual Studio Code documentation [Электронный ресурс]. URL: <https://code.visualstudio.com/docs/> (дата обращения: 06.03.2025).

14. SQL Language Documentation [Электронный ресурс]. URL: <https://www.sqlite.org/docs.html> (дата обращения: 06.03.2025).
15. WTForms — WTForms Documentation (3.2.x) [Электронный ресурс]. <https://wtforms.readthedocs.io/en/main/> (дата обращения: 24.09.2024).
16. Jinja2 Documentation [Электронный ресурс]. URL: <http://mitsuhiko.pocoo.org/Jinja2.pdf> (дата обращения: 06.03.2025).
17. Ubuntu Server tutorial [Электронный ресурс]. URL: <https://documentation.ubuntu.com/server/tutorial/> (дата обращения: 06.03.2025).
18. Technical Specs [Электронный ресурс]. URL: <https://docs.nginx.com/nginx/technical-specs/> (дата обращения: 06.03.2025).
19. Gunicorn - WSGI server — Gunicorn 23.0.0 documentation [Электронный ресурс]. URL: <https://docs.gunicorn.org/en/latest/> (дата обращения: 06.03.2025).
20. Ручное тестирование веб-сайтов: основы и примеры [Электронный ресурс]. URL: <https://sky.pro/wiki/profession/ruchnoe-testirovanie-veb-sajtov-osnovy-i-primery/> (дата обращения: 06.03.2025).