**Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**

**Лабораторная работа №1**

**вариант - Airbnb**

Выполнил:

Брагин Роман Андреевич

Проверил:

Райла Мартин

г. Санкт-Петербург

2025

**Software Requirements Specification**

**(Спецификация требований к продукту)**

**“Airbnb”**

**1.Introduction (Введение)**

**1.1. Purpose**

Настоящий документ определяет функциональные и нефункциональные требования к платформе Airbnb. Он используется как руководство для проектирования, разработки, тестирования и внедрения системы, а также для согласования ожиданий между всеми участниками проекта, включая заказчиков, разработчиков, тестировщиков и других заинтересованных сторон.

**1.2. Scope (Область применения)**

Документ определяет требования к разработке и совершенствованию платформы Airbnb — онлайн-сервиса для поиска, бронирования и управления арендой жилья. Ключевые функции системы:

1.2.1 Поиск и бронирование жилья: Пользователи могут находить объекты аренды, используя фильтры (локация, даты, цена, удобства), и бронировать их.

1.2.2 Управление бронированиями: Хозяины могут редактировать информацию о своих объектах, а гости — управлять своими бронированиями.

1.2.3 Оплата: Интеграция с платежными системами для безопасного проведения транзакций.

1.2.4 Коммуникация: Встроенный чат для общения между Хозяевами и гостями.

1.2.5 Рейтинги и отзывы: Возможность оставлять отзывы и оценки после завершения бронирования.

1.2.6 Целевая аудитория:

1.2.6.1 Гости: Пользователи, которые ищут и бронируют жилье для временного проживания.

1.2.6.2 Хозяины: Владельцы недвижимости, сдающие свои объекты через платформу.

1.2.6.3 Администраторы: Сотрудники Airbnb, отвечающие за поддержку и управление платформой.

1.2.7 Границы системы:

1.2.7.1 В функционал системы входит:

1.2.7.1.1 Поиск и бронирование жилья.

1.2.7.1.2 Управление объектами аренды и бронированиями.

1.2.7.1.3 Интеграция с платежными системами.

1.2.7.1.4 Коммуникация между пользователями.

1.2.7.1.5 Система рейтингов и отзывов.

1.2.7.2 В функционал системы не входит:

1.2.7.2.1 Управление персоналом компании Airbnb.

1.2.7.2.2 Финансовый учет и отчетность для Хозяинов

1.2.7.2.3 Физическое обслуживание объектов аренды (например, уборка или ремонт).

**1.3. Definitions, Acronyms and Abbreviations (Определения и аббревиатуры)**

В данном документе используются следующие термины, аббревиатуры и определения:

1. Airbnb – онлайн-платформа для поиска, бронирования и управления арендой жилья.
2. Гость (Guest) – пользователь, который ищет и бронирует жилье для временного проживания.
3. Хозяин (Host) – пользователь, который сдает свое жилье через платформу Airbnb.
4. Объект аренды (Listing) – жилье или помещение, предлагаемое для аренды на платформе.
5. Бронирование (Booking) – процесс резервирования объекта аренды гостем на определенные даты.
6. Чат (Chat) – встроенный инструмент для общения между гостями и Хозяевами.
7. Отзыв (Review) – оценка и комментарий, оставленные пользователем после завершения бронирования.

Технические термины и аббревиатуры

1. ReactJS – библиотека JavaScript для создания пользовательских интерфейсов.
2. JWT (JSON Web Token) – стандарт для создания токенов, используемых для аутентификации и авторизации.
3. Redux – библиотека для управления состоянием приложения на стороне клиента.
4. Java – язык программирования, используемый для серверной разработки.
5. Spring – фреймворк для разработки приложений на Java.
6. Ruby – язык программирования, используемый для серверной разработки.
7. Ruby on Rails – фреймворк для разработки веб-приложений на Ruby.
8. Python – язык программирования, используемый для обработки данных и аналитики.
9. Swift – язык программирования для разработки приложений под iOS.
10. Kotlin – язык программирования для разработки приложений под Android.
11. GraphQL – язык запросов для API, позволяющий клиентам запрашивать только необходимые данные.
12. PostgreSQL – реляционная база данных для хранения структурированных данных (пользователи, бронирования, объекты аренды и т.д.).
13. Redis – база данных в памяти, используемая для кэширования данных.
14. Google Cloud Storage – облачное хранилище для хранения файлов, таких как изображения и документы.
15. Apache Kafka – распределенная система для обработки потоков данных и обмена сообщениями между сервисами.
16. API (Application Programming Interface) – интерфейс для взаимодействия между различными компонентами системы.
17. MIT License – лицензия на свободное программное обеспечение, разрешающая использование, модификацию и распространение кода.
18. Apache License 2.0 – лицензия на свободное программное обеспечение, разрешающая использование, модификацию и распространение кода с указанием авторства.
19. BSD 3-Clause – лицензия на свободное программное обеспечение, разрешающая использование, модификацию и распространение кода с минимальными ограничениями.
20. RSAL (Redis Source Available License) – лицензия, ограничивающая использование Redis в некоторых коммерческих продуктах.
21. GitHub – платформа для Хозяининга и совместной разработки программного кода.
22. Google Maps – сервис карт и геолокации от Google
23. Google Translate – сервис автоматического перевода текста от Google.
24. SSD (Solid State Drive) – тип накопителя для хранения данных.
25. CPU (Central Processing Unit) – центральный процессор, основной компонент для обработки данных.
26. SMS (Short Message Service) – сервис для отправки текстовых сообщений.
27. REST API (Representational State Transfer API) – архитектурный стиль для создания веб-сервисов.
28. HTTP/HTTPS (HyperText Transfer Protocol/Secure) – протоколы для передачи данных в сети.

Дополнительные термины

1. Кэширование (Caching) – процесс временного хранения данных для ускорения доступа к ним.
2. Микросервисы (Microservices) – архитектурный подход, при котором приложение состоит из небольших независимых сервисов.
3. Аутентификация (Authentication) – процесс проверки подлинности пользователя.
4. Авторизация (Authorization) – процесс предоставления пользователю доступа к определенным ресурсам.

Платежные системы и сервисы

1. VISA – международная платежная система для проведения транзакций.
2. MasterCard – международная платежная система для проведения транзакций.
3. Amex (American Express) – международная платежная система для проведения транзакций.
4. Discover – платежная система, популярная в США.
5. PayPal – онлайн-платежная система для проведения транзакций.
6. Google Pay – мобильная платежная система от Google.

**1.4. References (Ссылки)**

1. Многое было просмотрено и взято у Microsoft из этого [источника.](https://learn.microsoft.com/ru-ru/windows/)
2. Шаблон был взят из этого [источника.](https://docs.google.com/document/d/1aneQQb_lH27JNSE5U6KMmOIhJgpHCXbW/edit)
3. Полезная [ссылка](https://habr.com/ru/companies/itsumma/articles/786228/)
4. ReactJS (MIT License):

[Официальный репозиторий ReactJS на GitHub](https://github.com/facebook/react)

[Лицензия MIT](https://opensource.org/licenses/MIT)

1. Java и Spring Framework:

[Лицензия Oracle Binary Code License Agreement](https://www.oracle.com/java/technologies/javase/jdk-license.html)

[Spring Framework License (Apache 2.0)](https://spring.io/projects/spring-framework)

5. Ruby on Rails (MIT License):

[Официальный репозиторий Ruby on Rails на GitHub](https://github.com/rails/rails)

6. Python (Python Software Foundation License):

[Официальный сайт Python](https://www.python.org/)

[Python License](https://docs.python.org/3/license.html)

7. Swift (Apache License 2.0):

[Официальный репозиторий Swift на GitHub](https://github.com/apple/swift)

[Лицензия Apache 2.0](https://www.apache.org/licenses/LICENSE-2.0)

8. Kotlin (Apache License 2.0):

[Официальный сайт Kotlin](https://kotlinlang.org/)

9. GraphQL:

[Официальный сайт GraphQL](https://graphql.org/)

[Apollo Server (MIT License)](https://github.com/apollographql/apollo-server)

10. PostgreSQL (PostgreSQL License):

[Официальный сайт PostgreSQL](https://www.postgresql.org/)

[PostgreSQL License](https://www.postgresql.org/about/licence/)

11. Redis (BSD 3-Clause и RSAL):

[Официальный сайт Redis](https://redis.io/)

[Лицензия BSD 3-Clause](https://redis.io/docs/about/license/)

[Redis Source Available License (RSAL)](https://redis.com/legal/licenses/)

12. Google Cloud Storage:

[Условия использования Google Cloud Platform](https://cloud.google.com/terms/)

13. Apache Kafka (Apache License 2.0):

[Официальный сайт Apache Kafka](https://kafka.apache.org/)

**1.5. Overview (Обзор документа)**

Software Requirements Specification (Спецификация требований к продукту)

**1. Введение (Introduction)**

1.1 Назначение документа (Purpose)

1.2 Область применения (Scope)

1.3 Определения и аббревиатуры (Definitions, Acronyms and Abbreviations)

1.4 Ссылки (References)

1.5 Обзор документа (Overview)

**2. Общее описание (Overall Description)**

2.1 Функционал продукта (Product functions)

2.2 Описание пользователей (User characteristics)

2.3 Влияющие факторы и зависимости (Assumptions and dependencies)

2.4 Ограничения (Constraints)

**3. Спецификация требований (Specific Requirements)**

3.1 Функциональные требования (Functionality)

3.2 Требования к удобству использования (Usability)

3.3 Требования к надежности (Reliability)

3.4 Требования к производительности (Performance)

3.5 Ограничения разработки (Design Constraints)

3.6 Интерфейсы (Interfaces)

3.6.1 Пользовательские интерфейсы (User Interfaces)

3.6.2 Аппаратные интерфейсы (Hardware Interfaces)

3.6.3 Программные интерфейсы (Software Interfaces)

3.6.4 Сетевые интерфейсы (Communications Interfaces)

3.7 Требования к лицензированию (Licensing Requirements)

**4. Requirements Attributes (Атрибуты требований)**

4.1 Функциональные требования (Functionality)

4.2 Нефункциональные требования (Non-Functionality)

**5. Functional modeling (Функциональное моделирование)**

5.1 Use case диаграмма (Use case diagram)

5.2 Прецеденты системы (Use cases of the system)

**2.** **Overall Description (Общее описание)**

**2.1. Product functions (Функционал продукта)**

2.1.1Просмотр карточек жилья, а также их подробного описания с картинками, отзывами, сроком аренды, ценой и профилем арендодателя.

2.1.2 Поиск жилья по заданному региону, стране, городу. Причем начальные карточки жилья должны предлагаться на основе местоположения пользователя.

2.1.3 Поиск жилья по заданному промежутку желаемых дат.

2.1.4 Поиск жилья с учетом возраста и количества гостей.

2.1.5 Возможность выбрать удобный язык (или включить автоперевод) для всего интерфейса и отзывов.

2.1.6 Возможность авторизоваться через Google, Apple, Facebook, электронную почту, номер телефона.

2.1.7 Возможность поделиться жильем/ мероприятием.

2.1.8 Поиск жилья по указанным удобствам (телевизор, бесплатная парковка и т.п.)

2.1.9 Поиск по типу жилья (замок, комната, дом, квартира и т.п.)

2.1.10 Просмотр карты с доступными вариантами жилья и указанными ценами на каждом из них.

2.1.11 Возможность оплаты через VISA, MasterCard, Amex, Discover, PayPal, Google Pay.

2.1.12 Возможность оплаты по частям.

2.1.13 Выбор жилья по удобствам доступной среды (парковочное место, удобства для людей с инвалидностью).

2.1.14 Расчет стоимости аренды за выбранный промежуток аренды.

2.1.15 Возможность увидеть стоимость до уплаты налогов.

2.1.16 Выбор арендодателя по языкам, которыми он владеет.

2.1.17 Возможность оплаты через сайт, и выбор удобной валюты.

2.1.18 Возможность забронировать различные виды активного отдыха.

2.1.19 Широкий фильтр (город, тип природы и т.п.)

2.1.20 Список рекомендации.

2.1.21 Добавление жилья/ мероприятия в избранное

2.1.22 Выставление своих мероприятий и жилья.

2.1.21 Просмотр истории ранее просмотренных вариантов жилья и мероприятий.

2.1.22 Просмотр платежей и управление транзакциями, налоговой информацией.

2.1.22 Аннулирование (удаление) аккаунта.

2.1.23 Инструменты для управления выставленными жильем и мероприятиями.

2.1.24 Управление параметрами своего аккаунта (конфиденциальность, почта, адрес, деловой адрес).

2.1.25 Управление сообщениями.

2.1.26 Система реферальных бонусов и купонов.

2.1.27 Система оценивания арендодателя.

2.1.28 Система рейтинга арендодателя и услуг которые он предоставляет.

**2.2. User characteristics (Описание пользователей)**

Неавторизованные пользователи - пользователи, которые либо не имеют аккаунт, или не вошли в него. Они взаимодействуют с системой так:

1. Просматривают доступные варианты жилья и мероприятий, а также их описания, отзывы и мероприятия на карте или в списке предложений.
2. Деляться мероприятием или жильем.
3. Фильтруют предложенные варианты по своим критериям.
4. Выбирают удобны язык и валюту.
5. Смотрят цену до налогов и после.
6. Смотрят цену за выбранный промежуток.
7. Выбирают интересующий их тип жилья или мероприятия.
8. Авторизуются для продолжения.

Авторизованные пользователи - эти пользователи делятся в свою очередь на 2 категории - гости и хозяева.

Гости - пользователи, которые хотят арендовать жилье или забронировать мероприятие или активным отдыхом. Они могут взаимодействуют с системой так же неавторизованные пользователи, и еще:

1. Бронируют жилье или мероприятие на определенное кол-во людей и на интересующие даты.
2. Пишут отзывы (комментарии), выставляют оценки.
3. Управляют бронированиями, включая изменением даты и отмену.
4. Добавляют жилье и мероприятия в избранное.
5. Добавляют/ Изменяют данные об аккаунте.
6. Оплачивают жилье или мероприятие удобными им способами.
7. Пользуются реферальными программами и бонусами.
8. Смотрят вишлисты (история просмотренных мероприятий и жилья)

Хозяева - пользователи (арендодатели), которые выставляют жилье и/или мероприятия.Они могут взаимодействуют с системой так же авторизованные пользователи, и еще:

1. Размещают жилье и/ мероприятия, для этого подробно описывают их согласно критериям.
2. Настраивают параметры жилья и мероприятий, включая цену, доступность, удобства, правила проживания.
3. Выбирают предпочитаемые способы оплаты и настройки налогов.
4. Коммуницируют с гостями.
5. Управляют бронированиями (принимают или отклоняют).
6. Управляют выставленными мероприятиями и жильем.
7. Арендодатели могут выставлять свои ограничения, которые не противоречат правилам платформы.

**2.3. Assumptions and dependencies (Влияющие факторы и зависимости)**

1. Регистрация и авторизация (3.1.2)

Зависимость от Google, Facebook, Apple: Для удобства пользователей система должна поддерживать регистрацию и авторизацию через аккаунты в социальных сетях. Это требует интеграции с API этих платформ.

Соответствие GDPR: Система обязана запрашивать согласие пользователей на обработку их персональных данных, а также предоставлять возможность удаления аккаунта и данных.

#### 2. Просмотр карты (3.1.4)

Зависимость от Google Maps: Для отображения объектов на карте, построения маршрутов и интеграции с фильтрами система должна использовать API Google Maps. Это обеспечивает корректное отображение и функциональность карты.

#### 3. Локализация и многоязычность (3.1.10)

Зависимость от Google Translate: Для автоматического перевода интерфейса и сообщений система должна интегрироваться с API Google Translate. Это позволяет поддерживать несколько языков и упрощает коммуникацию между пользователями.

#### 4. Инструменты для хозяев (3.1.12)

Зависимость от местных законов: Система должна адаптироваться к законодательным требованиям разных стран и городов, включая налоги, лицензии и ограничения на срок аренды. Это требует проверки объявлений на соответствие местным правилам.

Примеры:

1. В некоторых городах, таких как Нью-Йорк или Сан-Франциско, хозяева обязаны зарегистрироваться или получить лицензию для сдачи жилья на короткий срок. Airbnb интегрирует эти требования в свои процессы, проверяя, соответствуют ли объявления местным законам. Например, в Нью-Йорке запрещено сдавать жилье на срок менее 30 дней без специального разрешения, и Airbnb блокирует такие объявления. [ссылка](https://www.tourdom.ru/hotline/konflikty/megapolisy-protiv-airbnb/)
2. В некоторых регионах Airbnb автоматически рассчитывает, собирает и перечисляет налоги на проживание от имени хозяев. Например, в Японии хозяева должны соблюдать Закон о частном жилье , который регулирует налогообложение и лицензирование. Airbnb предоставляет информацию о таких требованиях и помогает хозяевам соблюдать их. [ссылка](https://www.airbnb.ru/help/article/920)
3. В Берлине и Барселоне действуют строгие ограничения на краткосрочную аренду. Airbnb адаптирует свои правила, чтобы объявления соответствовали этим ограничениям. Например, в Берлине запрещено сдавать жилье на срок менее 30 дней без специального разрешения, и Airbnb удаляет объявления, нарушающие это правило.

#### 5. Модерация контента (3.1.14)

Зависимость от законов и репутации: Система должна проверять объявления, фотографии, видео и отзывы на соответствие правилам платформы и отсутствие запрещенного контента. Это включает автоматическую и ручную модерацию.

#### 6. Система рекомендаций (3.1.7)

Зависимость от вычислительных мощностей: Для предоставления персонализированных рекомендаций система должна анализировать историю поиска и бронирований пользователей.

#### 7. Система поддержки пользователей (3.1.11)

Система должна включать раздел FAQ, который будет автоматически обновляться на основе анализа наиболее частых запросов пользователей.

Система должна предоставлять функционал онлайн-чата с поддержкой, включая возможность автоматического ответа на простые запросы (чат-бот) и переключения на живого оператора при необходимости.

Система должна предоставлять номер телефона для связи с поддержкой, а также возможность автоматического перенаправления звонка на соответствующего специалиста.

**2.4. Constraints (Ограничения)**

#### Зарегистрироваться можно только с 18 лет.

### Влияние на функционал:

### Система проверяет возраст пользователя при регистрации.

### Блокирует регистрацию для пользователей младше 18 лет.

### Последствия несоблюдения:

### Аккаунт пользователя младше 18 лет блокируется.

### Все бронирования отменяются.

### Пользователь не сможет восстановить аккаунт до достижения 18 лет.

#### Запрещено выставлять фальшивые объявления.

### Влияние на функционал:

### Система проверяет объявления на достоверность (фото, описание, контактные данные).

### Внедрена модерация объявлений.

### Последствия несоблюдения:

### Фальшивые объявления удаляются.

### Хозяин получает предупреждение или блокировку аккаунта.

### Пользователи получают полный возврат средств.

### Хозяин может быть оштрафован или привлечен к юридической ответственности.

#### Запрещено использовать ненормативную лексику.

### Влияние на функционал:

### Система фильтрует тексты (описания, отзывы, сообщения) на наличие ненормативной лексики.

### Последствия несоблюдения:

### Тексты с ненормативной лексикой блокируются или удаляются.

### Пользователь получает предупреждение или блокировку аккаунта.

### Отзывы с ненормативной лексикой удаляются, и пользователь теряет возможность оставлять отзывы.

#### Один пользователь может иметь только один аккаунт.

### Влияние на функционал:

### Система проверяет уникальность аккаунтов (через email, телефон, соцсети).

### Последствия несоблюдения:

### Дополнительные аккаунты блокируются.

### Пользователь получает уведомление о необходимости использовать один аккаунт.

### При повторных нарушениях основной аккаунт блокируется.

#### В разных странах и городах действуют свои правила аренды жилья.

### Влияние на функционал:

### Система учитывает местные законы (налоги, лицензии, ограничения на срок аренды).

### Последствия несоблюдения:

### Объявления, нарушающие местные правила, удаляются.

### Хозяин может быть оштрафован местными властями.

### Платформа может быть заблокирована в определенных регионах.

### Пользователи получают полный возврат средств.

### 

**3. Specific Requirements (Спецификация требований)**

**3.1 Functionality (Функциональные требования)**

Требования пользователей сайта:

### 3.1.2 Регистрация и Авторизация: Система должна предоставлять возможность:

1. Создание аккаунта:
   1. Платформа должна позволять пользователям создавать аккаунт через стандартную форму регистрации с указанием личных данных (имя, фамилия, электронная почта, пароль).
   2. Также возможна регистрация с использованием аккаунтов в социальных сетях: Google, Facebook,Apple
2. Подтверждение электронной почты:
   1. После регистрации, система отправляет письмо с подтверждением на указанный электронный адрес.
   2. Для завершения регистрации необходимо перейти по ссылке в этом письме, подтверждая свой адрес электронной почты.
3. Подтверждение номера телефона:Для повышения безопасности и улучшения взаимодействия с пользователем, система требует ввести номер телефона.
   1. После ввода номера телефона на указанный номер отправляется SMS с кодом подтверждения, который необходимо ввести на платформе.
4. Авторизация через социальные сети:
   1. Система должна поддерживать возможность авторизации через аккаунты в Google, Facebook или Apple.
   2. При авторизации через социальные сети пользователю не требуется вводить пароль и логин, так как доступ осуществляется через привязанные аккаунты.
5. Подтверждение личности:
   1. Для повышения безопасности и доверия, система может требовать верификации личности пользователя.
   2. Это может включать загрузку фотографии документа, удостоверяющего личность, и/или селфи, чтобы подтвердить, что пользователь соответствует предоставленным данным.
6. Дополнительные меры безопасности:
   1. При подозрении на недобросовестные действия система может запросить дополнительные меры безопасности, такие как двухфакторная аутентификация или подтверждение через другие каналы.

3.1.2 Просмотр карточек жилья/мероприятий :Система должна корректно отображать карточки объектов (жилья или мероприятий) и предоставлять пользователю возможность просмотра подробной информации о каждом объекте. Карточка должна включать следующие элементы:

1. Фотографии и медиа
   1. Галерея фотографий объекта с возможностью просмотра в полноэкранном режиме.
   2. Поддержка видео или 3D-туров для более детального ознакомления с объектом.
2. Отзывы и рейтинги
   1. Отзывы предыдущих гостей с указанием даты, текста отзыва и рейтинга.
   2. Общий рейтинг объекта на основе оценок гостей.
   3. Возможность фильтрации отзывов (например, по дате, языку или ключевым словам).
3. Цены и расчет стоимости
   1. Отображение цены за ночь или за указанный промежуток времени.
   2. Детализация стоимости с учетом налогов и дополнительных сборов (например, сервисный сбор, уборка).
   3. Возможность расчета итоговой стоимости для выбранных дат.
4. Доступные даты аренды
   1. Календарь с указанием доступных и занятых дат.
   2. Возможность выбора дат для бронирования с автоматическим обновлением стоимости.
5. Местоположение
   1. Точный адрес объекта (после подтверждения бронирования).
   2. Отображение объекта на карте с указанием ближайших достопримечательностей, транспорта и инфраструктуры.
   3. Возможность построения маршрута до объекта.
6. Профиль хозяина
   1. Информация о хозяине (имя, фото, рейтинг, количество объектов).
   2. Возможность связаться с хозяином через встроенную систему сообщений.
   3. Отзывы о хозяине от предыдущих гостей.
7. Дополнительная информация
   1. Подробное описание объекта (удобства, правила, особенности).
   2. Список доступных удобств (например, Wi-Fi, кухня, парковка, бассейн).
   3. Правила объекта (например, время заезда/выезда, запрет на курение, возможность размещения с животными).

3.1.3 Фильтр:Система должна позволять искать жильё/ мероприятие по следующим категориям:

1. Местоположение: Страна, Город, Континент, Регион)
2. Стоимость (без налогов и за ночь)
3. Удобства: Wi-Fi, Кухня, Стиральная машина, Сушильная машина, Кондиционер, Отопление, Рабочая зона, Телевизор, Фен, Утюг.
4. Характеристики: Бассейн, Джакузи, Бесплатная парковка, Зарядка для электромобиля, Кроватка, Кровать кинг-сайз, Спортзал, Зона барбекю, Завтрак, Камин, Можно курить.
5. Безопасность: Датчик дыма, Датчик угарного газа.
6. Язык Хозяина: хозяин указывает какими он языками он владеет.
7. Элементы доступной среды: Доступ без ступеней, Парковочное место для людей с инвалидностью, Гостевой вход шириной от 81 см, Доступ в спальню без ступеней, Вход в спальню шириной от 81 см, Доступ в ванную без ступеней, Вход в ванную шириной от 81 см, Поручень в туалете, Поручень в душе, Доступ в душ без ступеней, Стул для душа или ванной, Потолочный или передвижной подъемник.
8. Категории жилья: Дизайн, Суперхозяева, Отличные виды, Вау!, Пещеры, Домики, У пляжа, Особняки, Города мечты, Тропики, Замки, Популярные, Вершины мира, Супербассейны, Новинки, Luxe, Острова, У озера, Рёканы, Загородные дома.
9. По количеству гостей: Желаемое количество взрослых от 13 лет, детей от 2 до 12, и младенцев до 2 лет.
10. По виду занятия: Искусство и культура, Развлечения, Еда и напитки, Спорт.

3.1.4 Просмотр карты : Система должна предоставлять пользователю возможность просматривать варианты жилья и мероприятия на интерактивной карте. Функционал карты должен включать следующие возможности:

1. Отображение объектов на карте
   1. Варианты жилья и мероприятия должны отображаться на карте в виде маркеров или значков с возможностью их детализации (например, при наведении или клике).
   2. Карта должна поддерживать масштабирование, перемещение и изменение режимов отображения (схема, спутник, гибрид).
2. Интеграция с фильтрами
   1. Карта должна динамически обновляться при применении фильтров, указанных в разделе 3.1.3 (например, фильтры по цене, типу жилья, датам, рейтингу и другим параметрам).
   2. Пользователь должен видеть только те объекты, которые соответствуют выбранным фильтрам.
3. Детализация объектов
   1. При клике на маркер объекта должна отображаться краткая информация о нем (например, фото, название, цена, рейтинг).
   2. Должна быть возможность перехода к полному описанию объекта из карточки на карте.
4. Поиск и навигация
   1. Пользователь должен иметь возможность искать объекты на карте по адресу, названию или другим параметрам.

3.1.5. Cистема Бронирования: Система должна предоставлять пользователям возможность бронировать жилье или мероприятия (впечатления) с учетом следующих функциональных требований:

#### Процесс бронирования жилья:

* 1. Выбор дат:
     1. Пользователь должен иметь возможность выбрать даты заезда и выезда через календарь.
     2. Система должна автоматически проверять доступность объекта на выбранные даты.
  2. Указание количества гостей:
     1. Пользователь должен указать количество взрослых, детей и младенцев.
     2. Система должна учитывать ограничения по количеству гостей, установленные хозяином.
  3. Расчет стоимости:
     1. Система должна автоматически рассчитывать общую стоимость бронирования, включая:
        1. Цену за ночь.
        2. Налоги и сборы (например, сервисный сбор, уборка).
        3. Дополнительные услуги (например, завтрак, парковка).
     2. Должна отображаться детализация стоимости.
  4. Подтверждение бронирования:
     1. Пользователь должен подтвердить бронирование, согласившись с правилами объекта и условиями платформы.
     2. Система должна запрашивать подтверждение через электронную почту или SMS.

#### Процесс бронирования мероприятий (впечатлений):

#### Выбор даты и времени:

* + 1. Пользователь должен выбрать подходящую дату и время мероприятия из доступных вариантов.
  1. Указание количества участников:
     1. Пользователь должен указать количество участников .
  2. Расчет стоимости:
     1. Система должна рассчитывать общую стоимость с учетом количества участников и дополнительных опций.
  3. Подтверждение бронирования:
     1. Пользователь должен подтвердить бронирование, согласившись с условиями мероприятия и платформы.

#### Оплата бронирования:

* 1. Выбор способа оплаты:
     1. Пользователь должен иметь возможность выбрать способ оплаты (кредитная карта, PayPal, Apple Pay, Google Pay и т.д.).
  2. Безопасность платежей:
     1. Система должна использовать защищенные методы оплаты с шифрованием данных.
  3. Подтверждение оплаты:
     1. После успешной оплаты система должна отправлять подтверждение на электронную почту и в личный кабинет пользователя.

#### Управление бронированиями:

* 1. Просмотр текущих бронирований:
     1. Пользователь должен иметь доступ к списку текущих бронирований с деталями (даты, стоимость, статус).
  2. Изменение бронирования:
     1. Пользователь должен иметь возможность изменить даты или количество гостей (если это разрешено хозяином).

1. Отмена бронирования:
   1. Пользователь должен иметь возможность отменить бронирование в соответствии с политикой отмены хозяина.
   2. Система должна автоматически рассчитывать возврат средств (если применимо).
2. Уведомления:
   1. Пользователь должен получать уведомления о подтверждении бронирования, напоминания о предстоящем заезде и изменениях в бронировании.

3.1.6 Управление отзывами и рейтингами. Система должна позволять пользователям:

Оставлять отзывы хозяевам после завершения бронирования.

Просматривать рейтинги и отзывы других пользователей для принятия решений.

Просматривать рейтинги и отзывы на жилье/ мероприятие.

### 3.1.7 Система рекомендаций: Система должна предоставлять пользователю рекомендации на основе его местоположения, предпочтений и оценок.

1. Рекомендации на основе местоположения:  
   1. Система должна учитывать текущее местоположение пользователя для предложения жилья поблизости. Например, если пользователь находится в определенном городе, Airbnb может предложить ему варианты жилья в этом городе или рядом с ним.
2. Рекомендации на основе предпочтений:  
   1. Система должна отслеживать предпочтения пользователя, такие как типы жилья (квартира, дом, комната и т.д.), удобства (бассейн, кухня, Wi-Fi и т.д.), а также предпочтения по ценовому диапазону. Это позволяет рекомендовать только те карточки, которые соответствуют интересам пользователя.
3. Рекомендации с наивысшими оценками:  
   1. Airbnb будет предлагать пользователю варианты жилья с наивысшими оценками и отзывами от других пользователей, чтобы помочь ему выбрать наиболее качественные и проверенные предложения.
4. Персонализированные предложения:
   1. Система будет учитывать историю поиска и бронирования пользователя, чтобы предлагать ему похожие варианты, которые могут ему понравиться, основываясь на предыдущем опыте.

3.1.8 Управление аккаунтом:Система должна предоставлять пользователю возможность полного контроля над своим аккаунтом, включая следующие функции:

1. История бронирований и сохраненные списки
   1. Пользователь может просматривать историю своих бронирований, включая текущие и завершенные поездки, с деталями (даты, стоимость, статус).
   2. Пользователь может создавать и управлять сохраненными списками (wishlists) для понравившихся вариантов жилья.
2. Управление платежами и налоговой информацией
   1. Пользователь может просматривать историю платежей, включая даты, суммы и статусы транзакций.
   2. Система должна позволять управлять методами оплаты (добавление, редактирование, удаление банковских карт или других платежных данных).
   3. Пользователь может загружать и редактировать налоговую информацию.
3. Аннулирование (удаление) аккаунта
   1. Пользователь может удалить или временно деактивировать свой аккаунт.
   2. При удалении аккаунта система должна запрашивать подтверждение и уведомлять о последствиях (например, отмена бронирований, потеря данных и т.д.).
4. Управление параметрами аккаунта
   1. Пользователь может редактировать личные данные, такие как имя, адрес электронной почты, телефонный номер и профильное фото.
   2. Система должна предоставлять настройки конфиденциальности, включая управление видимостью профиля и доступом к данным.
   3. Пользователь может настраивать уведомления (например, о новых сообщениях, подтверждении бронирования и т.д.).

3.1.9 Система обмена сообщениями Система должна позволять пользователям:

1. Обмен сообщениями между пользователями:
   1. Система должна обеспечивать возможность обмена текстовыми сообщениями между зарегистрированными пользователями (гостями и хозяевами).
   2. Сообщения должны доставляться в режиме реального времени или с минимальной задержкой.
2. Уведомления о новых сообщениях:
   1. Пользователи должны получать уведомления о новых сообщениях через push-уведомления, электронную почту или встроенные оповещения в интерфейсе системы.
   2. Уведомления должны содержать краткую информацию о сообщении (например, имя отправителя и часть текста).
3. Фильтрация и поиск сообщений:
   1. Пользователи должны иметь возможность фильтровать сообщения по дате, отправителю, ключевым словам или другим параметрам.
   2. Система должна предоставлять функцию поиска по истории сообщений для быстрого нахождения нужной информации.
4. Прикрепление файлов и фотографий:
   1. Пользователи должны иметь возможность прикреплять к сообщениям файлы (документы, изображения, видео и другие форматы) в пределах установленных ограничений по размеру и типу файлов.
5. Жалобы и блокировка пользователей:
   1. Пользователи должны иметь возможность пожаловаться на другого пользователя в случае нарушения правил общения (спам, оскорбления и т.д.).
   2. Система должна предоставлять возможность блокировки пользователей, после чего заблокированный пользователь не сможет отправлять сообщения или видеть активность блокирующего.
   3. Жалобы должны автоматически направляться модераторам или администраторам системы для дальнейшего рассмотрения.

### 3.1.10 Локализация и многоязычность: Система должна поддерживать:

1. Множество языков для удобства пользователей из разных стран:  
   1. Интерфейс на нескольких языках: Airbnb предоставляет интерфейс на различных языках, включая русский, английский, испанский, французский и другие, что обеспечивает удобство использования платформы для пользователей по всему миру.
   2. Автоматический переводчик: В случае необходимости, Airbnb использует автоматический переводчик для перевода сообщений между гостями и хозяевами, что упрощает коммуникацию между пользователями, говорящими на разных языках.
2. Локализацию валюты и единиц измерения:
   1. Автоматическое отображение цен в местной валюте: Airbnb автоматически отображает цены на жилье в местной валюте пользователя, учитывая его местоположение и настройки профиля. Это упрощает процесс бронирования и оплаты для пользователей.
   2. Использование соответствующей валюты: При оплате бронирования Airbnb использует валюту, соответствующую местоположению пользователя и страны проживания хозяина, обеспечивая удобство и прозрачность финансовых операций.

3.1.11 Система поддержки пользователей. Система должна предоставлять пользователям:

1. Центр помощи: раздел с часто задаваемыми вопросами и ответами для самостоятельного решения проблем.
2. Связь с поддержкой: возможность связаться с командой поддержки через чат, электронную почту или телефон для решения возникших вопросов.

Требования владельцев сайта:

3.1.12 Инструменты для хозяев:Система должна предоставлять хозяевам (владельцам жилья или организаторам мероприятий) набор инструментов для эффективного управления своими объектами. Функционал должен включать следующие возможности:

1. Добавление новых объектов
   1. Хозяин должен иметь возможность добавлять новые объекты жилья или мероприятия, заполняя необходимую информацию (описание, фото, цена, доступные даты, правила и т.д.).
   2. Система должна поддерживать загрузку медиафайлов (фото, видео, 3D-туры) для наглядного представления объекта.
   3. Должна быть предусмотрена возможность предварительного просмотра объекта перед публикацией.
2. Редактирование информации о существующих объектах
   1. Хозяин может вносить изменения в информацию об объекте (например, обновлять описание, фото, цены, доступные даты или правила).
   2. Изменения должны сохраняться и отображаться в реальном времени для потенциальных гостей.
3. Управление бронированиями
   1. Хозяин должен иметь доступ к календарю бронирований, где отображаются все забронированные даты и статусы бронирований (подтвержденные, ожидающие подтверждения, отмененные).
   2. Должна быть возможность подтверждать, отклонять или отменять бронирования.
   3. Система должна уведомлять хозяина о новых бронированиях и изменениях в существующих.
4. Управление ценами и акциями
   1. Хозяин может устанавливать базовые цены, а также настраивать динамическое ценообразование (например, повышение цен в высокий сезон или скидки для длительных бронирований).
5. Аналитика и отчеты
   1. Хозяин должен иметь доступ к статистике по своим объектам (например, количество просмотров, бронирований, доходы, рейтинги и отзывы).
   2. Система должна предоставлять отчеты за выбранные периоды времени.
6. Общение с гостями
   1. Хозяин должен иметь возможность общаться с гостями через встроенную систему сообщений.
   2. Должна быть функция отправки автоматических уведомлений (например, приветствие, напоминание о заезде, инструкции по выезду).
7. Управление отзывами
   1. Хозяин может просматривать отзывы гостей и отвечать на них.
   2. Должна быть возможность запроса отзыва у гостя после завершения бронирования.

### 3.1.13 Управление платежами: Система должна обеспечивать:

1. Безопасные платежи:  
   1. Защищенные методы оплаты: Airbnb использует современные технологии шифрования и безопасные платежные системы для защиты финансовой информации пользователей, обеспечивая конфиденциальность и безопасность транзакций.
2. Поддержку различных способов оплаты*:*  
   1. Кредитные и дебетовые карты: Airbnb принимает платежи с помощью основных кредитных и дебетовых карт, включая Visa, MasterCard, American Express и JCB.
   2. Электронные кошельки и другие платежные системы: Платформа поддерживает оплату через электронные кошельки, такие как PayPal, а также через системы Apple Pay и Google Pay, предоставляя пользователям гибкость в выборе способа оплаты.
3. Автоматические расчеты с хозяевами:  
   1. Перевод средств после успешного бронирования: Airbnb автоматически переводит средства хозяевам после успешного завершения бронирования, удерживая установленную комиссию. Это обеспечивает своевременные выплаты и упрощает финансовые операции для хозяев.

#### 3.1.14. Модерация контента:

1. Проверка объявлений:
   1. Система должна автоматически проверять новые объявления на соответствие правилам платформы (например, запрет на незаконную деятельность, корректность описания, качество фотографий).
   2. При необходимости объявления должны отправляться на ручную модерацию администраторам.
2. Модерация фотографий и видео:
   1. Фотографии и видео должны проверяться на соответствие стандартам качества (четкость, отсутствие запрещенного контента).
   2. Система должна автоматически отклонять материалы низкого качества или нарушающие правила.
3. Модерация отзывов:
   1. Отзывы пользователей должны проверяться на наличие оскорбительных выражений, спама или fake-отзывов.
   2. Система должна предоставлять возможность хозяевам жаловаться на некорректные отзывы для дальнейшей проверки модераторами.
4. Блокировка контента:
   1. В случае обнаружения нарушений система должна блокировать контент до устранения проблем.
   2. Хозяева должны получать уведомления о причинах блокировки и инструкции по исправлению.

**3.2 Usability (Требования к удобству использования)**

1) Время обучения:

1. Новый пользователь должен освоить базовый функционал в среднем за 10 минут.
2. Опытный пользователь должен выполнять нужные ему действия в среднем менее чем за 3 минуты.
3. При вводе некорректных данных система должна давать всплывающие подсказки.

2) Простота пользовательского интерфейса:

1. Базовый функционал должен быть доступен пользователю на главной странице или доступен в менее чем в 4 клика.
2. Должен быть Центр Помощи, где написаны ответы на самые частые вопросы, и в целом описано как пользоваться функционалом системы. Должен быть реализован поиск по эти вопросам.
3. Система должна включать интерактивную карту, которая упрощает выбор жилья и мероприятий. На карте должны отображаться их цены, а при нажатии на них должна всплывать их краткая информация. Если пользователю требуется больше деталей, он может перейти на страницу с полным описанием и отзывами.
4. Для простоты бронирования и поиска должен быть реализован календарь, где жирным цветов доступные даты, а бледным недоступные. Выбранные даты и промежуток между ними должны быть помечены другим цветом.

3) Локализация:

1. Должны поддерживаться большинство языков, или функция автоперевода через API.
2. Должны поддерживаться большинство мировых валют.

4) Оптимизация:

Для обеспечения стабильной и удобной работы платформы Airbnb на различных устройствах и в современных браузерах система должна соответствовать следующим требованиям:

1. Адаптивный веб-дизайн:
   1. Интерфейс должен автоматически подстраиваться под размеры экрана пользователя, обеспечивая удобство работы как на настольных компьютерах, так и на мобильных устройствах.
2. Оптимизация для мобильных устройств:
   1. Для того чтобы сайт правильно отображался на разных устройствах — мобильных телефонах, планшетах и компьютерах — нужно настроить его отображение так, чтобы он подстраивался под размер экрана
   2. Интерактивные элементы (кнопки, ссылки) должны иметь достаточный размер и расстояние между собой для удобного взаимодействия на сенсорных экранах.
   3. В системе не должны использоваться плагины, требующие дополнительной установки (например, Flash), для обеспечения совместимости с мобильными браузерами.
3. Кросс-браузерная совместимость:
   1. Платформа должна корректно работать в следующих браузерах: Google Chrome (версия 100 и выше), Mozilla Firefox (версия 95 и выше), Safari (версия 14 и выше), Microsoft Edge (версия 100 и выше)

5) Удобный поиск:

1. Поиск должен быть разделен на 2 категории. Одна из категории должна реализовывать горизонтальный скроллинг. А вторая более детальная, выпадающим списком.
2. Все виды фильтрации должны сопровождаться визуальным представлением или ассоциативным элементом рядом, чтобы пользователи могли интуитивно воспринимать их значение. Это может быть иконка, цветовая индикация или графический символ, который сразу передает суть фильтра.

6) Поддержка пользователей:

1. Каналы поддержки: Платформа должна предоставлять несколько каналов связи для поддержки пользователей, включая:
   1. Онлайн-центр помощи: Обширный раздел с часто задаваемыми вопросами и статьями, охватывающими различные аспекты использования платформы.
   2. Чат с поддержкой: Возможность оперативно связаться с представителем службы поддержки через встроенный чат.
   3. Телефонная поддержка: Предоставление горячей линии с приемлемым временем ожидания ответа.
   4. Чат-бот: Интеграция с ботом, который помогает пользователям с базовыми запросами, такими как поиск жилья, бронирование и решение стандартных вопросов. Бот должен уметь направлять более сложные запросы к живому оператору.
2. Разрешение проблем: В случае возникновения проблем с бронированием, жилья или платежами, поддержка должна стремиться к разрешению вопросов в течение 24 часов.
3. Обратная связь: Регулярный сбор отзывов пользователей о качестве поддержки с целью постоянного улучшения сервиса.

**3.3 Reliability (Требования к надежности)**

Для обеспечения высокой надежности “Airbnb”, требуется выполнить следующие пункты:

1. Доступность системы: доступность системы должна составлять не менее 99,9 % , что около 8, 76 часов простоя в год, что соответствует современным стандартам для современных сервисов.
2. Среднее время между отказами: среднее время работы системы между сбоями. Система должна стремиться к показателю 2’000 часов, что соответствует менее 5 ошибкам в год.
3. Среднее время восстановления: среднее время необходимое для восстановления системы после сбоя. Нужно стремиться как можно быстрее решать сбои, при этом не пренебрегая качеством.
4. Количество критических ошибок: это крупные ошибки, которые могут приводить к потере данных, недоступности сервера или не работе элементов базового функционала. Нормой является не более 5 ошибок в год.

**3.4 Performance (Требования к производительности)**

1. Время отклика системы:
   1. **First Content Full Paint (**Это время, за которое браузер отображает первый элемент контента на странице**)** – 0,7 секунды
   2. **Largest Content Full Paint (**Это время, за которое отображается самый крупный элемент на странице**)** – 2,5 секунды
   3. **Speed Index(**Это метрика, которая показывает, насколько быстро визуально завершается загрузка страницы**)** – 2,4 секунды .
2. Пропускная способность: не менее 10’000 транзакций в секунду.
3. Максимальное кол-во активных пользователей 1’000’000 пользователей
4. Использования ресурсов: разделим на три блока
   1. Память: должно быть оптимизированное использование оперативной памяти. Необходимо использовать кэширование данных, для частых запросов, через Redis, Memcached.
   2. Процессор: должна быть разделена нагрузка между ядрами процессора и серверами. Должна быть горизонтальная и вертикальная масштабируемость. Должны применяться самые эффективные и быстрые алгоритмы. Средний уровень загрузки CPU не должен превышать 70**%,** чтобы при резкой активности система работала как надо.
   3. Хранилище: необходимо использованы высокоростные SSD для Баз Данных (с важной информацией). Также необходимо использовать облачное хранилище Google Cloud Storage для (остальной информации). Нужно иметь систему резервного копирования для важной информации.

**3.5 Design Constraints (Ограничения разработки)**

Для разработки системы использовать:

Следующие языки программирования

1. JavaScript - язык для основной фронтенд части сервиса. Используется с библиотекой ReactJS. Также использовать JWT токены для авторизаций, Redux библиотеку - для управления состояниями объектов.
2. Для серверной части использовать Java в связке с фреймворком Spring. И Ruby с фреймворком Ruby on Rails.
3. Для обработки данных использовать Python.
4. Для мобильного приложения использовать Swift для iOS и Kotlin для Android

Следующие подходы и архитектуры:

1. Микросервисы - для облегченного расширения и устойчивости.
2. Для взаимодействиями между микросервисами использовать HTTPS
3. Для взаимодействия между клиентом и сервером использовать язык запросов и среду выполнения для работы с API GraphQL

Базы Данных и хранилища:

1. PostgreSQL - хранение пользователей, бронирований, жилья и т.д.
2. Redis - используется для кэширования данных.
3. Google Cloud Storage - для хранения изображений и других не самых важных данных.
4. Apache Kafka - использовать для сбора, хранения и передачи сообщений между различными сервисами и приложениями.

Ограничения:

1. ReactJS: Распространяется под лицензией MIT, которая является свободной и разрешает использование, копирование, модификацию и распространение.
2. Java является зарегистрированным товарным знаком Oracle и распространяется под лицензией Oracle Binary Code License Agreement. Spring Framework распространяется под лицензией Apache License 2.0, которая позволяет свободное использование, модификацию и распространение с некоторыми условиями.
3. Ruby on Rails: Распространяется под лицензией MIT, предоставляющей широкие права на использование, модификацию и распространение.
4. Python: Имеет собственную Python Software Foundation License, которая является свободной и открытой, позволяя использование, модификацию и распространение.
5. Swift: Разработан Apple и распространяется под лицензией Apache License 2.0.
6. Kotlin: Разработан JetBrains и распространяется под лицензией Apache License 2.0.
7. GraphQL: Является спецификацией, а его реализации могут иметь разные лицензии. Например, Apollo Server распространяется под лицензией MIT.
8. PostgreSQL: Распространяется под собственной PostgreSQL License, которая является свободной и открытой.
9. Redis: Исходный код Redis распространяется под лицензией BSD 3-Clause. Однако модули могут иметь различные лицензии, и некоторые из них используют Redis Source Available License (RSAL), ограничивающую использование в некоторых коммерческих продуктах.
10. Google Cloud Storage: Это проприетарный сервис от Google, и его использование регулируется условиями предоставления услуг Google Cloud Platform.
11. Apache Kafka: Распространяется под лицензией Apache License 2.0.

**3.6 Interfaces (Интерфейсы)**

3.6.1 User Interfaces (Пользовательские интерфейсы)

1) Цветовая палитра:

****

Основной цвет: Carnation (#fd5c63) — насыщенный кораллово-розовый.

Дополнительные оттенки (справа):

Темный бордовый (вверху) — глубокий красно-коричневый оттенок.

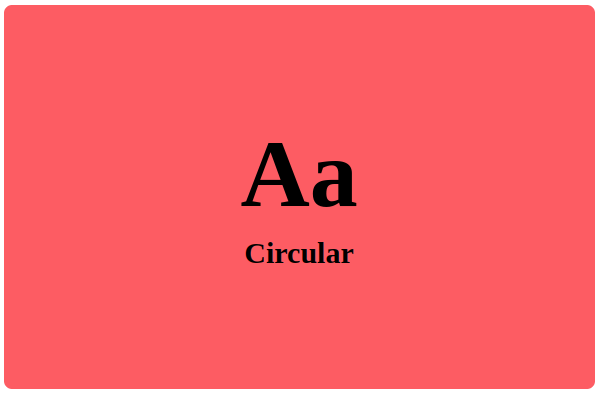
Темно-красный — более приглушенный насыщенный красный.

Бледно-розовый — мягкий пастельный оттенок.

Очень светлый розовый (внизу) — почти пудровый цвет.

А также белый цвет.

2) Шрифт:



3) Логотип:

****

Этот логотип используется на сайте, а остальные в соцсетях. Но возможны и изменения.

4) Предыдущие цвета должны фигурировать и на сайте.

5) Так выглядит макет десктопной версии сайта:



Вот его описание:

Стандартная сетка сайта будет использоваться для полноэкранных макетов (например, P0, P1 и PDP), обеспечивая удобное расположение контента. Она будет гибкой, но фиксироваться при ширине 1440px, включать 12 колонок с отступами (gutter) и внешними полями (margin) для удобного размещения элементов. Дизайн адаптируется под различные разрешения экранов, сохраняя четкую структуру и удобство восприятия.

И мобильное представление:



3.6.2 User Interfaces (Пользовательские интерфейсы)

Не предоставляется

3.6.3 Software Interfaces (Программные интерфейсы)

1. Предоставление собственного API для интеграции с партнерами такими как туристические агентства, системы управления бронированиями и другие сервисы.
2. Интеграция с платежными API: PayPal, Visa, MasterCard
3. Интеграция с картографическими сервисами: Google Maps.
4. SDK для iOS и Android, которые позволяют разработчикам создавать приложения с функциональностью Airbnb.

3.6.4 Software Interfaces (Сетевые интерфейсы)

Не предоставляется

**3.7 Licensing Requirements (Требования к лицензированию)**

Разрабатываемый продукт Airbnb, включая его программное обеспечение, API, мобильные приложения и веб-платформу, распространяется под проприетарной лицензией. Это означает, что исходный код и интеллектуальная собственность Airbnb защищены авторским правом и не могут быть свободно использованы, модифицированы или распространяемы третьими лицами без явного разрешения компании.

Основные аспекты лицензирования:

1. Проприетарная лицензия: Весь код, дизайн, интерфейсы и функциональность платформы Airbnb являются собственностью компании.
2. Пользователи и разработчики могут использовать платформу только в рамках условий, указанных в Пользовательском соглашении и Условиях использования.
3. Использование API: Airbnb предоставляет доступ к своим API для партнеров и разработчиков, но использование API регулируется отдельным соглашением.
4. Разработчики обязаны соблюдать ограничения, такие как количество запросов в минуту, запрет на использование данных в коммерческих целях без разрешения и другие условия.
5. Открытый исходный код: Некоторые инструменты и библиотеки, разработанные Airbnb, могут быть выпущены под открытыми лицензиями (например, MIT, Apache 2.0). Примером является библиотека Airbnb JavaScript Style Guide. Такие проекты доступны на платформах, таких как GitHub, и могут свободно использоваться сообществом в соответствии с условиями их лицензий.
6. Сторонние лицензии:Airbnb использует сторонние библиотеки и компоненты, которые могут распространяться под различными лицензиями (например, MIT, GPL, Apache).Компания обязана соблюдать условия этих лицензий, включая указание авторства и предоставление исходного кода, если это требуется.
7. Коммерческое использование:Любое коммерческое использование платформы Airbnb, включая интеграцию с другими сервисами, требует согласования с компанией и заключения соответствующего договора.
8. Требования к пользователям:Пользователи платформы Airbnb обязаны соблюдать Условия использования и Политику конфиденциальности.Запрещено копирование, модификация или распространение контента платформы (фотографии, описания, отзывы) без разрешения.

**4. Requirements Attributes (Атрибуты требований)**

**4.1 Functional requirements (Функциональные требования)**

| **№ Требования** | **Описание требования** | **Приоритетность** | ***Трудоемкость***  ***в часах*** |
| --- | --- | --- | --- |
| 3.1.1 | Просмотр карточек жилья/мероприятия: фотографии, отзывы, цена, срок аренды, местоположение, профиль хозяина, доступные удобства | 10 | 500 |
| 3.1.2 | Фильтр по местоположению (страна, город, континент, регион) | 9 | 300 |
| 3.1.3 | Фильтр по стоимости (с налогами и без) | 9 | 200 |
| 3.1.4 | Фильтр по удобствам (Wi-Fi, кухня, стиральная машина, кондиционер и т.д.) | 9 | 400 |
| 3.1.5 | Фильтр по характеристикам (бассейн, джакузи, парковка, зарядка для электромобиля и т.д.) | 9 | 400 |
| 3.1.6 | Фильтр по безопасности (датчик дыма, датчик угарного газа) | 8 | 200 |
| 3.1.7 | Фильтр по языкам хозяина | 7 | 200 |
| 3.1.8 | Фильтр по элементам доступной среды (парковочное место для инвалидов, поручни и т.д.) | 8 | 300 |
| 3.1.9 | Фильтр по категориям жилья (дизайн, суперхозяева, у пляжа и т.д.) | 9 | 400 |
| 3.1.10 | Фильтр по количеству гостей (взрослые, дети, младенцы) | 9 | 300 |
| 3.1.11 | Фильтр по виду занятия (искусство и культура, спорт и т.д.) | 8 | 200 |
| 3.1.12 | Просмотр карты с интерактивными фильтрами и обновлением в реальном времени | 10 | 600 |
| 3.1.13 | Бронирование жилья или мероприятий | 10 | 500 |
| 3.1.14 | Система оценивания: комментарии, избранное, рейтинг карточек и хозяев | 9 | 400 |
| 3.1.15 | Рекомендации на основе местоположения, предпочтений и рейтингов | 9 | 500 |
| 3.1.16 | Инструменты для хозяев: добавление, редактирование, управление бронированиями | 10 | 600 |
| 3.1.17 | Управление аккаунтом: история просмотров, платежи, налоговая информация | 9 | 400 |
| 3.1.18 | Аннулирование (удаление) аккаунта | 7 | 200 |
| 3.1.19 | Управление параметрами аккаунта (конфиденциальность, почта, адрес) | 9 | 300 |
| 3.1.20 | Управление сообщениями | 9 | 300 |
| 3.1.21 | Реферальные программы и купоны | 7 | 200 |
| Сумма человеко-часов | | | 5600 |

**4.1.1 Non-functional requirements (Нефункциональные требования)**

| **Требование** | **Описание требования** | **Приоритетность** | **Стабильность** |
| --- | --- | --- | --- |
| 3.2.1 | Время обучения: новый пользователь должен освоить базовый функционал за 10 минут, опытный пользователь — менее чем за 3 минуты. | 10 | Высокая |
| 3.2.2 | Простота пользовательского интерфейса: базовый функционал доступен в менее чем 4 клика, наличие Центра Помощи с поиском по вопросам. | 9 | Средняя |
| 3.2.3 | Локализация: поддержка большинства языков и мировых валют, функция автоперевода через API. | 8 | Высокая |
| 3.2.4 | Оптимизация: система должна быть оптимизирована под различные устройства и современные браузеры. | 10 | Средняя |
| 3.2.5 | Доступность: интерфейс адаптирован для мобильных устройств, планшетов, ПК и людей с ограниченными способностями. | 7 | Высокая |
| 3.2.6 | Сообщения: фильтрация сообщений по категориям, автоответчик, запланированные сообщения, архивирование, подсказки для ответов. | 6 | Низкая |
| 3.2.7 | Удобный поиск: разделение на две категории (горизонтальный скроллинг и выпадающий список), визуальное представление фильтров. | 9 | Высокая |
| 3.2.8 | Надежность: доступность системы не менее 99.9%, среднее время между отказами — 2000 часов, среднее время восстановления — минимальное. | 10 | Высокая |
| 3.2.9 | Производительность: время отклика не более 0.5 секунды, пропускная способность — 10’000 транзакций в секунду, поддержка 1’000’000 активных пользователей. | 10 | Высокая |
| 3.2.10 | Использование ресурсов: оптимизация оперативной памяти, кэширование, нагрузка на CPU не более 70%, использование SSD и облачного хранилища. | 9 | Высокая |
| 3.2.11 | Ограничения разработки: использование JavaScript (ReactJS), Java (Spring), Ruby (Ruby on Rails), Python, Swift, Kotlin, GraphQL, PostgreSQL, Redis, Google Cloud Storage, Apache Kafka. | 8 | Высокая |

**4.2 Прецеденты системы**

**4.2.1 Название прецедента: Поиск жилья/мероприятий**

| Поле | Описание |
| --- | --- |
| Краткое описание: | Гость ищет жилье/мероприятия |
| Главные актёры: | Гость |
| Второстепенные актеры: | Нет |
| Предусловия: | Гость авторизован или использует платформу без авторизации |
| Основной поток: | 1. Гость открывает главную страницу или раздел поиска. |
|  | 2. Система отображает поле для ввода параметров поиска (местоположение, даты, количество гостей и т.д.). |
|  | 3. Гость вводит параметры поиска и нажимает кнопку "Найти". |
|  | 4. Система отображает список доступных вариантов жилья/мероприятий. |
|  | 5. Гость может применять дополнительные фильтры (цена, удобства и т.д.). |

**4.2.1 Название прецедента: Бронирование жилья/мероприятия**

| Поле | Описание |
| --- | --- |
| Краткое описание: | Гость бронирует жилье/мероприятие |
| Главные актёры: | Гость |
| Второстепенные актеры: | Хозяин (для подтверждения бронирования) |
| Предусловия: | Гость авторизован и выбрал жилье/мероприятие для бронирования |
| Основной поток: | 1. Гость выбирает жилье/мероприятие и нажимает "Забронировать". |
|  | 2. Система запрашивает подтверждение данных (даты, количество гостей и т.д.). |
|  | 3. Гость подтверждает данные и переходит к оплате. |
|  | 4. Система обрабатывает оплату и отправляет подтверждение бронирования. |
|  | 5. Хозяин получает уведомление о новом бронировании. |

**4.2.2 Название прецедента: Добавление нового жилья/мероприятия**

| Поле | Описание |
| --- | --- |
| Краткое описание: | Хозяин добавляет новое жилье/мероприятие |
| Главные актёры: | Хозяин |
| Второстепенные актеры: | Нет |
| Предусловия: | Хозяин авторизован и имеет доступ к панели управления |
| Основной поток: | 1. Хозяин заходит в панель управления и выбирает "Добавить объект". |
|  | 2. Система отображает форму для ввода информации о новом объекте (описание, фото, цена и т.д.). |
|  | 3. Хозяин заполняет форму и нажимает "Сохранить". |
|  | 4. Система проверяет данные и добавляет объект в список доступных для бронирования. |

**4.2.3 Название прецедента: Управление бронированиями**

| Поле | Описание |
| --- | --- |
| Название прецедента: | Управление бронированиями |
| Краткое описание: | Хозяин управляет бронированиями своего жилья/мероприятий |
| Главные актёры: | Хозяин |
| Второстепенные актеры: | Гость (для уведомлений) |
| Предусловия: | Хозяин авторизован и имеет доступ к панели управления |
| Основной поток: | 1. Хозяин заходит в панель управления и выбирает "Бронирования". |
|  | 2. Система отображает список текущих и будущих бронирований. |
|  | 3. Хозяин может подтвердить, отменить или изменить бронирование. |
|  | 4. Система отправляет уведомление гостю об изменениях. |

**4.2.4 Название прецедента: Оставление отзывов и оценок**

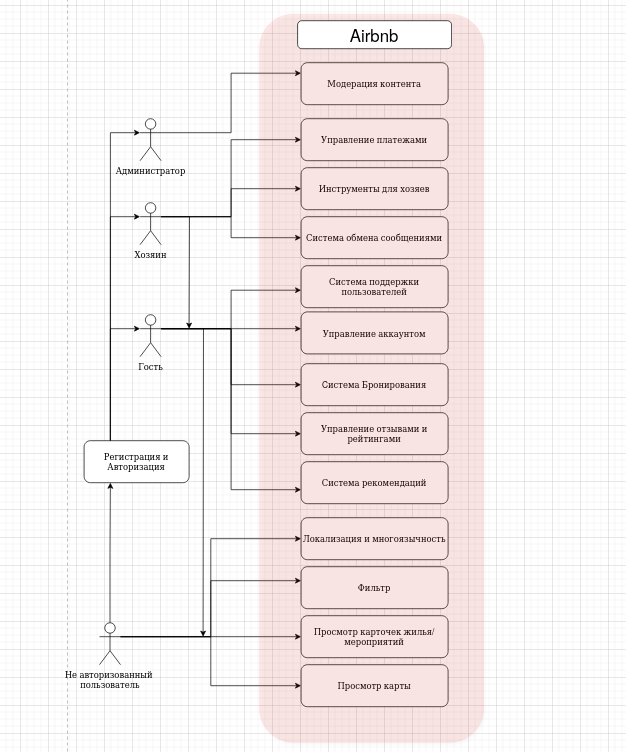
| Поле | Описание |
| --- | --- |
| Краткое описание: | Гость оставляет отзыв и оценку после завершения бронирования |
| Главные актёры: | Гость |
| Второстепенные актеры: | Хозяин (для получения отзывов) |
| Предусловия: | Гость завершил бронирование и авторизован |
| Основной поток: | 1. Гость заходит в раздел "Мои бронирования". |
|  | 2. Система предлагает оставить отзыв и оценку для завершенного бронирования. |
|  | 3. Гость заполняет форму отзыва и нажимает "Отправить". |
|  | 4. Система сохраняет отзыв и уведомляет хозяина. |

4.3 Риски

| **№** | **Название риска** | **Описание риска** | **Вероятность появления** |
| --- | --- | --- | --- |
| 1. Ресурсные риски | | | |
| 1.1. Нехватка квалифицированных разработчиков | Сложность найма и удержания специалистов для разработки и поддержки платформы. | Инвестиции в обучение, привлекательные условия труда, удаленная работа. | Средняя |
| 1.2. Ошибки при разработке | Недостаток опыта команды в реализации сложных функций, таких как система бронирования или интеграция с платежными системами. | Проведение тестирования на каждом этапе разработки, привлечение опытных консультантов. | Высокая |
| 1.3. Нехватка времени | Задержки в разработке из-за сложности реализации функций, таких как локализация или интеграция с картографическими сервисами. | Использование Agile-методологий для гибкого управления проектом. | Средняя |
| 2. Бизнес риски | | | |
| 2.1. Конкуренция (Booking.com, Vrbo) | Потеря пользователей из-за более привлекательных предложений конкурентов. | Постоянное улучшение функционала, уникальные предложения (например, опытные мероприятия). | Высокая |
| 2.2. Зависимость от хозяев | Сбои в работе из-за ненадежности хозяев (например, отмена бронирований или некачественное жилье). | Внедрение системы рейтингов и отзывов, строгая модерация объявлений. | Средняя |
| 2.3. Неправильное управление бюджетом | Перерасходы бюджета, необоснованно увеличенные расходы на маркетинг или разработку. | Строгий контроль выполнения бюджета, регулярный аудит расходов. | Средняя |
| 2.4. Изменение законодательства | Новые законы или налоги, которые могут повлиять на бизнес-модель Airbnb. | Постоянный мониторинг законодательных изменений, адаптация бизнес-процессов. | Средняя |
| 2.5. Потеря ключевых партнеров | Прекращение сотрудничества с важными партнерами, такими как платежные системы или картографические сервисы. | Диверсификация партнеров, заключение долгосрочных контрактов. | Низкая |
| 3. Технические риски | | | |
| 3.1. Недостаточная масштабируемость | Ухудшение эффективности системы с ростом количества пользователей и бронирований. | Использование облачных технологий для горизонтального масштабирования, оптимизация баз данных, регулярное тестирование под нагрузкой. | Высокая |
| 3.2. Уязвимости безопасности | Утечка персональных данных пользователей или платежной информации. | Внедрение шифрования данных, двухфакторной аутентификации, регулярные аудиты безопасности. | Высокая |
| 3.3. Проблемы с интеграцией API | Сбои в работе из-за ненадежности сторонних API (например, Google Maps, платежные системы). | Тщательный выбор партнеров, резервные решения на случай сбоев. | Средняя |
| 3.4. Сбои в работе серверов | Потеря данных или недоступность платформы из-за технических сбоев. | Использование резервных серверов, регулярное тестирование на отказоустойчивость. | Средняя |
| 3.5. Устаревание технологий | Использование устаревших технологий, которые могут замедлить развитие платформы. | Регулярное обновление технологического стека, мониторинг новых технологий. | Средняя |
| 4. Маркетинговые риски | | | |
| 4.1. Неэффективная рекламная кампания | Низкая окупаемость инвестиций в маркетинг. | Анализ эффективности рекламных каналов, использование таргетированной рекламы. | Средняя |
| 4.2. Негативные отзывы в СМИ | Ухудшение репутации из-за негативных публикаций или скандалов. | Активная работа с PR, оперативное реагирование на негатив. | Низкая |
| 4.3. Недостаточное присутствие на рынке | Низкая узнаваемость бренда в новых регионах. | Расширение маркетинговых кампаний, сотрудничество с локальными influencers. | Средняя |
| 5. Риски, связанные с пользователями | | | |
| 5.1. Мошенничество со стороны пользователей | Фальшивые бронирования или мошеннические действия со стороны гостей и хозяев. | Внедрение системы верификации пользователей, мониторинг подозрительных действий. | Средняя |
| 5.2. Низкая удовлетворенность пользователей | Уход пользователей из-за неудовлетворенности сервисом. | Регулярный сбор обратной связи, улучшение функционала на основе отзывов. | Высокая |
| 5.3. Недостаточная активность пользователей | Низкая частота использования платформы. | Внедрение программ лояльности, уведомления о новых предложениях. | Средняя |
| 6. Риски, связанные с внешними факторами | | | |
| 6.1. Экономический кризис | Снижение спроса на аренду жилья из-за экономического спада. | Диверсификация предложений, снижение цен в периоды кризиса. | Низкая |
| 6.2. Пандемии или природные катаклизмы | Резкое снижение спроса на путешествия и аренду жилья. | Разработка гибких условий бронирования, страхование рисков. | Низкая |
| 6.3. Политическая нестабильность | Ограничения на международные поездки или аренду жилья в определенных регионах. | Мониторинг политической ситуации, адаптация приложений под локальные условия. | Низкая |
| 7. Риски, связанные с данными | | | |
| 7.1. Потеря данных | Утрата важных данных из-за технических сбоев или атак. | Регулярное резервное копирование данных, использование надежных систем хранения. | Средняя |
| 7.2. Неправильная обработка данных | Ошибки в обработке данных, ведущие к некорректным рекомендациям или расчетам. | Внедрение систем контроля качества данных, регулярный аудит. | Средняя |
| 8. Риски, связанные с персоналом | | | |
| 8.1. Текучесть кадров | Частая смена сотрудников, ведущая к потере знаний и опыта. | Создание комфортных условий труда, программы мотивации и развития. | Средняя |
| 8.2. Недостаток квалификации сотрудников | Низкий уровень профессиональной подготовки сотрудников. | Регулярное обучение и повышение квалификации сотрудников. | Средняя |

**5. Functional modeling (Функциональное моделирование)**

**5.1 Use-case диаграмма**

****

**\_**

**Вывод:** В процессе выполнения лабораторной работы я изучил структуру документа SRS (Software Requirements Specification), а также попрактиковался в создании UML-диаграммы на основе анализа SRS для платформы Airbnb.