**Содержание**

[Введение 2](#_Toc183541817)

[1 Анализ 3](#_Toc183541818)

[1.1 Особенности организации соревнований по боксу 3](#_Toc183541819)

[1.1.1 Существующие решения 3](#_Toc183541820)

[1.1.2 Проблемы, которые возникают при организации вручную 4](#_Toc183541821)

[1.1.3 Постановка задачи 4](#_Toc183541822)

[1.1.4 Вывод по обзору 4](#_Toc183541823)

[1.2 Обзор существующих программных средств 5](#_Toc183541824)

[1.2.1 Система "Спорт Бокс" 5](#_Toc183541825)

[1.2.2 "СКС: Спортивные соревнования" 6](#_Toc183541826)

[1.2.3 "ЦОРС: Центр организации соревнований" 8](#_Toc183541827)

[1.2.4 Итоги сравнения существующих платформ 9](#_Toc183541828)

[1.3 Процессы As-Is и To-Be 9](#_Toc183541829)

[1.3.1 AS-IS 9](#_Toc183541830)

[1.3.2 TO-BE 11](#_Toc183541831)

[1.4 Описание вариантов использования 13](#_Toc183541832)

[1.5 Выработка требований и постановка задачи 14](#_Toc183541833)

[2 Проектирование 17](#_Toc183541834)

[2.1 Выбор и обоснование средств проектирования и реализации 17](#_Toc183541835)

[2.1.1 Средства проектирования 17](#_Toc183541836)

[2.1.2 Средства реализации 19](#_Toc183541837)

[2.2 Проектирование архитектуры приложения 20](#_Toc183541838)

[2.2.1 Описание компонентов архитектуры 20](#_Toc183541839)

[2.2.2 Схема архитектуры приложения и её описание 22](#_Toc183541840)

[2.3 Проектирование хранилища данных 23](#_Toc183541841)

[2.4 Проектирование пользовательского интерфейса 27](#_Toc183541842)

[2.4.1 Верхнеуровневое определение экранов 27](#_Toc183541843)

[2.4.2 Назначение экранов 28](#_Toc183541844)

[2.4.3 Отрисовка и описание макетов ваших экранов 32](#_Toc183541845)

[2.4.4 Карта экранов и состояний 35](#_Toc183541846)

[3 Разработка 39](#_Toc183541847)

[3.1 Итоговые результаты работы 39](#_Toc183541848)

[3.1.1 Скриншоты готовых экранов: 39](#_Toc183541849)

[3.1.2 Фрагменты кода: 41](#_Toc183541850)

[3.2 Вывод 46](#_Toc183541851)

# Введение

Бокс — один из самых древних и популярных видов единоборств, история которого насчитывает несколько тысячелетий. Со временем он эволюционировал в структурированный и строго регламентированный вид спорта, требующий высокого уровня организации как на локальных, так и на международных соревнованиях. Турниры по боксу характеризуются большим количеством участников и строгими правилами, что делает их организацию сложным и трудоемким процессом.

Соревнования по боксу включают множество этапов: регистрацию участников, процесс взвешивания и распределения по весовым категориям, жеребьевку, составление турнирных сеток и финальные бои. Каждый этап требует тщательного контроля и точного выполнения, так как ошибки в организации могут повлиять на результаты турнира и создать проблемы для участников и организаторов.

Сегодня ручная организация турниров становится все более сложной задачей из-за увеличения числа участников и необходимости точного учета множества данных. Информационные технологии могут значительно облегчить эти процессы, сделав их более эффективными и прозрачными. Разработка автоматизированной системы для проведения соревнований по боксу способна решить многие проблемы, возникающие при организации турниров, и значительно упростить работу организаторов.

Целью данного проекта является упрощение и оптимизация процесса организации соревнований по боксу за счет создания автоматизированной системы. Система должна минимизировать человеческие ошибки и повысить эффективность управления турнирами.

Для достижения цели были поставлены следующие задачи:

1. Исследовать существующие системы для управления спортивными соревнованиями.
2. Проанализировать требования, предъявляемые к системе для боксерских турниров.
3. Спроектировать и реализовать базовую функциональность системы.
4. Провести тестирование системы на примере гипотетического турнира.

# 1 Анализ

## 1.1 Особенности организации соревнований по боксу

Организация соревнований по боксу включает в себя множество этапов, каждый из которых требует тщательного планирования и выполнения:

**Регистрация участников**. На этапе регистрации собираются данные о спортсменах, включая их имя, возраст, весовую категорию и предыдущие спортивные достижения. Этот процесс важен для дальнейшего распределения участников по категориям и формирования турнирной сетки.

**Процесс взвешивания и распределения по весовым категориям**. В боксе существует несколько весовых категорий, в рамках которых спортсмены соревнуются между собой. Перед началом соревнований каждый участник должен пройти взвешивание, по результатам которого он будет распределен в соответствующую категорию. Неправильное распределение по категориям может привести к несправедливым боям и нарушению правил турнира.

**Жеребьевка и составление сетки боев**. Жеребьевка проводится для распределения спортсменов в турнирную сетку и определения последовательности боев. Это важный этап, так как от него зависит дальнейший ход соревнований. Составление расписания должно учитывать временные интервалы между боями и распределение судей по площадкам.

**Судейство и критерии оценки боя**. В боксе существуют четкие правила, по которым судьи оценивают действия боксеров в ринге. Основные критерии включают количество нанесенных ударов, их точность и сила, а также защитные действия бойца. Важно, чтобы система учета результатов обеспечивала быстрый и точный ввод данных от судей и предоставляла возможность их дальнейшего анализа.

### 1.1.1 Существующие решения

Существует несколько решений, которые уже применяются для организации спортивных мероприятий, однако они имеют ограничения, которые не всегда позволяют эффективно использовать их в боксе:

**Системы для управления спортивными мероприятиями общего назначения**. Такие системы, как Tournament Software, позволяют автоматизировать основные этапы организации турниров: регистрацию участников, составление турнирной сетки и учет результатов. Однако они не всегда учитывают специфику бокса, например, динамическое распределение по весовым категориям, требующее точного учета веса участников в день взвешивания.

**Приложения для регистрации и жеребьевки**. Программы для регистрации участников и автоматической жеребьевки также существуют, но они часто имеют ограниченный функционал. Например, такие системы могут не предусматривать возможность интеграции с судейскими протоколами и не обеспечивают должного уровня прозрачности для участников и зрителей.

**Ручные методы организации**. Несмотря на доступность современных технологий, многие турниры по боксу продолжают организовываться с использованием Excel или других подобных инструментов для ведения учета участников и составления расписания боев. Такой подход приводит к увеличению времени на подготовку турнира и повышает вероятность ошибок.

### 1.1.2 Проблемы, которые возникают при организации вручную

Основные проблемы, с которыми сталкиваются организаторы боксерских турниров при отсутствии автоматизированных систем, включают:

**Трудности в распределении участников по категориям**. Вручную распределение по весовым категориям занимает много времени и может сопровождаться ошибками, что нарушает спортивный принцип соревнований.

**Ошибки при жеребьевке и составлении расписания.** Вручную составление турнирной сетки может привести к неверному распределению боев и неправильному составлению расписания, что затрудняет координацию работы судей и участников.

**Проблемы с учётом результатов**. Введение результатов вручную увеличивает вероятность ошибок, особенно при больших масштабах соревнований. Также это требует значительных временных затрат, что замедляет процесс подведения итогов.

**Затруднённая обратная связь для участников и зрителей**. При отсутствии автоматизированной системы результаты турнира и расписание могут обновляться с задержками, что создает неудобства для всех участников процесса.

### 1.1.3 Постановка задачи

Основываясь на выявленных проблемах, автоматизированная система должна решить следующие задачи:

**Автоматизация регистрации и распределения по категориям**. Необходимо предусмотреть систему регистрации участников с возможностью ввода данных и автоматического распределения по весовым категориям на основании взвешивания.

**Автоматизация жеребьевки и составления расписания**. Система должна автоматически проводить жеребьевку участников и составлять расписание боев с учетом временных интервалов между поединками и работы судей.

**Учет результатов и прозрачность для всех участников**. Введение результатов боев должно происходить в реальном времени с минимальными временными затратами. Также система должна обеспечить доступ к результатам для участников, судей и зрителей через веб-интерфейс.

**Интеграция с судейскими протоколами**. Система должна позволять вводить и обрабатывать данные от судей для формирования итоговых результатов и дальнейшего анализа.

### 1.1.4 Вывод по обзору

На основании проведенного анализа, можно сделать вывод, что автоматизация процессов организации боксерских турниров является актуальной задачей. Современные системы управления соревнованиями, несмотря на свою эффективность в ряде других видов спорта, не всегда способны удовлетворить специфические потребности бокса. Разработка специализированного программного обеспечения позволит повысить качество управления турнирами, минимизировать ошибки и значительно сократить время на подготовку и проведение мероприятий.

Внедрение автоматизированной системы для организации боксерских соревнований позволит улучшить взаимодействие между участниками, судьями и организаторами, а также сделает процесс проведения турниров более прозрачным и удобным для зрителей.

## 1.2 Обзор существующих программных средств

В России существует несколько программных решений, которые ориентированы на организацию спортивных мероприятий и могут применяться для проведения боксерских турниров. В этом разделе рассмотрим их основные функции, особенности и оценим их актуальность для реализации проекта автоматизации боксерских соревнований.

### 1.2.1 Система "Спорт Бокс"

**Спорт Бокс** — это специализированное программное обеспечение, разработанное для организации и проведения боксерских соревнований. Программа включает все необходимые модули для ведения турнирных таблиц, учета участников и результатов боев. На рисунке 1 представлена главная страница сайта «Sportbox.ru»

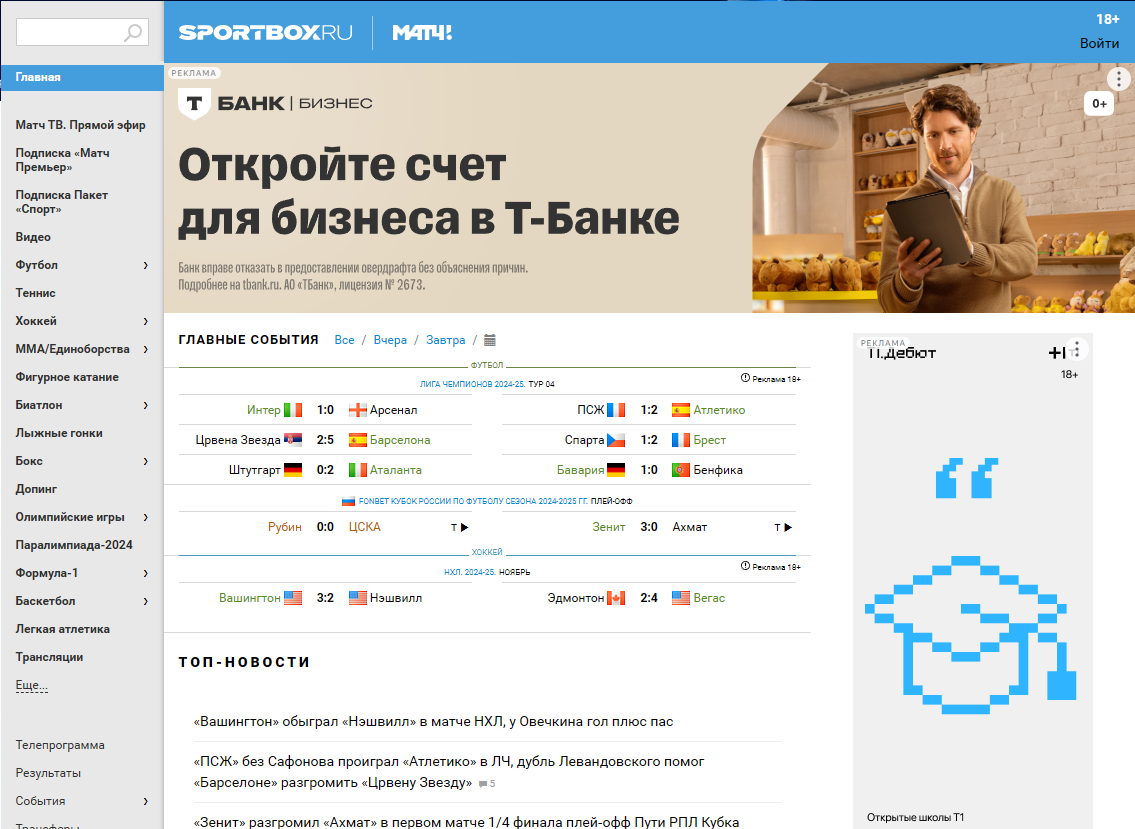


Рисунок 1 – Главная страница Sportbox.ru

* **Регистрация участников**. Система позволяет регистрировать участников как вручную, так и автоматически с использованием данных из предыдущих соревнований.
* **Учет весовых категорий**. Программа поддерживает динамическое распределение участников по весовым категориям на основании данных о взвешивании.
* **Создание турнирных сеток**. Система автоматизирует процесс жеребьевки и формирования боев, учитывая специфику бокса.

**Учет результатов и судейских оценок**. Программа поддерживает ввод судейских оценок в реальном времени, что упрощает ведение и подведение итогов.

**Недостатки**:

* Программа ориентирована исключительно на бокс, что ограничивает возможности ее использования в других видах спорта.
* Интерфейс программы может показаться устаревшим и не всегда интуитивно понятным для новых пользователей.

Таблица с отмеченными главными требованиями представлена в таблице 1.

Таблица 1 – Функции сайта Sportbox.ru

|  |  |
| --- | --- |
| Функция | Наличие в "Спорт Бокс" |
| Регистрация участников | Да |
| Автоматизация жеребьевки | Да |
| Учет результатов | Да |
| Интеграция с судейскими протоколами | Да |
| Поддержка весовых категорий | Да |

### 1.2.2 "СКС: Спортивные соревнования"

**СКС: Спортивные соревнования** — это программное обеспечение для организации и проведения различных спортивных мероприятий, разработанное в России. Система поддерживает множество видов спорта, включая единоборства, такие как бокс и кикбоксинг. Главная страница представлена на рисунке 2.

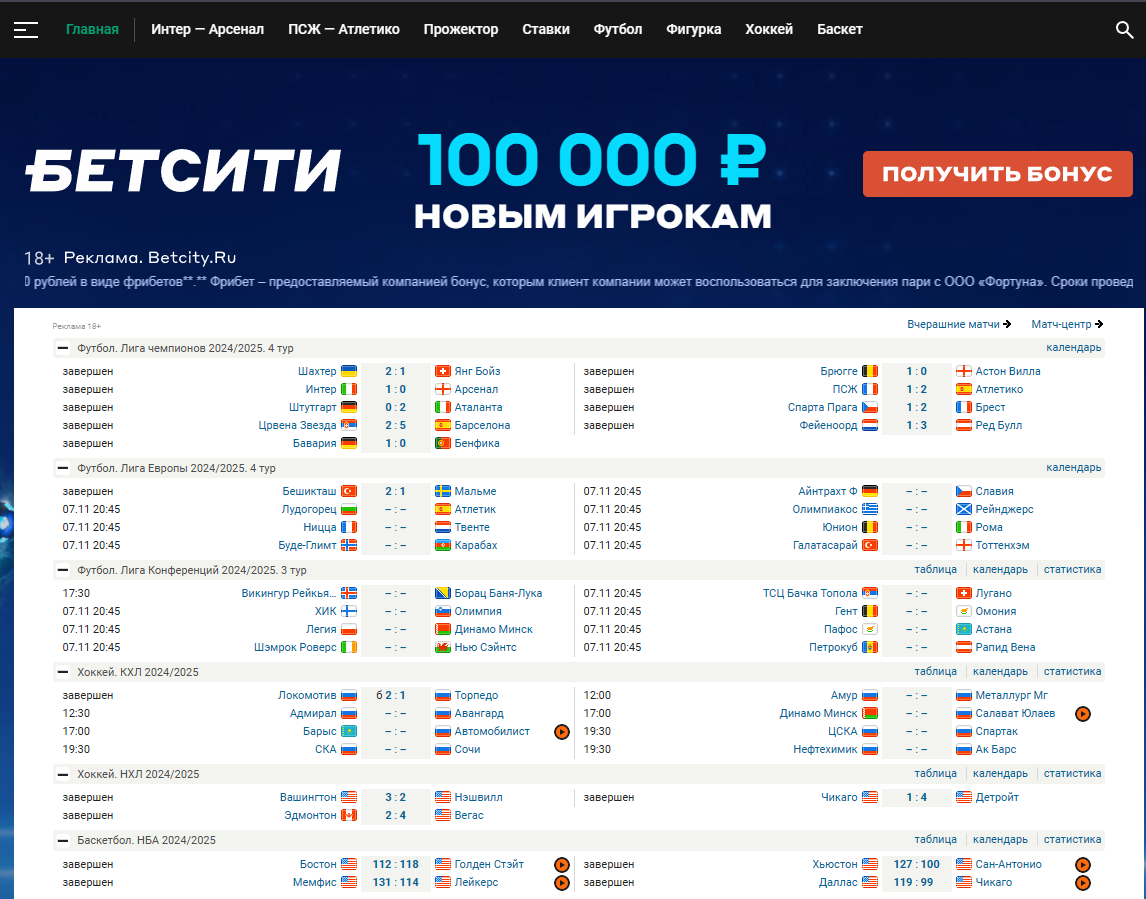


Рисунок 2 – Главная страница СКС: Спортивные соревнования

**СКС: Спортивные соревнования** — это программное обеспечение для организации и проведения различных спортивных мероприятий, разработанное в России. Система поддерживает множество видов спорта, включая единоборства, такие как бокс и кикбоксинг.

* **Регистрация участников и взвешивание**. Система позволяет регистрировать участников онлайн и автоматически распределять их по весовым категориям на основе данных о взвешивании.
* **Создание турнирных сеток**. Программа автоматизирует жеребьевку и составление расписания, что позволяет значительно ускорить процесс организации.
* **Учет результатов**. Система предоставляет возможность в реальном времени вносить результаты боев и публиковать их на веб-портале для участников и зрителей.
* **Модуль судейства**. Программа поддерживает ввод судейских протоколов с возможностью дальнейшего анализа результатов.

**Недостатки**:

* Ограниченная поддержка специализированных боксерских функций, таких как интеграция с весовыми категориями и автоматизация взвешивания.
* Высокая стоимость для небольших соревнований.

Таблица 2 – Функции сайта "СКС: Спортивные соревнования".

|  |  |
| --- | --- |
| Функция | Наличие в "СКС" |
| Регистрация участников | Да |
| Автоматизация жеребьевки | Да |
| Учет результатов | Да |
| Интеграция с судейскими протоколами | Да |
| Поддержка весовых категорий | Да |

### 1.2.3 "ЦОРС: Центр организации соревнований"

**ЦОРС** — это комплексная система для организации спортивных соревнований, которая активно используется на региональном уровне для проведения различных спортивных мероприятий, включая бокс. Программа разработана с целью поддержки соревнований в командных и индивидуальных видах спорта.

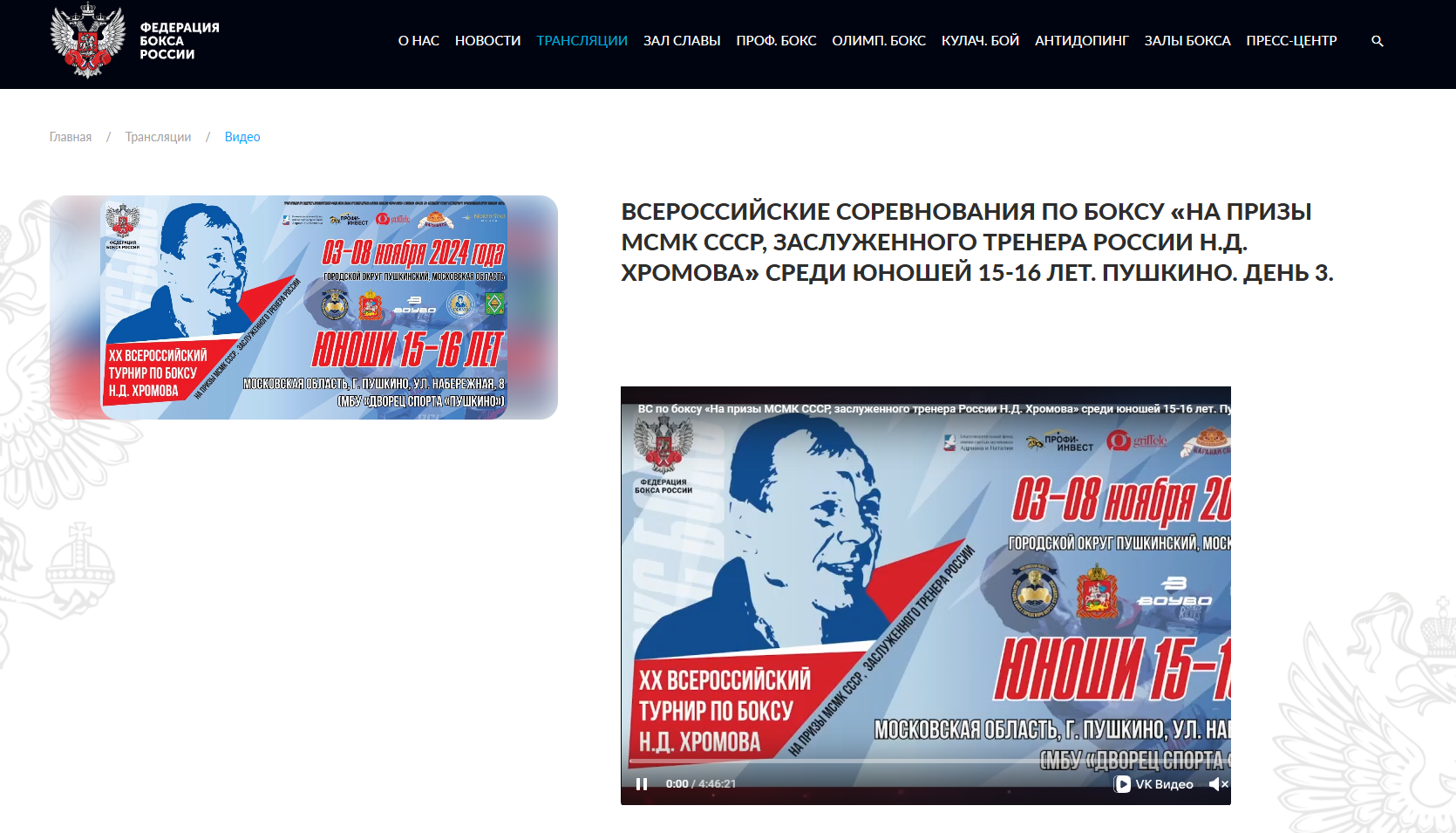


Рисунок 3 – Главная странциа "ЦОРС: Центр организации соревнований"

* **Управление турнирами**. ЦОРС предлагает широкую функциональность для управления соревнованиями, включая регистрацию участников, автоматизацию жеребьевки и составление расписания.
* **Модуль для судейства**. Программа позволяет вносить результаты боев и вводить судейские оценки. Эти данные сразу синхронизируются с общей системой, что исключает возможность ошибок.
* **Формирование отчётных документов**. Система автоматически генерирует отчеты о ходе турнира, что облегчает работу организаторов.

**Недостатки**:

* Программа больше ориентирована на командные виды спорта и может потребовать доработки для боксерских турниров.
* Интерфейс и функциональность могут быть перегружены для проведения небольших соревнований.

Таблица 3 – Функции сайта "ЦОРС: Центр организации соревнований"

|  |  |
| --- | --- |
| Функция | Наличие в "ЦОРС" |
| Регистрация участников | Да |
| Автоматизация жеребьевки | Да |
| Учет результатов | Да |
| Интеграция с судейскими протоколами | Да |
| Поддержка весовых категорий | Ограничено |

### 1.2.4 Итоги сравнения существующих платформ

На российском рынке присутствуют несколько решений для организации спортивных мероприятий, которые могут быть использованы для проведения боксерских соревнований. Однако каждое из этих решений имеет свои ограничения. Например, специализированная система "Спорт Бокс" полностью ориентирована на бокс, но имеет ограничения в интерфейсе и требует дополнительного времени на обучение. Универсальные решения, такие как "СКС" и "ЦОРС", предоставляют более широкую функциональность, но могут быть излишне сложны и дорогие для небольших соревнований.

Разработка специализированной автоматизированной системы для проведения боксерских турниров, которая будет учитывать особенности процесса регистрации, взвешивания, жеребьевки и судейства, позволит устранить существующие недостатки и повысить эффективность организации соревнований.

## 1.3 Процессы As-Is и To-Be

### 1.3.1 AS-IS

AS IS представляет текущие функциональные процессы сайта бокса, который включает в себя базовые операции для организации боксерских соревнований. На данном этапе мы фиксируем проблемы и недостатки, которые мешают эффективному функционированию сайта и требуют улучшения.

Основные функции в AS IS:

* Регистрация участников — процесс регистрации участников на соревнования выполняется вручную через форму на сайте, информация отправляется в виде заявки и обрабатывается администратором.
* Управление расписанием соревнований — создание и публикация расписания на сайте происходит вручную, без автоматического уведомления участников о предстоящих боях.
* Сбор и хранение данных о боях — результаты боев заносятся в базу данных после их проведения, обновление статистики осуществляется вручную.
* Поддержка рейтингов и статистики — рейтинг участников обновляется после обработки результатов боев вручную. Отсутствует автоматическое обновление и подсчет очков на основе правил ранжирования.

Для примера мы рассмотрим процесс регистрации участников.

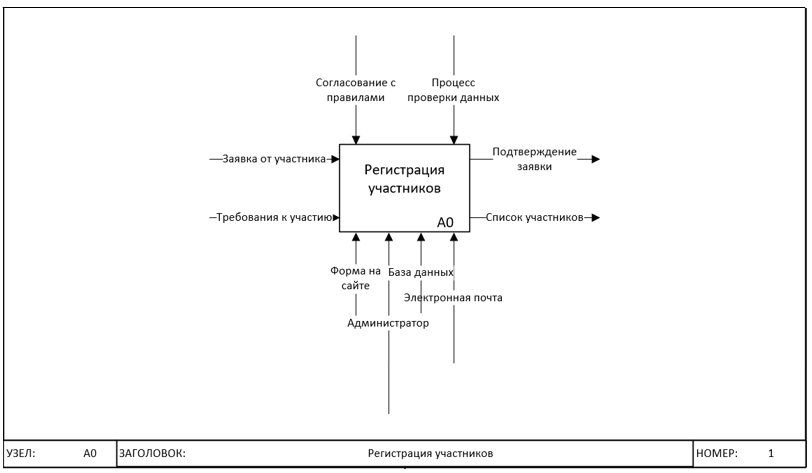


Рисунок 4 – Контекстная диаграмма AS-IS

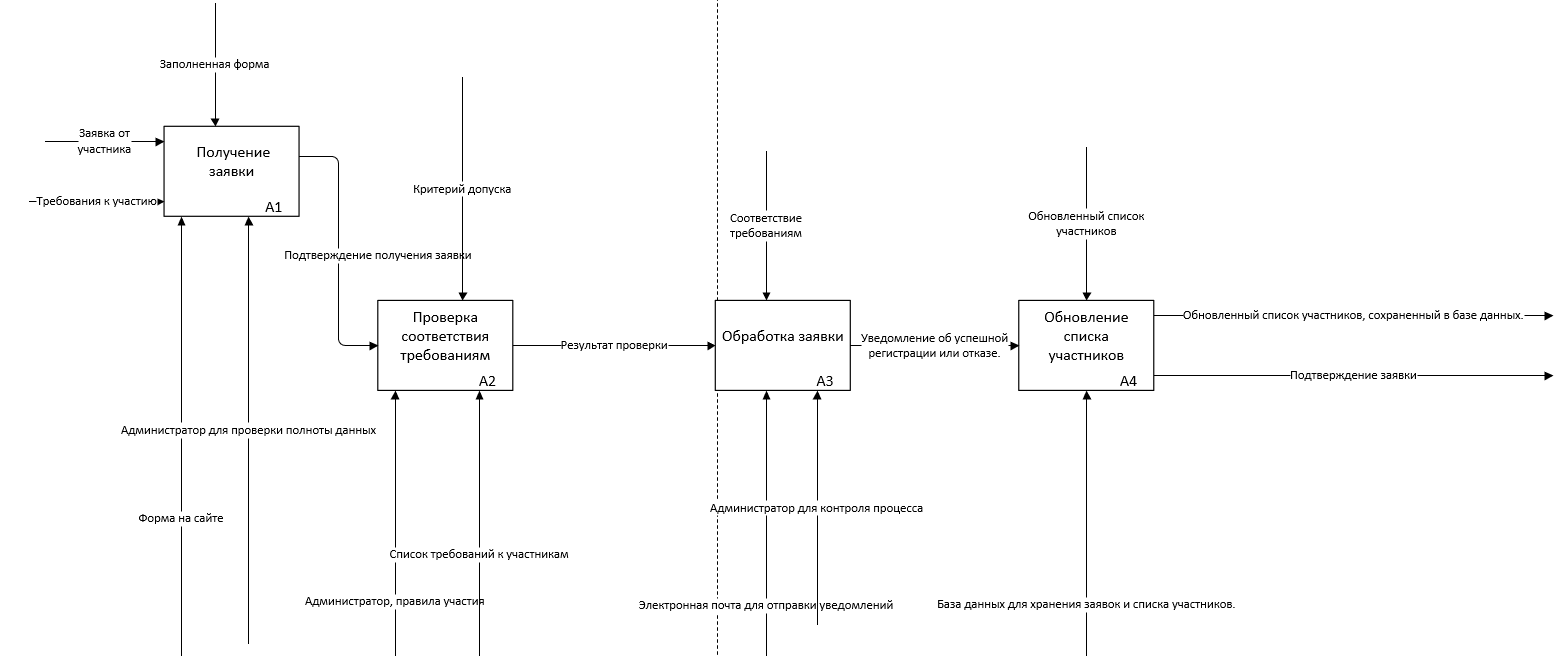


Рисунок 5 – Диаграмма декомпозиции AS-IS

### 1.3.2 TO-BE

TO BE представляет целевую функциональную модель системы, где процессы автоматизированы для улучшения взаимодействия с пользователями и повышения эффективности. Ниже описаны улучшенные функции, включенные в целевую структуру.

Основные функции в TO BE:

* Автоматическая регистрация участников — процесс подачи заявки полностью автоматизирован, участники могут регистрироваться и получать уведомления о статусе заявки сразу через сайт.
* Автоматизация расписания и уведомлений — расписание соревнований формируется и публикуется автоматически; участники и организаторы получают уведомления о грядущих боях.
* Автоматическое обновление данных о боях и расчет рейтингов — после каждого боя данные автоматически обновляются в базе, статистика и рейтинги пересчитываются по заранее настроенным алгоритмам.

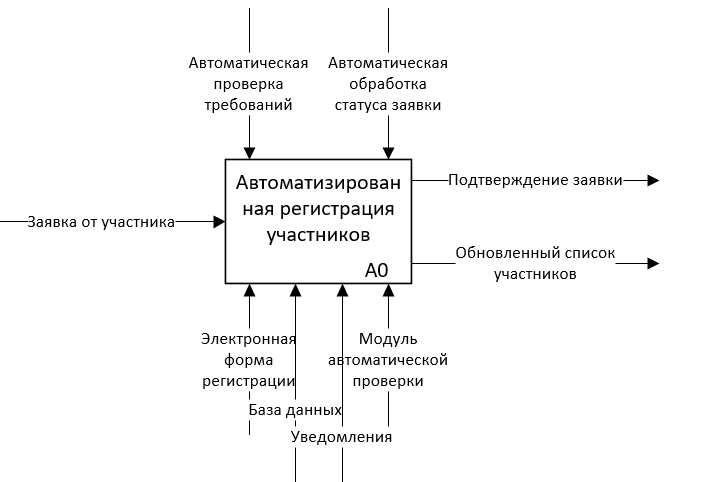


Рисунок 6 – Контекстная диаграмма TO-BE

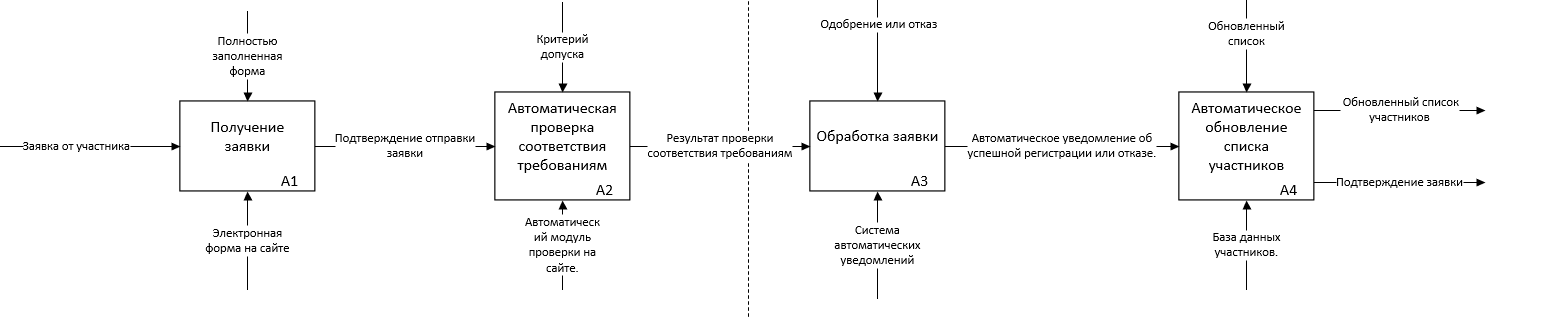


Рисунок 7 – Диаграмма декомпозиции TO-BE

В ходе анализа текущего процесса регистрации участников на боксерском сайте (**AS IS**) и разработки его целевого состояния (**TO BE**) была проделана работа по выявлению основных проблем и разработке более эффективной схемы работы.

1. **Текущий процесс (AS IS)**:

* Весь процесс регистрации в текущем виде ручной и требует значительного участия администратора.
* Администратор вручную проверяет каждую заявку, что занимает много времени и повышает вероятность ошибок при обработке.
* Уведомления участникам отправляются вручную, что также замедляет процесс и требует дополнительных усилий.
* Недостаточная автоматизация снижает общую эффективность системы и требует от администратора значительных временных и трудовых затрат.

1. **Целевой процесс (TO BE)**:
   * Все ключевые этапы регистрации автоматизированы, начиная с получения заявки и заканчивая ее проверкой и уведомлением участника.
   * Автоматическая проверка требований позволяет сократить участие администратора, ускоряя процесс и уменьшая вероятность ошибок.
   * Уведомления участникам отправляются автоматически, что улучшает пользовательский опыт и позволяет оперативно информировать участников о статусе их заявки.
   * База данных обновляется в режиме реального времени, что позволяет администратору получать актуальные данные без необходимости вручной обработки.

Переход к целевому состоянию позволит достичь следующих улучшений:

* **Снижение нагрузки на администратора**: благодаря автоматизации проверки заявок и отправки уведомлений.
* **Скорость и эффективность**: автоматический процесс регистрации позволяет участникам получать обратную связь значительно быстрее.
* **Повышение точности**: автоматическая проверка минимизирует ошибки и повышает достоверность данных.
* **Прозрачность и доступность**: обновление базы данных в реальном времени обеспечивает администратору и организаторам соревнований доступ к актуальной информации.

Автоматизация процесса регистрации участников на боксерском сайте приведет к значительному повышению эффективности и удобства системы. Внедрение предложенной модели TO BE позволит сократить время обработки заявок, повысить точность данных и снизить количество ручной работы, необходимой администратору. Это улучшение сделает процесс регистрации более прозрачным и удобным для пользователей, а также повысит общий уровень обслуживания и удовлетворенность участников.

## 1.4 Описание вариантов использования

Диаграмма вариантов использования (Use Case Diagram) отображает основные функциональные требования к системе с точки зрения пользователя. Эта диаграмма поможет определить, какие задачи могут выполнять различные типы пользователей на боксерском сайте и как система будет взаимодействовать с ними.

Основные участники (Actors):

Участник — пользователь, который заходит на сайт, чтобы зарегистрироваться на соревнование.

Администратор — пользователь, который управляет заявками, проверяет их соответствие требованиям и обновляет информацию о соревнованиях.

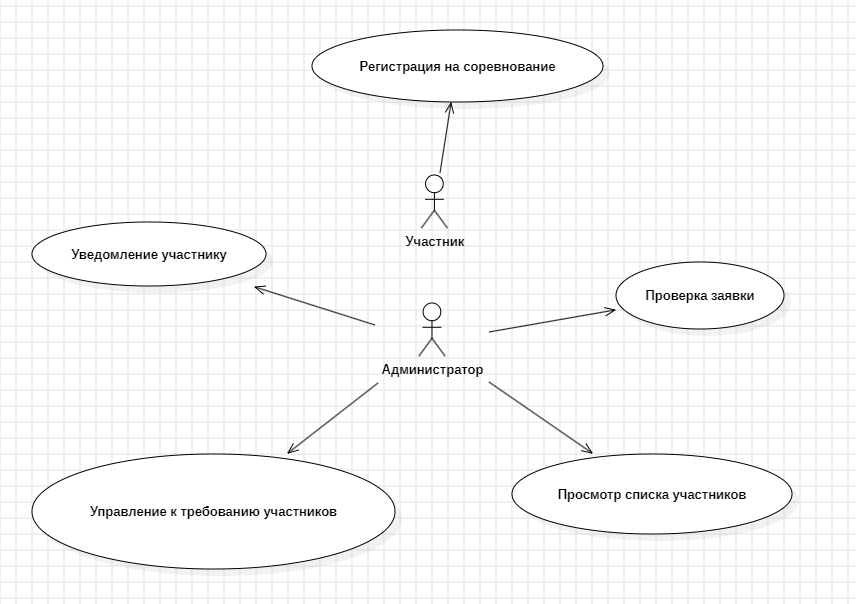


Рисунок 8 – Диаграмма прецендентов

Варианты использования (Use Cases) и их описание

1. **Регистрация на соревнование**
   * **Актёр**: Участник
   * **Описание**: Участник заполняет форму на сайте для регистрации на соревнование.
   * **Предусловие**: Участник должен иметь доступ к интернету и быть на сайте.
   * **Результат**: Заявка отправляется в систему для проверки.
2. **Проверка заявки на соответствие требованиям**
   * **Актёр**: Администратор
   * **Описание**: Система автоматически проверяет данные, указанные в заявке, на соответствие требованиям соревнования (например, возраст, категория).
   * **Предусловие**: Заявка должна быть получена от участника.
   * **Результат**: Заявка либо одобряется, либо отклоняется.
3. **Отправка уведомления участнику**
   * **Актёр**: Администратор
   * **Описание**: Система автоматически отправляет уведомление участнику о результате проверки заявки (одобрено или отклонено).
   * **Предусловие**: Завершение проверки заявки.
   * **Результат**: Участник получает уведомление о статусе своей заявки.
4. **Обновление списка участников**
   * **Актёр**: Администратор
   * **Описание**: Система автоматически добавляет одобренные заявки в список участников соревнования.
   * **Предусловие**: Заявка должна быть одобрена.
   * **Результат**: Список участников обновлен.
5. **Просмотр списка участников**
   * **Актёр**: Администратор
   * **Описание**: Администратор имеет доступ к списку участников, чтобы просмотреть и управлять заявками.
   * **Предусловие**: у администратора должен быть доступ к системе.
   * **Результат**: Администратор видит список текущих участников и может управлять ими.
6. **Управление требованиями к участию**
   * **Актёр**: Администратор
   * **Описание**: Администратор может обновлять требования к участию, такие как возрастные ограничения и категории.
   * **Предусловие**: у администратора есть права на изменение параметров системы.
   * **Результат**: Требования к участию обновлены.

## 1.5 Выработка требований и постановка задачи

Анализ предметной области показал, что в современном управлении процессами спортивных соревнований, особенно боксерских, возрастает необходимость в автоматизации основных функций системы. Традиционные методы регистрации участников, требующие ручной обработки заявок и проверки данных, сталкиваются с проблемами в условиях увеличивающихся объемов информации и высоких требований к точности и оперативности. Ручная обработка заявок увеличивает время проверки, повышает вероятность ошибок и требует значительных временных затрат от администратора.

Существующие решения для регистрации участников, как правило, ограничены в функционале и не всегда позволяют автоматизировать проверку заявок и быстро уведомлять участников о статусе их регистрации. В ходе анализа также была выявлена потребность в надежной системе, которая обеспечит удобный доступ к информации, защиту данных участников и автоматизацию всего процесса регистрации.

Современные технологии, включая автоматизированные системы проверки данных и управления уведомлениями, создают основу для разработки новой системы регистрации, ориентированной на повышение эффективности и точности обработки данных.

На основе проведенного анализа предметной области и выявленных особенностей управления заявками на соревнования, сформулирована следующая задача: разработать автоматизированную систему для регистрации участников на боксерские соревнования. В отличие от традиционных методов, требующих ручной обработки заявок, разрабатываемая система будет направлена на автоматическую проверку данных участников, управление заявками и уведомление пользователей, что позволит ускорить процесс регистрации и снизить нагрузку на администратора.

При решении поставленной задачи необходимо учесть следующие аспекты:

1. Система должна поддерживать автоматическую проверку данных участников на соответствие установленным критериям, что требует разработки алгоритмов, способных эффективно обрабатывать входящие данные.
2. Автоматизация уведомлений участников о статусе их заявки должна быть реализована для оперативного информирования пользователей и повышения качества обслуживания.
3. Система должна обеспечивать защиту данных участников, с соблюдением всех норм по хранению и доступу к личной информации.
4. Интерфейс системы должен быть интуитивно понятным как для участников, так и для администратора, чтобы сократить время на обработку заявок и обеспечить удобство использования.
5. Система должна быть масштабируемой и поддерживать возможность добавления новых функций, например, расширение функционала для управления различными этапами соревнований.

Средством достижения поставленной задачи является создание программного решения, способного автоматически обрабатывать заявки участников, проверять их соответствие критериям, уведомлять пользователей и обновлять список участников в режиме реального времени, что обеспечит устойчивую и эффективную работу системы в условиях увеличения количества заявок и расширения функционала сайта для боксерских соревнований.

# 2 Проектирование

## 2.1 Выбор и обоснование средств проектирования и реализации

### 2.1.1 Средства проектирования

1) **StarUML**

Описание: StarUML — это инструмент для создания UML-диаграмм (Unified Modeling Language), поддерживающий такие диаграммы, как диаграммы классов, последовательности и состояний. Он позволяет детализировать архитектуру системы и отображать взаимосвязи между её компонентами.

Обоснование выбора: StarUML будет использоваться для разработки структуры системы, поскольку поддерживает стандарты UML 2.0 и позволяет визуализировать сложные процессы. UML-диаграммы помогут наглядно представить взаимодействие компонентов системы и упростят процесс документирования. Интерфейс данного приложения представлен на рисунке 9.

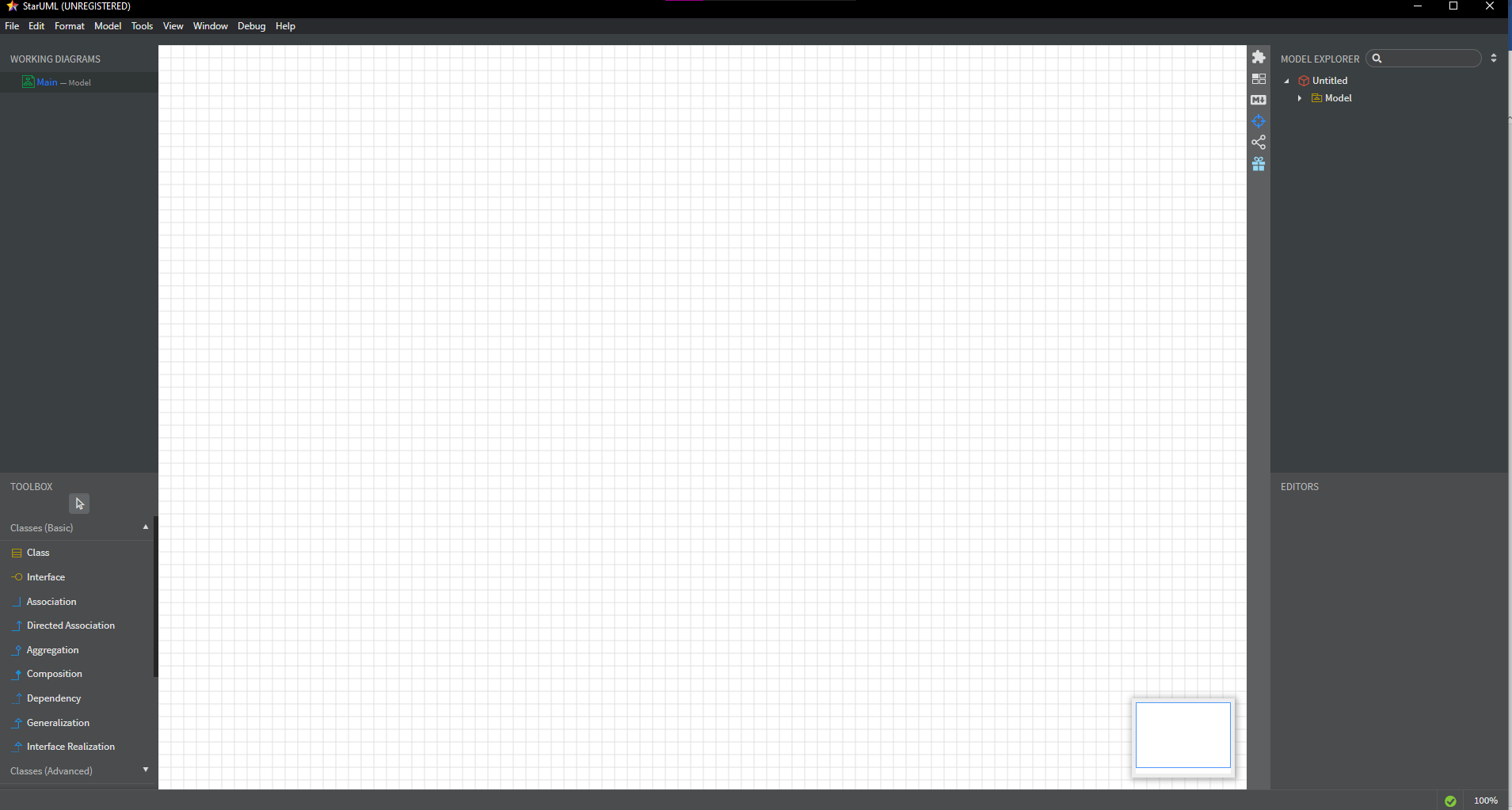


Рисунок 9 – Интерфейс StarUML.

2) **Microsoft Visio**

Описание: Microsoft Visio — это графический редактор для создания разнообразных схем и диаграмм. Visio предоставляет обширный набор шаблонов для проектирования процессов и систем.

Обоснование выбора: Microsoft Visio будет применяться для создания IDEF0-диаграмм, отображающих функциональную структуру системы. Благодаря поддержке IDEF0, Visio позволит представить процессы на уровнях AS-IS и TO-BE, что необходимо для анализа и улучшения функций системы. Интерфейс данного приложения представлен на рисунке 10.

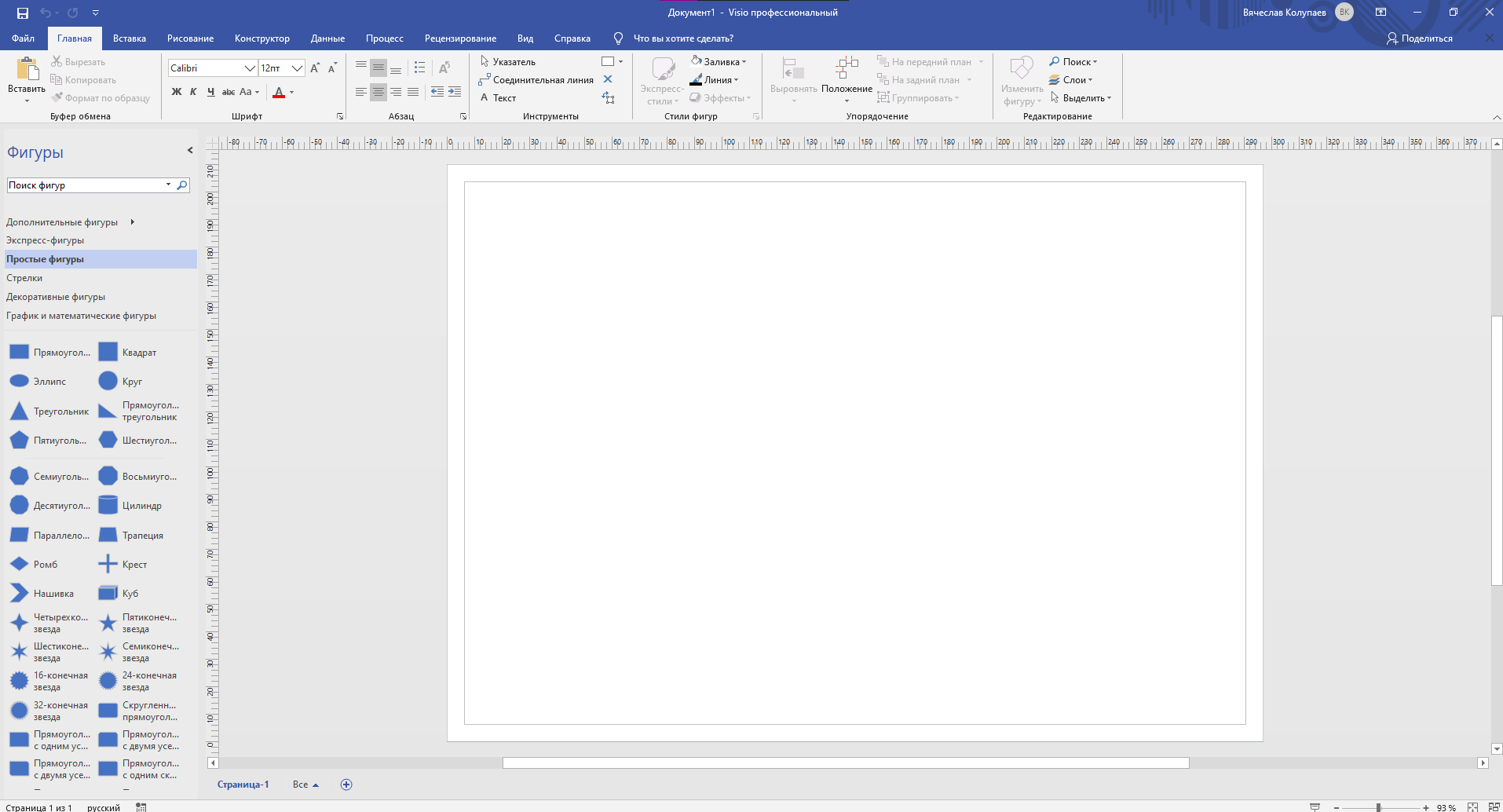


Рисунок 10 – Интерфейс Visio.

3) **DBDesigner**

Описание: DBDesigner — это инструмент для проектирования баз данных, который позволяет визуально моделировать структуру данных и задавать связи между таблицами.

Обоснование выбора: DBDesigner будет использоваться для разработки структуры базы данных системы. Это облегчит проектирование схем хранения данных для регистрации участников, распределения по категориям и учёта результатов соревнований. Интерфейс представлен на рисунке 11.

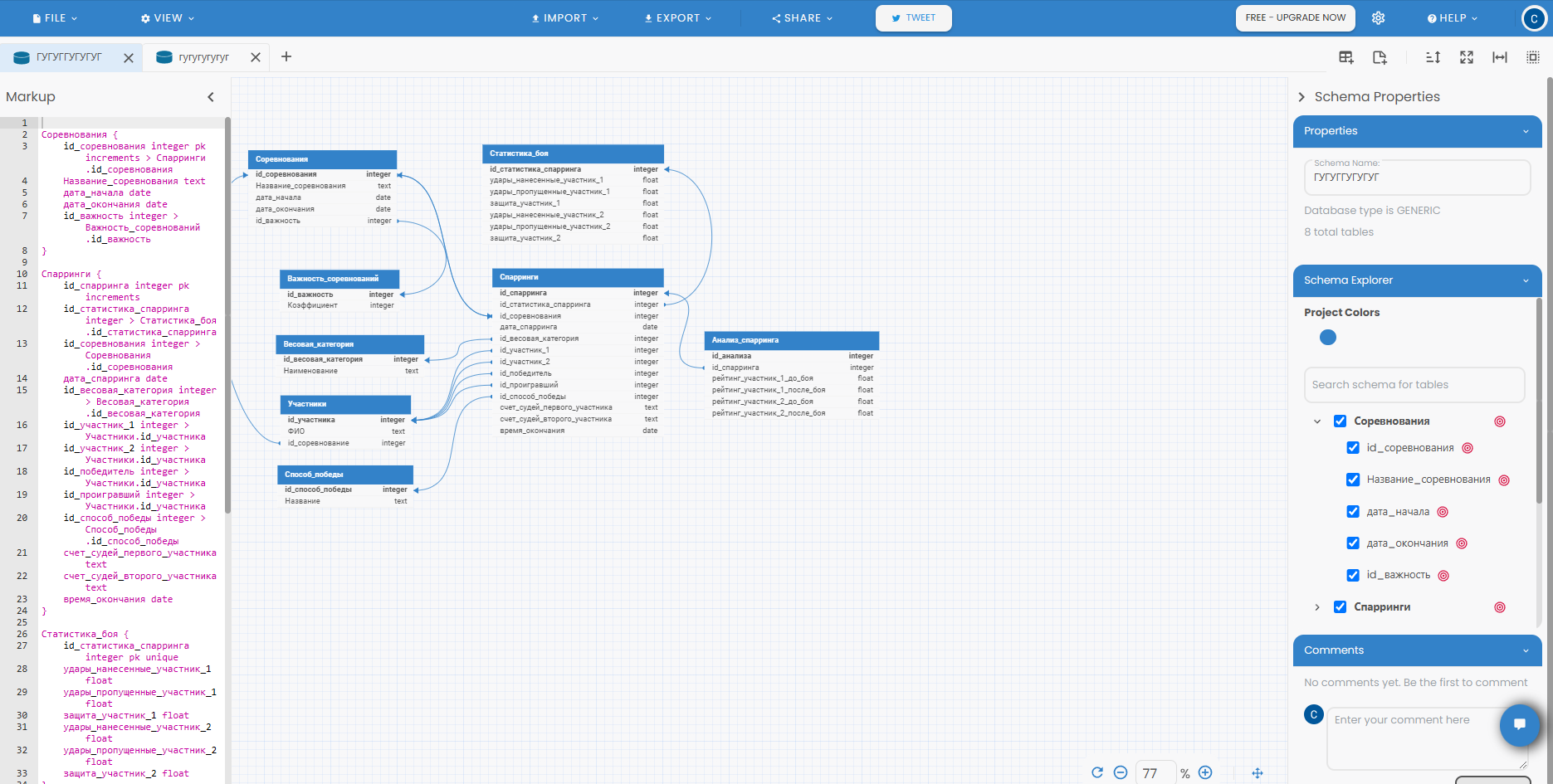


Рисунок 11 – Интерфейс DBDesigner.

### 2.1.2 Средства реализации

1. Python

Описание: Python — это высокоуровневый язык программирования, который широко используется для создания прикладных систем, так как имеет множество библиотек и поддерживает интеграцию с различными инструментами.

Обоснование выбора: Python будет выбран как основной язык программирования для реализации системы. Его универсальность и богатая библиотека модулей облегчат создание необходимой функциональности, а также тестирование и модификацию приложения.

1. PyQt5

Описание: PyQt5 — это библиотека для создания графических интерфейсов на Python. Она позволяет разрабатывать приложения с элементами управления и интерфейсами, поддерживающими кроссплатформенность.

Обоснование выбора: PyQt5 будет использоваться для создания интерфейса пользователя, обеспечивая интуитивно понятное взаимодействие с системой. С помощью PyQt5 планируется реализовать интерфейс для регистрации участников, жеребьевки, учёта и просмотра результатов боев.

1. VS Code

Описание: Visual Studio Code (VS Code) — это текстовый редактор с расширенными функциями для программирования, поддерживающий Python и различные расширения.

Обоснование выбора: VS Code запланирован в качестве основной среды разработки для Python и PyQt5. Этот редактор обладает инструментами для дебаггинга, управления проектом и работы с кодом, что позволит эффективно организовать процесс разработки и упростит тестирование системы. Интерфейс данного приложения продемонстрирован на рисунке 12.

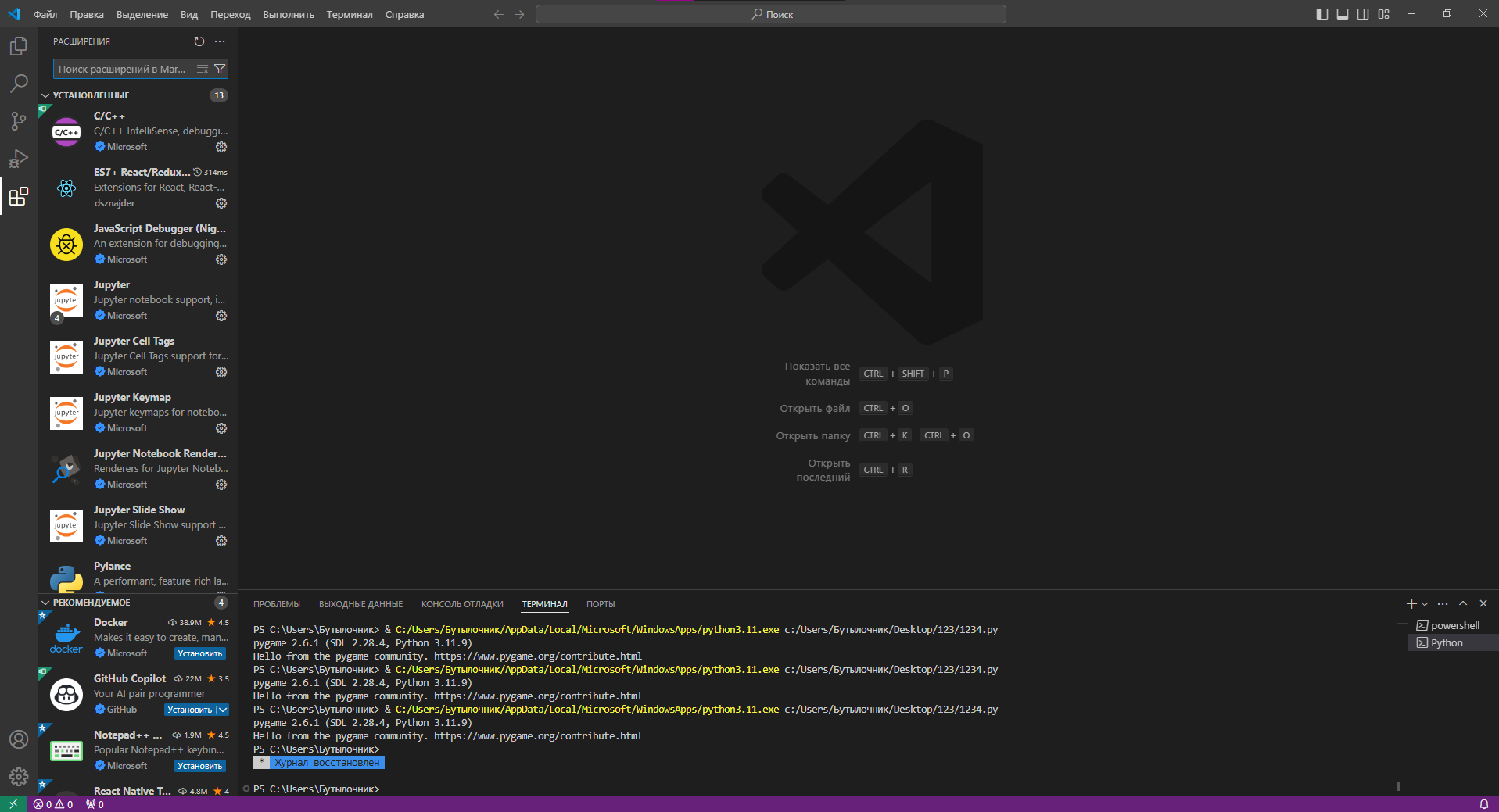


Рисунок 12 – Интерфейс приложения VS Code.

1. SQLite

Описание: SQLite — это встроенная реляционная СУБД, известная своей простотой и возможностью использования без сложной настройки сервера.

Обоснование выбора: SQLite будет использоваться для локального хранения данных о турнире, так как она подходит для систем со средними объёмами данных и предоставляет высокую скорость обработки данных. Это позволит надёжно сохранять и обрабатывать данные участников, распределение по категориям и результаты боев.

## 2.2 Проектирование архитектуры приложения

Для автоматизированной системы организации боксерских соревнований планируется архитектура с разделением на клиентскую и серверную части, а также базы данных для хранения информации о турнире. Данная архитектура позволит гибко управлять процессом регистрации, жеребьевки, распределения по весовым категориям и учёта результатов боев.

Для реализации системы была выбрана клиент-серверная архитектура с локальной базой данных, что обеспечивает следующие преимущества:

* Гибкость и масштабируемость: архитектура позволяет в дальнейшем интегрировать дополнительные модули, например, для вывода данных на веб-интерфейс.
* Упрощение разработки и тестирования: использование SQLite как встроенной базы данных позволяет разрабатывать и тестировать систему на одном устройстве.
* Надежность данных: с локальным хранением данных в базе данных вероятность потерь информации минимизируется, что особенно важно при работе с результатами и статистикой турниров.

### 2.2.1 Описание компонентов архитектуры

1. **Клиентская часть (Frontend):**

Описание: Клиентское приложение будет реализовано на базе PyQt5, что обеспечивает простой и удобный графический интерфейс (GUI) для организаторов, судей и других пользователей. Это приложение устанавливается на рабочие станции организаторов турниров и позволяет пользователям взаимодействовать с системой.

Функции:

* Регистрация участников: Форма ввода данных, которая позволяет организаторам добавлять новых участников в базу данных, указывая такие параметры, как имя, возраст, весовую категорию, спортивные достижения и т.д.
* Жеребьевка: Механизм автоматической жеребьевки, определяющий порядок боев и распределение участников в турнирной сетке.
* Ввод результатов: Интерфейс для ввода результатов каждого боя, включая счет, результаты по раундам, технику боя и другие параметры.
* Просмотр статистики: Возможность отображения статистики боев и участников, а также текущего состояния турнира (например, турнирная таблица и рейтинг участников).

1. **Серверная часть (Backend):**

Описание: Серверная часть будет реализована с использованием Python. Она будет обрабатывать запросы от клиентской части и взаимодействовать с базой данных SQLite для хранения и получения информации. Сервер также будет включать логику распределения по весовым категориям, автоматическую жеребьевку и алгоритмы для расчета статистики боев.

Функции:

* Валидация данных: Проверка корректности вводимой информации, например, проверка, что участники добавлены с правильной весовой категорией.
* Алгоритм жеребьевки: Автоматическое распределение участников в турнирной сетке на основе их весовой категории и случайного выбора.
* Обработка результатов: Прием данных от клиента о результатах боев и расчет турнирной таблицы.
* Обновление базы данных: Передача обновленных данных в базу, например, добавление новых участников или обновление информации по результатам боев.

1. **База данных (Database):**

Описание: База данных на основе SQLite обеспечивает локальное хранение данных. Она хранит всю информацию о турнире, включая участников, расписание боев, результаты и статистику.

Функции:

* Хранение данных участников: Полный список всех зарегистрированных участников, их характеристики и достижения.
* Жеребьевка и расписание: Структурированные данные о турнирной сетке, расписании боев и распределении участников.
* Результаты и статистика: Информация о каждом бое, включая победителей, счет, рейтинг участников.

Более подробный пример структуры данных БД, представлен на рисунке 13.

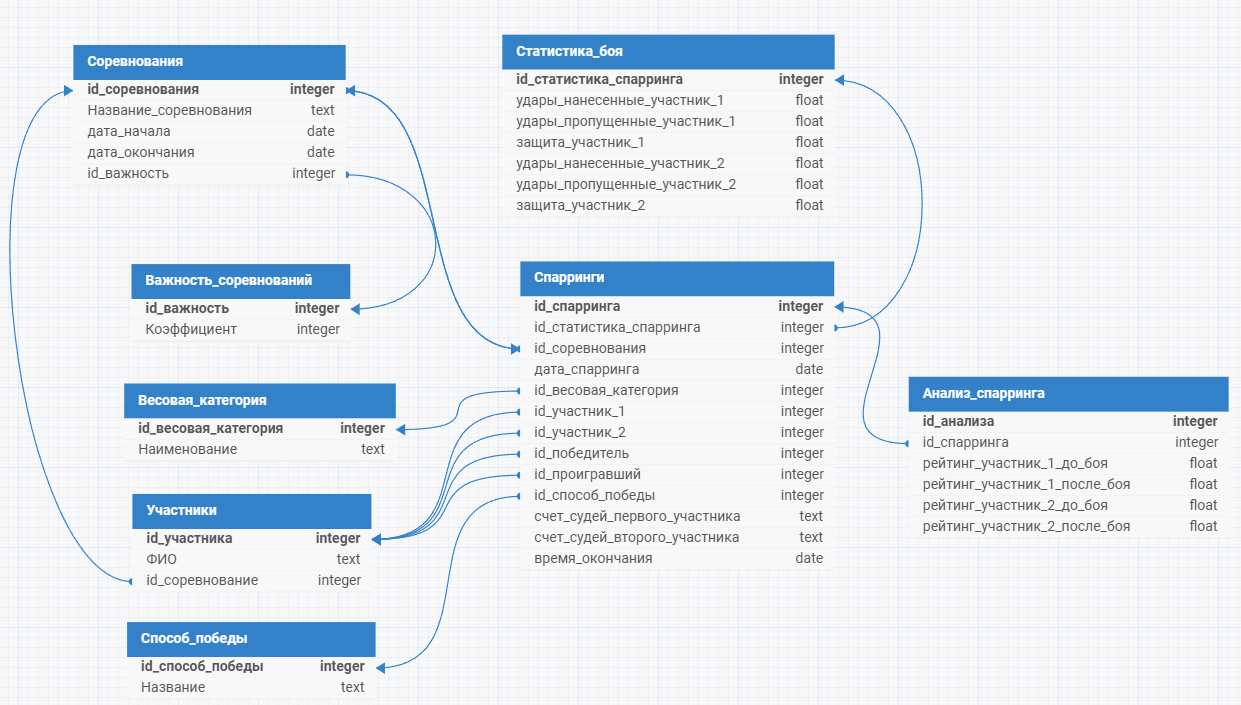


Рисунок 13 – Структура данных в БД

### 2.2.2 Схема архитектуры приложения и её описание

Архитектура приложения для автоматизации боксерских соревнований построена на многоуровневой схеме, чтобы обеспечить модульность, удобство в эксплуатации и масштабируемость. Система разделена на несколько слоев, каждый из которых выполняет конкретные функции, что обеспечивает гибкость и простоту поддержки приложения.

Описание модулей и взаимодействий:

**Пользовательский интерфейс (UI):**

* Этот модуль предоставляет интерфейс для участников, судей и администраторов. Он включает регистрационные формы для участников, интерфейс жеребьевки, ввода и отслеживания результатов боев, а также раздел для управления расписанием соревнований.
* Данные, введенные через UI, отправляются на сервер для обработки и сохранения в базе данных, что позволяет системе обновлять данные в реальном времени.

**Слой логики приложения:**

* Модуль регистрации участников: обрабатывает данные, введенные пользователями во время регистрации, автоматически распределяя участников по весовым категориям на основе предоставленных данных.
* Модуль жеребьевки и расписания: позволяет администратору настраивать сетку турнира и распределять бои с учетом интервалов времени и распределения судей.
* Модуль управления результатами: обеспечивает сбор и сохранение данных о результатах боев, их автоматическую обработку и отображение через интерфейс.
* Система оповещений: отправляет уведомления участникам и администраторам о предстоящих боях, результатах и изменениях в расписании.

**База данных:**

* В базе данных хранятся данные об участниках, боевых категориях, расписании, результатах боев, а также данные пользователей системы и их учетные записи.
* База данных обеспечивает хранение и быстрый доступ к данным, а также поддерживает систему бэкапа и восстановления.
* Система аналитики и отчетов:
* Модуль аналитики анализирует данные о результатах соревнований, формирует рейтинги участников, а также предоставляет подробную статистику для последующего анализа.
* Результаты анализов могут использоваться для улучшения распределения участников по категориям, оптимизации расписания, а также для отчетности перед организаторами соревнований.

**Система безопасности и аутентификации:**

* Этот модуль управляет учетными записями пользователей и их доступом к различным функциям системы. Поддерживается аутентификация для обеспечения безопасности данных.
* Важной частью является система защиты от несанкционированного доступа и фиксация действий пользователей, что позволяет отслеживать все изменения.

**Описание взаимодействия компонентов:**

* Интерфейс UI взаимодействует со слоем логики приложения, передавая данные пользователей для регистрации, жеребьевки и учета результатов боев. Результаты обработки отображаются пользователю в режиме реального времени.
* Слой логики приложения отправляет запросы к базе данных для извлечения и сохранения данных, таких как информация об участниках, жеребьевке и результатах. Логика приложения также взаимодействует с системой оповещений и аналитикой для отправки уведомлений и генерации отчетов.
* API позволяет внешним приложениям получать доступ к функционалу, таким как результаты соревнований и расписание боев, что повышает гибкость системы.
* Система оповещений и система безопасности интегрированы с логикой приложения, обеспечивая защиту данных и своевременные уведомления участников.

Схема архитектуры приложения представлена на рисунке 14.

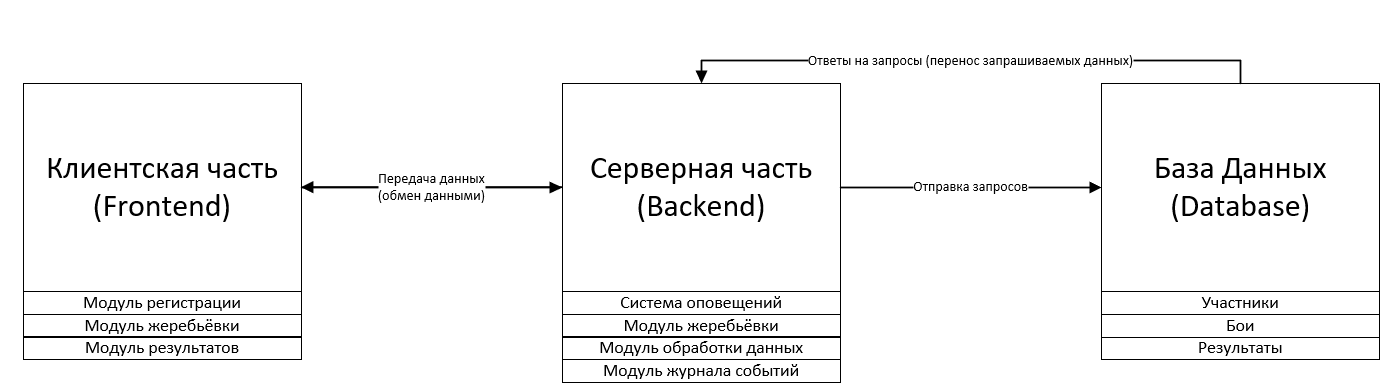


Рисунок 14 – Схема архитектуры приложения

## 2.3 Проектирование хранилища данных

Для системы управления соревнованиями по боксу предусмотрено использование реляционной базы данных для структурированного хранения информации о соревнованиях, участниках, спаррингах, статистике боёв и других элементах, необходимых для проведения и анализа соревнований.

**Основные сущности и атрибуты**

В системе предусмотрены следующие сущности (см. Таблица 4):

Таблица 4 – Основные сущности и атрибуты системы

|  |  |  |
| --- | --- | --- |
| **Сущность** | **Атрибуты** | **Описание** |
| Соревнования | id\_соревнования (PK), Название\_соревнования, дата\_начала, дата\_окончания, id\_важность | Информация о соревнованиях, включая название, даты проведения и важность. |
| Спарринги | id\_спарринга (PK), id\_статистика\_спарринга, id\_соревнования, дата\_спарринга, id\_весовая\_категория, id\_участник\_1, id\_участник\_2, id\_победитель, id\_проигравший, id\_способ\_победы, счет\_судей\_первого\_участника, счет\_судей\_второго\_участника, время\_окончания | Сведения о спаррингах в рамках соревнований, включая участников, результат и способ победы. |
| Статистика боя | id\_статистика\_спарринга (PK), удары\_нанесенные\_участник\_1, удары\_пропущенные\_участник\_1, защита\_участник\_1, удары\_нанесенные\_участник\_2, удары\_пропущенные\_участник\_2, защита\_участник\_2 | Детальная статистика по каждому бою, включая удары и защиту участников. |
| Анализ спарринга | id\_анализа (PK), id\_спарринга, рейтинг\_участник\_1\_до\_боя, рейтинг\_участник\_1\_после\_боя, рейтинг\_участник\_2\_до\_боя, рейтинг\_участник\_2\_после\_боя | Анализ рейтинга участников до и после боя. |
| Способ победы | id\_способ\_победы (PK), Название | Способы победы в спарринге, например, "нокаут" или "решение судей". |
| Весовая категория | id\_весовая\_категория (PK), Наименование | Категории веса для участников, такие как "легкий вес", "средний вес". |
| Участники | id\_участника (PK), ФИО | Личные данные участников соревнований. |
| Важность соревнований | id\_важность (PK), Коэффициент | Важность соревнований, влияющая на их приоритет или значимость. |

**Логическая модель данных**

Логическая модель данных описывает, как основные сущности системы связаны друг с другом, и позволяет сохранить структуру базы данных, обеспечивая целостность данных через внешние ключи. Рассмотрим детально основные взаимосвязи между сущностями:

1. **Связь "Соревнования — Спарринги"**

* id\_соревнования: В таблице Спарринги присутствует внешний ключ id\_соревнования, который ссылается на первичный ключ таблицы Соревнования. Эта связь определяет, какие спарринги проходят в рамках конкретного соревнования, и позволяет сопоставлять бои с конкретными турнирами. Таким образом, при выводе данных по соревнованиям мы можем видеть все спарринги, которые в них состоялись.

1. **Связь "Спарринги — Участники"**

* id\_участник\_1 и id\_участник\_2: Таблица Спарринги содержит два внешних ключа, id\_участник\_1 и id\_участник\_2, которые ссылаются на таблицу Участники. Это позволяет хранить информацию о двух участниках каждого спарринга.
* id\_победитель и id\_проигравший: Эти поля также ссылаются на таблицу Участники и определяют исход боя. Это дает возможность отдельно фиксировать как участников боя, так и его результат, связывая его с конкретными спортсменами.

1. **Связь "Спарринги — Статистика\_боя"**

* id\_статистика\_спарринга: Таблица Статистика\_боя содержит подробную информацию о ходе спарринга, включая количество нанесенных и пропущенных ударов, защиту участников и т.д. Внешний ключ id\_статистика\_спарринга в таблице Спарринги позволяет связывать каждый спарринг с его статистикой, храня всю аналитическую информацию отдельно для удобства расчётов и анализа.

1. **Связь "Спарринги — Весовая\_категория"**

* id\_весовая\_категория: Весовая категория указывается для каждого спарринга через внешний ключ id\_весовая\_категория, ссылающийся на таблицу Весовая\_категория. Это позволяет фильтровать и анализировать спарринги по весовым категориям, что важно для ведения статистики и соблюдения правил соревнований.

1. **Связь "Спарринги — Способ\_победы"**

* id\_способ\_победы: Таблица Способ\_победы хранит информацию о возможных способах завершения боя, таких как "нокаут" или "решение судей". Внешний ключ id\_способ\_победы в таблице Спарринги позволяет фиксировать, каким образом был завершен бой, что упрощает анализ победных стратегий участников.

1. **Связь "Спарринги — Анализ\_спарринга"**

* id\_спарринга: Таблица Анализ\_спарринга включает данные о рейтингах участников до и после боя, что дает возможность отслеживать изменение показателей после каждого боя. Внешний ключ id\_спарринга в таблице Анализ\_спарринга связывает эту таблицу с Спаррингами и помогает анализировать влияние боев на рейтинги участников.

1. **Связь "Соревнования — Важность\_соревнований"**

* id\_важность: Поле id\_важность в таблице Соревнования ссылается на таблицу Важность\_соревнований. Это позволяет классифицировать турниры по уровню значимости (например, национальные, международные), что может влиять на стратегическое планирование и подготовку к соревнованиям.

Эти взаимосвязи позволяют создавать связные запросы к базе данных, получая данные по каждому бою, участникам и результатам турниров, обеспечивая структурированность и удобство работы с информацией.

**Дополнительные ограничения целостности**

1. **Целостность внешних ключей**

Все внешние ключи обеспечивают целостность данных. Они предотвращают несогласованность данных между таблицами, например, наличие записи в Спаррингах без привязки к существующему Соревнованию или Участнику. Если запись в одной из связанных таблиц удаляется, то внешний ключ не позволит нарушить связь или удалить запись, если для этого не установлены каскадные действия.

1. **Уникальные ограничения**

Каждая сущность имеет первичный ключ, который обеспечивает уникальность каждой записи. Например, id\_соревнования в Соревнованиях и id\_участника в Участниках являются уникальными идентификаторами, исключая возможность дублирования данных. Это позволяет хранить уникальные данные о соревнованиях и участниках.

1. **Обязательные поля**

Ключевые поля, такие как id\_соревнования, id\_участника, и id\_спарринга, обязательны для создания корректных связей между данными. Наличие обязательных полей позволяет поддерживать целостность базы, исключая появление "пустых" связей и ошибок.

1. **Контроль каскадных операций**

Важно, чтобы база данных корректно обрабатывала каскадные операции при удалении или обновлении данных. Например, если запись о Соревновании удаляется, каскадное удаление удалит все Спарринги, связанные с этим соревнованием, или просто запретит удаление записи для защиты от потери данных.

## 2.4 Проектирование пользовательского интерфейса

### 2.4.1 Верхнеуровневое определение экранов

Для проектирования пользовательского интерфейса создадим верхнеуровневую структуру экранов, отражающую основные разделы и назначения приложения. Группировка экранов по разделам помогает пользователям легко ориентироваться и находить нужную информацию.

1. **Главная**

Основной экран, который содержит ссылки на все ключевые разделы приложения. Здесь могут отображаться последние новости, анонсы соревнований и общая информация о платформе. Этот экран служит отправной точкой для пользователей.

1. **Соревнования**

2.1 Просмотр соревнований (View Competitions): Экран, где пользователь может увидеть список всех текущих и запланированных соревнований. Доступны фильтры для сортировки по дате, уровню значимости и другим критериям.

2.2 Выбор соревнований (Select Competition): Позволяет пользователю выбрать конкретное соревнование для детального просмотра. Экран включает подробную информацию о выбранном событии, в том числе место проведения, дату и список участников.

1. **Участникам**

3.1 Правила регистрации (Registration Rules): Экран с правилами и требованиями для регистрации на соревнования. Содержит описание условий участия, требований к участникам и регламент.

3.2 Регистрация на соревнования (Competition Registration): Форма регистрации, где участники могут подать заявку на участие в соревновании, заполнив необходимые данные, такие как имя, категория веса и контактная информация.

1. **Новости**

Экран, содержащий последние новости, статьи и анонсы, связанные с соревнованиями и событиями в мире бокса. Информация структурирована в виде ленты новостей, чтобы пользователи могли следить за обновлениями.

1. **Контакты**

Экран для отображения контактной информации. Здесь пользователи могут найти номер телефона, email и, возможно, форму обратной связи для связи с администрацией платформы или организаторами соревнований.

1. **Профиль**

6.1 Зарегистрироваться/Войти (Register/Login): Экран для авторизации и регистрации пользователей. Включает поля для ввода email, пароля и, при необходимости, других данных.

6.2 Профиль (View Profile): Экран профиля, где пользователь может видеть свои личные данные, историю регистрации на соревнования и другую информацию.

6.3 Редактировать профиль (Edit Profile): Форма для редактирования личной информации, такой как имя, контактные данные и настройки профиля.

6.4 Выйти из профиля (Logout): Опция для выхода из учетной записи. После нажатия пользователь будет перенаправлен на экран авторизации.

Верхнеуровневая структура данных представлена на рисунке 15.

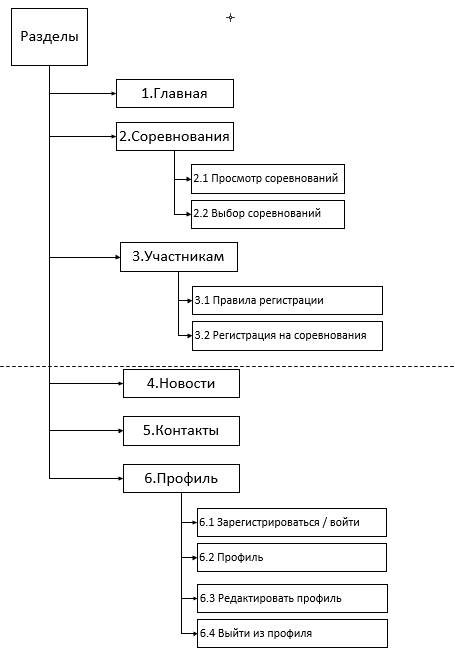


Рисунок 15 – Верхнеуровневая структура данных

### 2.4.2 Назначение экранов

Для описания экранов и их взаимодействий создадим таблицу, которая охватывает основные характеристики и поведение каждого экрана. Эта структура помогает понять функциональные аспекты каждого экрана и необходимые элементы для тестирования и разработки. Результаты представлены в таблице 5.

Таблица 5.1 – Описание экранов и их назначение

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Название экрана** | **Состояния** | **Поля ввода для валидации** | **Описание экрана и его поведения** | **Список уведомлений** | **Идентификаторы UI-контролов** | **Стили** |
| 1 | Главная (Home) | Normal | - | Основной экран с кнопками навигации: Главная, Соревнования, Участникам, Новости, Контакты и Профиль. При клике на каждый элемент происходит переход к соответствующему разделу. | - | HomeButton, CompetitionButton | Основной стиль приложения |
| 2 | Просмотр соревнований (View Competitions) | Loading, Normal | - | Отображает список соревнований с возможностью фильтрации по дате, значимости и другим параметрам. При выборе соревнования осуществляется переход к экрану "Выбор соревнований". | - | CompetitionList, FilterButton | Карточки с тенями, список |
| 2.1 | Выбор соревнований (Select Competition) | Normal | - | Показывает подробную информацию о выбранном соревновании (место, дата, участники). Есть возможность вернуться к списку соревнований. | Уведомление при отсутствии данных | BackButton, DetailView | Карточка с детальной информацией |

Таблица 5.2 – Продолжение таблицы описание экранов и их назначение

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | Правила регистрации (Registration Rules) | Normal | - | Экран с информацией о правилах и требованиях к участникам соревнований. Только текстовое описание, без интерактивных элементов. | - | - | Стандартный текстовый стиль |
| 3.1 | Регистрация на соревнования (Competition Registration) | Normal, Error | Имя (>=2 chars), Категория веса (обязательное поле) | Форма регистрации участника. При неверном вводе данных отображается сообщение об ошибке, а при успешной регистрации — подтверждение. | Уведомление об успешной регистрации | RegisterButton, NameField, WeightCategoryDropdown | Стиль для форм ввода |
| 4 | Новости (News) | Loading, Normal | - | Лента новостей и обновлений. Новости выводятся в виде списка, при клике на новость — переход на страницу подробной информации. | - | NewsList, NewsItem | Список с картинками и заголовками |
| 5 | Контакты (Contacts) | Normal | - | Экран с контактной информацией (телефон, email) и возможностью отправить сообщение через форму обратной связи. | Уведомление об отправке сообщения | ContactForm, SendMessageButton | Стиль для текстовых полей |

Таблица 5.3 – Продолжение таблицы описание экранов и их назначение

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | Зарегистрироваться/Войти (Register/Login) | Normal, Loading, Error | Email (>5 и <160), Пароль (>=8 и <24) | Экран для авторизации/регистрации пользователя. При вводе неверных данных показывается сообщение об ошибке. | Ошибки авторизации | LoginButton, RegisterButton, EmailField, PasswordField | Стиль формы авторизации |
| 6.1 | Профиль (View Profile) | Normal, Loading | - | Отображает информацию о пользователе, историю регистраций на соревнования и настройки аккаунта. Возможность перейти на экран редактирования профиля. | - | EditProfileButton, ViewHistoryButton | Карточка профиля |
| 6.2 | Редактировать профиль (Edit Profile) | Normal, Error | Имя (>=2 chars), Email (>5 и <160), Телефон (11 digits) | Форма для редактирования личных данных. При некорректном вводе отображается сообщение об ошибке, при успешном сохранении — подтверждение изменений. | Уведомление об успешном изменении | SaveButton, NameField, EmailField, PhoneField | Стиль для редактирования профиля |
| 6.3 | Выйти из профиля (Logout) | Confirm, Canceled | - | Подтверждение выхода из профиля. Если пользователь соглашается, происходит выход и возврат на экран входа. | Диалог подтверждения выхода | ConfirmLogoutButton, CancelButton | Стандартный стиль приложения |

### 2.4.3 Отрисовка и описание макетов ваших экранов

Главная страница представлена на рисунке 16.

На главной странице отображаются основные разделы интерфейса, такие как: "Домой", "Соревнования", "Участники", "Новости", и "Контакты". В верхней части экрана также предусмотрена область личного кабинета, содержащая кнопки "Вход" и "Регистрация". Кнопки позволяют пользователю перейти к авторизации или регистрации. Главная страница служит точкой входа в систему, предоставляя пользователям доступ к основным функциям и навигацию по разделам.



Рисунок 16 – Макет главной страницы

Форма регистрации Представлена на рисунке 17.

При нажатии на кнопку "Регистрация" на главной странице, основной фон затемняется, и открывается форма для ввода регистрационных данных. Форма включает следующие поля:

* Поле для ввода ФИО (например, «Колупаев Вячеслав Андреевич»).
* Список для выбора спортивной организации (например, «ИРНИТУ»).
* Поле для ввода даты рождения (например, «11.04.2003»).
* Выпадающий список для выбора роли (Участник, Тренер, Судья).
* Поле для ввода телефона или электронной почты.
* Поля для ввода и подтверждения пароля (оба поля должны совпадать).

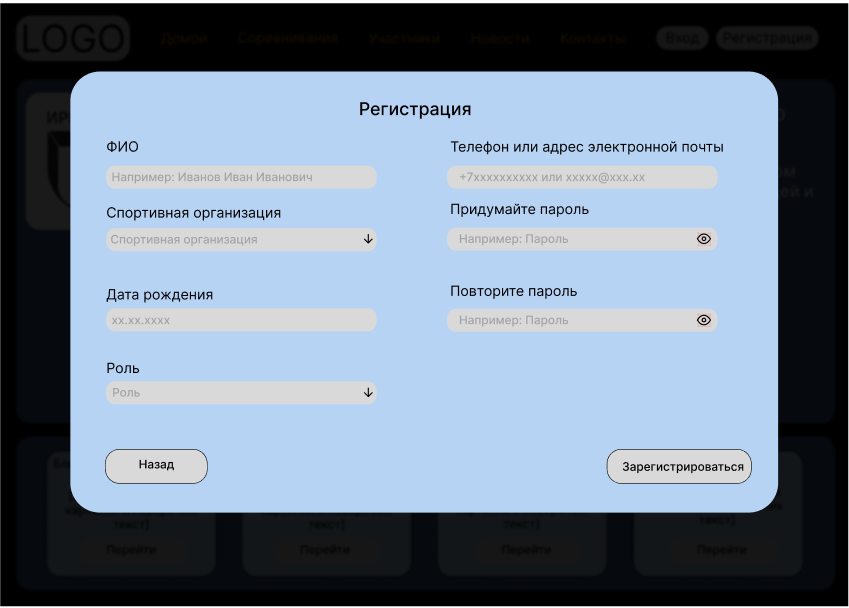


Рисунок 17 – Макет регистрации пользователя

Форма также проверяет корректность введенных данных, например, соответствие паролей, формат электронной почты, наличие всех обязательных полей и т.д. После успешного заполнения форма закрывается, и пользователь получает доступ к своему профилю.

Страница анализа спаррингов представлен на рисунке 18 .

* На странице анализа спаррингов пользователь может выбрать:
* Конкретное соревнование из списка (например, «Первенство национальной студенческой лиги среди юниоров»).
* Дату спарринга (например, «24 октября 2024 года»).
* Весовую категорию (например, «60 кг»).
* Конкретный спарринг, проведенный в выбранный день.

Выбор параметров позволяет пользователю просмотреть информацию об интересующем спарринге и его результатах, что помогает провести глубокий анализ выступления участников.

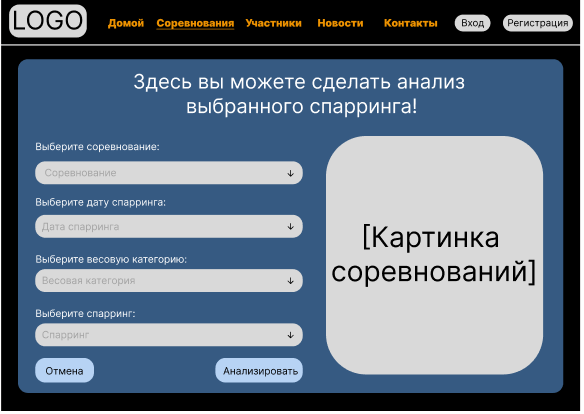


Рисунок 18 – Макет страницы с выбором конкретного спарринга под анализ

Макет экрана результатов анализа спарринга представлен на рисунке 19.

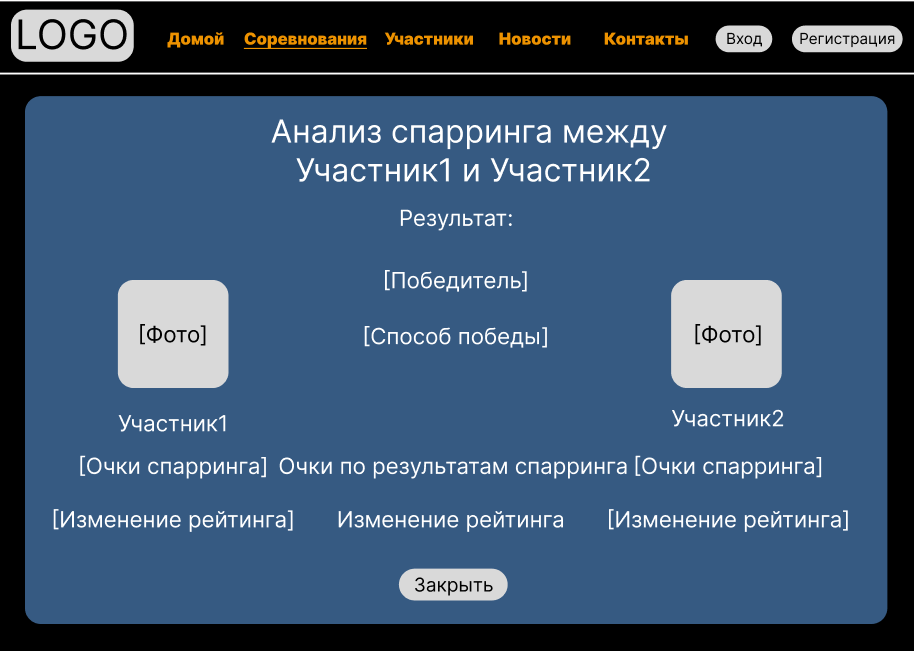


Рисунок 19 – Макет анализа страницы спаррингов

Назначение: Этот экран отображает результаты анализа выбранного спарринга, предоставляя детализированную информацию о ходе и итогах боя.

Элементы управления:

* Поля для отображения информации о выбранном соревновании, дате и весовой категории.
* Данные о статистике боя, такие как количество ударов, оценки судей, статус победителя и рейтинги участников до и после боя.

Описание поведения: Экран загружается с подробной информацией о выбранном спарринге. Пользователь может просматривать результаты, анализировать статистику боя и сравнивать рейтинги участников.

### 2.4.4 Карта экранов и состояний

Для разработки системы управления соревнованиями по боксу было создано верхнеуровневое определение экранов и их состояний, которое помогает структурировать взаимодействие пользователя с системой. Карта экранов и состояний описывает основные разделы и экраны интерфейса, а также переходы между ними, что позволяет обеспечить логичную навигацию и удобство использования.

Каждый экран имеет определенные состояния и возможные действия, которые пользователь может выполнить. Например, с главной страницы пользователь может перейти к регистрации, входу в систему, просмотру списка соревнований, а также к новостям, контактам и анализу спаррингов.

Состояния экранов включают в себя:

* Нормальное отображение — стандартное состояние, в котором отображается содержимое экрана.
* Заполненная форма — состояние при вводе данных пользователем.
* Состояние валидации — состояние, при котором система проверяет корректность введенных данных и выводит соответствующие уведомления.

Кроме того, карта экранов включает все основные переходы между экранами и описывает различные возможные состояния, чтобы помочь разработчикам при создании блок-схем и при проектировании маршрутов взаимодействия, она представлена на рисунке 20.

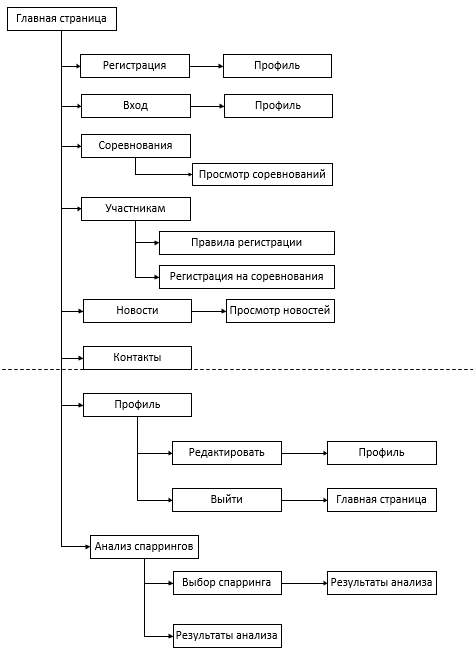


Рисунок 20 – Карта экранов и состояний

**Описание экранов и их переходов**

1. **Главная страница**

Состояния: Нормальное отображение, затемнение при переходе к регистрационным формам.

Переходы:

* Нажатие на кнопку "Регистрация" → Переход на экран регистрации с затемнением главной страницы.
* Нажатие на кнопку "Вход" → Переход на экран входа.
* Переход в разделы "Соревнования," "Участникам," "Новости," "Контакты," или "Профиль" → Соответствующий экран раздела.

1. **Экран регистрации**

Состояния: Пустая форма, заполненная форма, состояние валидации (ошибки при неправильном заполнении).

Переходы:

* Успешная регистрация → Переход на экран профиля.
* Ошибка при заполнении → Появление уведомлений об ошибках (например, "пароли не совпадают").

1. **Экран входа**

Состояния: Пустая форма, заполненная форма, состояние валидации (ошибка при неправильных данных).

Переходы:

* Успешный вход → Переход на экран профиля.
* Ошибка при входе → Появление уведомления об ошибке (например, "неверный логин или пароль").

1. **Профиль**

Состояния: Просмотр профиля, режим редактирования.

Переходы:

* Нажатие на кнопку "Редактировать" → Переход в режим редактирования профиля.
* Нажатие на кнопку "Выйти" → Переход на главную страницу.

1. **Редактирование профиля**

Состояния: Заполненная форма, состояние валидации (ошибка при неправильных данных).

Переходы:

* Успешное обновление данных → Переход обратно на экран профиля.
* Ошибка при обновлении данных → Появление уведомления об ошибке.

1. **Раздел "Соревнования"**

Состояния: Список соревнований, просмотр выбранного соревнования.

Переходы:

* Выбор соревнования → Переход на экран выбранного соревнования с деталями.

1. **Раздел "Участникам"**

Состояния: Просмотр правил, регистрация на соревнования.

Переходы:

* Нажатие на "Правила регистрации" → Переход на экран с правилами.
* Нажатие на "Регистрация на соревнования" → Переход на форму регистрации для участия.

1. **Новости**

Состояния: Список новостей, просмотр выбранной новости.

Переходы:

* Выбор новости → Переход на экран с полным текстом новости.

1. **Контакты**

Состояния: Стандартное отображение.

Переходы: нет переходов на другие экраны.

1. **Анализ спаррингов - Экран выбора спарринга**

Состояния: Выбор соревнования, даты, весовой категории, конкретного спарринга.

Переходы:

* Нажатие на кнопку подтверждения выбора → Переход на экран результатов анализа спарринга.

1. **Анализ спаррингов - Экран результатов спарринга**

Состояния: Просмотр статистики спарринга.

Переходы: нет переходов на другие экраны.

# 3 Разработка

## 3.1 Итоговые результаты работы

В процессе разработки веб-приложения для анализа спортивных соревнований были реализованы следующие ключевые компоненты и функциональные модули. Основное внимание было уделено пользовательскому интерфейсу, серверной части и взаимодействию с базой данных.

### 3.1.1 Скриншоты готовых экранов:

1. **Главное меню сайта:**

* Навигационное меню обеспечивает доступ к основным разделам: «Домой», «Соревнования», «Участники», «Новости», «Контакты».
* Также доступны кнопки для «Входа» и «Регистрации».
* Особое внимание уделено дизайну меню для удобства пользователя и его адаптивности.

Главная страница представлена на рисунке 21.

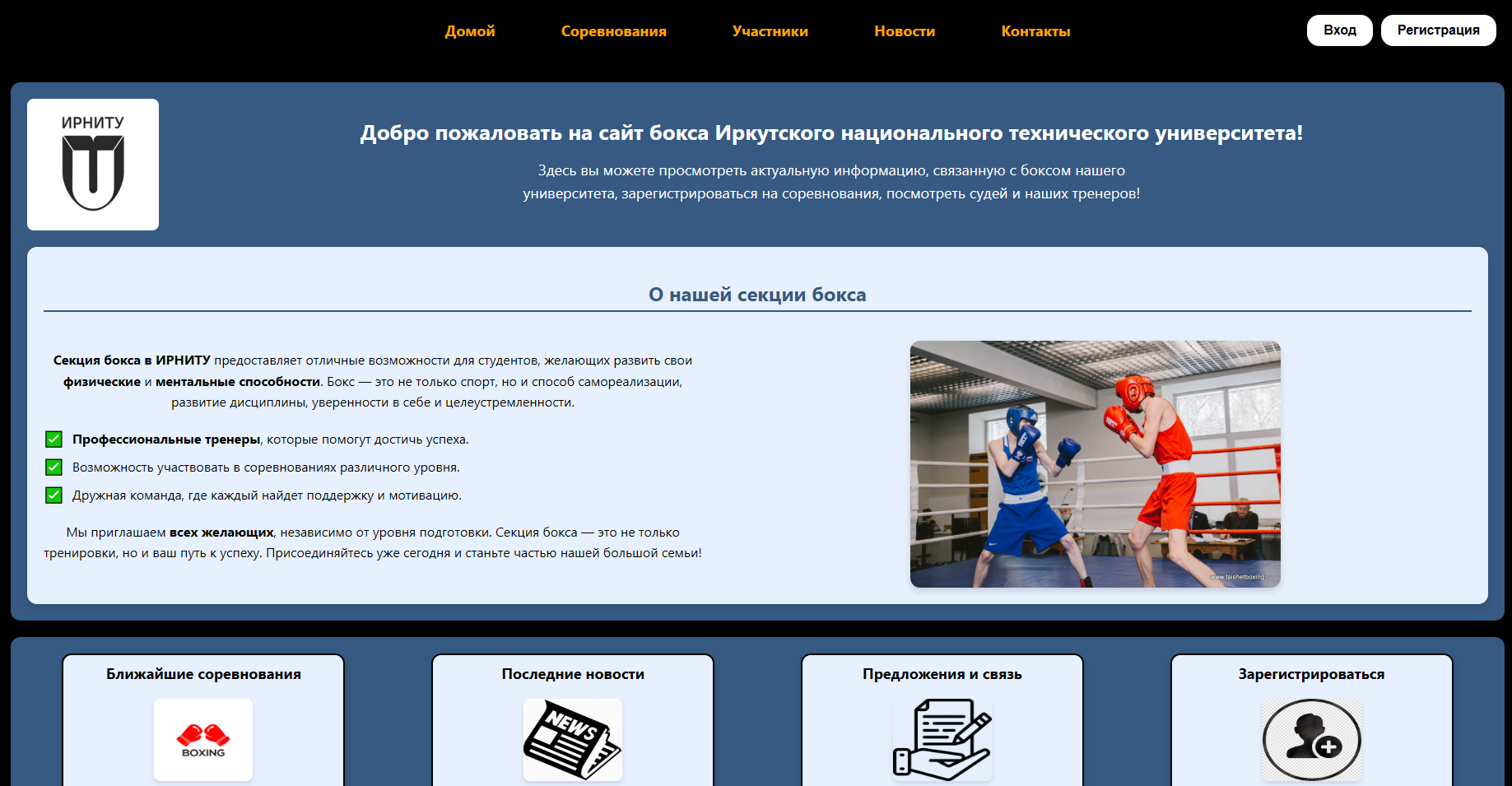


Рисунок 21 – Главная страница сайта

1. **Форма входа:**

* Реализован функционал аутентификации с вводом логина (телефонного номера) и пароля.
* Для удобства пользователя добавлены ссылки для регистрации нового пользователя и восстановления пароля.

Форма входа представлена на рисунке 22.

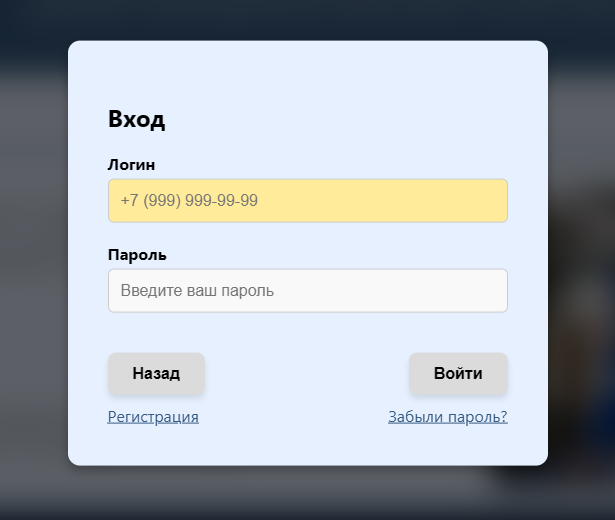


Рисунок 22 – Форма входа в личный кабинет на сайте

1. **Форма регистрации:**

* Пользователь может зарегистрироваться, указав свои данные: ФИО, спортивную организацию, дату рождения, роль (участник, тренер, судья), а также телефон и пароль.
* Реализована проверка правильности ввода данных и уведомление об ошибках.

Форма регистрации представлена на рисунке 23.

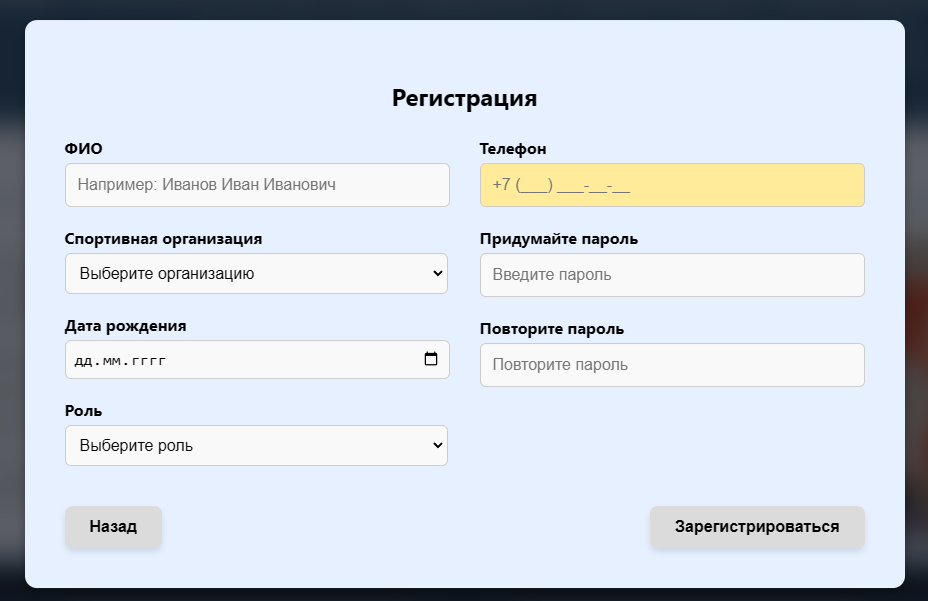


Рисунок 23 – Форма регистрации нового пользователя

1. **Анализ спаррингов:**

* Пользователь выбирает соревнование, дату, весовую категорию и конкретный спарринг для анализа.
* После выбора система рассчитывает изменение рейтинга участников в зависимости от результата спарринга.
* Добавлен функционал уведомления, если анализ уже выполнен ранее.

Страница анализа спаррингов представлена на рисунке 24.

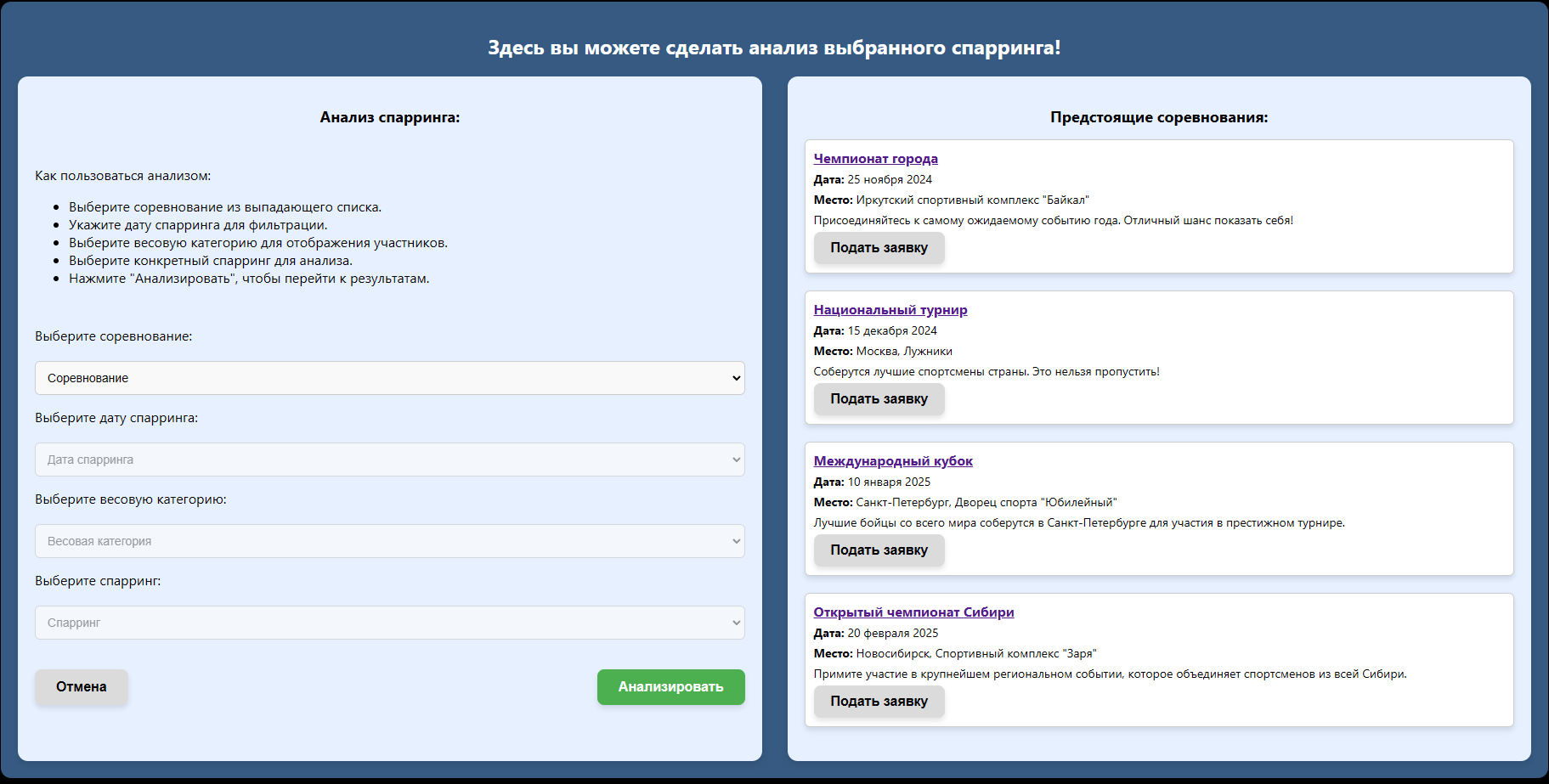


Рисунок 24 – Страница анализа спаррингов

### 3.1.2 Фрагменты кода:

**Подключение к базе данных SQLite:**

Вся логика работы с базой данных вынесена в отдельный модуль, что упрощает управление соединениями и обработку ошибок. Пример подключения:

const express = require('express');

const sqlite3 = require('sqlite3').verbose();

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

const PORT = 3001;

const db = new sqlite3.Database('./database.db', (err) => {

  if (err) {

    console.error('Ошибка подключения к базе данных:', err.message);

  } else {

    console.log('Успешное подключение к базе данных.');

  }

});

**Обработка анализа спарринга на сервере:**

Этот фрагмент кода отвечает за анализ данных о спарринге, вычисление изменений рейтинга и их сохранение в базу данных:

app.post('/analyze-rating/:sparringId', (req, res) => {

  const { sparringId } = req.params;

  const sparringQuery = `

    SELECT

      Sparrings.id\_sparring,

      Sparrings.id\_participant1,

      Sparrings.id\_participant2,

      Participants1.rating AS rating\_do1,

      Participants2.rating AS rating\_do2,

      Sparrings.id\_win,

      Sparrings.Points1, -- Очки первого участника

      Sparrings.Points2, -- Очки второго участника

      Competitions.id\_importance,

      Importance.value AS K -- Коэффициент важности

    FROM Sparrings

    JOIN Participants AS Participants1 ON Sparrings.id\_participant1 = Participants1.id\_participant

    JOIN Participants AS Participants2 ON Sparrings.id\_participant2 = Participants2.id\_participant

    JOIN Competitions ON Sparrings.id\_competition = Competitions.id\_competition

    JOIN Importance ON Competitions.id\_importance = Importance.id\_importance

    WHERE Sparrings.id\_sparring = ?;

  `;

  db.get(sparringQuery, [sparringId], (err, sparringData) => {

    if (err) {

      console.error('Ошибка получения данных о спарринге:', err.message);

      return res.status(500).json({ error: 'Ошибка базы данных.' });

    }

    if (!sparringData) {

      return res.status(404).json({ error: 'Спарринг не найден.' });

    }

    const {

      id\_participant1,

      id\_participant2,

      rating\_do1,

      rating\_do2,

      id\_win,

      Points1,

      Points2,

      K

    } = sparringData;

**SQL-запрос для создания таблицы Analyze**

Таблица Analyze предназначена для хранения данных анализа каждого спарринга, включая рейтинги участников до и после боя. Она служит связующим звеном между таблицей спаррингов и модулем анализа.

CREATE TABLE Analyze (

  id INTEGER PRIMARY KEY AUTOINCREMENT,

  id\_sparring INTEGER NOT NULL,

  rating\_do1 INTEGER NOT NULL,

  rating\_do2 INTEGER NOT NULL,

  rating\_po1 INTEGER,

  rating\_po2 INTEGER,

  FOREIGN KEY (id\_sparring) REFERENCES Sparrings(id\_sparring)

);

**Структура базы данных**

Структура базы данных включает следующие основные таблицы:

* Sparrings — информация о проведенных боях.
* Participants — данные об участниках.
* Competitions — информация о соревнованиях.
* Importance — коэффициенты важности соревнований.
* Analyze — данные анализа спаррингов.

Пример визуального отображения структуры:

* Таблица Participants содержит id\_participant, имя, текущий рейтинг.
* Таблица Competitions включает поле важности, связанное с таблицей Importance.
* Analyze связывает спарринги с расчетами рейтингов.

**Математическое обоснование**

Описание алгоритма:

Задача анализа спаррингов и последующего пересчета рейтингов основывается на наборе правил и математических зависимостей, которые позволяют точно определить влияние каждого спарринга на рейтинг участников. Алгоритм построен с использованием трех базовых структур:

1. Линейная структура: обрабатывает данные по каждому спаррингу.
2. Разветвляющаяся структура: определяет победителя на основе исхода боя.
3. Циклическая структура: пересчитывает рейтинги участников после каждого поединка.

Алгоритм пересчета рейтингов:

1. Входные данные:

* Название соревнования.
* Дата конкретного спарринга.
* Весовая категория.
* Участники спарринга.
* Результат боя (победа, поражение, ничья).
* Статистические данные о каждом бойце (количество ударов, пропущенных ударов, тип победы и т.д.).

1. Ожидаемый результат (E) для каждого бойца вычисляется на основе рейтингов перед боем. Формула расчета:

Где:

* — текущий рейтинг участника.
* ​ — текущий рейтинг соперника.

1. Фактический результат боя (S):

* S = 1 - победа.
* S = 0.5 - ничья.
* S = 0 - поражение.

1. Обновление рейтинга ():

Где:

* K - коэффициент важности соревнования.

**Пример анализа спарринга:**

1. Данные:

* Участник 1:
* Участник 1:
* Коэффициент важности:

1. Шаги расчета:

* Ожидаемый результат участника 1:
* Ожидаемый результат участника 2:
* Новый рейтинг участника 1:
* Новый рейтинг участника 2:

**SQL-запрос для анализа:**

Для получения данных анализа по определенному спаррингу, в данном случае для спарринга с идентификатором 1, используется следующий запрос:

SELECT rating\_do1, rating\_do2, rating\_po1, rating\_po2

FROM Analyze

WHERE id\_sparring = 1;

**Клиентская часть: Компонент анализа**

Этот компонент отвечает за отображение информации о соревнованиях и анализ спаррингов:

import React, { useState, useEffect } from "react";

import "./Analyze.css";

const Analyze = () => {

  const [ratingAnalysis, setRatingAnalysis] = useState(null);

  const [error, setError] = useState(null);

  useEffect(() => {

    fetch("http://localhost:3001/analyze-rating/1", { method: "POST" })

      .then((res) => res.json())

      .then((data) => setRatingAnalysis(data))

      .catch((err) => setError(err.message));

  }, []);

  if (error) return <div>Ошибка: {error}</div>;

  return (

    <div>

      <h1>Результаты анализа</h1>

      {ratingAnalysis && (

        <div>

          <p>Рейтинг до: {ratingAnalysis.rating\_do1}</p>

          <p>Рейтинг после: {ratingAnalysis.rating\_po1}</p>

        </div>

      )}

    </div>

  );

};

export default Analyze;

**Серверная часть: Обработка анализа**

Этот код выполняет расчет рейтингов и возвращает данные на клиент:

app.post('/analyze-rating/:sparringId', (req, res) => {

  const sparringId = req.params.sparringId;

  const query = `

    SELECT Participants1.rating AS rating\_do1, Participants2.rating AS rating\_do2

    FROM Sparrings

    JOIN Participants AS Participants1 ON Sparrings.id\_participant1 = Participants1.id\_participant

    JOIN Participants AS Participants2 ON Sparrings.id\_participant2 = Participants2.id\_participant

    WHERE Sparrings.id\_sparring = ?

  `;

  db.get(query, [sparringId], (err, data) => {

    if (err) return res.status(500).send("Ошибка базы данных");

    const { rating\_do1, rating\_do2 } = data;

    const E1 = 1 / (1 + Math.pow(10, (rating\_do2 - rating\_do1) / 400));

    const E2 = 1 - E1;

    const R\_new1 = Math.round(rating\_do1 + 30 \* (1 - E1));

    const R\_new2 = Math.round(rating\_do2 + 30 \* (0 - E2));

    res.json({

      rating\_do1,

      rating\_do2,

      rating\_po1: R\_new1,

      rating\_po2: R\_new2,

    });

  });

});

## 3.2 Вывод

**В ходе работы над проектом были реализованы**:

1. Структура базы данных:

* Уникальный дизайн таблиц с учетом связей между спаррингами, участниками и их рейтингами.
* Реализация дополнительных таблиц для хранения аналитических данных.

1. Логика расчета:

* Корректное применение формул для перерасчета рейтинга.
* Учет коэффициента важности соревнований.

1. Клиентская часть:

* Дружественный интерфейс для выбора соревнований и анализа данных.
* Уведомления о выполнении анализа.

1. Серверная часть:

* Оптимизация SQL-запросов.
* Обработка ошибок и уведомление пользователя.

Проект предоставляет пользователю возможность быстро и удобно анализировать спортивные данные, сохраняя результаты для последующего использования.