

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук
Кафедра информационных систем

Учет статистики соревнований по биатлону

*Курсовой проект
по дисциплине
Технологии программирования*

*09.03.02 Информационные системы и технологии
Информационные системы в телекоммуникациях*

6 семестр 2022/2023 учебного года

Зав. кафедрой	_____ к. т. н., доцент Д.Н. Борисов
Обучающийся	_____ ст. 3 курса оч. отд. А. С. Тихонов
Обучающийся	_____ ст. 3 курса оч. отд. А. С. Новикова
Обучающийся	_____ ст. 3 курса оч. отд. Б. В. Шульга
Руководитель	_____ В.С. Тарасов, ст. преподаватель _____.20__

Воронеж 2023

Содержание

Введение	4
1 Постановка задачи.....	5
1.1 Требования к разрабатываемой системе	5
1.1.1 Функциональные требование	5
1.1.2 Технические требования.....	5
1.1.3 Требования к интерфейсу.....	5
1.1.4 Задачи, решаемые в процессе разработки	6
2 Анализ предметной области	7
2.1 Глоссарий.....	7
2.2 Обзор аналогов	8
2.2.1 Сайт olympteka.ru	8
2.2.2 Сайт rusbiathlon.ru	10
2.3 Графическое описание работы системы	14
2.3.1 Диаграмма IDEF0	14
2.3.2 Диаграмма прецедентов	15
2.3.3 Диаграмма классов	16
2.3.4 Диаграмма состояний.....	17
2.3.5 Диаграмма активностей	18
2.3.6 Диаграмма последовательности	19
2.4 Продуктовые воронки	21
3 Реализация	22
3.1 Средства реализации	22
3.2 Реализация серверной части приложения.....	23
3.2.1 Схема базы данных	23

3.2.2 Архитектура серверной части приложения	28
3.3 Реализация клиентской части приложения.....	29
3.3.1 Графический интерфейс.....	30
4 Тестирование	37
4.1 Дымовое тестирование.....	37
4.2 UI тесты	37
4.3 Юзабилити тесты	37
Заключение	38
Список используемых источников.....	39

Введение

В последние годы в информационном обществе всё больше изменяется уклад жизни, система ценностей: возрастает спрос на знания, от человека требуется способность к интеллектуальному труду и творчеству. Возможности человека по восприятию и переработке информации ограничены наряду с существующими массивами хранящихся и передаваемых данных. Возникло большое число избыточной информации, в которой иногда трудно сориентироваться и выбрать нужные сведения. Для решения подобных проблем применяются автоматизированные информационные системы и базы данных. Они стали неотъемлемы для всех сфер жизни человека, таких как наука, образование, искусство и т.д. И конечно же это не могло обойти такую часть нашей жизни, как спорт.

Спорт в настоящее время становится все более популярным. Он широко пропагандируется и рекламируется как один из компонентов здорового образа жизни. Спортивные достижения, рейтинг спортсменов и спортивная статистика становятся интересны все более широкому кругу людей, не имеющих отношения к профессиональному спорту, а использование интернета приведет к увеличению популярности спорта как такового. Актуальность работы выражается в возможности широкого применения ее результатов. Поскольку интернет является неотъемлемой частью современного общества, то взаимодействие интернета и спорта заставит последнего модернизироваться и соответствовать требованиям времени. Доступность интернета делает статистические данные, метод их сбора и обработки более прозрачными. Разрабатываемое интернет-приложение позволит усовершенствовать процесс сбора данных, минимизировать человеческий труд, создать удобный интерфейс для пользователя.

1 Постановка задачи

Целью курсового проекта является разработка программного продукта, позволяющего вести статистику спортивных мероприятий по биатлону.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

Система должна отвечать таким функциональным требованиям как:

- регистрация и авторизация пользователей;
- разделять пользователя по ролям “Гость” / “Редактор”;
- реализация всех ролей системы;
- выдавать статистические данные по спортивным мероприятиям по биатлону;
- поиск интересующего чемпионата по названию или дате;
- подсчет количества медалей определенной страны за определенный чемпионат по выбранным категориям (среди мужчин, среди женщин, среди мужчин и женщин и т. д.).

1.1.2 Технические требования

Программный продукт должен обеспечить:

- авторизацию пользователей с использованием логина и пароля;
- шифрование логина и пароля при записи в БД;
- хранение данных в БД.

1.1.3 Требования к интерфейсу

Система должна отвечать на такие требования к интерфейсу, как:

- логически понятный для пользователя интерфейс;
- интерфейс, выполненный в единой цветовой гамме и стиле;
- все надписи должны быть легко читабельны;
- содержание только необходимой пользователю информации;

- основные элементы управления должны быть заметны для пользователя;
- сайт должен быть оптимизирован для изменения размеров экрана.

1.1.4 Задачи, решаемые в процессе разработки

Перед проектом были поставлены следующие задачи:

- анализ предметной области;
- анализ аналогов;
- написание технического задания;
- проектирование веб-сервиса средствами языка UML;
- определение используемой платформы;
- подключение внешнего модуля для хранения данных;
- разработка БД;
- реализация ролей;
- реализация функциональных возможностей ролей;
- разработка функциональных возможностей сайта;
- создание макета дизайна и его реализация;
- реализация интерфейса;
- проведение тестирования;
- описание процесса разработки и результата.

2 Анализ предметной области

2.1 Глоссарий

Таблица 1 -Глоссарий

Проект	Это некоторая задача с определенными исходными данными и требуемыми результатами (целями), обуславливающими способ ее решения.
Гость	Это учетная запись, предназначенная для пользователей, не имеющих постоянной учетной записи на компьютере или в домене.
Пользователь	Это человек, который посетил ресурс или совершил на нем какое-либо действие.
Администратор	Это специалист отвечает за производство уникального контента, который размещается на сайте.
Сервер. Серверная часть	Это компьютер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.
Клиент, клиентская часть	Это устройство, основным приложением которого (с точки зрения разработчика устройства или маркетолога) является браузер.
Back-end	Это логика работы сайта, скрытая от пользователя.
Front-end	Это разработка пользовательского интерфейса и функций, которые работают на клиентской стороне веб-сайта или приложения.

GitHub	Это такая социальная сеть для разработчиков, которая помогает удобно вести коллективную разработку IT-проектов.
--------	---

2.2 Обзор аналогов

Для создания программного средства, необходимо изучить представленные аналоги и выделить их основные недостатки и преимущества, определить ключевые особенности тенденций в данном направлении.

В качестве исследуемых аналогов были выбраны программные продукты, связанные с учетом статистики соревнований по биатлону. Основным критерием для выбора служила актуальность данных программных средств, частота их использования, представленный функционал. Источником информации послужили электронные базы в сети Интернет. В результате поиска были выявлены 2 программных продукта:

- Сайт olympteka.ru [1];
- Сайт rusbiathlon.ru [2].

2.2.1 Сайт olympteka.ru

На Рисунке 1 представлен интерфейс Главной страницы сайта olympteka.ru. На этом сайте представлен весь необходимый функционал для просмотра статистики по биатлону.

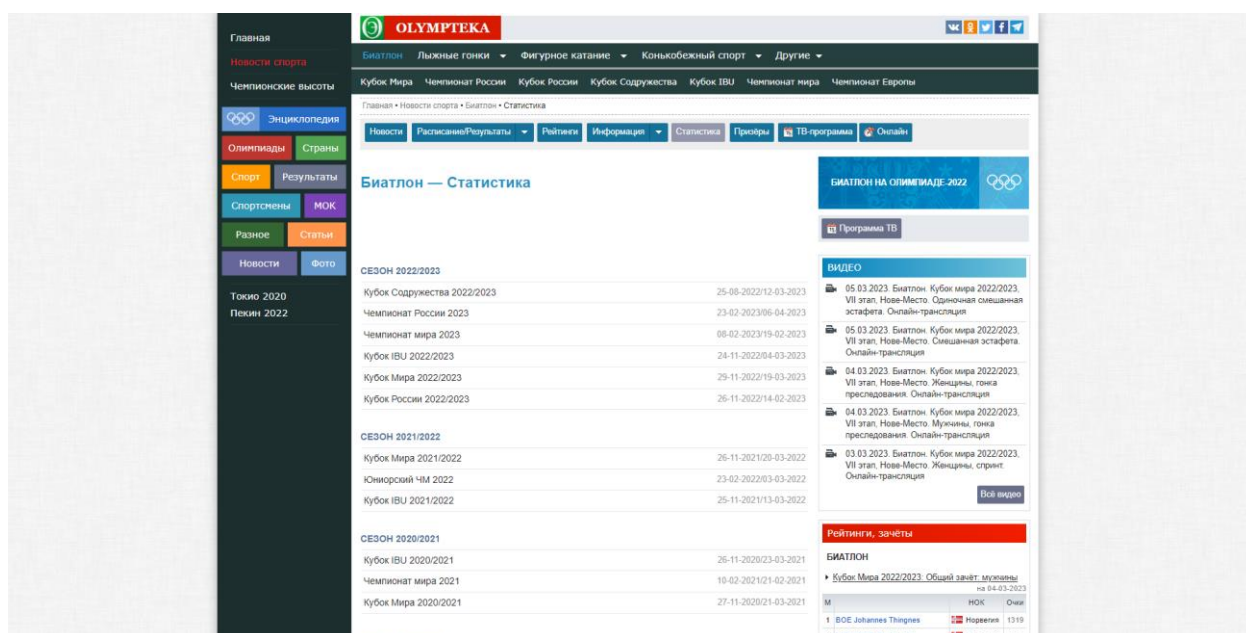


Рисунок 1 - Главная страница сайта olympteka.ru

На главной странице сайта можно увидеть список чемпионатов по биатлону, отсортированные по сезонам. Выбрав интересующий нас чемпионат, мы переходим на страницу с таблицами с разными категориями (Рисунок 2), в которых содержится информация о призовом месте, стране, количестве золотых, серебряных и бронзовых медалей, общем количестве медалей, полученные конкретной страной и количеством попаданий в 20, 30, 40.

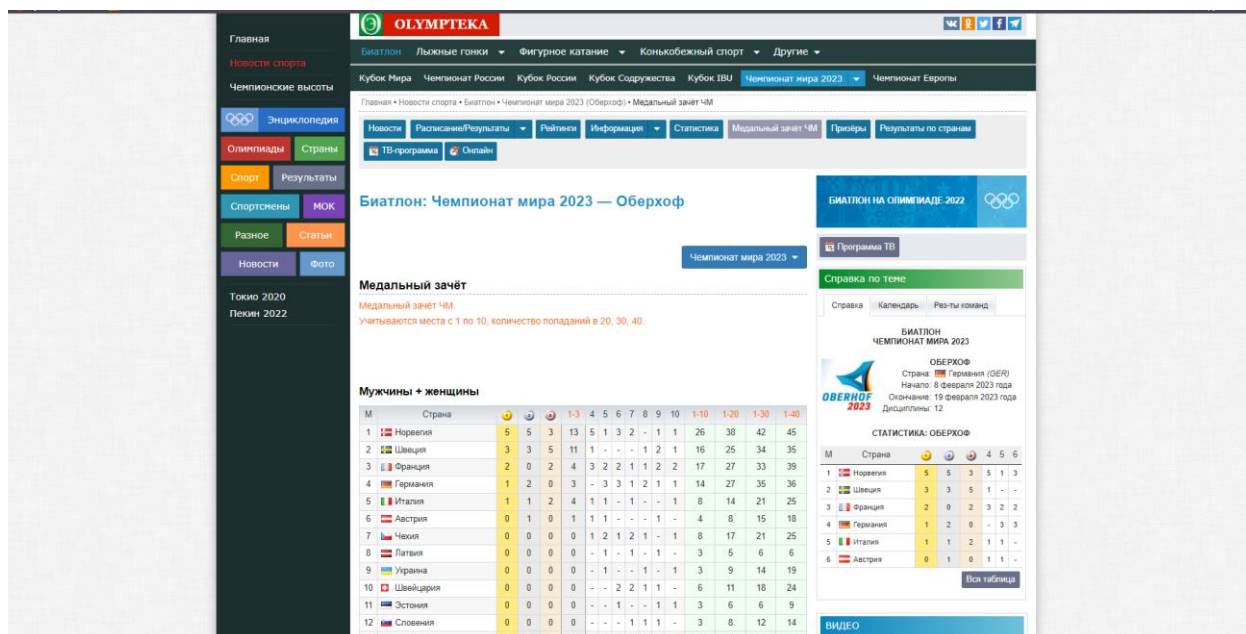


Рисунок 2 - Страница просмотра статистики на сайте olymppteka.ru

Достоинства:

- понятный интерфейс, отсутствие перенасыщенного дизайна (отсутствие динамических элементов, анимации);
- многостраничность;
- разбиение на информационные блоки;
- отсутствие излишней информации;
- в представленных таблицах столбцы с медалями удобно разделены по цветам, что упрощает просмотр интересующих нас значений.

Недостатки:

- несовременный UI;
- затруднено использование сайта людьми с ограниченными способностями (наличие маленьких элементов);
- отсутствие регистрации пользователя и чатов для обсуждения;

2.2.2 Сайт rusbiathlon.ru

На Рисунке 3 представлена главная страница сайта, которая содержит информацию о текущих чемпионатах, ссылки на статистические данные чемпионатов, которые проходили в этом сезоне и разделы для обсуждения новостей.

The screenshot shows the homepage of the Russian Biathlon website (rusbiathlon.ru). The header includes navigation links for various sports and biathlon-specific sections. Below the header, there are links for registration and login. The main content area is divided into several sections:

- Статистика биатлона**: This section contains two tables showing the overall standings for the IBU Cup and the IBU Cup for Women.
- Спортивная статистика**: This section provides links to various statistical data, including the overall standings for the IBU Cup and the IBU Cup for Women.
- Новости**: This section contains a list of recent news articles, including updates on the IBU Cup and the IBU Cup for Women.

The tables in the 'Статистика биатлона' section show the following data:

№	Мужчины	Страна	Очки
1	BOE Johannes Thingnes	NOR	1319
2	LAEGREID Sturla Holm	NOR	920
3	CHRISTIANSEN Vete Sjaastad	NOR	682
4	BOE Torge	NOR	596
5	PONSLIUDMA Martin	SWE	567
6	DOLL Benedikt	GER	534
7	REES Roman	GER	510
8	FILLON MAILLET Quentin	FRA	510
9	DALE Johannes	NOR	500
10	CLAUDE Fabien	FRA	499

№	Женщины	Страна	Очки
1	SIMON Julia	FRA	893
2	OEBERG Elvira	SWE	735
3	VITTOZZI Lisa	ITA	720
4	WIERER Dorothea	ITA	679
5	HERRMANN-WICK Denise	GER	668
6	TANDREVDOL Ingrid Landmark	NOR	627
7	HAUSER Lisa Theresa	AUT	574
8	DAVIDOVA Marketa	CZE	554
9	CHEVALIER-BOUCHET Anais	FRA	523
10	PERSSON Linn	SWE	518

Рисунок 3 - Главная страница сайта rusbiathlon.ru

Выбрав интересующий нас чемпионат, мы переходим на страницу с одной общей таблицей (Рисунок 4), где содержится информация только о странах и общим количеством очков.

№	Страна	Очки
1	Norway	7730
2	France	6834
3	Germany	6552
4	Sweden	6353
5	Switzerland	5430
6	Czech Republic	5374
7	Austria	5263
8	Italy	5230
9	Ukraine	5078
10	Finland	5036
11	Slovenia	5001
12	United States	4262
13	Lithuania	3883
14	Romania	3874
15	Estonia	3530
16	Canada	3429
17	Poland	3351
18	Latvia	2901
19	Bulgaria	2900
20	Kazakhstan	2664
21	Belgium	2564
22	Moldova	2451
23	Japan	1796
24	Slovakia	1709
25	Korea	1585
26	New Zealand	841
27	Croatia	762
28	Peoples Rep. of China	481
29	Greece	128
30	Spain	109
31	Turkey	34
32	Denmark	34
33	Bosnia & Herzegovina	16
34	Great Britain	13
35	Hungary	8

Рисунок 4 - Страница просмотра статистической информации на сайте rusbiathlon.ru

При регистрации сайт предлагает вам выбрать интересующий вас статус, ввести имя и почту, на которую в дальнейшем будет выслана информация для авторизации пользователя (Рисунок 5).

Регистрация (для активных участников)

Ваш статус: Я болельщик! (фан-клуб)

Ваше имя: _____

Ваш e-mail: _____

☐ Я принимаю пользовательские соглашения и согласен с политикой конфиденциальности

Отправить

Рисунок 5 - Регистрация на сайте rusbiathlon.ru

Но в процессе авторизации, сайт потребовал авторизацию через соц. сети, что нельзя не отметить, как большой минус для пользователей, которые заботятся о собственной конфиденциальности (Рисунок 6).

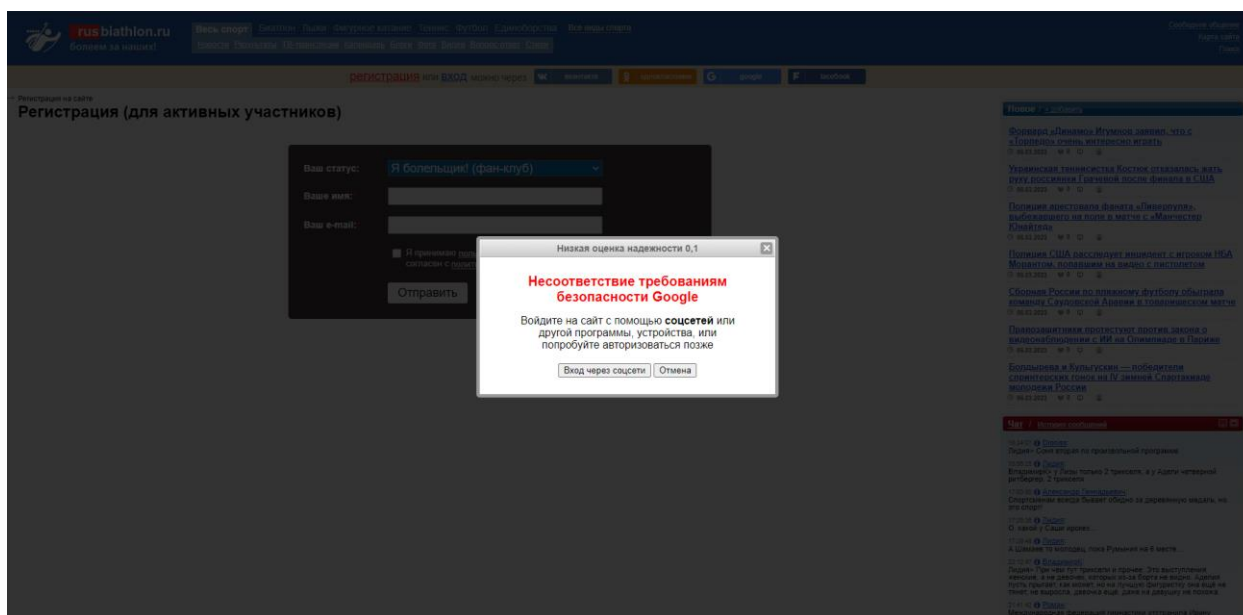


Рисунок 6 - Ошибка при авторизации на сайте rusbiathlon.ru

Достоинства:

- наличие поиска;
- наличие общих чатов для обсуждения интересующих вопросов;
- наличие разбиения на информационные блоки.

Недостатки:

- несовременный UI;
- затруднено использование сайта людьми с ограниченными способностями (наличие маленьких элементов);
- большая информационная загруженность, непонятный и сложный графический интерфейс;
- используется сторонние ресурсы для регистрации участников;
- недостаточное количество статистической информации по чемпионатам;

— при изменении размеров сайта информационные блоки смещаются в хаотичном порядке.

2.3 Графическое описание работы системы

2.3.1 Диаграмма IDEF0

Рассмотрим основные бизнес – процессы на примере диаграммы IDEF0 диаграммы, которая представлена на Рисунке 7.

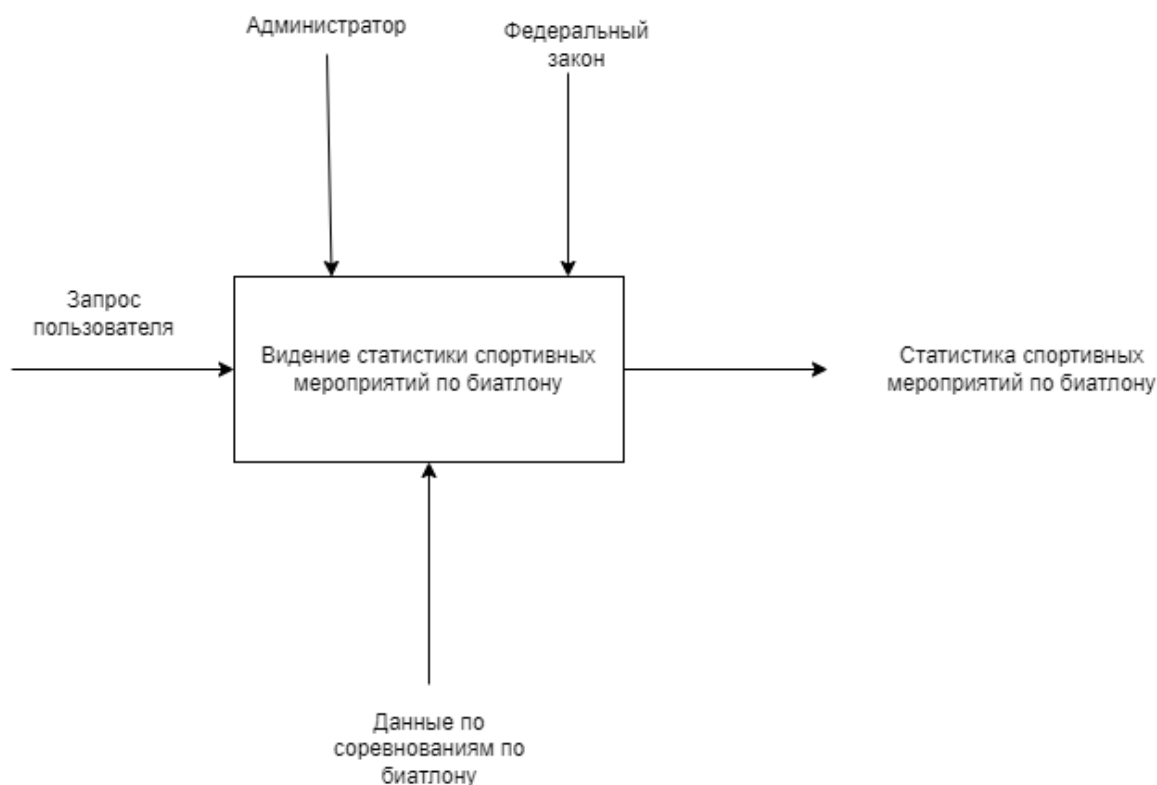


Рисунок 7 - IDEF0 диаграмма

Данная диаграмма используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции.

Работа сервиса регулируется Федеральным законом. На вход в систему поступает Пользователь (авторизованный или нет), на выходе система выдает информацию о статистических данных по биатлону. Работу обеспечивает Администратор сервиса и Исходная БД по статистическим данным о соревнованиях по биатлону.

2.3.2 Диаграмма прецедентов

На Рисунке 8 продемонстрирована диаграмма Use Case, которая показывает какие сценарии использования приложения доступны различным пользователям веб-приложения.



Рисунок 8 - Диаграмма USE CASE

В этой системе можно выделить следующие группы пользователей:

- незарегистрированный пользователь;
- зарегистрированный пользователь;
- администратор.

Каждая из групп пользователей может пользоваться разрабатываемой системой по-своему.

Незарегистрированные пользователи могут:

- просматривать таблицы соревнований;
- просматривать чат;
- пользоваться калькулятором медалей.

Зарегистрированные пользователи могут делать все то же самое, что и незарегистрированные пользователи, но дополнительно:

- оставлять сообщение в чате.

Администратор может делать все то же самое, что и зарегистрированные пользователи, но дополнительно:

- редактировать данные о сезонах;
- редактировать данные о соревнованиях;
- редактировать данные в таблицах с результатами.

2.3.3 Диаграмма классов

На Рисунке 9 представлена диаграмма классов. Диаграмма классов UML используется для описания структуры объектно-ориентированной системы. Она показывает классы, их атрибуты и методы, а также связи между классами.

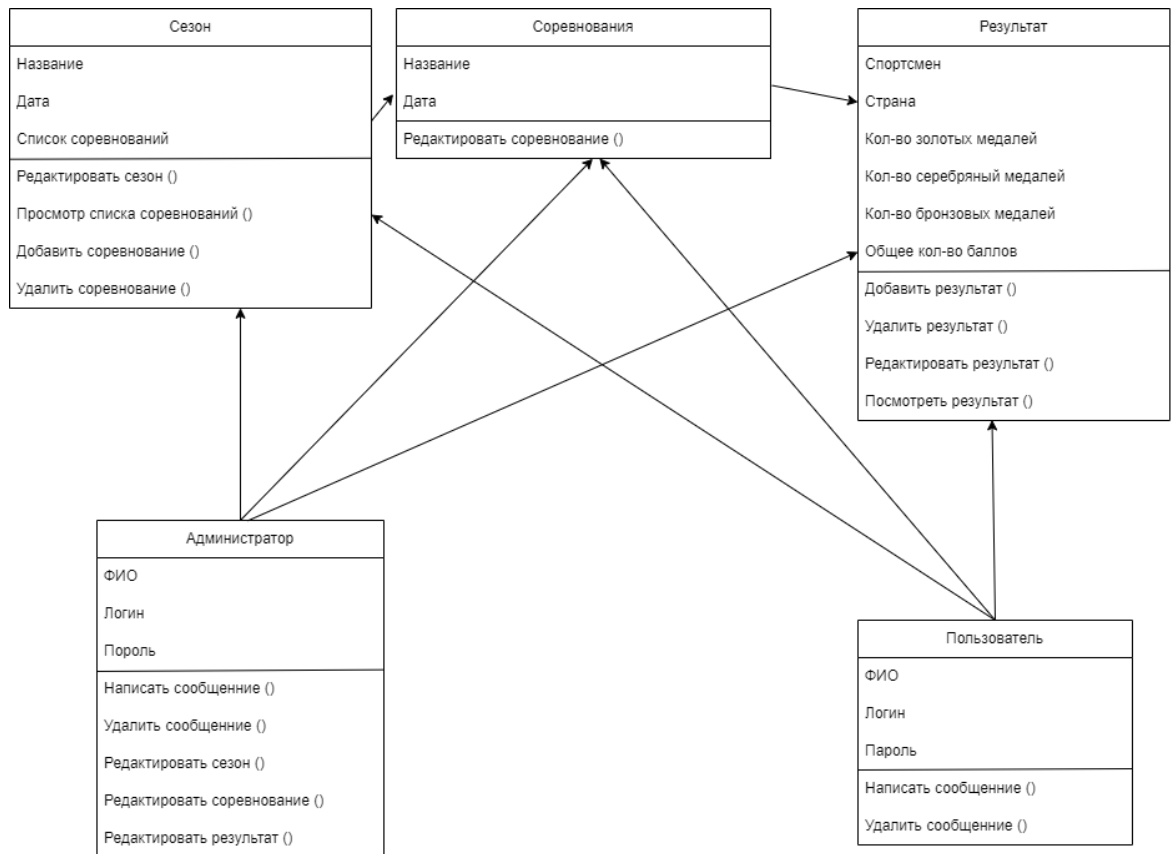


Рисунок 9 - Диаграмма классов

2.3.4 Диаграмма состояний

На Рисунке 10 представлена Диаграмма состояний. Она используется для описания поведения объекта или системы в различных состояниях, а также переходы между ними.

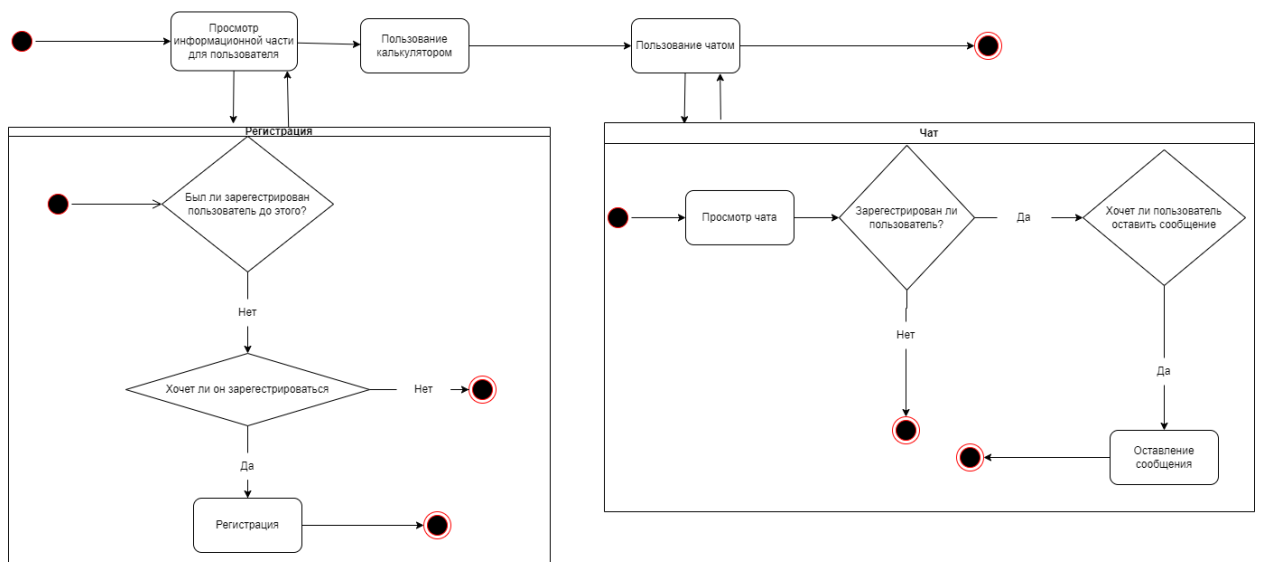


Рисунок 10 - Диаграмма состояний

После входа в систему пользователь может просматривать интересующую его информацию и вести подсчет медалей с помощью калькулятора без авторизации. Пользователь сервиса также может пройти процесс регистрации, и это дает ему возможность оставление сообщений чате для обсуждения интересующих вопросов.

2.3.5 Диаграмма активностей

На Рисунке 11 представлена диаграмма активностей. Она используется для моделирования процессов и последовательностей действий, которые должны быть выполнены для достижения определенной цели.

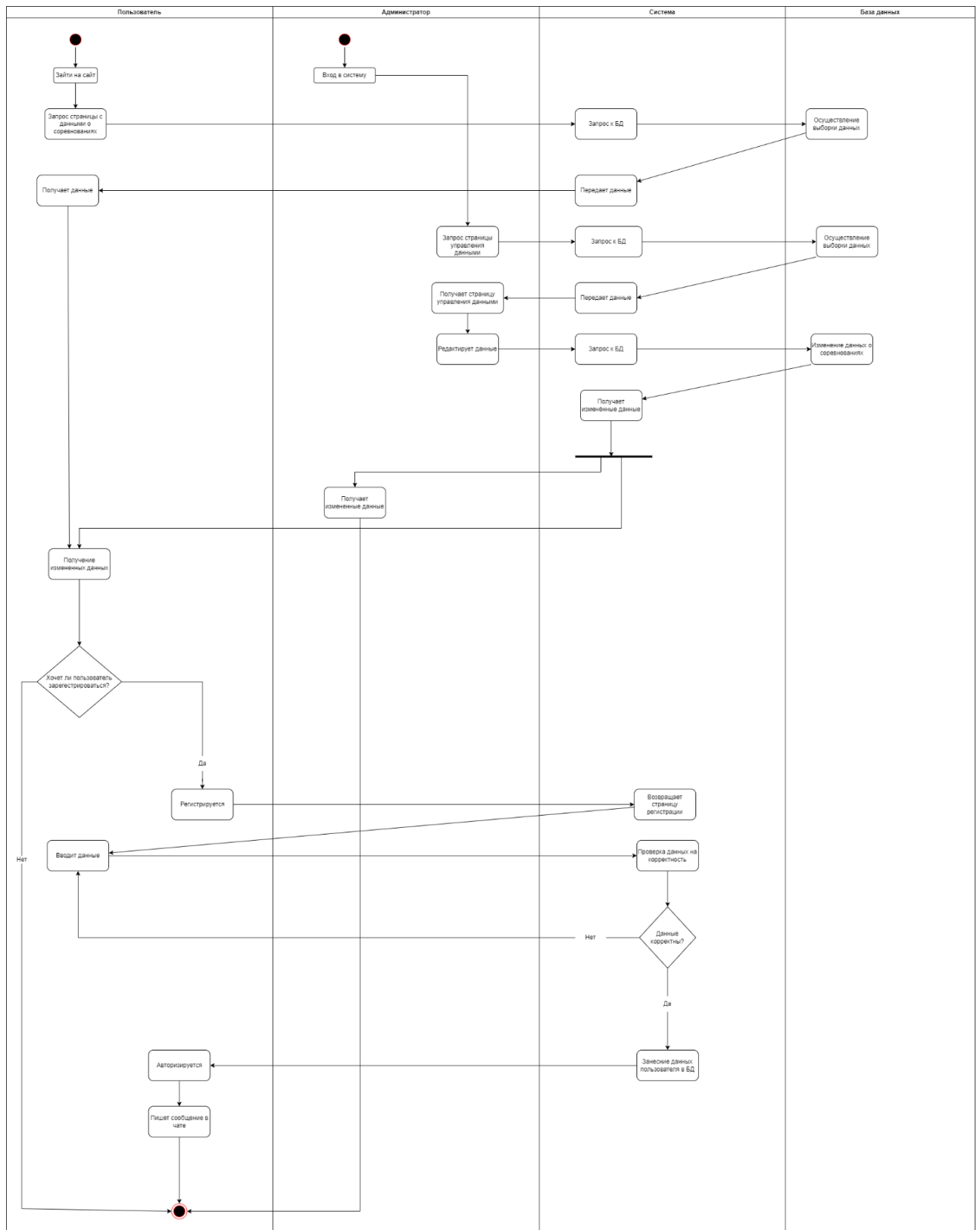


Рисунок 11 - Диаграмма активностей

2.3.6 Диаграмма последовательности

На Рисунке 12 представлена диаграмма последовательностей. Она используется для визуализации последовательности действий, которые выполняются в ходе взаимодействия различных объектов и компонентов в системе.

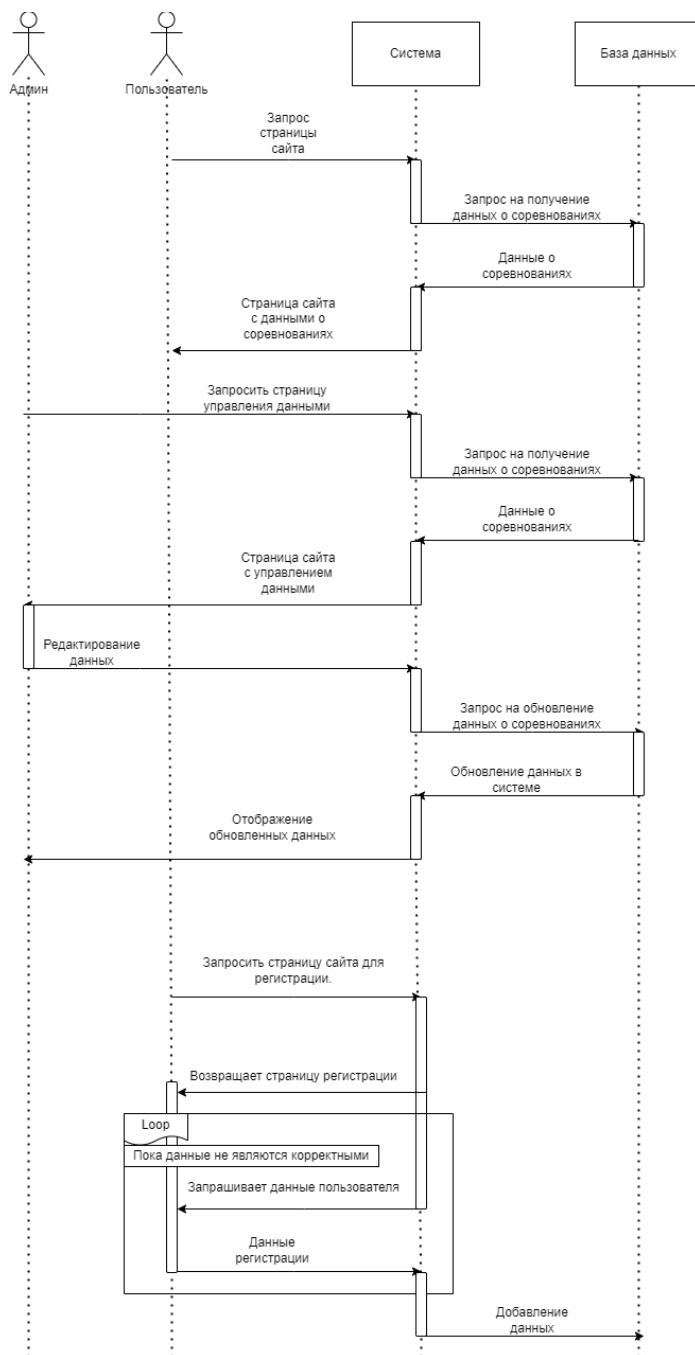


Рисунок 12 - Диаграмма последовательностей

На диаграмме показано упорядоченное во времени взаимодействие объектов и системы при просмотре статистической информации, редактирование их, регистрации и авторизации пользователей в систему.

2.4 Продуктовые воронки

3 Реализация

3.1 Средства реализации

Веб-приложение должно соответствовать модели клиент-серверного приложения.

В качестве средств реализации для серверной части приложения были выбраны:

- ОС Windows 10;
- frameworks – Django 4.2;
- язык разработки – Python 3.10.11;
- база данных – PostgreSQL 14;
- система контроля версий – Git 2.40.0.

Для серверной части была выбрана связка Python и Django, так как фреймворк Django имеет большое количество преимуществ:

- множество библиотек, которые включают готовые решения;
- автоматическое создание административной панели. Она генерируется при создании приложения и адаптируется под проект с помощью сторонних приложений;
- ORM (Объектно-реляционное отображение) упрощает взаимодействие приложения с базами данных. Зачастую программисту для этого даже не требуется знать язык, который для этого используется, что значительно ускоряет процесс разработки прототипов и веб-приложений.

В качестве средств реализации для клиентской части приложения были выбраны:

- ОС Windows 10;
- frameworks – Vue 2;
- язык разработки – TypeScript 4.5.5;
- среда исполнения Node.js v19.4.0;

- язык гипертекстовой разметки - HTML;
- формальный язык описания внешнего вида документа – CSS;
- система контроля версий – Git 2.40.0.

В качестве фреймворка для клиентской части приложения был выбран Vue, поскольку он имеет такие преимущества, как:

- библиотека довольно проста и функциональна;
- требования к стеку отсутствуют, поэтому Vue.JS можно использовать на любом проекте;
- довольно высокая скорость разработки. Благодаря использованию любых шаблонов и доступности документации, большинство возникающих проблем решаются довольно быстро.

3.2 Реализация серверной части приложения

3.2.1 Схема базы данных

На Рисунке 13 представлена ER - диаграмма базы данных веб-приложения.

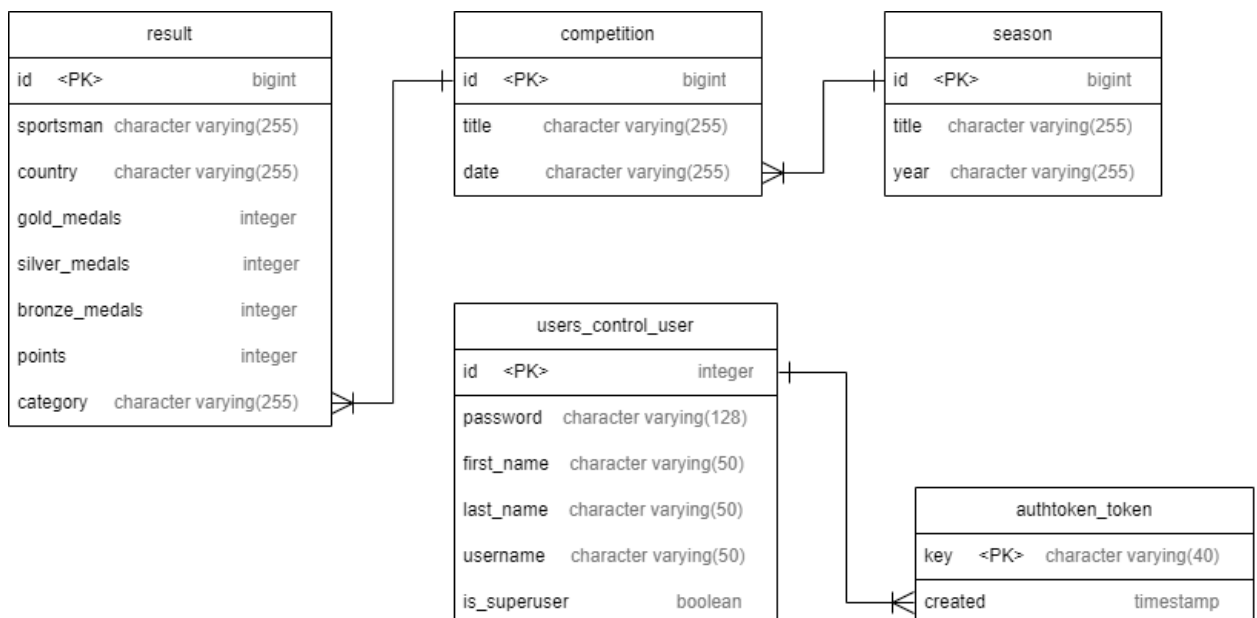


Рисунок 13 - ER - диаграмма базы данных

В базе данных представлены 3 сущности:

- result;
- competition;
- season;
- users_control_user;
- authToken_token.

Таблица “result” предназначена для хранения данных о категориях, спортсменах, количестве медалей и общем результате для таблиц, которые будут представлены пользователям на странице веб-приложения.

Таблица имеет такие атрибуты как:

- “id”. Главный ключ. Домен: целые числа. Уникальный идентификатор результата. Назначается автоматически при помощи СУБД;
- “sportsman”. Домен: строки. ФИО спортсмена;
- “country”. Домен: строки. Название страны, которую представлял спортсмен;
- “gold_medals”. Домен: целые числа. Количество золотых медалей, которое было завоевано спортсменом;
- “silver_medals”. Домен: целые числа. Количество серебряных медалей, которое было завоевано спортсменом;
- “bronze_medals”. Домен: целые числа. Количество бронзовых медалей, которое было завоевано спортсменом;
- “points”. Домен: целые числа. Общий счет спортсмена за соревнование;
- “category”. Домен: строки. Категория, в которой участвовал спортсмен;

Таблица “competition” предназначена для хранения общих данных о соревнованиях.

Таблица имеет такие атрибуты как:

- “id”. Главный ключ. Домен: целые числа. Уникальный идентификатор соревнования. Назначается автоматически при помощи СУБД;
- “title”. Домен: строки. Название соревнования;
- “date”. Домен: строки. Дата проведения соревнования.

Таблица “season” предназначена для хранения общих данных о сезонах.

Таблица имеет такие атрибуты как:

- “id”. Главный ключ. Домен: целые числа. Уникальный идентификатор сезона. Назначается автоматически при помощи СУБД;
- “title”. Домен: строки. Название сезона;
- “date”. Домен: строки. Сроки проведения сезона.

Таблица “users_control_user” предназначена для хранения данных о авторизованных пользователях.

Таблица имеет такие атрибуты как:

- “id”. Главный ключ. Домен: целые числа. Уникальный идентификатор пользователя. Назначается автоматически при помощи СУБД;
- “password”. Домен: строки. Пароль пользователя;
- “first_name”. Домен: строки. Имя пользователя;
- “last_name”. Домен: строки. Фамилия пользователя;
- “username”. Домен: строки. Уникальный атрибут. Никнейм, под которым зарегистрировался пользователь;
- “silver_medals”. Домен: целые числа. Количество серебряных медалей, которое было завоевано спортсменом;
- “is_superuser”. Домен: логический тип. Атрибут для определения, является ли пользователь администратором;

Таблица “authtoken_token” предназначена для хранения данных о токенах.

Таблица имеет такие атрибуты как:

- “key”. Главный ключ. Домен: строки. Уникальный идентификатор токена;
- “created”. Домен: бинарные данные. Время создания токена;

На Рисунке 10 представлена физическая схема базы данных.

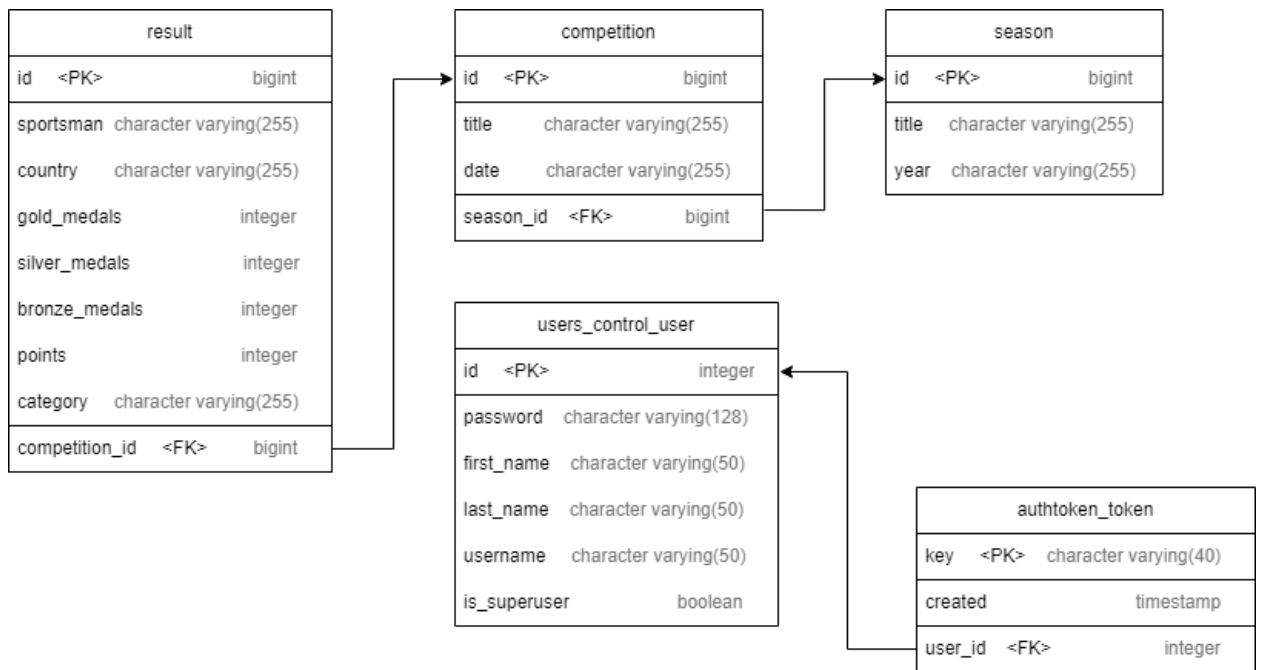


Рисунок 14 - Схема базы данных серверной части приложения

Сущности “result” и “competition”.

Связь между сущностями 1: m. Класс принадлежности “competition” к “result” и “result” к “competition” обязательный.

На ER-диаграмме представлены две сущности “competition” и “result”, связанные связью один-ко-многим с обязательным классом принадлежности сущностей по обеим сторонам связи. То есть результат должен принадлежать только одному соревнованию, а соревнование должен иметь несколько результатов.

Реляционная схема, соответствующая данной ER-диаграмме, содержит два отношения – “competition” и “result”, одноименные исходным сущностям.

Идентифицирующие свойства этих отношений становятся их первичными ключами (<PK>). Это соответственно, атрибут ID (“competition”) и атрибут ID (“result”). В отношение, соответствующее сущности с множественной связностью (“result”), вводится атрибут ID (“competition”), который становится внешним ключом, ссылающимся на одноименный первичный ключ отношения “competition”.

Сущности “competition” и “season”.

Связь между сущностями 1: m. Класс принадлежности “season” к “competition” и “competition” к “season” обязательный.

На ER-диаграмме представлены две сущности “season” и “competition”, связанные связью один-ко-многим с обязательным классом принадлежности сущностей по обеим сторонам связи. То есть соревнование должно принадлежать только одному сезону, а сезон должен иметь несколько соревнований.

Реляционная схема, соответствующая данной ER-диаграмме, содержит два отношения – “season” и “competition”, одноименные исходным сущностям.

Идентифицирующие свойства этих отношений становятся их первичными ключами (<PK>). Это соответственно, атрибут ID (“season”) и атрибут ID (“competition”). В отношение, соответствующее сущности с множественной связностью (“competition”), вводится атрибут ID (“season”), который становится внешним ключом, ссылающимся на одноименный первичный ключ отношения “season”.

Сущности “authtoken_token” и “users_control_user”.

Связь между сущностями 1: m. Класс принадлежности “users_control_user” к “authtoken_token” и “authtoken_token” к “users_control_user” обязательный.

На ER-диаграмме представлены две сущности “users_control_user” и “authtoken_token”, связанные связью один-ко-многим с обязательным классом принадлежности сущностей по обеим сторонам связи. То есть токен должен принадлежать только одному пользователю, а пользователь должен иметь несколько токенов.

Реляционная схема, соответствующая данной ER-диаграмме, содержит два отношения – “users_control_user” и “authtoken_token”, одноименные исходным сущностям.

Идентифицирующие свойства этих отношений становятся их первичными ключами (<PK>). Это соответственно, атрибут ID (“users_control_user”) и атрибут KEY (“authtoken_token”). В отношение, соответствующее сущности с множественной связностью (“authtoken_token”), вводится атрибут ID (“users_control_user”), который становится внешним ключом, ссылающимся на одноименный первичный ключ отношения “users_control_user”.

3.2.2 Архитектура серверной части приложения

В рамках данного приложения серверная часть отвечает за обработку запросов от клиентов, управление базой данных и предоставление необходимой функциональности для работы с данными биатлонистов, сезонов, соревнований и других связанных событий.

Основные компоненты серверной части приложения включают следующие файлы и функциональность:

- файлы маршрутизации: эти файлы определяют маршруты URL, к которым можно обратиться для выполнения различных операций. Например, файлы маршрутизации определяют маршруты для создания, чтения, обновления и удаления данных, таких как данные о сезонах или результатах соревнований.
- модели данных: в этой части приложения определены модели данных, описывающие сущности, такие как пользователи, сезоны,

соревнования и результаты. Модели данных определяют структуру данных и связи между ними.

- контроллеры: контроллеры отвечают за обработку запросов от клиентов и взаимодействие с моделями данных. Они содержат функции для создания, чтения, обновления и удаления данных, а также другие операции, связанные с бизнес-логикой приложения.
- база данных: серверная часть приложения взаимодействует с базой данных для хранения и извлечения данных. В серверной части приложения содержатся файлы, связанные с настройкой базы данных и файлы миграции, которые определяют структуру базы данных и её обновления.
- другие служебные файлы: это служебные файлы, такие как файлы конфигурации, файлы для обработки аутентификации и авторизации пользователей или файлы для настройки среды выполнения серверной части приложения.

3.3 Реализация клиентской части приложения

Клиентская часть отвечает за интерфейс пользователя и взаимодействие с серверной частью приложения для отображения данных и выполнения операций. Основные компоненты клиентской части приложения включают следующие файлы и функциональность:

- vue файлы: они определяют структуру и внешний вид пользовательского интерфейса. Файлы состоят из 3 блоков:
 1. `<template>` - блок, который содержит HTML-разметку компонента, используя синтаксис шаблонизатора. В этом блоке определяется внешний вид компонента.
 2. `<script>` - блок, который содержит TypeScript-код компонента. Здесь определяются методы и свойства компонента, а также взаимодействие с другими компонентами и приложением в целом.

3. `<style>` - блок, который содержит CSS-стили компонента. В этом блоке можно определить стили для всех элементов компонента.

- TypeScript файлы: они обрабатывают события пользователя, отправляют запросы на серверную часть приложения, хранят данные о состоянии приложения, обновляют модель данных
- файлы конфигурации: репозиторий содержит файлы конфигурации, которые определяют настройки клиентской части приложения, такие как URL сервера, порты и другие параметры. Файлы конфигурации обеспечивают гибкость и возможность настройки приложения для различных окружений.
- библиотеки и зависимости: в файле зависимостей (например, `package.json` или `package-lock.json`) указаны сторонние библиотеки и модули, необходимые для работы клиентской части приложения.

3.3.1 Графический интерфейс

При входе в веб-приложение пользователь может зарегистрироваться (Рисунок 15). Для этого ему необходимо ввести следующие данные:

- имя;
- фамилия;
- логин;
- пароль;
- подтверждение пароля.

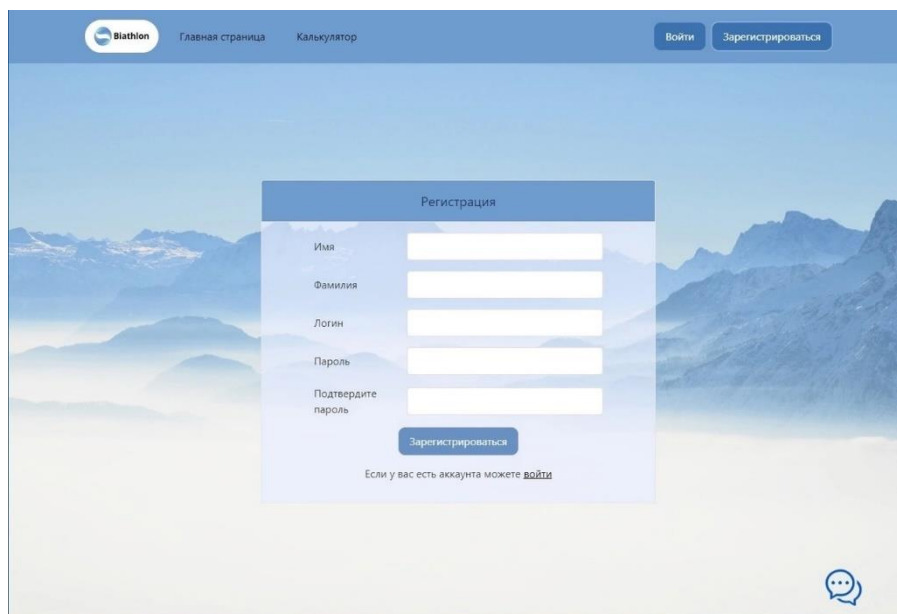


Рисунок 15 - Страница регистрации

После процесса регистрации, пользователь может воспользоваться личным аккаунтом. Для этого ему необходимо ввести необходимую информацию на странице авторизации (Рисунок 16).

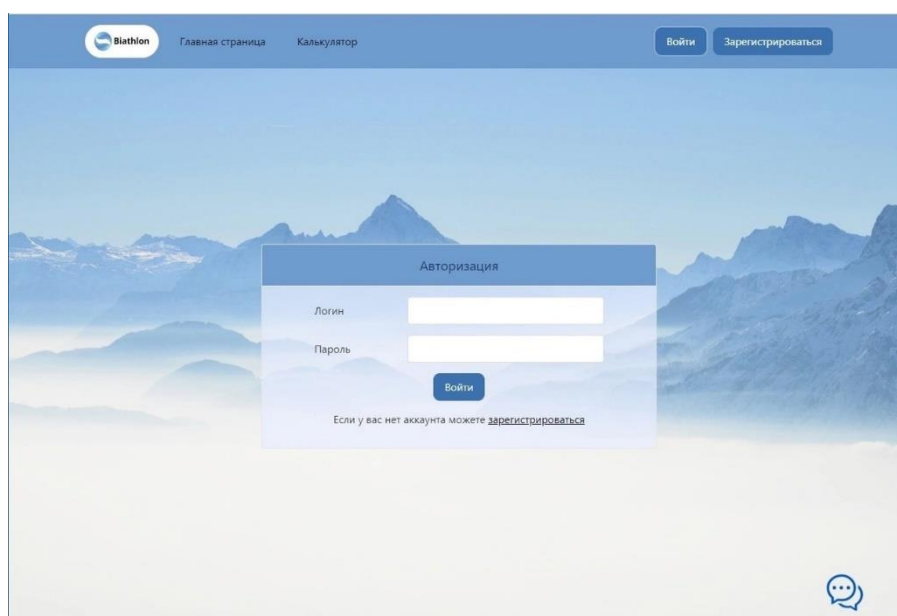


Рисунок 16 - Страница авторизации

Авторизованный пользователь, также гость сайта, может просматривать результаты интересующих соревнований. На главной странице сайта (Рисунок

17) есть список из сезонов и входящих в них соревнований с указанием дат их проведения.

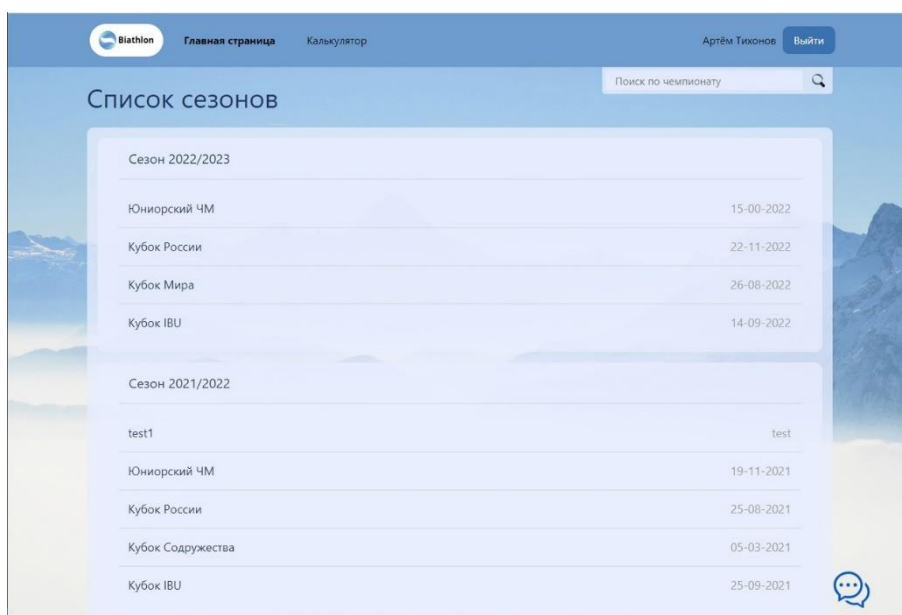


Рисунок 17 - Страница просмотра сезонов

Список сезонов достаточно объемный, и, если пользователь ищет результаты конкретного соревнования, он может воспользоваться поиском, который отсортирует список по введенным данным (Рисунок 18).

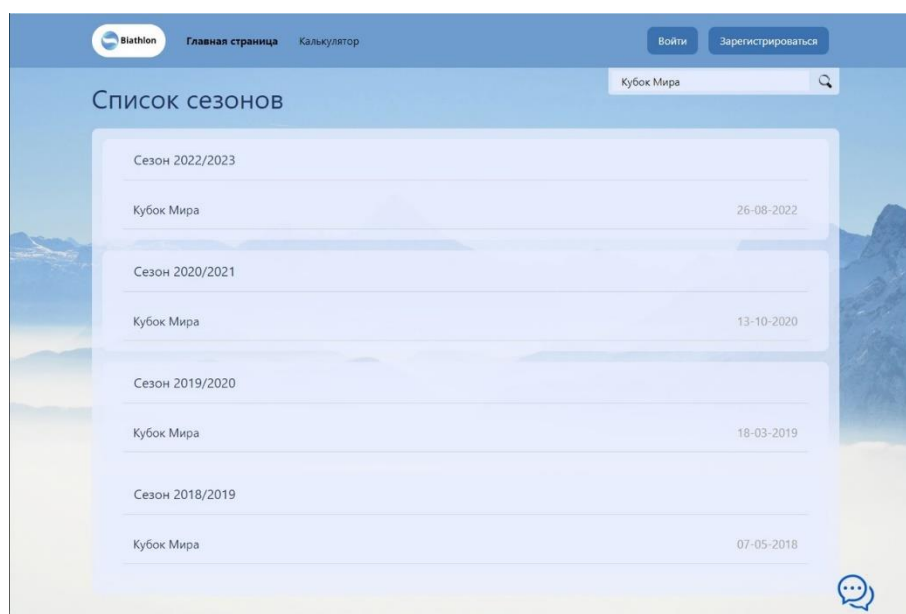


Рисунок 18 - Поиск по сайту

Если же данные были введены некорректно, сайт выведет сообщение об ошибке (Рисунок 19).

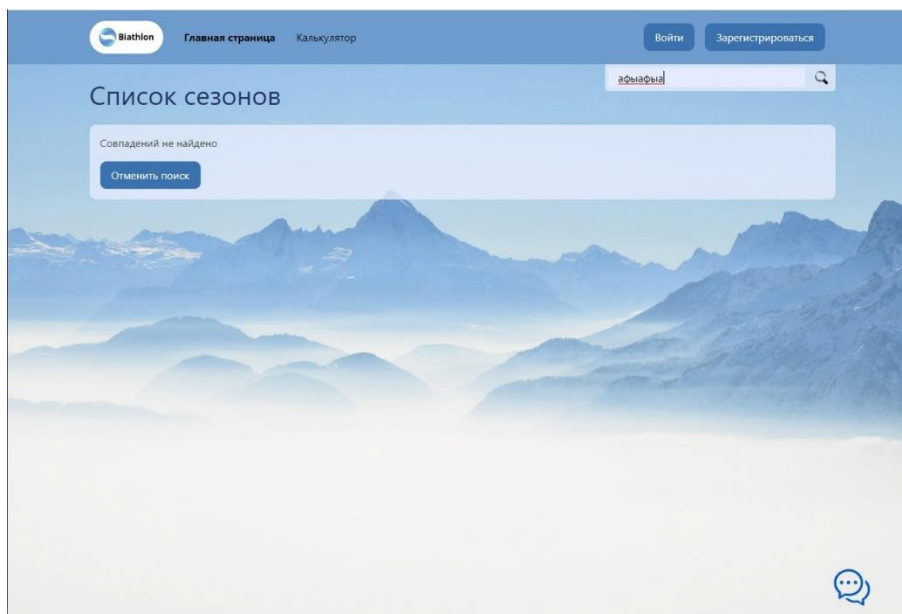


Рисунок 19 - Некорректно введенные данные в строке поиска

При просмотре результатов соревнований пользователь переходит на страницу интересующего соревнования (Рисунок 20). На ней представлены таблицы, разбитые по разным категориям, в которых содержится информация о:

- ФИО атлета;
- стране, которую представляет спортсмен;
- количестве золотых медалей;
- количестве серебряных медалей;
- количестве бронзовых медалей;
- общем количестве очков, полученных в результате соревнования.

Для простоты восприятия колонки для медалей выделены в подходящие цвета.

Юниорский ЧМ

Мужчины

#	Атлет	Страна	Золото	Серебро	Бронза	Очки
1	Виталий Юриевич Крючатов	Швеция	7	18	6	232
2	Герасим Сергеевич Собаков	Беларусь	2	6	5	285
3	Игорь Олегович Фатьянов	Швеция	0	16	0	11
4	Виталий Данилович Лябушев	Россия	5	10	9	235
5	Станислав Викторович Здориков	Россия	5	11	0	21
6	Степан Андреевич Нурабаев	Австрия	18	19	13	91

Женщины

#	Атлет	Страна	Золото	Серебро	Бронза	Очки
1	Елизавета Федоровна Аминьева	Норвегия	14	17	15	8
2	Надежда Яковлевна Негибина	Россия	11	12	16	183
3	Вера Романовна Михаличева	Россия	19	10	17	133
4	Софья Егоровна Кумкина	Швеция	5	18	9	10

Рисунок 20 - Страница просмотра результатов соревнования

Любой пользователь сайта может воспользоваться разделом “Калькулятор” (Рисунок 21), для подсчета общего количества медалей для каждой страны. Для этого пользователю необходимо выбрать интересующие его параметры:

- сезон;
- соревнование;
- категория;
- страна.

Результат будет выведен справа в виде таблицы.

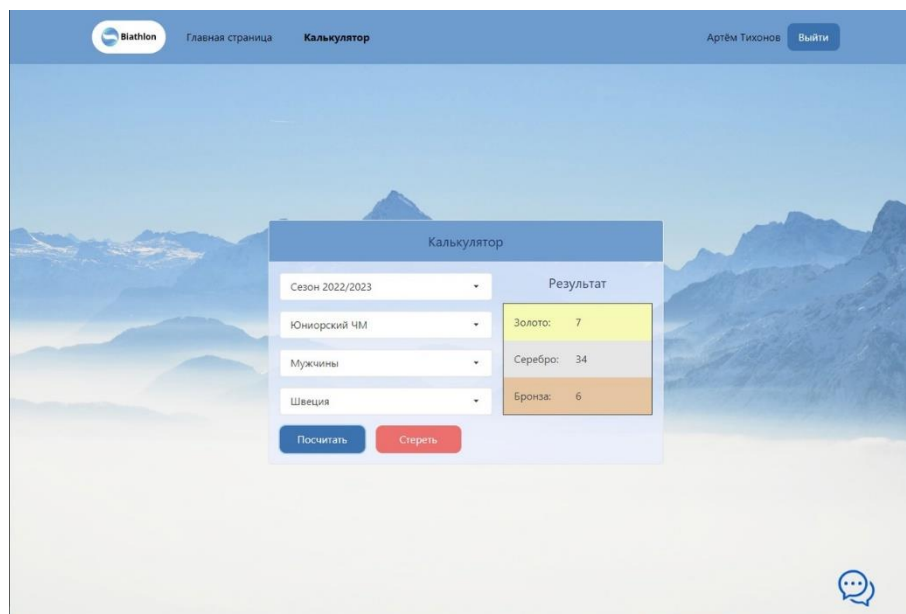


Рисунок 21 - Страница “Калькулятор”

Кроме того, авторизованный пользователь может ввести диалог с другими пользователями в разделе “Чат” (Рисунок 22).

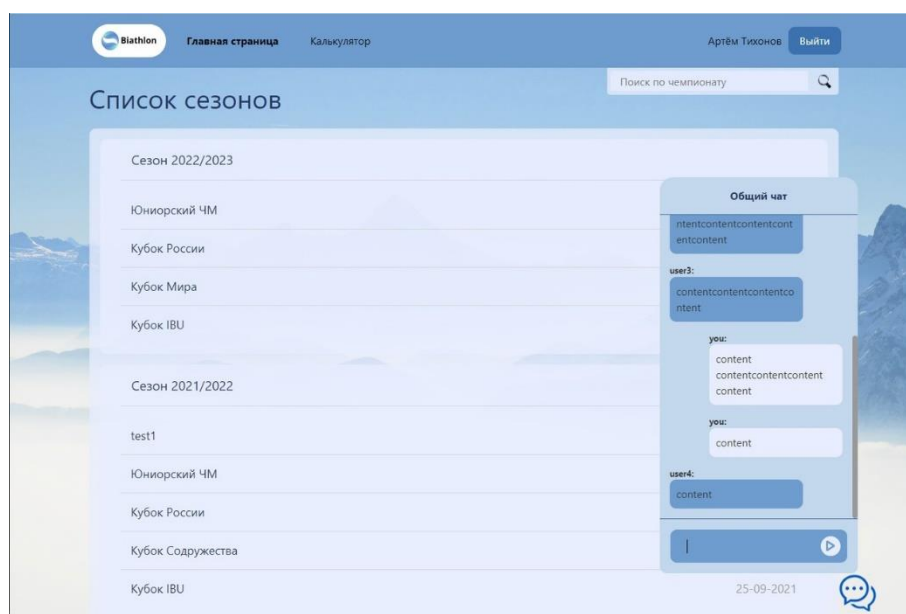


Рисунок 22 - Страница чата для авторизованного пользователя

Для неавторизованного пользователя данная функция будет закрыта, но он может видеть содержимое чата (Рисунок 23).

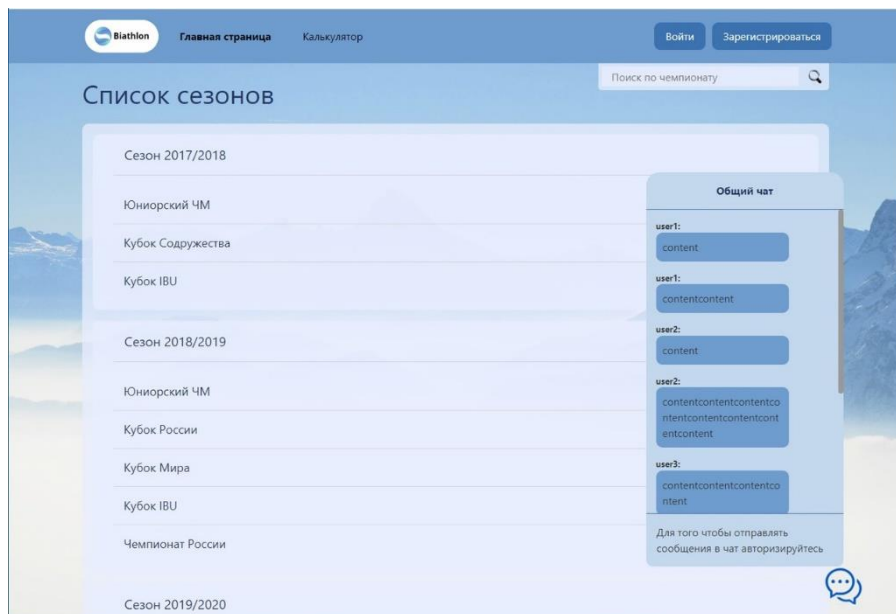


Рисунок 23 - Страница чата для неавторизованного пользователя

4 Тестирование

После реализации всех задач, был произведен запланированный набор тестов:

- дымовое тестирование;
- UI тесты;
- юзабилити тесты.

4.1 Дымовое тестирование

4.2 UI тесты

4.3 Юзабилити тесты

Заключение

В результате выполнения данного проекта, командой был разработан сервис для учета статистики по биатлону, которое предоставляет ряд преимуществ, связанных с автоматизацией, централизацией и визуализацией данных. Приложение позволяет автоматизировать процесс учета статистики по биатлону. Вместо ручного сбора и обработки данных, приложение обеспечивает хранение и управление информацией о сезонах, соревнованиях и их результатов в централизованной базе данных.

Для реализации данного проекта была выполнена следующая работа:

- спроектирована система с учетом требований, описанных в техническом задании;
- произведен анализ предметной области и существующих аналогов;
- спроектирован веб-сервис средствами языка UML;
- создан макет дизайна приложения в Figma;
- спроектирована база данных;
- разработан Back-end приложения;
- разработан Front-end приложения;
- создана связь между Front-end и Back-end частями приложения;
- собрана статистика с помощью сервиса “Яндекс Метрика”;
- произведено тестирование веб-приложения.

Список используемых источников

1. OLYMPTEKA // Сайт olympdeka.ru URL: <https://olympdeka.ru/sport/biathlon/rating/440.html> (дата обращения: 8.03.2023).
2. RUSBIATHLON // Сайт rusbiathlon.ru URL: <https://rusbiathlon.ru/statistic/biathlon/> (дата обращения: 9.03.2023).
3. Джеймс Рамбо, М. Блаха UML 2.0. Объектно-ориентированное моделирование и разработка. - 2-е изд. - СПб: Питер, 2007. - 545 с.
4. Documenting your API // Django REST framework URL: <https://www.django-rest-framework.org/topics/documenting-your-api/> (дата обращения: 8.05.2023).
5. Django REST framework // django-rest-framework-avsd URL: <https://django-rest-framework-avsd.readthedocs.io/en/stable/> (дата обращения: 8.05.2023).
6. Docs // Incline URL: <https://v2-incline-io.vercel.app/docs/introduction/getting-started> (дата обращения: 3.05.2023).
7. Introduction // Vue.js URL: <https://v2.vuejs.org/v2/guide/> (дата обращения: 1.05.2023).