

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

ПРОЕКТ
на тему
«Сервис бронирования жилья(AirBnB)»

Выполнил
Студент группы НФИбд-02-21
Студенческий билет №:1132210570

_____ Куденко М. Р.

(подпись)

«24» _____ 10 _____ 2023 г.

Москва 2023

Оглавление

1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	3
1.1 Предназначение технического задания	3
1.2 Описание проекта.....	3
1.2.1 Обзор рынка	4
1.2.2 Конкуренты	4
1.3 Функциональные требования к проекту.....	4
1.4 Требования к безопасности	7
1.5 Требования к дизайну.....	8
1.6 Критерии завершения проекта.....	10
2 USE CASE ДИАГРАММА	12
2.1 Описание основных вариантов использования:.....	13
2.1.1 Регистрация	13
2.1.2 Поиск жилья.....	13
2.1.3. Добавление жилья.....	14
2.1.4. Бронирование жилья	14
2.1.5. Просмотр и оставление отзывов	15
2.1.6. Управление ценами.....	15
2.1.7 Управление пользователями.....	16
2.1.8 Модерация объявлений	16
2.1.9 Решение конфликтов и жалоб.....	16
3 ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ.....	18
3.1. Пользовательский интерфейс (User Interface):	18
3.2. Приложение сервера	19
3.3. База данных	21
3.4 Сервер платежей.....	23
4 ДИАГРАММА КЛАССОВ И ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ	
25	
4.1 Диаграмма классов	25
4.2 Последовательность действий	26
ЗАКЛЮЧЕНИЕ	28

1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1 Предназначение технического задания

Назначением этого ТЗ является предоставление полной технической информации и требований разработки приложения для автоматизации деятельности сервиса бронирования жилья (AirBnB). Это документ, который будет использоваться разработчиками программного обеспечения для создания программы, которая будет полностью удовлетворять потребности пользователей и требования бизнес-процессов. Также этот документ может использоваться в качестве исходного документа при управлении проектом и контроле за процессом его разработки.

1.2 Описание проекта

Целью данного технического задания (ТЗ) является разработка и реализация сервиса для бронирования жилья, аналогичного Airbnb. Наш проект призван предоставить пользователю удобный и надежный механизм для поиска, бронирования и аренды жилья в различных уголках мира.

Рынок онлайн-бронирования жилья демонстрирует стабильный рост и остается одним из ключевых сегментов в индустрии туризма. Платформы, подобные Airbnb, с каждым годом привлекают все больше пользователей, предоставляя им широкий выбор жилья и уникальных возможностей для путешествий. Однако существует потребность в более гибких и инновационных решениях, способных соединить владельцев жилья и путешественников.

Наш проект стремится к созданию современного и удобного сервиса, который не только удовлетворяет запросы пользователей, но также предоставляет дополнительные функции для обогащения опыта и обеспечения безопасности как для арендодателей, так и для гостей.

1.2.1 Обзор рынка

На мировом рынке жилья для краткосрочной аренды существует растущий спрос со стороны путешественников, деловых людей и других категорий потребителей. Этот сегмент рынка переживает бурное развитие, и Airbnb, как один из крупнейших участников, успешно предоставляет платформу для бронирования разнообразных видов жилья: от апартаментов до домов и квартир.

1.2.2 Конкуренты

На момент разработки проекта существует несколько крупных конкурентов в сфере онлайн-бронирования жилья. Кроме Airbnb, наиболее значимыми являются платформы Booking.com, Vrbo, и Expedia. Эти конкуренты также предоставляют услуги по поиску и бронированию жилья, что создает конкуренцию в индустрии.

1.3 Функциональные требования к проекту

1. В ходе регистрации и авторизации должны выполняться следующие требования:

- у пользователей должна быть возможность создавать учетные записи, используя электронную почту или социальные сети;
- должно быть подтверждение электронной почты для новых пользователей;
- должно быть реализовано восстановление паролей через электронную почту;
- должна быть возможность входа в систему через социальные сети (например, Facebook, Google).

2. В приложении должны быть функции поиска и бронирование жилья, которые отвечают следующим требованиям:

- поиск жилья по различным критериям, таким как местоположение, дата, количество гостей;
- отображение результатов поиска с фильтрами и картой;
- просмотр подробной информации о жилье с фотографиями, описанием и удобствами;
- возможность отправки запросов на бронирование;
- определение цен и стоимости бронирования на основе выбранных параметров;
- интеграция с системами оплаты для осуществления бронирования;
- управление бронированиями (создание, редактирование, отмена);
- уведомления о статусе бронирования.

3. Профили пользователей должны соответствовать следующим условиям:

- личные профили пользователей должны содержать основную информацию о пользователе с фотографиями и контактными данными;
- каждый пользователь должен иметь рейтинги и отзывы других пользователей;
- каждый пользователь должен иметь возможность просматривать историю бронирований и активности.

4. Для владельцев жилья, должны быть разработаны методы управления жильем, которые отвечают следующим требованиям:

- должна быть возможность владельцев жилья добавлять свои объекты в систему;
- должна быть возможность редактирование информации о жилье, фотографий и календаря доступности;
- должна быть возможность установки цен и правил проживания;
- должна быть возможность скрытия/отключения объявлений.

5. В приложении необходимо реализовать функцию проведения оплаты, которая удовлетворяет следующим требованиям:

- должна быть реализована интеграция с платежными системами для приема платежей от клиентов;

- должна быть возможность оплаты бронирований и возможность оплаты в несколько этапов;

- в случае отмены бронирования должны быть реализована функция возврата средств.

6. В приложение необходимо внедрить систему уведомлений, таких как:

- уведомления о статусе бронирования (подтверждение, отклонение, отмена);

- уведомления о новых сообщениях и отзывах;

- уведомления о важных событиях и акциях.

7. Кроме пользовательского интерфейса, также необходимо реализовать административную панель со следующими требованиями:

- возможность управление пользователями, объявлениями и отзывами;

- возможность мониторинга бронирований и платежей;

- система жалоб и блокирования аккаунтов.

8. В приложении должны быть следующие интеграции:

- интеграция с внешними API для карт для отображения местоположения жилья и ближайших достопримечательностей, геолокации и прогноза погоды.

- интеграция с социальными сетями для обмена отзывами и активностями.

9. Одними из важных функций, повышающих пользовательский опыт, являются многоязычность и многовалютность, поэтому в системе должны быть реализованы такие функции:

- возможность выбора языка интерфейса и валюты.

- поддержка множества валют для цен и оплат.

10. Кроме того для удобного доступа к сервису должны быть разработаны мобильные приложения для iOS и Android с аналогичными функциональными возможностями.

1.4 Требования к безопасности

Требования к безопасности для сервиса бронирования жилья, подобного Airbnb, играют критическую роль, чтобы защитить данные пользователей и обеспечить безопасную среду для совершения транзакций и взаимодействия. Вот некоторые основные требования к безопасности:

1. Защита данных пользователей:

- все личные данные пользователей, включая личные и финансовые сведения, должны быть шифрованы с использованием протокола SSL/TLS.

2. Авторизация и аутентификация:

- необходимо реализовать механизм сильной аутентификации при регистрации и входе в систему;

- должно быть обеспечено безопасность хранения паролей, включая соль и хеширование паролей;

- должен быть разработан механизм восстановления пароля с безопасным сбросом.

3. Защита от несанкционированного доступа:

- необходимо реализовать механизм защиты от несанкционированного доступа к учетным записям пользователей;

- должен быть ограничен доступ к конфиденциальным данным и функциональности на основе ролей.

4. Защита от межсайтовой подделки запросов (CSRF):

- необходима реализация механизмов защиты от CSRF-атак, таких как генерация и проверка токенов запросов.

5. Защита от межсайтовых сценариев (XSS):

- должна быть фильтрация и санитаризация ввода данных, чтобы предотвратить внедрение вредоносных сценариев.

6. Обновление и безопасность зависимостей:

- должны проводится регулярные обновления и патчи операционной системы, серверного ПО и сторонних библиотек;
- необходимо обеспечить мониторинг уязвимостей и оперативное реагирование на новые угрозы.

7. Резервное копирование и восстановление:

- необходимо регулярное создание резервных копий данных и настройка механизмов восстановления данных.

8. Сетевая безопасность:

- необходима настройка брандмауэра и контроля доступа на сетевом уровне;
- должна быть внедрена защита от DDoS-атак.

9. Аудит безопасности:

- должно быть регулярное проведение аудитов безопасности системы для выявления уязвимостей и оценки соответствия стандартам безопасности.

10. Санкции и мониторинг действий пользователей:

- должны быть разработаны меры дисциплинарной ответственности и мониторинга действий пользователей, включая их активность на платформе.

11. Защита от фишинга и мошенничества:

- должны быть реализованы меры для защиты пользователей от фишинговых атак и мошенничества.

Эти требования к безопасности помогут обеспечить надежную защиту для сервиса бронирования жилья и снизить риски связанные с потенциальными угрозами безопасности.

1.5 Требования к дизайну

Требования к дизайну и пользовательскому интерфейсу (UI/UX) играют важную роль в создании успешного сервиса бронирования жилья. Ниже представлены основные требования:

1. Общая концепция дизайна:

- дизайн должен соответствовать бренду сервиса и передавать его уникальную идентичность;

- дизайн должен учитывать интересы и потребности целевой аудитории, чтобы создать удовлетворительный пользовательский опыт.

2. Пользовательский интерфейс:

- главное меню, элементы управления и функциональные элементы должны быть легко доступны и понятны;

- дизайн должен быть адаптирован для мобильных устройств, планшетов и десктопов;

- обеспечение удобного отображение карты с жильем и его расположением;

- создание интуитивной формы для регистрации, бронирования и фильтрации жилья;

- возможность выбора параметров поиска (даты, местоположение, количества гостей) с минимальным усилием.

3. Графика и мультимедиа

- необходимо позволить пользователям загружать фотографии жилья и обеспечить их качественное отображение;

- необходимо использовать графику для облегчения понимания информации и украшения интерфейса;

- необходимо предоставить возможность добавления видео-туров и 360° просмотров жилья.

4. Цветовая палитра и типография

- необходимо определить основные и дополнительные цвета, соответствующие бренду и создающие приятное восприятие;

- необходимо выбрать шрифты для заголовков, текста и других элементов, обеспечивая читаемость.

5. Анимации и переходы

- должно быть реализовано использование анимации для плавных переходов между страницами и интерактивными элементами;

- необходимо обеспечить анимированные индикаторы загрузки для уведомления пользователя о процессе.

6. Удобство использования

- нужно дать пользователям возможность легко находить жилье с помощью удобного поиска и фильтров;
- необходимо предоставить пользователям инструкции и подсказки для использования сервиса;
- необходимо уменьшить количество необходимых кликов для выполнения задач.

Эти требования к дизайну и пользовательскому интерфейсу помогут создать удовлетворительный и привлекательный опыт для пользователей сервиса бронирования жилья.

1.6 Критерии завершения проекта

Завершение проекта разработки сервиса бронирования жилья Airbnb означает успешную реализацию всех поставленных целей и требований. Ниже приведены критерии и процедуры завершения проекта:

1. Функциональные требования. Все функциональные требования, описанные в Техническом Задании (ТЗ), должны быть реализованы и проверены. Все функции сервиса должны работать корректно и соответствовать ожиданиям пользователей.
2. Тестирование. Проведение тестирования всех компонентов и функций сервиса. Все выявленные ошибки и несоответствия должны быть устраниены и подтверждены тестами.
3. Безопасность. Необходимо убедиться в соответствии сервиса стандартам безопасности, включая защиту данных пользователей и их конфиденциальность.

4. Производительность. Необходимо убедиться, что сервис обладает приемлемой производительностью и отзывчивостью, и что он способен обслуживать ожидаемое количество пользователей без значительных задержек.

5. Масштабируемость. Подтвердить, что сервис готов к масштабированию, и что его инфраструктура может поддерживать увеличение нагрузки при необходимости.

6. Дизайн и Пользовательский Интерфейс. Проверить соответствие дизайна и интерфейса исходным дизайн-макетам и ожиданиям описанным в этом ТЗ.

7. Интеграции. Удостовериться, что все интеграции с платежными системами, картами и социальными сетями работают надежно.

8. Документация. Подготовить документацию, которая описывает работу и использование сервиса для администраторов, пользователей и разработчиков.

9. Обучение и поддержка. Обеспечить обучение для администраторов и службы поддержки по работе с сервисом. Подготовить планы для обслуживания и поддержки пользователей.

10. Оценка и отчетностью. Провести оценку проекта и составить отчет о его завершении, включая бюджетные расходы и затраты времени.

11. Публичный Запуск. Провести публичный запуск сервиса, уведомив о нем пользователей и общественность.

12. Мониторинг и поддержка. Начать мониторинг и поддержку сервиса после его запуска, решая вопросы, связанные с его работоспособностью и безопасностью.

13. Послепродажное обслуживание. Обеспечить обслуживание сервиса в соответствии с согласованными сроками и условиями.

2 USE CASE ДИАГРАММА



Рис. 2.1 Use case диаграмма

2.1 Описание основных вариантов использования:

2.1.1 Регистрация

Краткое описание: позволяет пользователям создать аккаунт в системе.

Основной поток:

- Пользователь выбирает опцию "Регистрация" на главной странице.
- Система отображает форму регистрации, где пользователь вводит свое имя, адрес электронной почты и пароль.
- Пользователь отправляет заполненную форму.
- Система проверяет введенные данные на валидность.
- Если данные верны, система создает учетную запись пользователя и перенаправляет его в систему.
- Если данные недопустимы (например, адрес электронной почты уже занят), система отображает сообщение об ошибке и предлагает исправить данные.

Альтернативные потоки:

- Если пользователь ввел недопустимый адрес электронной почты или пароль, система будет требовать исправления.
- Если адрес электронной почты уже занят, система может предложить восстановить пароль или войти в систему существующим аккаунтом.

2.1.2 Поиск жилья

Краткое описание: позволяет гостям искать доступное жилье по заданным критериям.

Основной поток:

- Пользователь выбирает опцию "Поиск жилья" на главной странице.
- Система отображает форму поиска с параметрами, такими как местоположение, даты пребывания и количество гостей.
- Пользователь вводит параметры поиска и нажимает кнопку "Поиск".

- Система отображает результаты поиска, предлагая список доступных вариантов жилья.

Альтернативные потоки:

- Если нет доступного жилья, система должна уведомить пользователя об этом.
- Пользователь может внести изменения в параметры поиска и выполнить новый поиск.

2.1.3. Добавление жилья

Краткое описание: позволяет хозяину жилья добавить информацию о своем жилье в систему.

Основной поток:

- Хозяин выбирает опцию "Добавление жилья" в своем личном кабинете.
- Система отображает форму для добавления информации о жилье, включая фотографии, описание, удобства, цены и календарь доступности.
- Хозяин вводит информацию и загружает фотографии.
- Система проверяет введенные данные на валидность.
- Если данные верны, система сохраняет информацию о жилье и добавляет его в список доступных вариантов.

Альтернативные потоки:

- Если введенные данные недопустимы или отсутствуют обязательные поля, система будет требовать исправления.
- Хозяин может редактировать информацию о жилье в будущем.

2.1.4. Бронирование жилья

Краткое описание: позволяет гостям бронировать жилье, которое соответствует их критериям поиска.

Основной поток:

- Гость выбирает жилье из результатов поиска или подробной информации о жилье.

- Гость указывает даты пребывания и количество гостей.
- Система проверяет доступность жилья на выбранные даты.
- Гость подтверждает бронирование и вводит платежные данные.
- Система обрабатывает платеж и подтверждает бронирование.
- Система отправляет уведомление хозяину и гостю с подтверждением бронирования.

Альтернативные потоки:

- Если жилье недоступно на выбранные даты, система уведомит гостя и предложит альтернативные даты или жилье.

2.1.5. Просмотр и оставление отзывов

Краткое описание: позволяет гостям оценивать и оставлять отзывы о месте проживания и хозяевах.

Основной поток:

- После окончания пребывания, гость может оценить жилье и оставить отзыв.
- Гость выбирает количество звезд и пишет комментарий.
- Система сохраняет отзыв и рейтинг в профиле гостя и хозяина.

Альтернативные потоки:

- Гость может не оставлять отзыв.

2.1.6. Управление ценами

Краткое описание: позволяет хозяину жилья устанавливать цены и акции.

Основной поток:

- Хозяин выбирает опцию "Управление ценами" в своем личном кабинете.
- Система отображает календарь с доступными датами и ценами.
- Хозяин может изменить цены и устанавливать скидки для определенных периодов.

Альтернативные потоки:

- Хозяин может оставить цены без изменений.

2.1.7 Управление пользователями

Краткое описание: позволяет администраторам системы управлять учетными записями пользователей.

Основной поток:

- Администратор выбирает опцию "Управление пользователями" в административной панели.
- Система позволяет администратору создавать, блокировать и удалять учетные записи пользователей, а также назначать им роли.

Альтернативные потоки:

- Администратор может разблокировать или снять блокировку с учетных записей.

2.1.8 Модерация объявлений

Краткое описание: позволяет администраторам контролировать и модерировать объявления о жилье, прежде чем они будут опубликованы.

Основной поток:

- Администратор проверяет новые объявления, которые были добавлены хозяинами.
- Администратор может одобрить или отклонить объявление на основе соответствия правилам и требованиям системы.

Альтернативные потоки:

- Если объявление не соответствует правилам или требованиям, администратор может отправить уведомление хозяину с запросом на исправление.

2.1.9 Решение конфликтов и жалоб

Краткое описание: позволяет пользователям (гостям и хозяинам) разрешать конфликты и подавать жалобы в случае проблем или недовольства.

Основной поток:

- Пользователь отправляет жалобу или описание конфликта через систему, предоставляя детали и доказательства.
- Система передает информацию администратору для рассмотрения.
- Администратор анализирует жалобу и принимает меры для разрешения конфликта, включая взаимодействие с пользователями.

Альтернативные потоки:

- Если конфликт не решен администратором, пользователь может обратиться в органы защиты прав потребителей или правоохранительные органы.

3 ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ

3.1. Пользовательский интерфейс (User Interface):

Пользовательский интерфейс (User Interface) — это часть системы, с которой пользователи взаимодействуют для выполнения различных действий и использования функциональности сервиса бронирования жилья, похожего на Airbnb. Этот компонент предоставляет доступ к системе через веб-сайт и мобильные приложения. Давайте подробнее разберем пользовательский интерфейс:

1. Веб-интерфейс:

Веб-сайт — это главный интерфейс, который пользователи могут использовать для доступа к сервису с компьютеров и мобильных устройств с веб-браузерами. Веб-интерфейс включает в себя:

- главную страницу с поисковой строкой и основными действиями, такими как вход в систему и регистрация;
- страницы с результатами поиска жилья, предоставляющие информацию о доступных вариантах;
- страницы с подробной информацией о конкретном жилье, включая фотографии, описание и отзывы;
- страницы для бронирования жилья, включая календарь доступности и расчет стоимости;
- личные кабинеты пользователей, где они могут управлять своими профилями, бронированиями и настройками уведомлений;
- страницы для управления жильем для хозяев, где они могут добавлять, редактировать и удалять объявления.

2. Мобильные приложения:

Мобильные приложения доступны для операционных систем iOS и Android и предоставляют аналогичный функционал, что и веб-сайт, но оптимизированы для мобильных устройств. Это позволяет пользователям бронировать жилье,

получать уведомления и управлять своими аккаунтами, используя смартфоны и планшеты.

3. Интерфейс для администраторов:

Для администраторов системы разрабатывается интерфейс для управления пользователями, мониторинга действий на платформе и модерации контента. Этот интерфейс позволяет администраторам следить за безопасностью и правопорядком на платформе.

4. Дизайн и пользовательский опыт (UI/UX):

Важная часть пользовательского интерфейса - это дизайн и пользовательский опыт. Интерфейс должен быть интуитивно понятным, приятным в использовании и соответствовать бренду сервиса. Важно уделять внимание деталям, таким как навигация, цветовая схема, типография и анимации.

5. Многоязычность и локализация:

Поскольку Airbnb поддерживает пользователей из разных стран и языков, интерфейс должен быть многоязычным и поддерживать локализацию, чтобы обеспечить комфортное использование сервиса для всех пользователей.

6. Доступность и адаптивность:

Интерфейс должен быть доступным для пользователей с ограниченными возможностями, а также адаптирован к различным типам устройств и разрешениям экранов.

7. Уведомления и коммуникация:

Взаимодействие с пользователями через уведомления, электронную почту и систему мгновенных сообщений.

3.2. Приложение сервера

Приложение сервера — это центральный компонент архитектуры системы бронирования жилья, подобной Airbnb. Этот компонент обладает высокой степенью

важности, так как в нем содержится основная бизнес-логика и обработка всех запросов от пользователей. Вот более подробное описание этого компонента:

1. Бизнес-логика

Этот компонент включает в себя бизнес-логику системы, включая логику поиска жилья, бронирования, управления жильем и пользовательскими учетными записями. Он обеспечивает обработку запросов от пользователей и выполнение различных сценариев использования, таких как поиск жилья, создание бронирования и управление профилем пользователя.

Кроме того, это место, где происходит взаимодействие с базой данных для получения и сохранения данных.

2. Управление сессиями и аутентификацией

- Для обеспечения безопасности и идентификации пользователей, сервер приложений управляет сессиями пользователей и аутентификацией.

- Пользователи аутентифицируются, и им выдаются токены доступа, которые используются для выполнения запросов и обеспечивают безопасность данных.

3. Управление состоянием

Сервер приложений отслеживает состояние системы и жилья, включая информацию о доступности жилья, ценах и бронированиях. Он также обрабатывает операции, связанные с управлением состоянием, такие как создание новых объявлений о жилье, редактирование информации и обновление календарей доступности.

4. Обработка бизнес-правил

В данном компоненте реализуются бизнес-правила, определяющие правила для расчета стоимости бронирования, учет скидок и наценок, а также другие параметры, связанные с ценообразованием и правилами использования.

5. Взаимодействие с другими компонентами

Сервер приложений взаимодействует с различными компонентами системы, такими как серверами баз данных для запросов и обновлений данных, серверами безопасности для проверки прав доступа, и серверами медиа-контента для получения изображений и медиа-файлов.

6. Безопасность

Важная задача сервера приложений - обеспечить безопасность системы. Он должен защищать данные, аутентификацию пользователей и обрабатывать запросы с осторожностью, чтобы предотвратить возможные угрозы, такие как атаки вроде SQL-инъекций.

7. Масштабируемость и производительность

Сервер приложений должен быть спроектирован с учетом возможности масштабируемости, чтобы обеспечить высокую доступность и производительность системы даже при росте числа пользователей и нагрузки. Это может быть достигнуто, например, путем использования микросервисной архитектуры и горизонтального масштабирования.

8. Реализация API

Сервер приложений предоставляет API, который позволяет клиентам (веб-интерфейсам, мобильным приложениям и интеграционным компонентам) взаимодействовать с системой. API определяет доступные эндпоинты, форматы запросов и ответов, а также правила авторизации.

9. Мониторинг и отладка

Сервер приложений должен быть настроен на мониторинг производительности и регистрацию событий, чтобы обеспечить возможность отладки и анализа работы системы.

10. Логирование

Чтобы отслеживать действия пользователей и операции в системе, сервер приложений должен вести лог событий, который может использоваться для анализа и аудита.

3.3. База данных

База данных (Database) - это один из наиболее важных компонентов системы для сервиса бронирования жилья, подобного Airbnb. Она отвечает за хранение и управление

всей информацией, необходимой для функционирования системы. Ниже представлена более подробная информация о базе данных:

В контексте бронирования жилья используется как реляционная база данных (RDBMS), а именно MySQL, так и NoSQL база данных, а именно MongoDB.

Реляционные базы данных хорошо подходят для хранения структурированных данных, таких как информация о пользователях, бронированиях, жилье и отзывах. Они обеспечивают сильную целостность данных и могут быть масштабированы с использованием кластеризации.

NoSQL базы данных могут быть полезны для хранения полуструктурных данных, таких как медиа-контент (фотографии, видео) и большие объемы данных о доступности жилья.

Структура реляционной базы данных включает в себя несколько основных таблиц, связанных между собой. Таблицы в реляционной базе данных:

- Users (Пользователи): Хранит информацию о зарегистрированных пользователях, включая их имена, адреса электронной почты, пароли и другие сведения;
- Listings (Объявления): Содержит информацию о доступном жилье, включая его описание, цены, местоположение и фотографии.
- Bookings (Бронирования): хранит информацию о бронированиях жилья, включая даты, количество гостей и связанные с ними детали.
- Reviews (Отзывы): содержит отзывы пользователей о жилье и хозяевах.
- Payments (Платежи): хранит информацию о финансовых транзакциях, связанных с бронированиями, включая суммы и статус оплаты.:

Для обеспечения быстрого доступа к данным, база данных использует индексы на ключевых полях, таких как идентификаторы пользователей, идентификаторы жилья и даты бронирований. Оптимизация запросов и структуры базы данных играет важную роль в обеспечении высокой производительности системы.

Кроме того, важно уделять внимание регулярному созданию резервных копий базы данных и ее восстановлению в случае сбоев или потери данных. Резервные копии хранятся в облачных хранилищах.

Безопасность данных является критически важной частью базы данных. Данные пользователей должны быть зашифрованы, пароли хэшированы, и доступ к базе данных должен быть ограничен только необходимым сотрудникам. Также важно регулярно обновлять систему и базу данных, чтобы закрыть известные уязвимости.

3.4 Сервер платежей

Сервер платежей является критически важным компонентом для системы бронирования жилья, подобной Airbnb, так как обеспечивает обработку финансовых транзакций между гостями и хозяевами жилья. Вот более подробное описание этого компонента:

1. Взаимодействие с платежными системами: этот сервер взаимодействует с внешними платежными системами, такими как Stripe, PayPal, Braintree и другими. Он отправляет запросы на авторизацию и проведение платежей к соответствующим платежным шлюзам.
2. Обработка платежей и возвратов: Payment Server отслеживает статусы платежей и обрабатывает как успешные транзакции, так и возвраты средств. Он также должен поддерживать обработку различных видов платежей, включая кредитные и дебетовые карты, электронные кошельки и другие способы оплаты.
3. Безопасность: поскольку финансовые данные клиентов являются чрезвычайно чувствительными, сервер платежей должен строго соблюдать стандарты безопасности, включая стандарт PCI DSS (Payment Card Industry Data Security Standard), чтобы гарантировать защиту данных и предотвращение мошенничества.
4. Учет и отчетность: сервер платежей должен учитывать все финансовые операции, сохраняя детализированные записи о платежах и возвратах. Это необходимо для учета доходов и подготовки отчетов, а также для удовлетворения налоговых требований.

5. Управление транзакциями: сервер платежей обеспечивает управление транзакциями, включая подтверждение платежей, отслеживание статусов и решение возможных проблем, связанных с платежами.

6. Многоязычная и мультивалютная поддержка: для мирового приложения, как Airbnb, важно поддерживать множество валют и языков для удовлетворения потребностей разных пользователей.

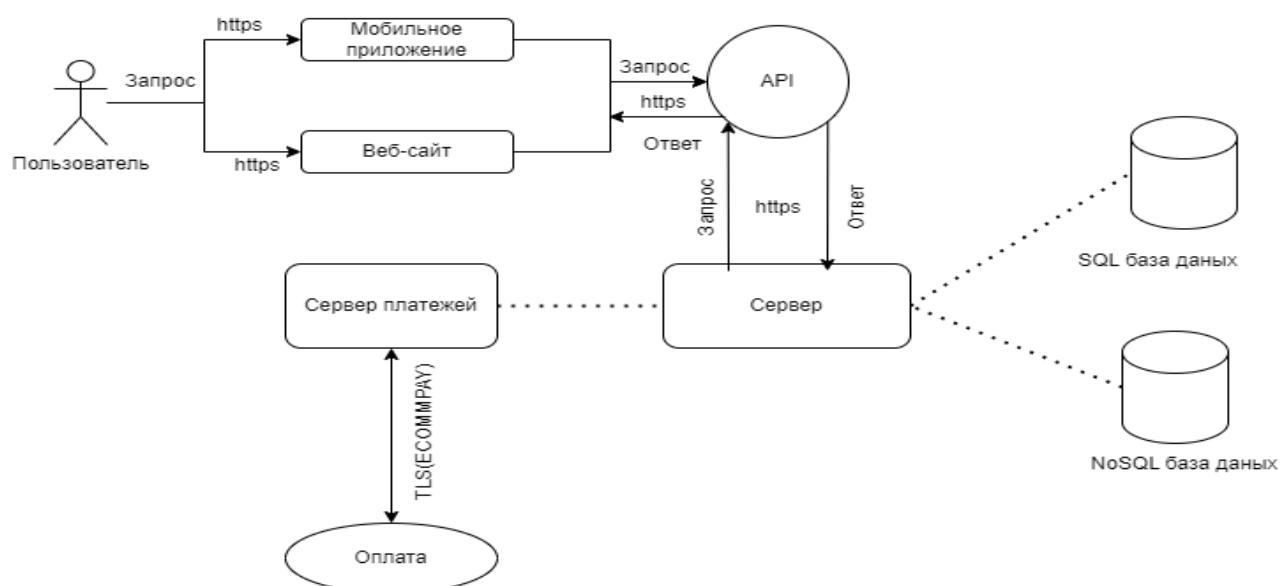
7. Управление сборами и комиссиями: в системе, подобной Airbnb, сервис может взымать сборы и комиссии с каждой успешной транзакции. Сервер платежей должен автоматически вычислять и удерживать соответствующие суммы.

8. Обработка отказов и конфликтов: Payment Server также отвечает за разрешение возможных споров и конфликтов между гостем и хозяином, связанных с оплатой и бронированием.

9. Интеграция с системой уведомлений: сервер платежей интегрируется с системой уведомлений для информирования пользователей о статусе и результатах платежей.

Важно отметить, что сервер платежей играет важную роль в обеспечении безопасности и надежности финансовых операций в системе. Его работа должна соответствовать мировым стандартам безопасности платежей, и он должен регулярно обновляться и аудитироваться для обеспечения защиты данных и пользователей.

Схема архитектуры системы:



4 ДИАГРАММА КЛАССОВ И ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ

4.1 Диаграмма классов

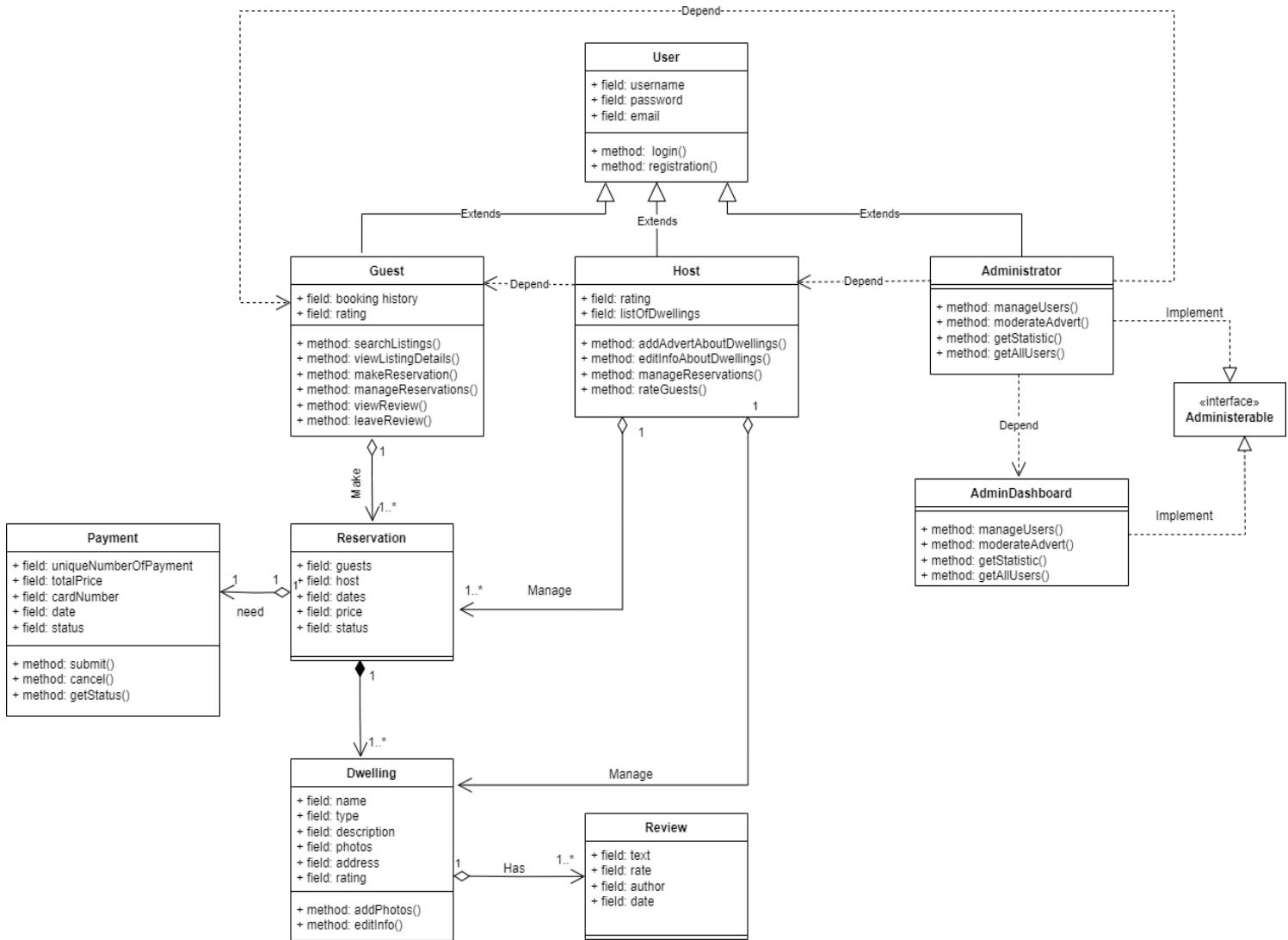


Рис 4.1 – Диаграмма классов

4.2 Последовательность действий

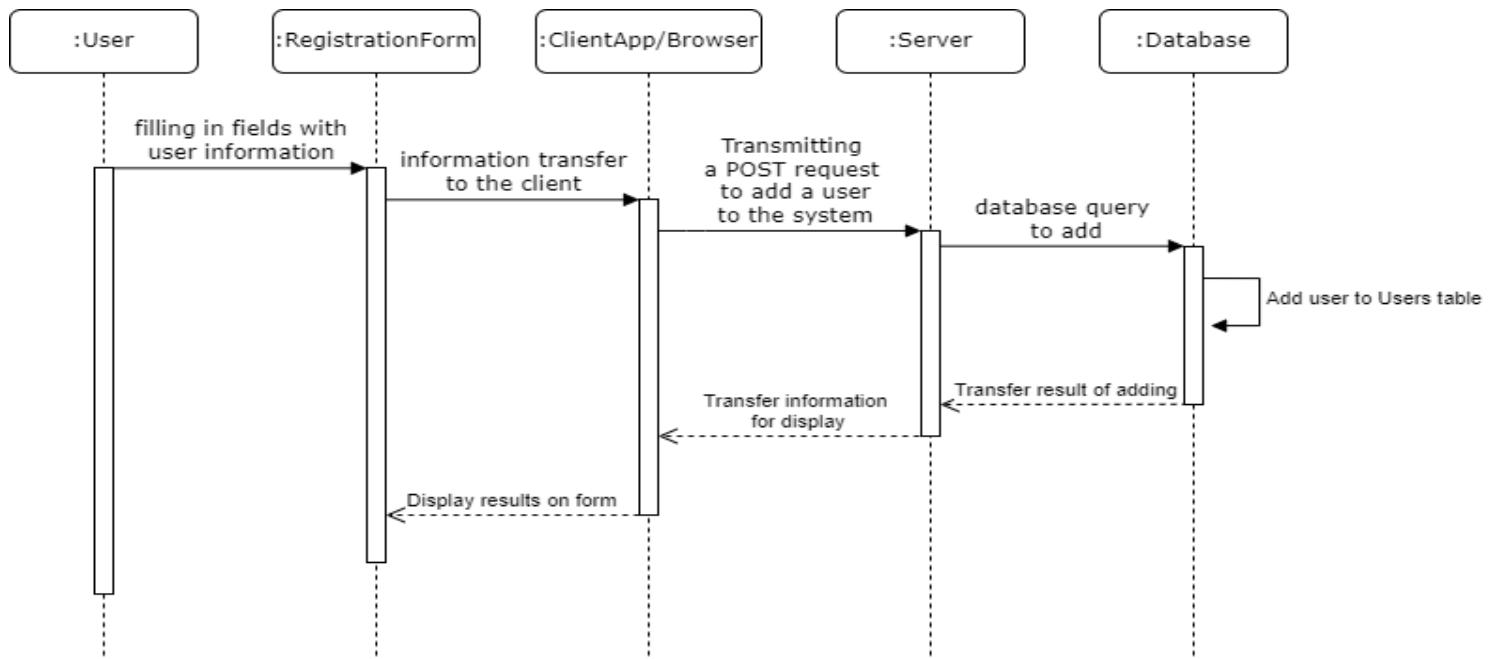


Рис 4.2 – Диаграмма последовательности для варианта
использование Регистрация в системе

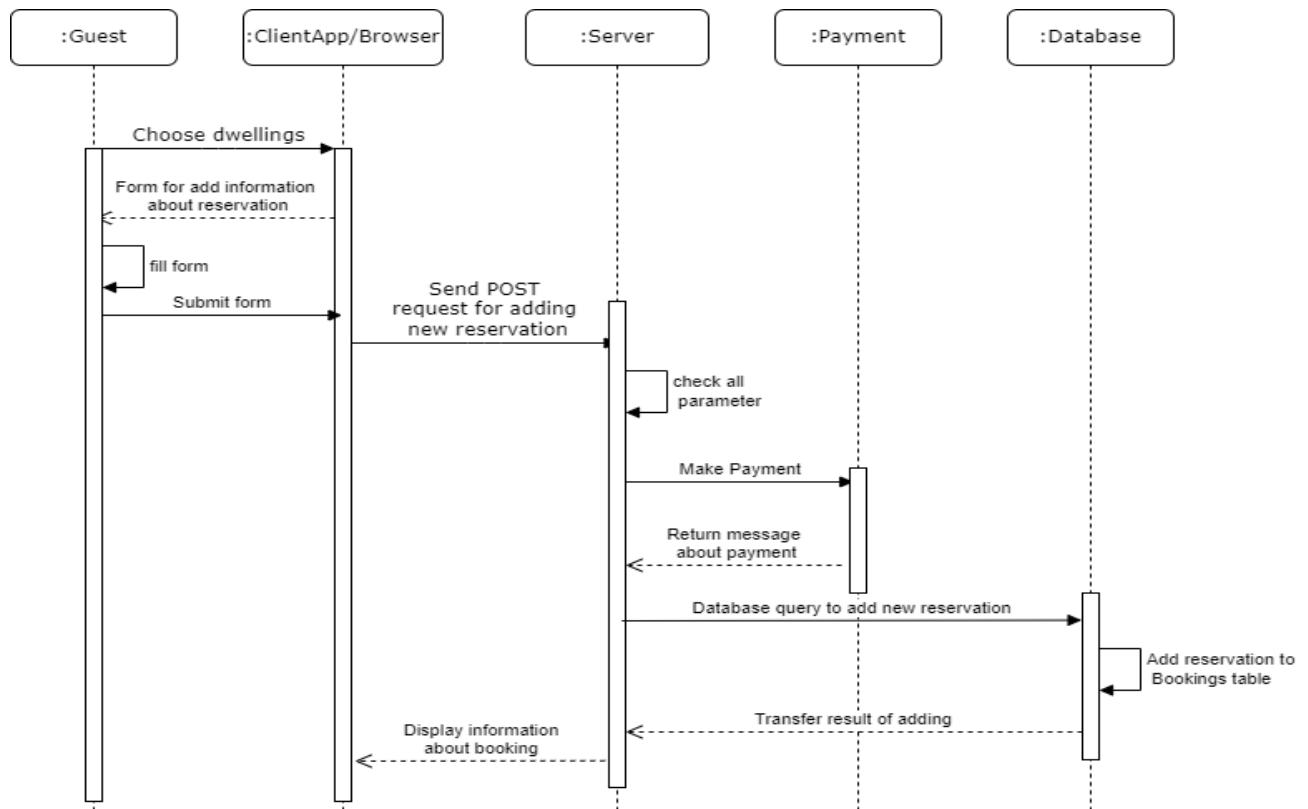


Рис 4.3 – Диаграмма последовательности для варианта использования
Бронирование жилья

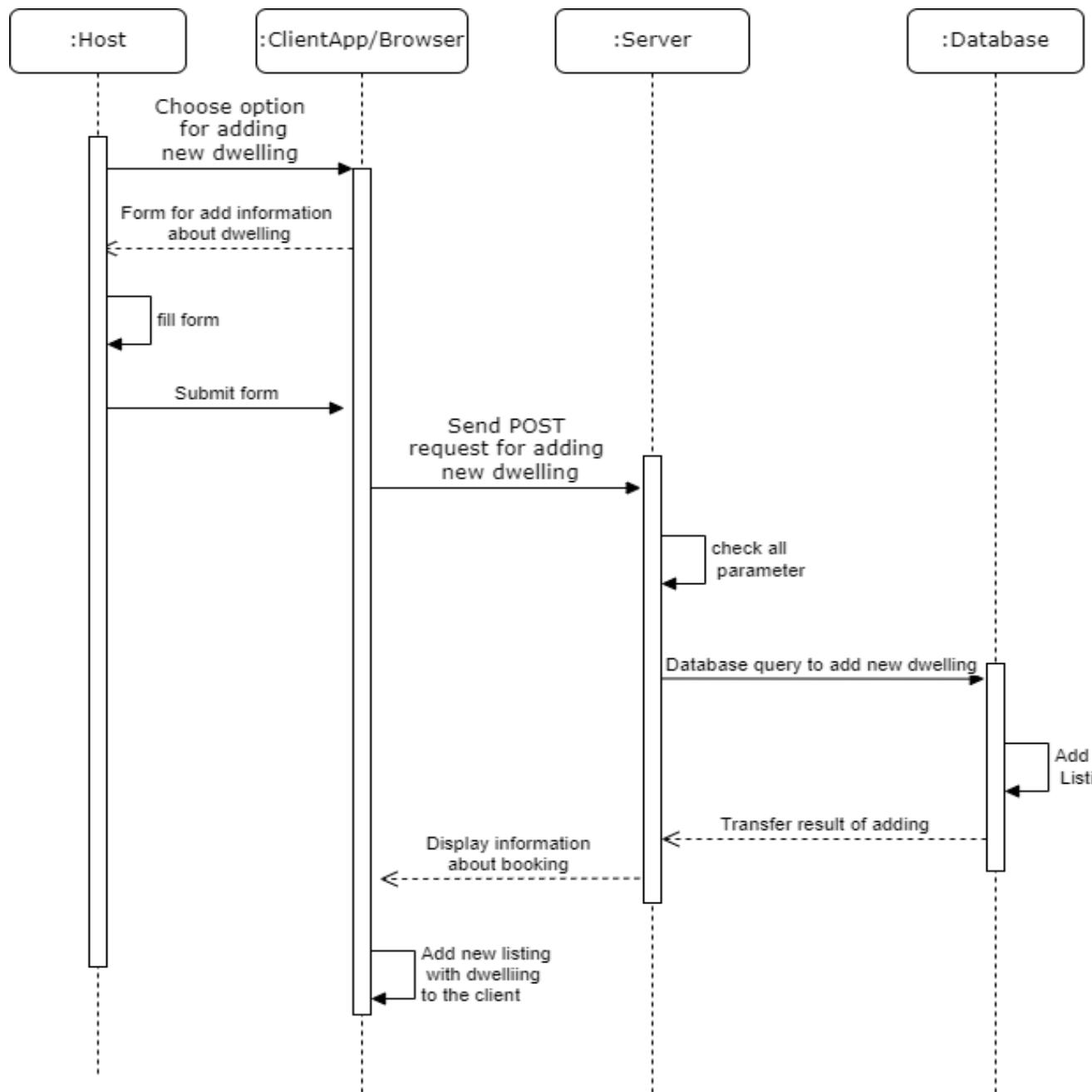


Рис 4.4 – Диаграмма последовательности для варианта использования
Добавление нового жилья Хозяином

ЗАКЛЮЧЕНИЕ

На протяжении данной работы были созданы различные диаграммы и технические документы для предметной области сервиса бронирования жилья, аналогичного Airbnb. Эти документы могут служить основой для разработки и внедрения подобного сервиса. Вот краткое подведение итогов работы:

1. Техническое задание (ТЗ): было представлено ТЗ, описывающее ключевые требования к функциональности, безопасности, производительности, масштабируемости и дизайну сервиса. ТЗ является важным документом, который определяет общие цели и требования проекта.

2. Use Case Диаграмма: была создана use case диаграмма, идентифицирующая актеров и сценарии использования в системе бронирования жилья. Это помогает понять, как различные пользователи будут взаимодействовать с системой.

3. Диаграмма Классов: была создана диаграмма классов, описывающая классы, их свойства, методы и взаимосвязи в системе. Эта диаграмма помогает структурировать объектную модель системы.

4. Диаграммы Последовательности: были созданы диаграммы последовательности для сценария "Регистрация", "Бронирование жилья" и "Добавление жилья". Это позволило визуализировать последовательность действий между актерами и системой.

Эти документы и диаграммы являются важными инструментами для разработки и внедрения сервиса бронирования жилья, а также для обеспечения понимания и согласования требований с заказчиком и командой разработки. Они могут быть использованы как отправная точка для создания проекта и дальнейшего его развития.