**Техническое задание**

**Проект: VR клуб – Поколение Z**

Оглавление

[1 Общие сведения 1](#__RefHeading___Toc536172748)

[1.1 Используемые термины и сокращения 1](#__RefHeading___Toc536172749)

[1.2 Цели изменений (доработок) системы 1](#__RefHeading___Toc536172750)

[1.3 Область применения 2](#__RefHeading___Toc536172751)

[1.4 Описание изменении бизнес-процессов (как будет) 3](#__RefHeading___Toc536172753)

[2 Требования к изменяемой подсистеме 3](#__RefHeading___Toc536172754)

[2.1 Требование к функционалу 3](#__RefHeading___Toc536172755)

[2.2 Технические требования 4](#__RefHeading___Toc536172756)

[2.3 Требования к ролям 4](#__RefHeading___Toc536172757)

[2.4 Требования к интерфейсам 4](#__RefHeading___Toc536172758)

[2.5 Требования к миграции и изменению потоков данных 5](#__RefHeading___Toc536172759)

[2.6 Нефункциональные требования (Качество ПО) 5](#__RefHeading___Toc536172760)

[2.7 Справочная информация по объектам (СИ) 6](#__RefHeading___Toc536172762)

[2.8 Порядок и виды тестирования 6](#__RefHeading___Toc536172763)

[3 План внедрения 6](#__RefHeading___Toc536172764)

# 1 Общие сведения:

## Используемые термины и сокращения

* VR (Virtual Reality / Виртуальная реальность) – технология, позволяющая создать иммерсивную цифровую среду, где пользователь может ощущать присутствие и взаимодействовать с этой средой через специализированные устройства – такие, как шлемы, перчатки или датчики движения.
* Поколение Z – поколение, родившееся в период с конца 1990-х по начало 2010-х годов. Это люди, которые росли в эру цифровых технологий, тесно связанную с интернетом, социальными сетями и современными гаджетами. Они склонны к использованию технологий для просторного общения, профессионального развития и развлечений.

## 1.2 Цели системы

|  |  |
| --- | --- |
| **Цель** | **Критерии достижения** |
| Обеспечить удобный просмотр информации по тарифам посредством реализации слайдера с тарифами. | Пользователь может просматривать тарифы с помощью удобного слайдера. |
| Повысить удобство выбора тарифа посредством реализации списка выбора тарифов в виде списка в форме. | Пользователь может выбрать нужный тариф из списка в форме. |
| Повысить удобство выбора бронирования посредством реализации соответствующих кнопок с доступным временем по датам. | |  | | --- | | Пользователь может выбрать доступное время для бронирования по заданным датам. |  |  | | --- | |  | |
| Обеспечить изменение, добавление, удаление информации о клиентах посредством реализации соответствующих кнопок и форм в админской панели. | |  | | --- | | Администратор может редактировать, добавлять и удалять информацию о клиентах через интерфейс админской панели. |  |  | | --- | |  | |
| Обеспечить изменение, добавление, удаление информации о тарифах посредством реализации соответствующих кнопок и форм в админской панели. | |  | | --- | | Администратор может редактировать, добавлять и удалять информацию о тарифах через интерфейс админской панели. |  |  | | --- | |  | |
| Обеспечить изменение, добавление, удаление информации о бронированиях посредством реализации соответствующих кнопок и форм в админской панели. | |  | | --- | | Администратор может редактировать, добавлять и удалять информацию о бронированиях через интерфейс админской панели. |  |  | | --- | |  | |
| Обеспечить просмотр статистической информации о клиентах посредством реализации таблиц со статистикой по клиентам. | |  | | --- | | Пользователь может просматривать статистические данные по клиентам в таблицах. |  |  | | --- | |  | |
| Обеспечить просмотр статистической информации о бронированиях посредством реализации таблиц со статистикой по бронированиям. | Пользователь может просматривать статистические данные по бронированиям в таблицах. |

## Область применения

Разрабатываемое приложение предназначено для использования в следующих информационных системах и технологических средах:

1. Веб-приложение – приложение доступно для настольных и мобильных браузеров с поддержкой следующих браузеров: Google Chrome, Mozilla Firefox, Safari, Edge.
2. Серверная инфраструктура – сервер развёрнут локально, обеспечивая хранение и обработку данных в пределах внутренней сети.

### 1.3.1 Используемые технологии

* СУБД :

- MongoDB: документоориентированная база данных, используемая для хранения информации о клиентах, тарифах и бронированиях. Позволяет эффективно обрабатывать запросы и управлять данными.

* Frontend:

- HTML, CSS: используются для создания структуры и стилизации веб-страниц.

- JavaScript: обеспечивает динамическое поведение интерфейса, а также реализацию логики взаимодействия с пользователем.

- DOM в JavaScript: используется для манипуляции элементами страницы, включая динамическое добавление и изменение контента на основе пользовательских действий.

* Backend:

- Golang: язык программирования для разработки серверной части приложения, обеспечивающий высокую производительность и масштабируемость.

- mongo-driver: библиотека для взаимодействия с базой данных MongoDB, реализующая операции чтения, записи и обновления данных.

- net/http: стандартный пакет для создания HTTP-сервера и обработки запросов от клиента.

- Хостинг: серверное приложение развернуто локально на `localhost`, обеспечивая тестирование и отладку функционала.

## 1.4 Текстовое описание бизнесс-процессов

В рамках разработки системы управления VR клуба «Поколение Z» реализованы бизнес-процессы, обеспечивающие функционал для пользователей и администраторов. Подробное описание всех процессов:

**1. Вход в администраторскую панель**

* Администратор использует форму авторизации для входа в панель управления, указывая логин и пароль. После успешного входа он получает доступ к основным разделам:
  + **Клиенты**;
  + **Тарифы**;
  + **Бронирования**;
  + **Статистика по тарифам**;
  + **Статистика по дням**.
* В случае неверных данных администратор получает сообщение об ошибке и может повторить попытку входа.

**2. Управление клиентами**

* В разделе «Клиенты» администратор видит таблицу со следующими столбцами:
  + **ФИО**;
  + **Телефон**;
  + **Email**.
* Функционал кнопок:
  + **Добавить клиента**: открывается форма для ввода данных нового клиента (ФИО, телефон, email). После сохранения информация отображается в таблице.
  + **Изменить запись**: при выборе клиента из списка становится доступна кнопка, позволяющая обновить его данные. Ввод изменений происходит в специальной форме.
  + **Удалить запись**: выбранная запись клиента удаляется из базы после подтверждения действия.

**3. Управление тарифами**

* Раздел «Тарифы» предоставляет список доступных тарифных планов:
  + **Базовый**;
  + **Плюс**;
  + **Премиум**;
  + **Чёрная пятница** (дополнительные акции, если применимо).
* Функционал кнопок:
  + **Добавить тариф**: открывается форма, где администратор может задать название тарифа, описание (например, доступные игры и устройства), стоимость и другие параметры. После сохранения тариф отображается в списке.

**4. Управление бронированиями**

* Раздел «Бронирования» позволяет администратору:
  + Просматривать таблицу с данными о бронированиях, включая:
    - **Имя клиента**;
    - **Выбранный тариф**;
    - **Дата бронирования**;
    - **Время бронирования**.
  + Добавлять бронирование через кнопку **Добавить бронирование**:
    - Администратор указывает клиента, тариф, дату и время бронирования в соответствующей форме.
  + Удалять бронирование с помощью кнопки **Удалить запись**.

**5. Просмотр статистики**

Система предоставляет два вида статистики для анализа работы VR клуба:

* **Статистика по тарифам**:
  + Для каждого тарифа отображается:
    - **Текущая прибыль**;
    - **Количество бронирований**.
* **Статистика по дням**:
  + Отображается таблица с количеством бронирований по конкретным датам.

**6. Функционал пользовательской стороны**

* Пользователь выбирает тариф на главной странице, где каждый тариф отображается с описанием, стоимостью и доступными функциями (игры, оборудование).
* Для бронирования пользователь:
  + Заполняет форму, указывая:
    - **Тариф** (выбирается из выпадающего списка);
    - **ФИО**;
    - **Телефон**;
    - **Email**;
    - **Желаемую дату**.
  + Подтверждает бронирование нажатием кнопки **Забронировать**.
  + Система проверяет корректность данных (например, формат email) и фиксирует запись.

**7. Работа с ошибками**

* Если введены некорректные данные, например, не выбран тариф или не указана дата, пользователю/администратору отображается соответствующее сообщение об ошибке с рекомендациями по исправлению.

# 2 Требования к изменяемой подсистеме

## 2.1 Требование к функционалу

Сервис «VR клуб – Поколение Z» был разработан для бронирования сеанса в VR клубе любым пользователем с возможностью выбора нужного тарифа и полноценным управлением всеми данными в базе данных через панель администратора.

**Сервис имеет следующие основные возможности:**

1. **Добавление пользователя и его записи бронирования:** Пользователи могут зарегистрироваться и забронировать сеанс на необходимую и свободную дату и время с предпочитаемым тарифом.
2. **Работа с пользователями:** Администратор может создавать/изменять/удалять пользователей, которые хранятся в базе через панель администратора.
3. **Работа с тарифами:** Администратор может создавать/изменять/удалять сами тарифы, а также делать аналогичные действия с играми и устройствами, которые хранятся внутри тарифа.
4. **Работа с бронированиями:** Администратор может добавлять/изменять/удалять записи бронирования пользователей.

2.1.1 **Добавление пользователя и его записи бронирования**

**Описание функционала: (остановился здесь!!!)**

Пользователь может создать рецепт, заполнив форму, которая состоит из следующих полей:

* Название рецепта (текстовое поле).
* Время подготовки (числовое поле).
* Время приготовления (числовое поле).
* Количество порций (числовое поле).
* Описание рецепта (текстовое поле).
* Шаги приготовления (многострочное текстовое поле).
* Ингредиенты и их количество (таблица из двух полей).
* Загрузка изображения рецепта.

Алгоритм работы формы:

1. Пользователь заполняет текстовые и числовые поля.
2. Добавляет шаги приготовления в виде списка (каждый шаг вводится с новой строки).
3. Вносит ингредиенты в таблицу: вводит название и количество.
4. Загружает изображение рецепта через кнопку загрузки.
5. После заполнения формы пользователь нажимает кнопку «Добавить», и данные отправляются на сервер.

2.1.2 Фильтрация и поиск рецептов

**Описание функционала:**

Пользователь может использовать строку поиска и фильтры для нахождения рецептов.

1. Поиск по ключевым словам в названии или описании рецепта.
2. Фильтрация рецептов по:
   * Количеству ингредиентов;
   * Числу порций;
   * Указанным ингредиентам.

Алгоритм работы поиска:

1. Пользователь вводит данные в строку поиска или задаёт параметры фильтров.
2. После нажатия кнопки «Показать» отправляется запрос на сервер.
3. Сервер возвращает список рецептов, соответствующих заданным параметрам.

Особенности интерфейса:

* Список фильтров можно очищать и задавать новые параметры поиска.

2.1.3 Просмотр и работа с рецептами

**Описание функционала:**  
На странице отображается список рецептов, каждый из которых представлен в виде карточки.

Элементы карточки рецепта:

* Название рецепта.
* Время подготовки и приготовления.
* Количество порций.
* Краткое описание.
* Список ингредиентов.
* Кнопка для добавления в избранное.

Алгоритм работы:

1. Карточки рецептов загружаются после выполнения поискового запроса или нажатия кнопки «Избранное».
2. При нажатии на рецепт пользователь переходит на страницу с его полной информацией.

**Дополнительно:**  
Рецепты в избранном выделяются визуально.

2.1.4 Система избранного

**Описание функционала:**  
Пользователи могут сохранять рецепты в избранное для быстрого доступа.

Алгоритм работы:

1. При нажатии на кнопку «Избранное» рецепт добавляется в базу данных пользователя.
2. Если рецепт уже находится в избранном, при повторном нажатии он удаляется.
3. Избранные рецепты доступны на отдельной вкладке интерфейса.

2.1.5 Загрузка и отображение изображений

**Описание функционала:**  
Каждый рецепт может содержать изображение, которое пользователь загружает через форму.

Требования к изображению:

* Поддержка форматов JPEG и PNG.
* Максимальный размер файла: 5 МБ.
* Автоматическая проверка типа файла перед загрузкой.

Алгоритм работы:

1. Пользователь загружает изображение через форму.
2. Изображение отображается в режиме предварительного просмотра.
3. После отправки формы файл загружается на сервер, где сохраняется с привязкой к соответствующему рецепту.

## 2.3 Требования к ролям