



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления (ИУ)»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

---

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К КУРСОВОЙ РАБОТЕ  
НА ТЕМУ:  
«Создание информационной системы для футбольного  
симулятора»

Студент ИУ7-63Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

К.Э. Ковалец  
(И. О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

Т.Н. Романова  
(И. О. Фамилия)

2022 г.

## РЕФЕРАТ

Данная курсовая работа представляет собой реализацию базы данных информационной системы для футбольного симулятора, а также создание Web-приложения, которое предоставляет интерфейс для взаимодействия с базой данных с возможностью создавать каждому отдельному пользователю свой индивидуальный состав футболистов, а также составлять рейтинги по различным параметрам. Пользователь может просматривать информацию о футболистах, тренерах и клубах, а также сортировать ее.

В качестве системы управления базой данных используется PostgreSQL. Платформой Web-разработки был выбран фреймворк ASP.NET Core. C# был выбран языком программирования.

Ключевые слова: футболист, состав, БД, СУБД, C#, PostgreSQL, ASP.NET Core, MVC.

Рассчетно-пояснительная записка к курсовой работе содержит 47 страниц, 34 иллюстрации, 3 таблицы, 10 источников.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 Аналитическая часть	7
1.1 Существующие решения . . . . .	7
1.2 Формализация задачи . . . . .	8
1.3 Формализация данных . . . . .	9
1.4 Формализация категорий пользователя . . . . .	10
1.5 Модели баз данных . . . . .	12
1.5.1 Дореляционные модели . . . . .	12
1.5.2 Реляционные модели . . . . .	13
1.5.3 Постреляционные модели . . . . .	13
1.6 Вывод . . . . .	13
2 Конструкторская часть	14
2.1 Проектирование базы данных . . . . .	14
2.2 Триггеры базы данных . . . . .	16
2.3 Функции базы данных . . . . .	18
2.4 Роли базы данных . . . . .	21
2.5 Вывод . . . . .	22
3 Технологическая часть	23
3.1 Средства реализации ПО . . . . .	23
3.2 Архитектура приложения . . . . .	24
3.3 Выбор СУБД . . . . .	24
3.4 Реализация триггеров . . . . .	25
3.5 Реализация функции . . . . .	26
3.6 Реализация ролевых моделей . . . . .	29
3.7 Демонстрация работы Web-приложения . . . . .	32
3.7.1 Демонстрация возможностей гося . . . . .	32
3.7.2 Демонстрация возможностей авторизованного пользова- теля . . . . .	36
3.7.3 Демонстрация возможностей администратора . . . . .	40
3.8 Вывод . . . . .	42

4	Исследовательская часть	43
4.1	Описание эксперимента . . . . .	43
4.2	Результаты эксперимента . . . . .	44
4.3	Вывод . . . . .	45
	ЗАКЛЮЧЕНИЕ	46
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	47

## ВВЕДЕНИЕ

В настоящее время огромное количество людей тратит много времени и денег в компьютерных играх. Одной из самых популярных игр и однозначным лидером в жанре спортивных симуляторов является FIFA 22 — 29-ая по счёту компьютерная игра из серии FIFA, разработанная компаниями EA Vancouver под издательством Electronic Arts.

Electronic Arts представила статистику футбольного симулятора FIFA 22 на 22-й день с момента релиза. Жители более 200 стран потратили на FIFA 22 почти 46 трлн минут, что равносильно 88 годам непрерывной игры. За это время пользователи провели более 2 млрд матчей и забили 5 млрд голов. В «Карьере» FIFA 22 игроки создали более 3 млн клубов.

По данным издания Eurogamer, за прошлый финансовый год EA заработала на режимах Ultimate Team в FIFA и NHL — \$ 1,62 млрд. С каждым годом выручка от внутриигровых покупок в спортивных симуляторах только увеличивается. В 2020 году EA заработала \$ 1,491 млрд, а в 2019-м прибыль составила \$ 1,369 млрд [1].

Учитывая все вышеуказанное, можно сделать вывод о том, что интерес к футбольным симуляторам огромный. Создание информационной системы для одного из них поможет пользователям организовать свою команду. Благодаря оригинальным подборкам, игроки смогут подобрать нужного футболиста или тренера.

Целью курсовой работы является разработка базы данных для информационной системы футбольного симулятора. Для достижения поставленной цели необходимо выполнить следующие задачи:

- проанализировать существующие решения;
- формализовать задачу и определить необходимый функционал;
- рассмотреть модели баз данных и выбрать подходящую;
- проанализировать существующие СУБД и выбрать нужную;
- спроектировать и разработать БД;
- спроектировать и разработать WEB-приложение.

## 1 Аналитическая часть

### 1.1 Существующие решения

Так как существует большой интерес к футбольным симуляторам, на рынке уже существуют решения, предоставляющие различный функционал для любителей таких игр.

Рассмотрим только самые популярные из них, такие как:

- futbin;
- futwiz;
- futhead;
- fifa companion.

Выделим следующие критерии для сравнения выбранных решений:

1. возможность поиска игроков по заданным параметрам;
2. возможность просмотра текущей цены игроков;
3. возможность собрать собственный состав футболистов;
4. возможность просмотра рейтинга составов других игроков;
5. наличие информации об игроках;
6. наличие информации о тренерах;
7. наличие информации о клубах.

Результаты сравнения выбранных решений по заданным критериям представлены в таблице 1.1.

Таблица 1.1 – Сравнение существующих решений

Критерий	Futbin	Futwiz	Futhead	FIFA Companion
1	+	+	+	+
2	+	+	-	-
3	+	+	+	+
4	-	-	-	+
5	+	+	+	+
6	-	-	-	+
7	-	-	+	-

Таким образом, ни одно из четырех рассмотренных решений не удовлетворяет всем семи критериям сравнения. Также стоит отметить, что все они являются зарубежными, отечественные аналоги либо отсутствуют, либо слишком непопулярны.

## 1.2 Формализация задачи

В ходе выполнения курсовой работы необходимо разработать базу данных для хранения информации о футболистах, тренерах, клубах для информационной системы футбольного симулятора. Также необходимо спроектировать и разработать Web-приложение, которое будет предоставлять интерфейс для взаимодействия с базой данных с возможностью создавать каждому отдельному пользователю свой индивидуальный состав футболистов, а также составлять рейтинги по различным параметрам.

Необходимо предусмотреть возможность добавления новых футболистов в базу данных, а также удаление уже существующих. Требуется реализовать функциональность для разных категорий пользователей, каждый из которых получит свой определённый набор прав.

### 1.3 Формализация данных

Разрабатываемая база данных для информационной системы футбольного симулятора должна содержать информацию о футболистах, тренерах, клубах, пользователях и их составах.

Таблица 1.2 – Категории данных в БД и информация о них

Категория	Информация
Футболист	Id футболиста, Id клуба, фамилия, рейтинг, страна, цена
Тренер	Id тренера, фамилия, страна
Клуб	Id клуба, название, страна, год основания
Пользователь	Id пользователя, логин, пароль категория пользователя
Состав	Id состава, Id тренера, название, рейтинг
Связь футболист-состав	Id связи, Id состава, Id футболиста

Также на рисунке 1.1 изображена ER-диаграмма системы в нотации Чена.

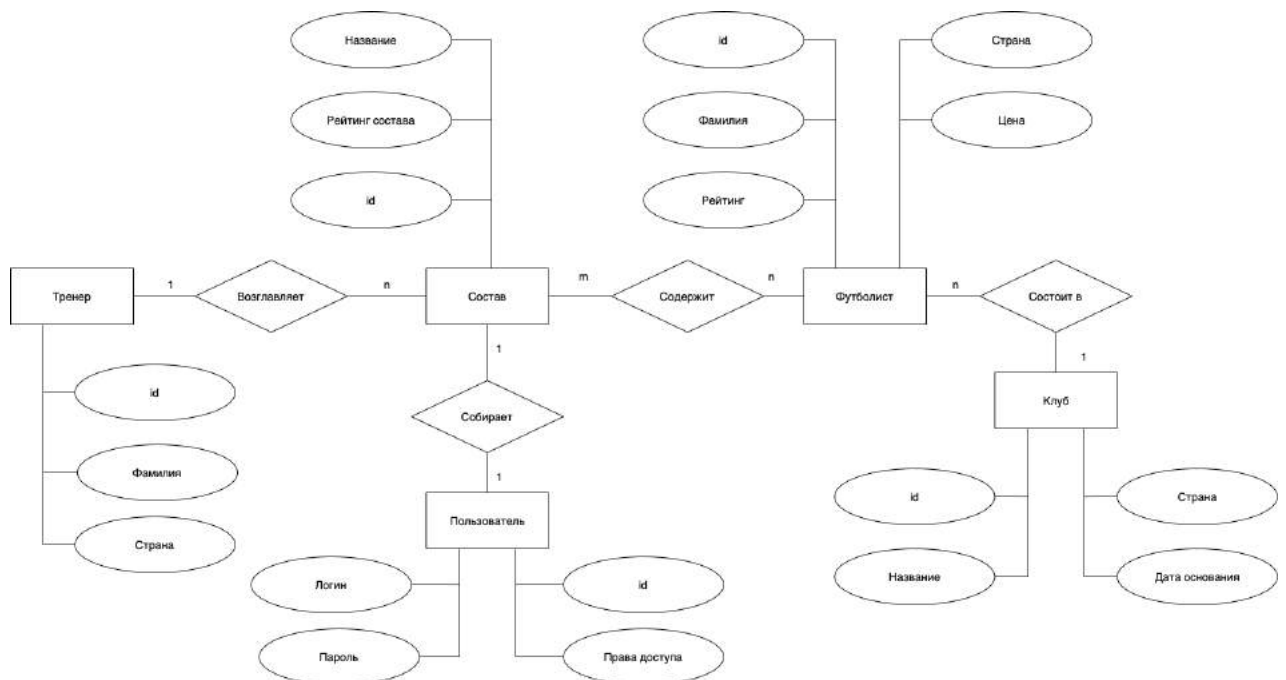


Рисунок 1.1 – ER-диаграмма в нотации Чена



## 1.4 Формализация категорий пользователя

Для взаимодействия с Web-приложением было выделено три категории пользователя: гость, авторизованный пользователь и администратор.

Гость (неавторизованный пользователь) сможет просматривать базовую информацию о футболистах, тренерах и клубах. Он сможет воспользоваться подборками для поиска нужных ему игроков. При просмотре списка клубов, гость сможет изучить игроков выбранного клуба.

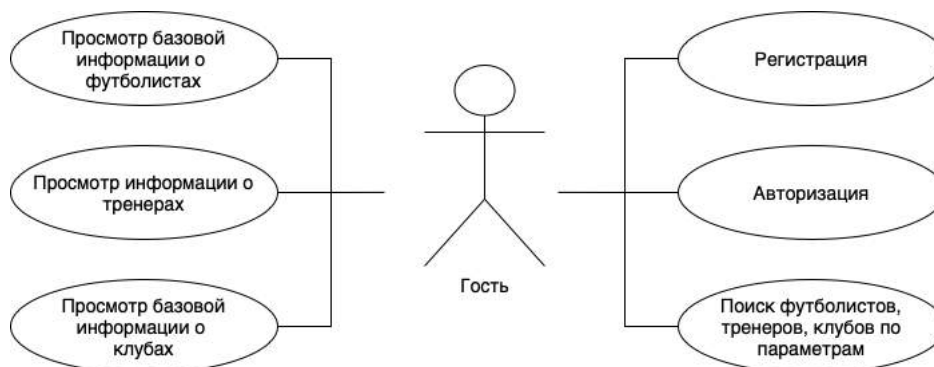


Рисунок 1.2 – Use-case диаграмма для неавторизованного пользователя

Для получения дополнительного функционала пользователю необходимо зарегистрироваться. Авторизованный пользователь в качестве дополнительного функционала получит возможность просматривать актуальную цену футболистов на рынке, а также организовывать свою команду путем добавления понравившихся игроков и тренера в свой состав. Также он сможет просматривать список составов других игроков.

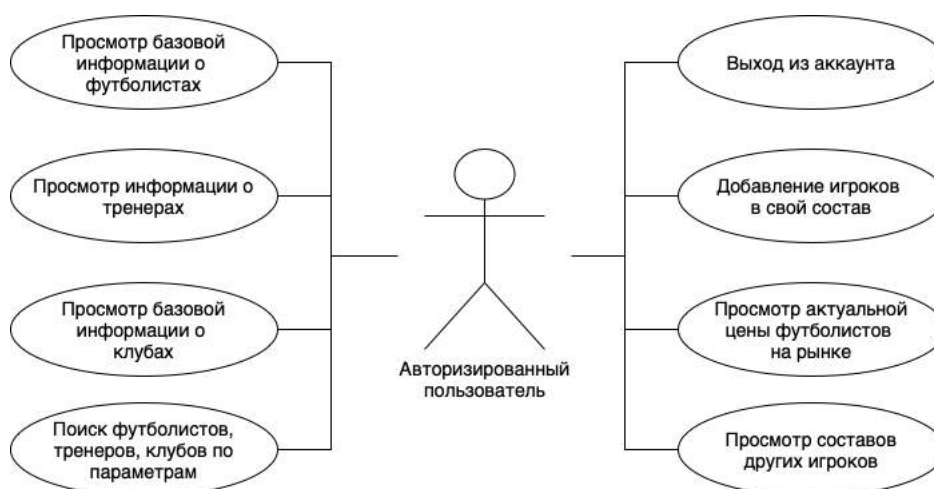


Рисунок 1.3 – Use-case диаграмма для авторизованного пользователя

Администратор — это авторизованный пользователь, который имеет права на изменение информации в базе данных (он может добавлять или удалять футболистов). Также администратор получит права просматривать список других пользователей Web-сайта, а также изменять их права доступа. В отличие от авторизованного пользователя, который может только просматривать список составов других игроков, администратор имеет права на просмотр футболистов других составов.

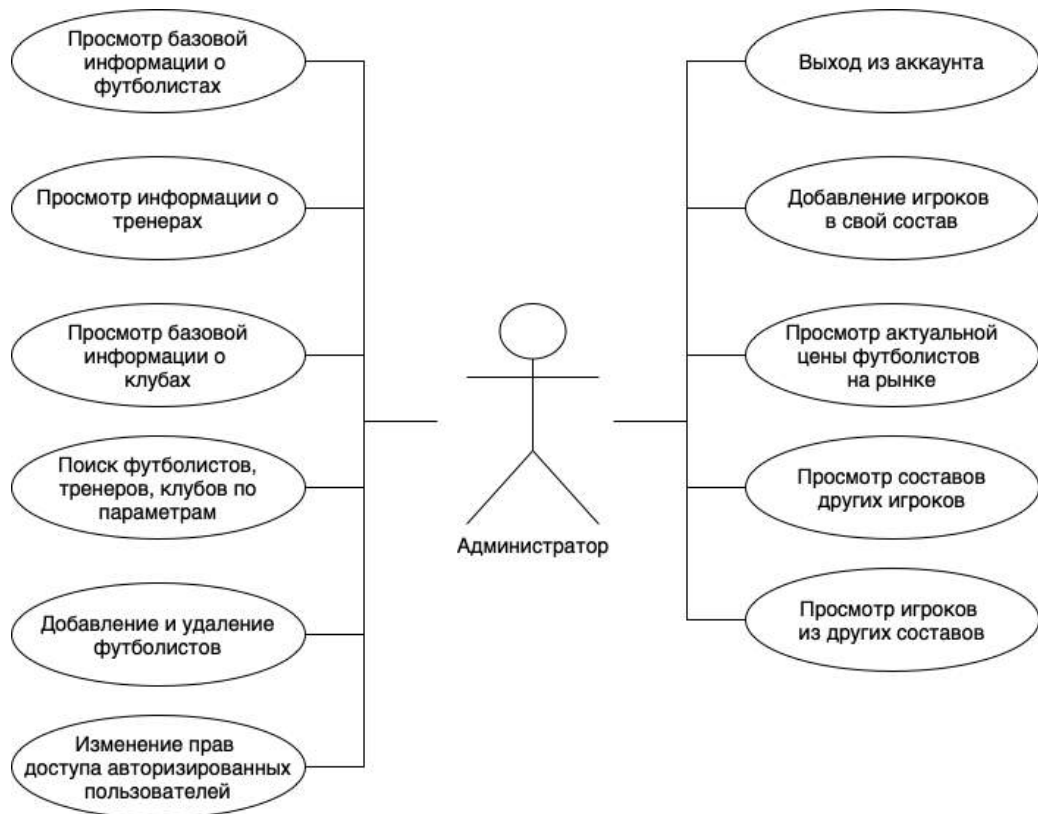


Рисунок 1.4 – Use-case диаграмма для администратора

## 1.5 Модели баз данных

База данных — это любая совокупность информации, которая была систематизирована для быстрого поиска с помощью компьютера. Базы данных используются для хранения, изменения, удаления и доступа к любым данным. Система управления базами данных (СУБД) использует запросы для извлечения информации из базы данных.

Модель базы данных определяет логическую структуру базы данных и то, каким образом данные могут храниться, организовываться и обрабатываться.

Существует три основных типа модели баз данных:

- дореляционные;
- реляционные;
- постреляционные.

### 1.5.1 Дореляционные модели

К дореляционным моделям баз данных относятся иерархическая и сетевая модели.

В иерархической модели записи связаны по принципу «родитель-потомок», при этом каждый родитель может быть связан с более чем одним дочерним элементом, но каждый дочерний элемент связан только с одним родителем. У такой модели есть проблемы с отношениями «многие ко многим». Кроме того, в иерархической модели добавление новых связей может привести к массовым изменениям существующей структуры.

Сетевая модель базы данных позволяет каждой записи иметь несколько родительских и несколько дочерних записей. Эта модель была создана на замену иерархической модели, так как она может обрабатывать отношения «многие ко многим».

Самая большая проблема с дореляционными моделями баз данных заключалась в отсутствии структурной независимости — внести структурные изменения в базу данных было очень сложно.

### 1.5.2 Реляционные модели

В реляционных моделях [2] данные организованы в набор двумерных взаимосвязанных таблиц. Каждая из которых представляет собой набор столбцов и строк, где столбец представляет атрибуты сущности, а строки представляют записи. Использование таблиц для хранения данных обеспечило простой и эффективный способ хранения структурированной информации, доступа к ней, а также легкую сортировку.

### 1.5.3 Постреляционные модели

Постреляционные базы данных [3] реализуют схему с использованием нереляционной модели данных. Такие БД поддерживают логическую модель данных, ориентированную на обработку транзакций. Постреляционные базы данных включают хранилища данных с ключевыми значениями, сетевые и графовые базы данных, а также базы данных, ориентированные на документы. Недостатком такой модели является сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.

## 1.6 Вывод

В данном разделе были проанализованы аналоги информационной системы футбольного симулятора, ни одно из рассмотренных решений не удовлетворило всем семи выдвинутых критериям сравнения. Также были формализованы поставленная задача, данные, а также категории пользователя. Были рассмотрены модели баз данных. Для решения поставленной задачи была выбрана реляционная модель, так как для разрабатываемой базы данных для информационной системы футбольного симулятора важна целостность хранимых данных, а также простота хранения структурированной информации и ее сортировка.

## 2 Конструкторская часть

### 2.1 Проектирование базы данных

На рисунке 2.1 изображена диаграмма разрабатываемой базы данных в соответствии с ER-диаграммой системы в нотации Чена на рисунке 1.1.

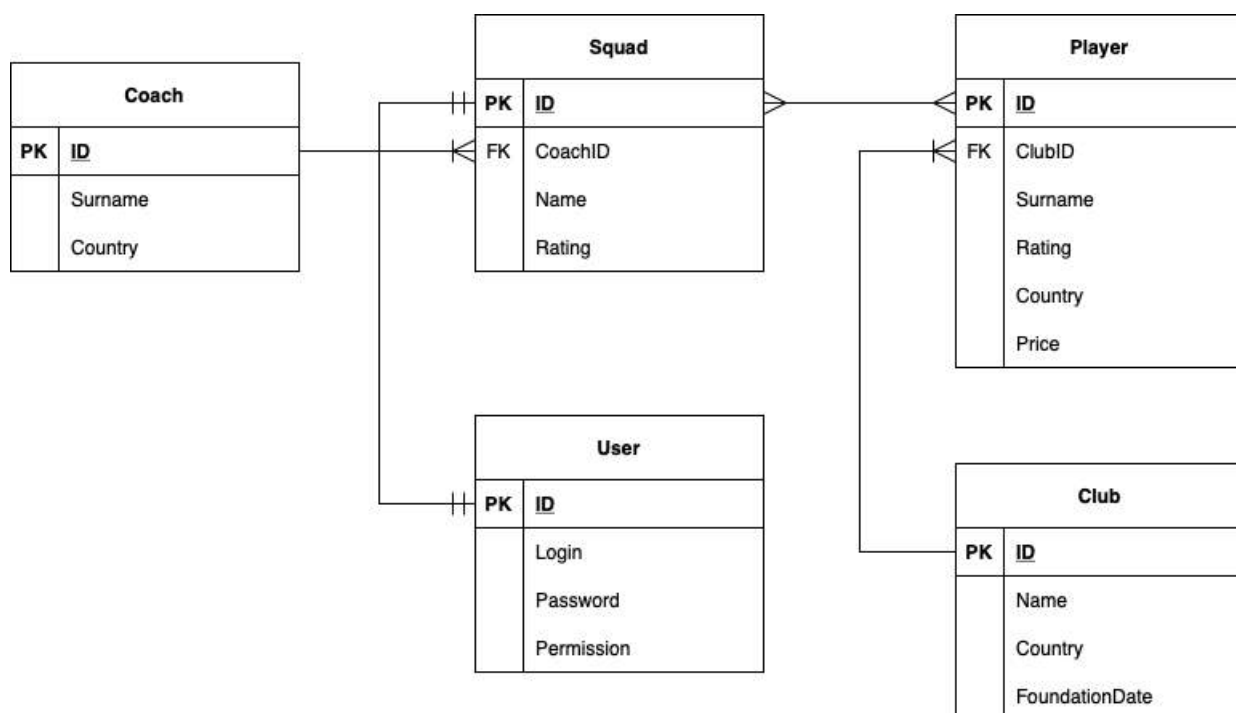


Рисунок 2.1 – Диаграмма разрабатываемой базы данных

Таблица Player содержит информацию о футболистах и имеет следующие поля:

- Id – первичный ключ таблицы (unsigned);
- ClubId – идентификатор клуба (unsigned);
- Surname – фамилия футболиста (string);
- Rating – рейтинг футболиста (unsigned);
- Country – страна футболиста (string);
- Price – цена футболиста (unsigned).

Таблица Coach содержит информацию о тренерах и имеет следующие поля:

- Id – первичный ключ таблицы (unsigned);
- Surname – фамилия тренера (string);
- Country – страна тренера (string).

Таблица Club содержит информацию о футбольных клубах и имеет следующие поля:

- Id – первичный ключ таблицы (unsigned);
- Name – название футбольного клуба (string);
- Country – страна расположения футбольного клуба (string);
- FoundationDate – год основания футбольного клуба (unsigned).

Таблица Squad содержит информацию о составах игроков и имеет следующие поля:

- Id – первичный ключ таблицы (unsigned);
- CoachId – идентификатор тренера (unsigned);
- Name – название состава (string);
- Rating – рейтинг состава (средний рейтинг игроков в составе) (unsigned).

Таблица User содержит информацию об игроках и имеет следующие поля:

- Id – первичный ключ таблицы (unsigned);
- Login – логин пользователя (string);
- Password – пароль пользователя (string);
- Permission – права доступа пользователя (string).

Таблица `SquadPlayer` содержит информацию о том, какие игроки в какие клубы были добавлены, имеет следующие поля:

- `Id` – первичный ключ таблицы (`unsigned`);
- `SquadId` – идентификатор состава (`unsigned`);
- `PlayerId` – идентификатор футболиста (`unsigned`).

## 2.2 Триггеры базы данных

При создании базы данных были определены два триггеры. Один срабатывает после добавления в таблицу `SquadPlayer` новой записи, а другой после удаления старой. Эти два триггера отвечают за обновление рейтинга состава, который определяется средним рейтингом футболистов в нем. Так при добавление игрока в состав вызовется триггер `insertRatingTrigger`, который вызовет функцию для пересчета рейтинга состава. Аналогично при удалении игрока вызовется триггер `deleteRatingTrigger`.

На рисунках 2.2 – 2.3 показаны схемы функций для обновления состава.

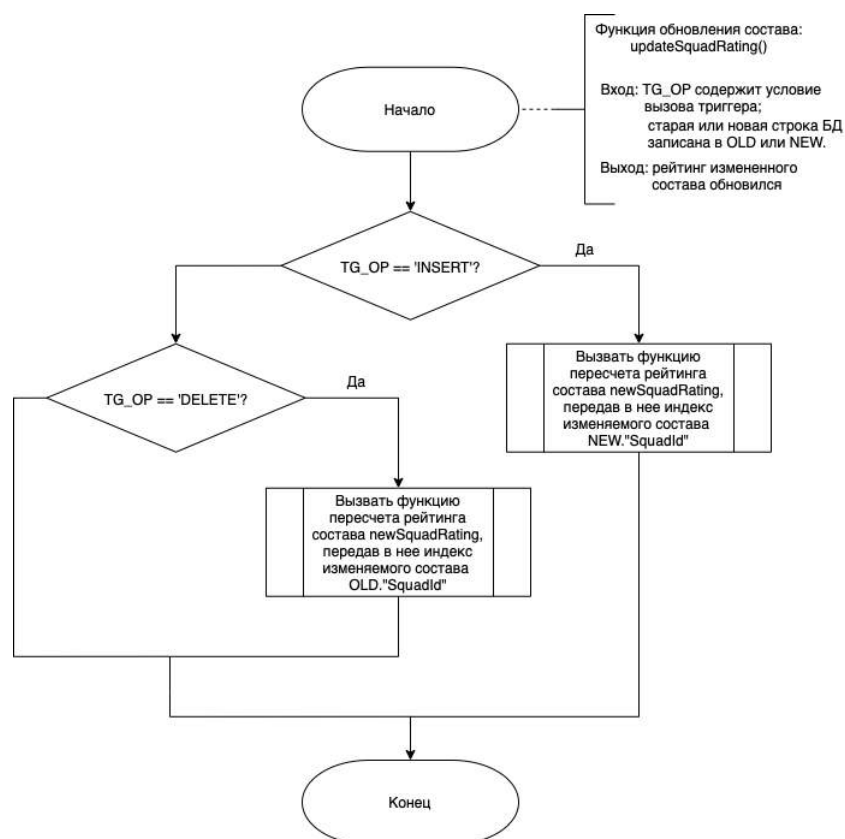


Рисунок 2.2 – Схема функции обновления состава

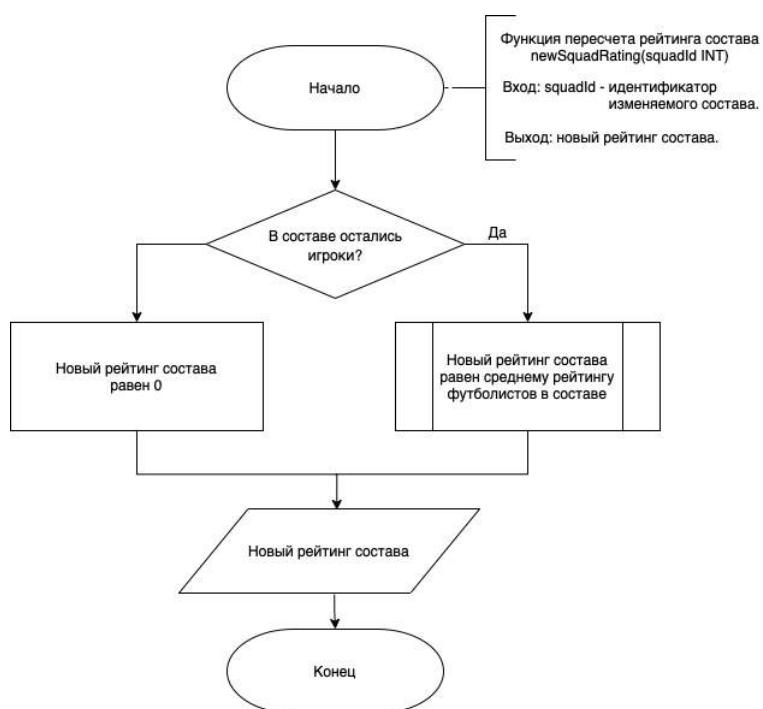


Рисунок 2.3 – Схема функции получения нового рейтинга состава



## 2.3 Функции базы данных

Кроме триггеров при создании базы данных была определена функция. Она отвечает за поиск футбольных клубов по заданным параметрам, таким как название футбольного клуба, страна, в которой он располагается, а также минимальная и максимальная даты основания клуба. На рисунке 2.4 показана схема функции поиска клубов по заданным параметрам.

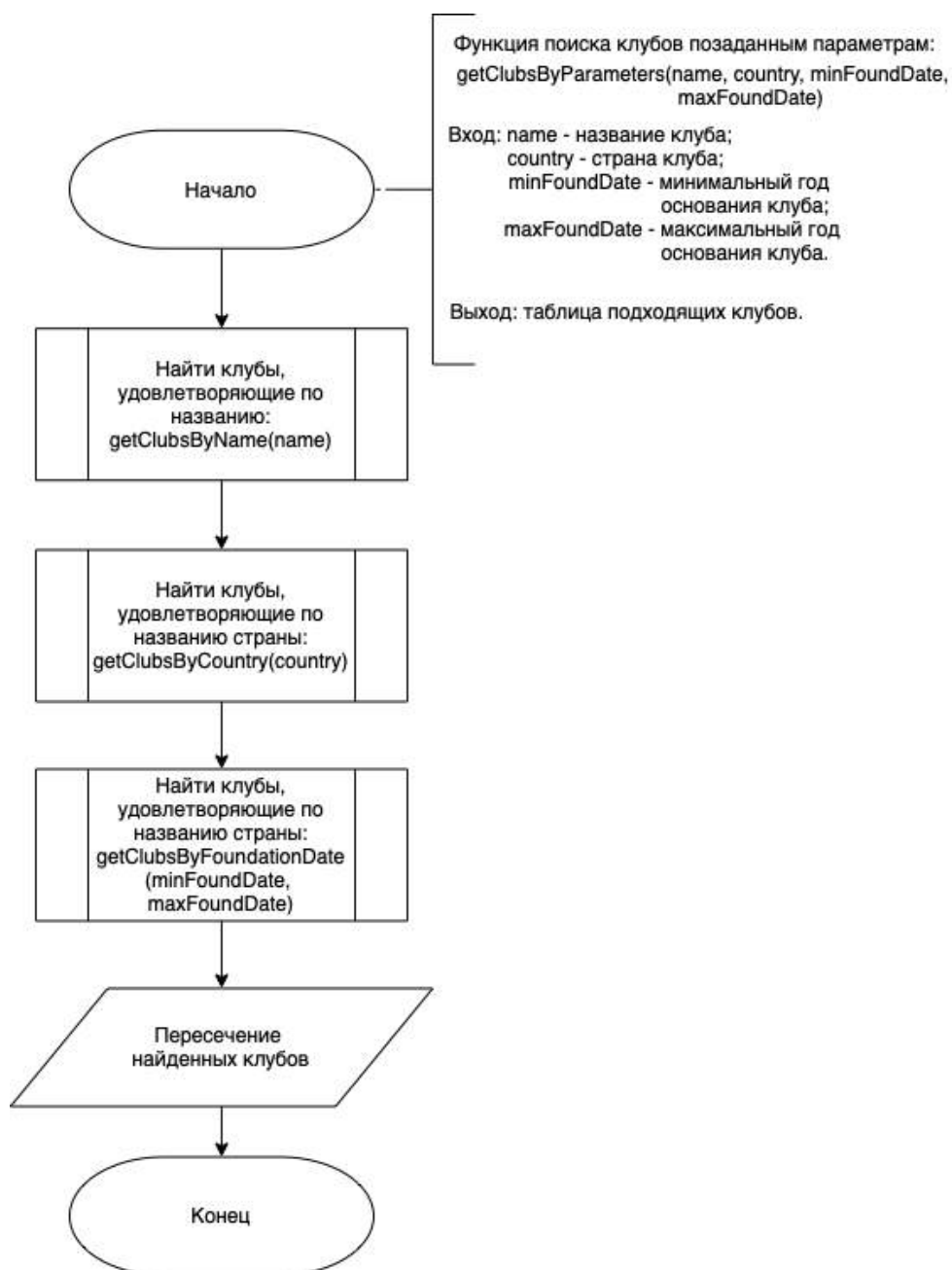


Рисунок 2.4 – Схема функции поиска клубов по заданным параметрам

На рисунках 2.5 – 2.6 показаны схемы функций поиска клубов по названию и стране.

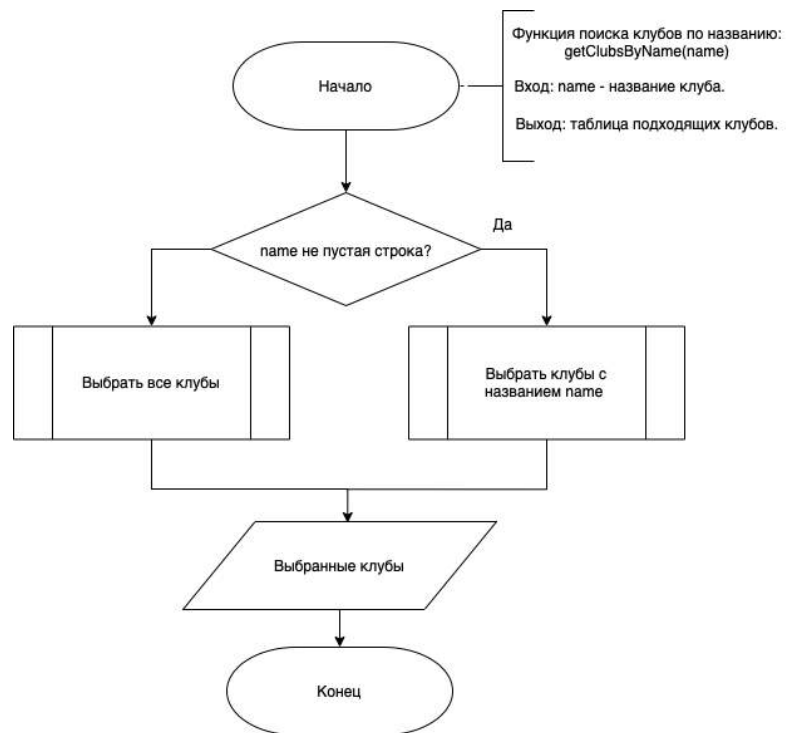


Рисунок 2.5 – Схема функции поиска клубов по имени

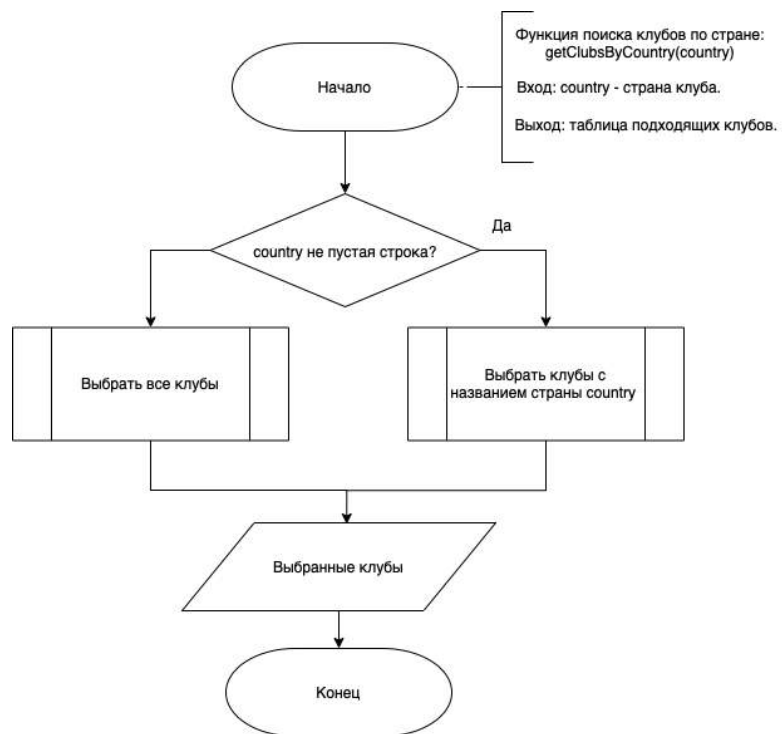


Рисунок 2.6 – Схема функции поиска клубов по названию страны

На рисунке 2.4 показана схема функции поиска клубов по дате основания.

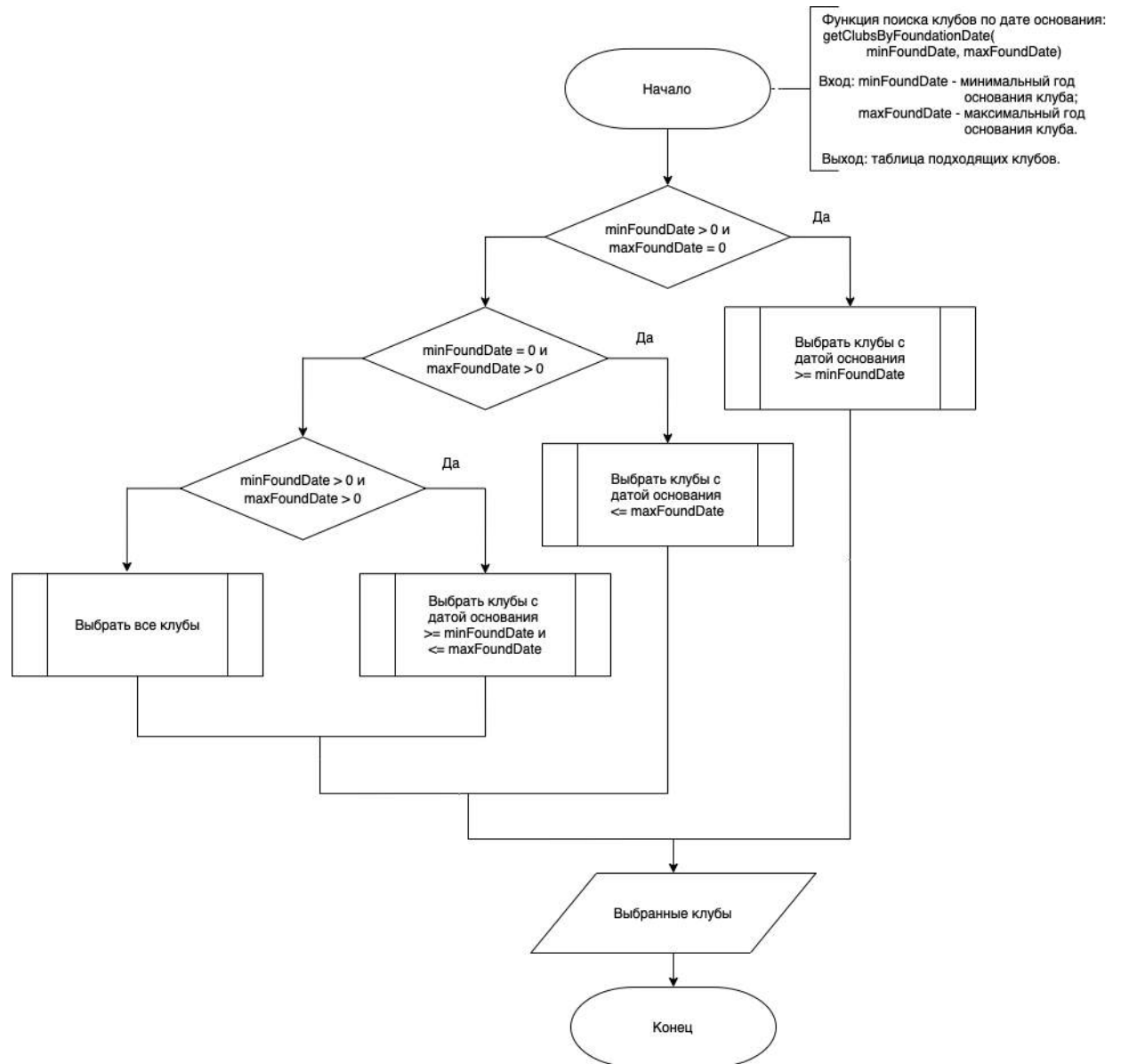


Рисунок 2.7 – Схема функции поиска клубов по дате основания

## 2.4 Роли базы данных

В аналитической части для взаимодействия с Web-приложением было выделено три категории пользователя: гость, авторизованный пользователь и администратор. Для каждой категории требуется создать отдельную роль в базе данных, которые будут называться аналогично.

1. Гость (неавторизованный пользователь) сможет просматривать базовую информацию о футболистах, тренерах и клубах, следовательно он должен иметь возможность просматривать таблицы Player, Coach, Club. Также гость сможет зарегистрироваться или авторизоваться на сайте, а значит ему понадобятся права на просмотр и изменение (только на добавление записей) таблиц User, Squad. Прав на добавление информации в другие таблицы, а также на удаление записей из всех таблиц неавторизованный пользователь иметь не будет.
2. Авторизованный пользователь, помимо привелегий гостя, сможет просматривать информацию о составах других игроков, а также добавлять понравившихся футболистов к себе в состав, следовательно он дополнительно должен иметь права на просмотр таблиц Squad и SquadPlayer, то есть авторизованный пользователь должен иметь права на просмотр всех таблиц в разрабатываемой базе данных. Так как пользователь сможет изменять свой состав, он должен иметь права на добавление и удаление записей из таблицы SquadPlayer. У каждого состава есть свой рейтинг, который отвечает за средний рейтинг игроков в нем. При добавлении или удалении футболиста из него рейтинг должен пересчитываться, следовательно авторизованный пользователь должен иметь права на обновление таблицы Squad.
3. Администратор имеет все права на все таблицы, так как он должен иметь возможность менять категорию пользователя у других игроков, добавлять новых или удалять старых футболистов из базы данных. Администратору разрешено все.

## 2.5 Вывод

В данном разделе было произведено проектирование базы данных, в ходе чего была построена диаграмма разрабатываемой базы данных, описаны все поля всех таблиц, а также определены два триггера и одна хранимая функция. На основе категорий пользователя были выделены три ролевые модели на уровне базы данных: гость, авторизованный пользователь и администратор. Подробно были описаны права доступа каждой из них.

## 3 Технологическая часть

### 3.1 Средства реализации ПО

Для написания Web-приложения был выбран язык программирования C# [4]. Такой решение обусловлено следующими причинами.

- Главной причиной выбора C# стала его Web-поддержка.
- Данный язык поддерживает объектно-ориентированную парадигму программирования, что было обязательным требованием для выбора языка разработки.
- C# многим похож на ранее знакомые мне языки программирования, такие как C++ и C, что также помогло склонить чашу весов в пользу C#.

В качестве среды разработки был выбран Visual Studio [5]. Такой решение обусловлено следующими причинами.

- Главным требованием при выборе среды разработки была Web-поддержка, которая имеется у Visual Studio.
- VS является бесплатным программным обеспечением, что также было важным требованием при выборе среды.
- Visual Studio доступна на macOS.

При выборе платформы Web-разработки рассматривались два фреймворка на платформе .NET. Это ASP.NET [6] и ASP.NET Core [7]. В пользу выбора второго из них повлияли следующие причины.

- Данная платформа имеет большую производительность по сравнению с ASP.NET.
- ASP.NET Core предназначена для Windows, macOS и Linux, в свою очередь как ASP.NET только для Windows.

Другие платформы не рассматривались, так как ранее в качестве языка программирования был выбран C#, а средой разработки Visual Studio.

## 3.2 Архитектура приложения

За основу архитектуры разрабатываемого Web-приложения была взята модель MVC (MODEL, VIEW, CONTROLLER). Эта модель помогает обеспечить разделение между бизнес-логикой программного обеспечения и отображением.

Фреймворк MVC включает в себя следующие три компонента.

- Модель – управляет данными и бизнес-логикой.
- Представление – используется для всей логики пользовательского интерфейса приложения.
- Контроллер – обеспечивает взаимосвязь между представлениями и моделью, поэтому он действует как посредник.

ASP.NET Core поддерживает модель разработки MVC. ASP.NET Core MVC framework [8] – это легкий, легко тестируемый фреймворк представления, который интегрирован с существующими ASP.NET Core функциями.

## 3.3 Выбор СУБД

В аналитическом разделе при выборе типа модели баз данных была выбрана реляционная модель, следовательно выбор СУБД будет производиться для реляционной модели баз данных. Рассмотрим только самые популярные из них: MySQL, Oracle, PostgreSQL [9], Microsoft SQL Server.

Выделим следующие критерии для сравнения выбранных СУБД:

1. бесплатное распространение СУБД, что будет главным требованием при выборе;
2. наличие подробной документации, которая поможет разобраться во всех возникших проблемах;
3. наличие высокого уровня оптимизации СУБД, что будет отличным плюсом при выборе;
4. опыт работы с СУБД, не является основным требованием, но очень желательным.

Результаты сравнения выбранных СУБД по заданным критериям представлены в таблице 3.1.

Таблица 3.1 – Сравнение выбранных СУБД

Критерий	MySQL	Oracle	PostgreSQL	Microsoft SQL Server
1	+	-	+	-
2	+	+	+	+
3	+	-	+	-
4	-	-	+	-

По результатам сравнения в качестве СУБД для реляционной модели баз данных был выбран PostgreSQL, так как он единственный удовлетворил всем четырем критериям сравнения.

### 3.4 Реализация триггеров

Триггеры обновления рейтинга состава после добавления или удаления футболиста представлены в листинге 3.1.

Листинг 3.1 — Триггеры обновления рейтинга состава

```

1  -- Получение нового рейтинга состава
2
3  CREATE OR REPLACE FUNCTION newSquadRating(squadId INT)
4  RETURNS DECIMAL
5  AS $$
6      BEGIN
7          IF ((
8              SELECT COUNT(*)
9              FROM public."SquadPlayer"
10             WHERE "SquadId" = squadId) > 0) THEN
11
12             RETURN (
13                 SELECT AVG("Rating")
14                 FROM public."SquadPlayer" AS sp
15                 JOIN public."Player" AS pl ON sp."PlayerId" = pl."Id"
16                 WHERE "SquadId" = squadId
17             );
18         ELSE
19             RETURN 0;
20         END IF;
21     END;

```



```

22  $$ LANGUAGE plpgsql;
23
24  -- Обновляет рейтинг состава после добавления или удаления футболиста (реализация с
   ↪ помощью триггеров)
25
26  CREATE OR REPLACE FUNCTION updateSquadRating()
27  RETURNS TRIGGER
28  AS $$
29      BEGIN
30          IF (TG_OP = 'INSERT') THEN
31              UPDATE public."Squad"
32              SET "Rating" = newSquadRating(NEW."SquadId")
33              WHERE "Id" = NEW."SquadId";
34
35              RETURN NEW;
36
37              ELSIF (TG_OP = 'DELETE') THEN
38                  UPDATE public."Squad"
39                  SET "Rating" = newSquadRating(OLD."SquadId")
40                  WHERE "Id" = OLD."SquadId";
41
42                  RETURN OLD;
43              END IF;
44      END;
45  $$ LANGUAGE plpgsql;
46
47
48  CREATE TRIGGER insertRatingTrigger AFTER INSERT ON public."SquadPlayer"
49  FOR EACH ROW EXECUTE PROCEDURE updateSquadRating();
50
51  CREATE TRIGGER deleteRatingTrigger AFTER DELETE ON public."SquadPlayer"
52  FOR EACH ROW EXECUTE PROCEDURE updateSquadRating();

```

### 3.5 Реализация функции

Функция поиска клубов по заданным параметрам представлены в листинге 3.2.

Листинг 3.2 — Функция поиска клубов по заданным параметрам

```

1  -- Поиск клубов по названию
2
3  CREATE OR REPLACE FUNCTION getClubsByName(name TEXT)
4  RETURNS TABLE (
5      _clubId INTEGER,
6      _name TEXT,

```

```

7      _country TEXT,
8      _foundationDate BIGINT
9  )
10 AS $$
11 BEGIN
12     IF (name IS NOT NULL) THEN
13         RETURN QUERY
14         SELECT *
15         FROM public."Club"
16         WHERE "Name" = name;
17     ELSE
18         RETURN QUERY
19         SELECT *
20         FROM public."Club";
21     END IF;
22 END;
23 $$ LANGUAGE plpgsql;
24
25 -- Поиск клубов по названию страны
26
27 CREATE OR REPLACE FUNCTION getClubsByCountry(country TEXT)
28 RETURNS TABLE (
29     _clubId INTEGER,
30     _name TEXT,
31     _country TEXT,
32     _foundationDate BIGINT
33 )
34 AS $$
35 BEGIN
36     IF (country IS NOT NULL) THEN
37         RETURN QUERY
38         SELECT *
39         FROM public."Club"
40         WHERE "Country" = country;
41     ELSE
42         RETURN QUERY
43         SELECT *
44         FROM public."Club";
45     END IF;
46 END;
47 $$ LANGUAGE plpgsql;
48
49 -- Поиск клубов по дате основания
50
51 CREATE OR REPLACE FUNCTION getClubsByFoundationDate(minFoundDate BIGINT, maxFoundDate
52 ↪ BIGINT)
53 RETURNS TABLE (

```

```

53     _clubId INTEGER,
54     _name TEXT,
55     _country TEXT,
56     _foundationDate BIGINT
57 )
58 AS $$
59 BEGIN
60     IF (minFoundDate > 0 AND maxFoundDate = 0) THEN
61         RETURN QUERY
62         SELECT *
63         FROM public."Club"
64         WHERE "FoundationDate" >= minFoundDate;
65
66     ELSEIF (minFoundDate = 0 AND maxFoundDate > 0) THEN
67         RETURN QUERY
68         SELECT *
69         FROM public."Club"
70         WHERE "FoundationDate" <= maxFoundDate;
71
72     ELSEIF (minFoundDate > 0 AND maxFoundDate > 0) THEN
73         RETURN QUERY
74         SELECT *
75         FROM public."Club"
76         WHERE "FoundationDate" >= minFoundDate AND
77             "FoundationDate" <= maxFoundDate;
78     ELSE
79         RETURN QUERY
80         SELECT *
81         FROM public."Club";
82     END IF;
83 END;
84 $$ LANGUAGE plpgsql;
85
86 -- Поиск клубов по заданным параметрам
87
88 CREATE OR REPLACE FUNCTION getClubsByParameters(_name TEXT, _country TEXT,
89 ↵ _minFoundDate BIGINT, _maxFoundDate BIGINT)
89 RETURNS TABLE (
90     Id INTEGER,
91     Name TEXT,
92     Country TEXT,
93     FoundationDate BIGINT
94 )
95 AS $$
96 BEGIN
97     RETURN QUERY
98     SELECT *

```

```

99         FROM getClubsByName(_name)
100     INTERSECT
101     SELECT *
102     FROM getClubsByCountry(_country)
103     INTERSECT
104     SELECT *
105     FROM getClubsByFoundationDate(_minFoundDate, _maxFoundDate);
106 END;
107 $$ LANGUAGE plpgsql;
108
109 -- Пример запроса с вызовом функции
110
111 SELECT *
112 FROM getClubsByParameters('FC Barcelona', 'Spain', 1800, 2000);

```

### 3.6 Реализация ролевых моделей

При создании базы данных были определены три роли: гость, авторизованный пользователь и администратор. Создание ролей, а также наделение их правами представлена в листинге 3.3.

Листинг 3.3 — Роли базы данных

```

1  -- роль Администратора
2
3  CREATE ROLE admin WITH
4      SUPERUSER
5      NOCREATEDB
6      CREATEROLE
7      NOINHERIT
8      NOREPLICATION
9      NOBYPASSRLS
10     CONNECTION LIMIT -1
11     LOGIN
12     PASSWORD 'admin';
13
14 -- права доступа
15
16 GRANT ALL PRIVILEGES
17     ON ALL TABLES IN SCHEMA public
18     TO admin;
19
20 -- роль Авторизованного пользователя
21
22 CREATE ROLE _user WITH

```

```

23     NOSUPERUSER
24     NOCREATEDB
25     NOCREATEROLE
26     NOINHERIT
27     NOREPLICATION
28     NOBYPASSRLS
29     CONNECTION LIMIT -1
30     LOGIN
31     PASSWORD 'user';
32
33     -- права доступа
34
35     GRANT SELECT
36         ON ALL TABLES IN SCHEMA public
37         TO _user;
38
39     GRANT INSERT
40         ON public."Squad",
41             public."SquadPlayer",
42             public."User"
43         TO _user;
44
45     GRANT DELETE
46         ON public."SquadPlayer"
47         TO _user;
48
49     GRANT UPDATE
50         ON public."Squad"
51         TO _user;
52
53     -- роль Гостя
54
55     CREATE ROLE guest WITH
56         NOSUPERUSER
57         NOCREATEDB
58         NOCREATEROLE
59         NOINHERIT
60         NOREPLICATION
61         NOBYPASSRLS
62         CONNECTION LIMIT -1
63         LOGIN
64         PASSWORD 'guest';
65
66     -- права доступа
67
68     GRANT SELECT
69         ON public."Club",

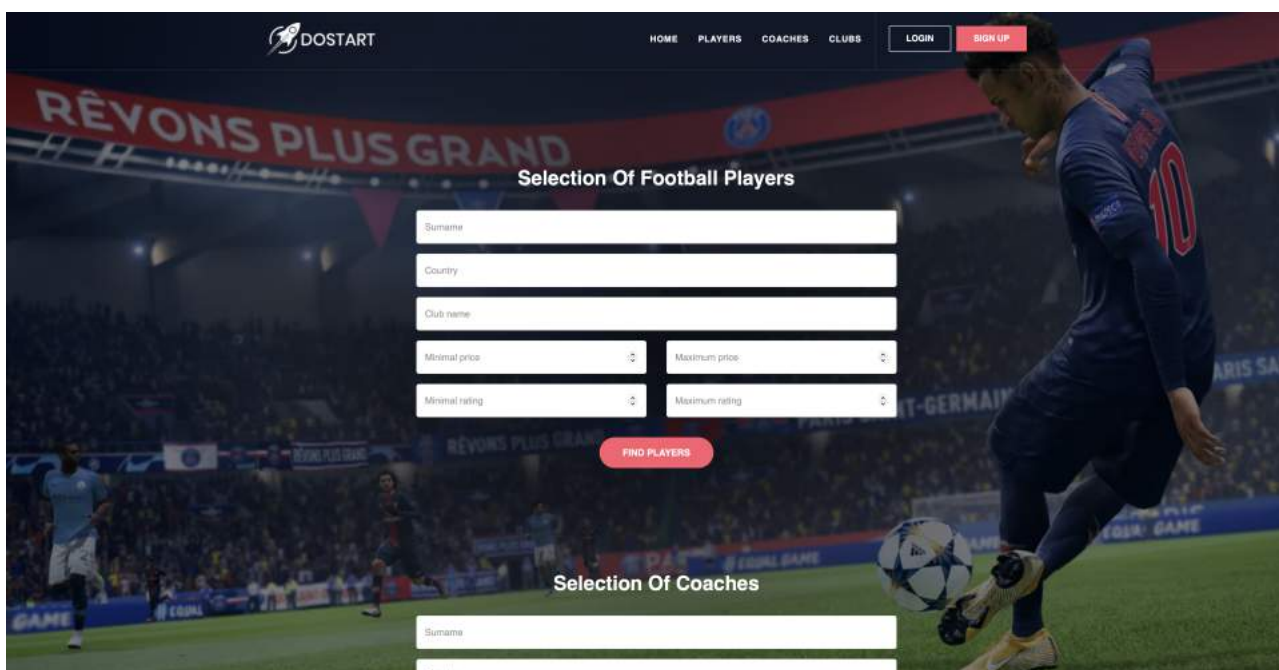
```

```
70         public."Coach",
71         public."Player",
72         public."Squad",
73         public."User"
74     TO guest;
75
76 GRANT INSERT
77     ON public."Squad",
78     public."User"
79     TO guest;
```

## 3.7 Демонстрация работы Web-приложения

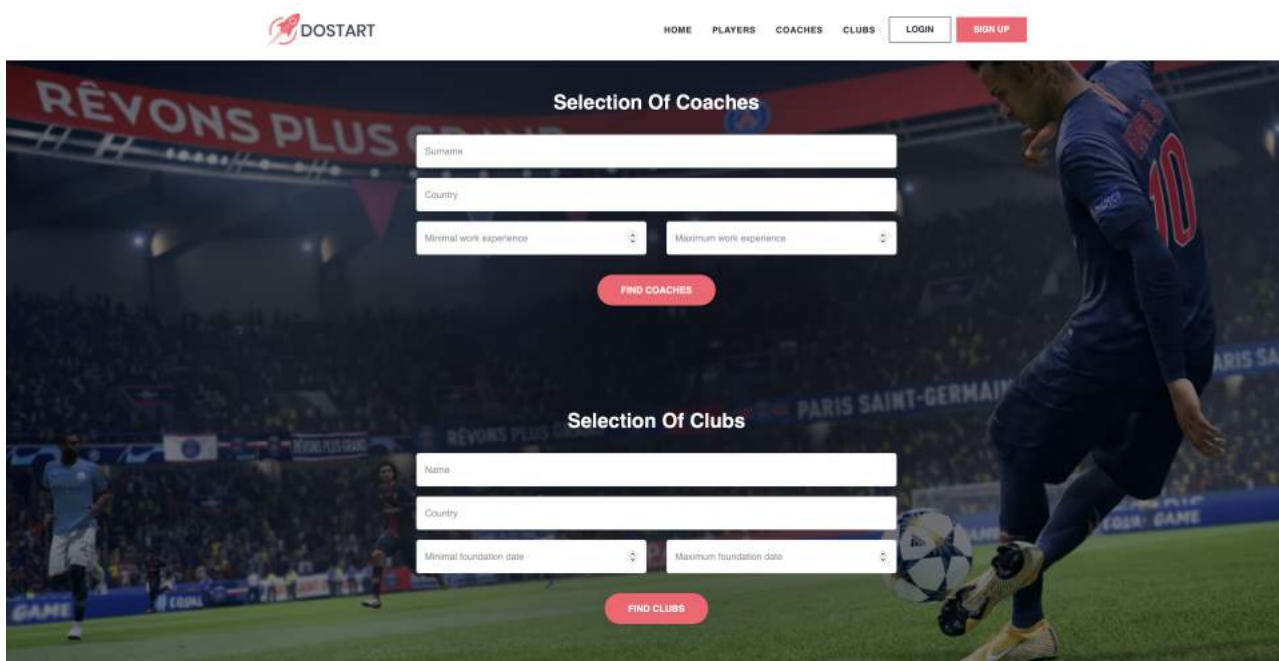
### 3.7.1 Демонстрация возможностей гося

Когда гость (неавторизованный пользователь) попадает на Web-сайт, его встречает домашняя страница (рисунки 3.1 – 3.2), на которой он может сделать поиск футболистов, тренеров или клубов по параметрам.



The screenshot shows the DOSTART website's home page for a guest user. The background is a soccer match scene with a player in a blue jersey (number 10) about to kick a ball. The navigation bar at the top includes 'HOME', 'PLAYERS', 'COACHES', 'CLUBS', 'LOGIN', and 'SIGN UP'. The main content area features two search sections. The first section, 'Selection Of Football Players', contains input fields for 'Surname', 'Country', 'Club name', 'Minimal price', 'Maximum price', 'Minimal rating', and 'Maximum rating', followed by a red 'FIND PLAYERS' button. The second section, 'Selection Of Coaches', has input fields for 'Surname' and 'Country', followed by a red 'FIND COACHES' button.

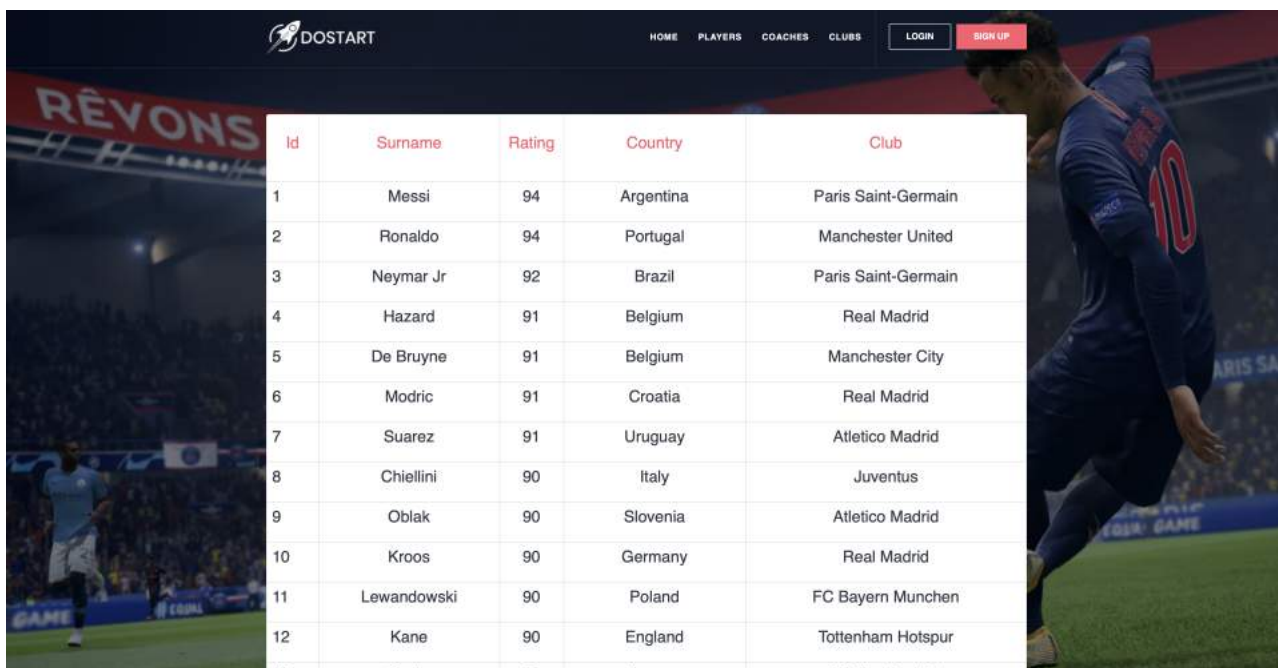
Рисунок 3.1 – Поиск футболистов по параметрам (для гостя)



This screenshot shows the same DOSTART website but with the 'Selection Of Clubs' form visible below the 'Selection Of Coaches' form. The 'Selection Of Coaches' form remains the same. The 'Selection Of Clubs' form includes input fields for 'Name', 'Country', 'Minimal foundation date', and 'Maximum foundation date', followed by a red 'FIND CLUBS' button. The background image and navigation bar are identical to the previous screenshot.

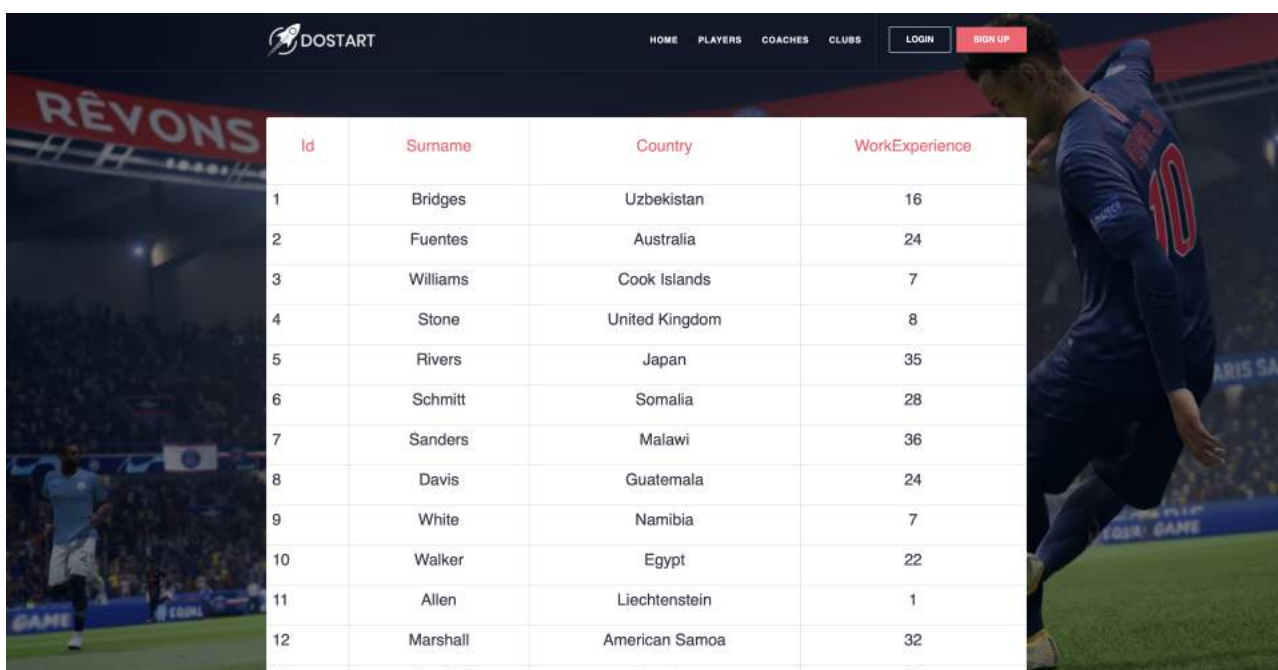
Рисунок 3.2 – Поиск тренеров и клубов по параметрам (для гостя)

Также гость может просматривать базовую информацию о футболистах (рисунок 3.3), тренерах (рисунок 3.4) и клубах (рисунок 3.5).



Id	Surname	Rating	Country	Club
1	Messi	94	Argentina	Paris Saint-Germain
2	Ronaldo	94	Portugal	Manchester United
3	Neymar Jr	92	Brazil	Paris Saint-Germain
4	Hazard	91	Belgium	Real Madrid
5	De Bruyne	91	Belgium	Manchester City
6	Modric	91	Croatia	Real Madrid
7	Suarez	91	Uruguay	Atletico Madrid
8	Chiellini	90	Italy	Juventus
9	Oblak	90	Slovenia	Atletico Madrid
10	Kroos	90	Germany	Real Madrid
11	Lewandowski	90	Poland	FC Bayern Munchen
12	Kane	90	England	Tottenham Hotspur

Рисунок 3.3 – Список всех футболистов (для гостя)

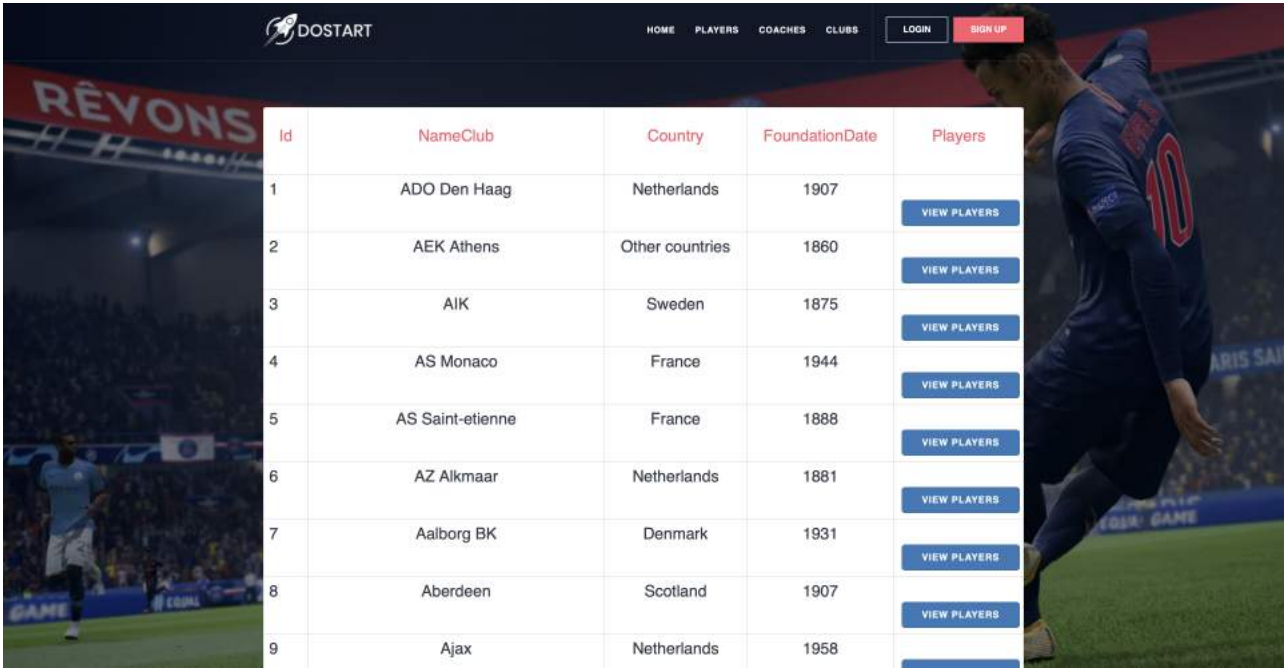


Id	Surname	Country	WorkExperience
1	Bridges	Uzbekistan	16
2	Fuentes	Australia	24
3	Williams	Cook Islands	7
4	Stone	United Kingdom	8
5	Rivers	Japan	35
6	Schmitt	Somalia	28
7	Sanders	Malawi	36
8	Davis	Guatemala	24
9	White	Namibia	7
10	Walker	Egypt	22
11	Allen	Liechtenstein	1
12	Marshall	American Samoa	32

Рисунок 3.4 – Список всех тренеров (для гостя)

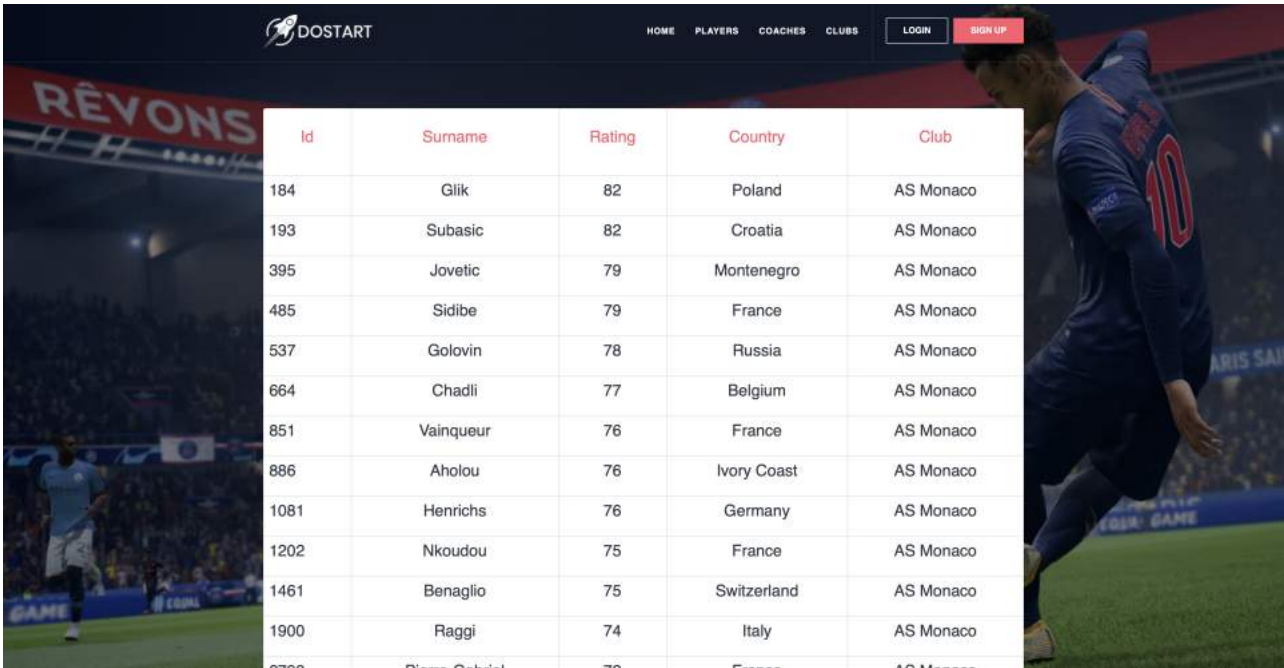


При просмотре списка клубов, гость может изучить игроков выбранного клуба (рисунок 3.6).



Id	NameClub	Country	FoundationDate	Players
1	ADO Den Haag	Netherlands	1907	<a href="#">VIEW PLAYERS</a>
2	AEK Athens	Other countries	1860	<a href="#">VIEW PLAYERS</a>
3	AIK	Sweden	1875	<a href="#">VIEW PLAYERS</a>
4	AS Monaco	France	1944	<a href="#">VIEW PLAYERS</a>
5	AS Saint-etienne	France	1888	<a href="#">VIEW PLAYERS</a>
6	AZ Alkmaar	Netherlands	1881	<a href="#">VIEW PLAYERS</a>
7	Aalborg BK	Denmark	1931	<a href="#">VIEW PLAYERS</a>
8	Aberdeen	Scotland	1907	<a href="#">VIEW PLAYERS</a>
9	Ajax	Netherlands	1958	<a href="#">VIEW PLAYERS</a>

Рисунок 3.5 – Список всех клубов (для гостя)



Id	Surname	Rating	Country	Club
184	Glik	82	Poland	AS Monaco
193	Subasic	82	Croatia	AS Monaco
395	Jovetic	79	Montenegro	AS Monaco
485	Sidibe	79	France	AS Monaco
537	Golovin	78	Russia	AS Monaco
664	Chadli	77	Belgium	AS Monaco
851	Vainqueur	76	France	AS Monaco
886	Aholou	76	Ivory Coast	AS Monaco
1081	Henrichs	76	Germany	AS Monaco
1202	Nkoudou	75	France	AS Monaco
1461	Benaglio	75	Switzerland	AS Monaco
1900	Raggi	74	Italy	AS Monaco

Рисунок 3.6 – Список футболистов AS Монасо (для гостя)

Также гость может авторизоваться (рисунок 3.7) или зарегистрироваться (рисунок 3.8).

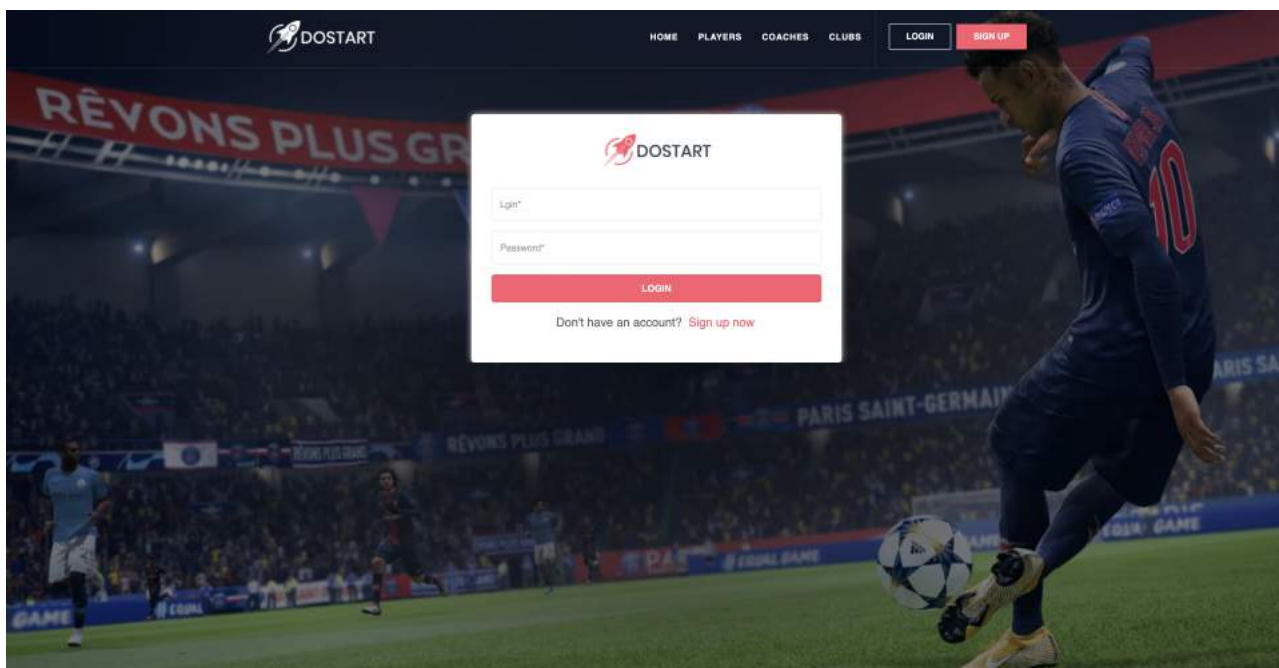


Рисунок 3.7 – Авторизация пользователя

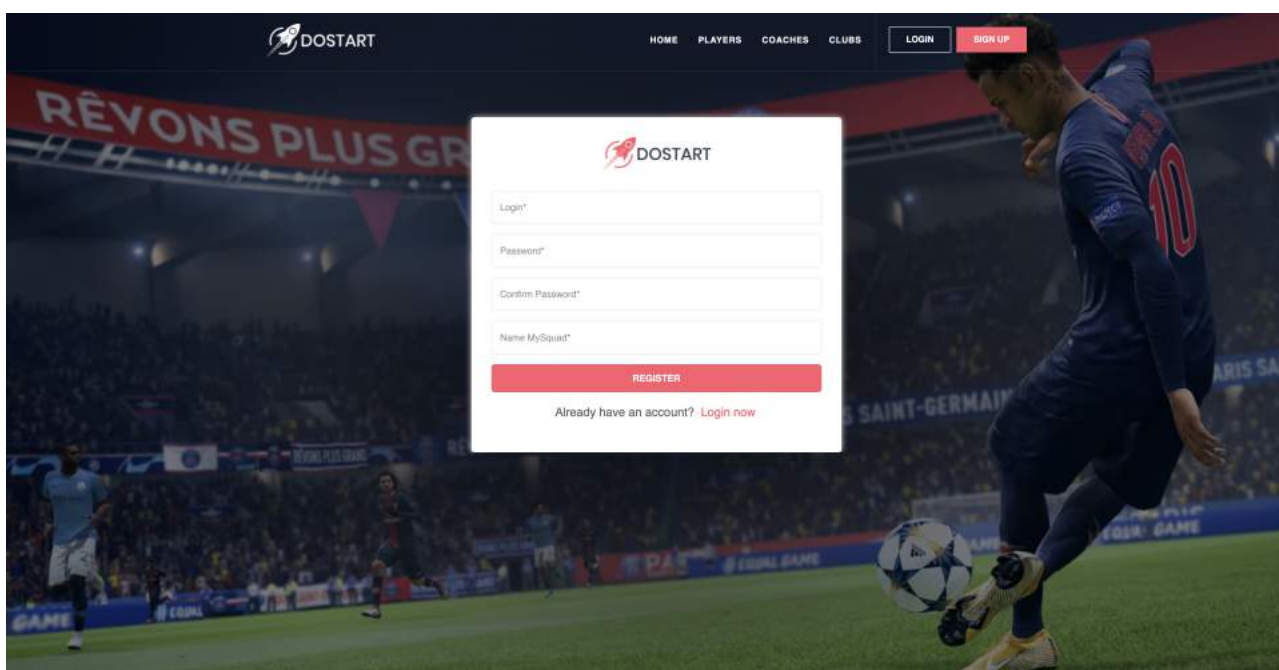
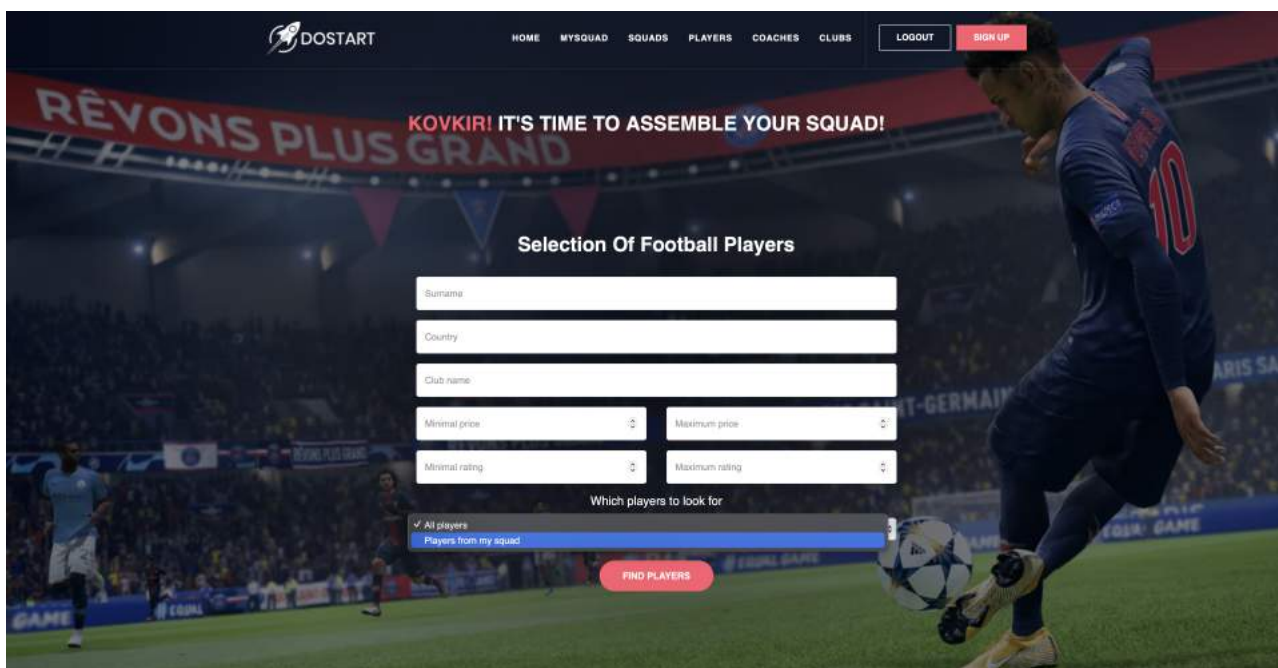


Рисунок 3.8 – Регистрация пользователя

### 3.7.2 Демонстрация возможностей авторизованного пользователя

Авторизованный пользователь при поиске футболистов по параметрам сможет выбирать, где их искать (в клубе или среди всех футболистов), также он сможет просматривать список составов других игроков (рисунки 3.9 – 3.10).



**Selection Of Football Players**

Surname:

Country:

Club name:

Minimal price:  Maximum price:

Minimal rating:  Maximum rating:

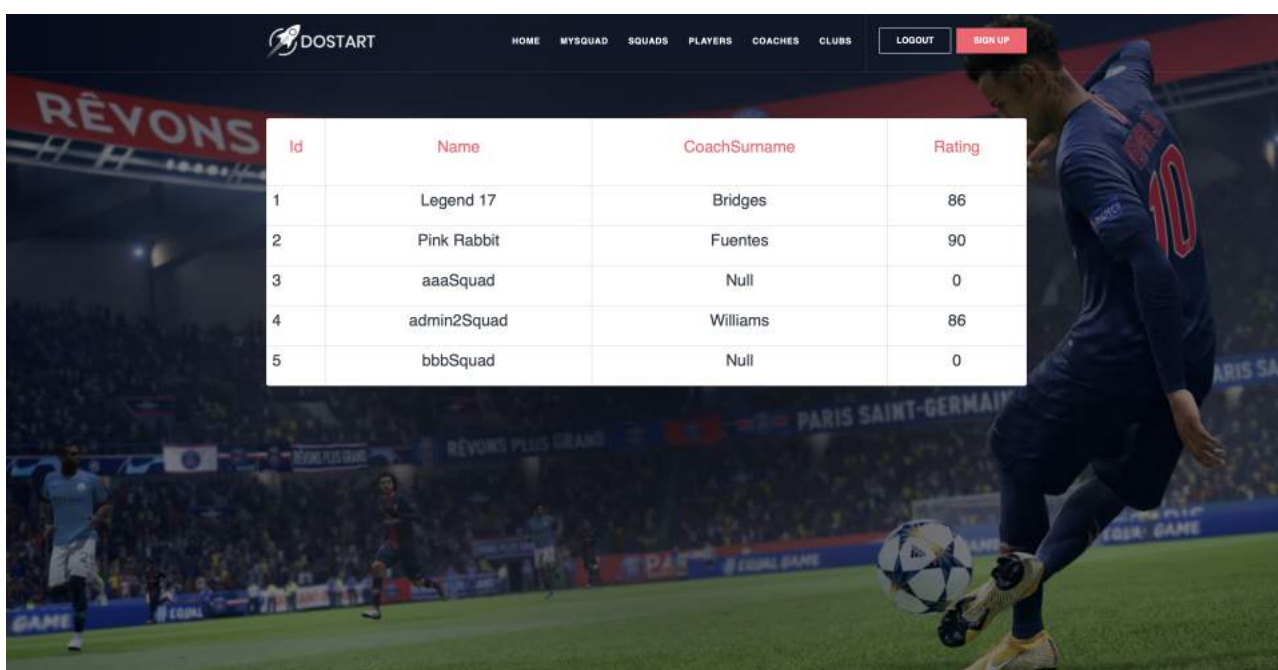
Which players to look for

☒ All players

☐ Players from my squad

**FIND PLAYERS**

Рисунок 3.9 – Поиск футболистов по параметрам (для пользователя)

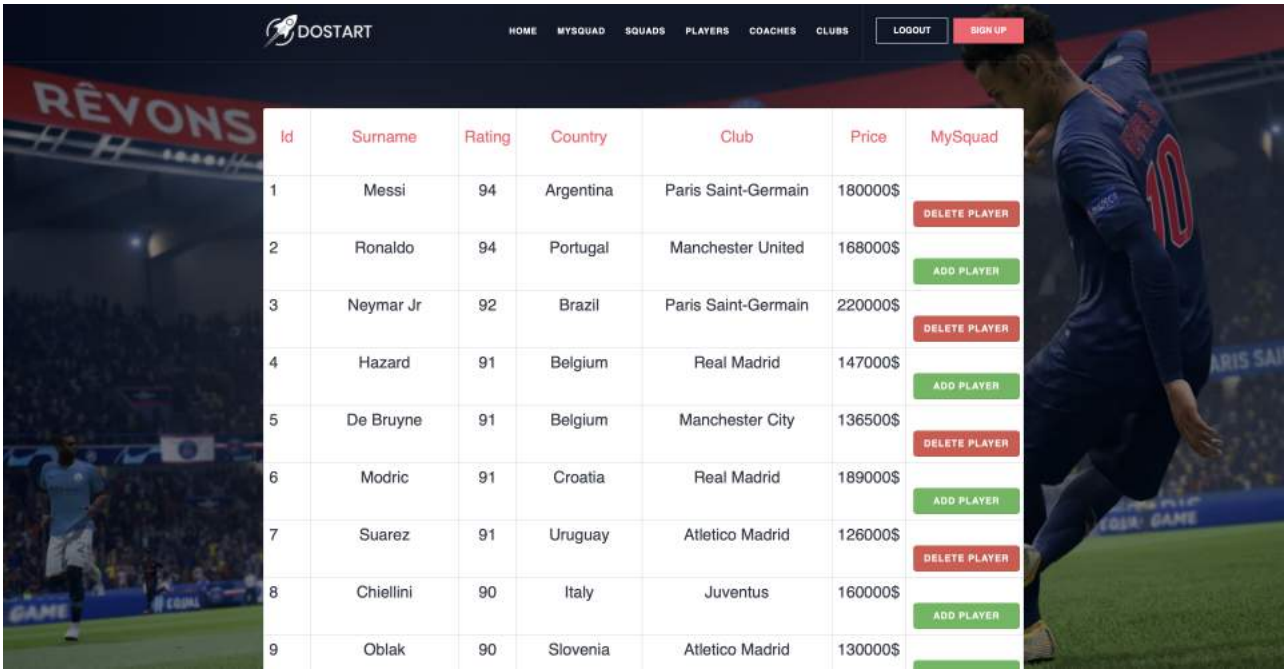


Id	Name	CoachSurname	Rating
1	Legend 17	Bridges	86
2	Pink Rabbit	Fuentes	90
3	aaaSquad	Null	0
4	admin2Squad	Williams	86
5	bbbSquad	Null	0

Рисунок 3.10 – Список составов других игроков (для пользователя)

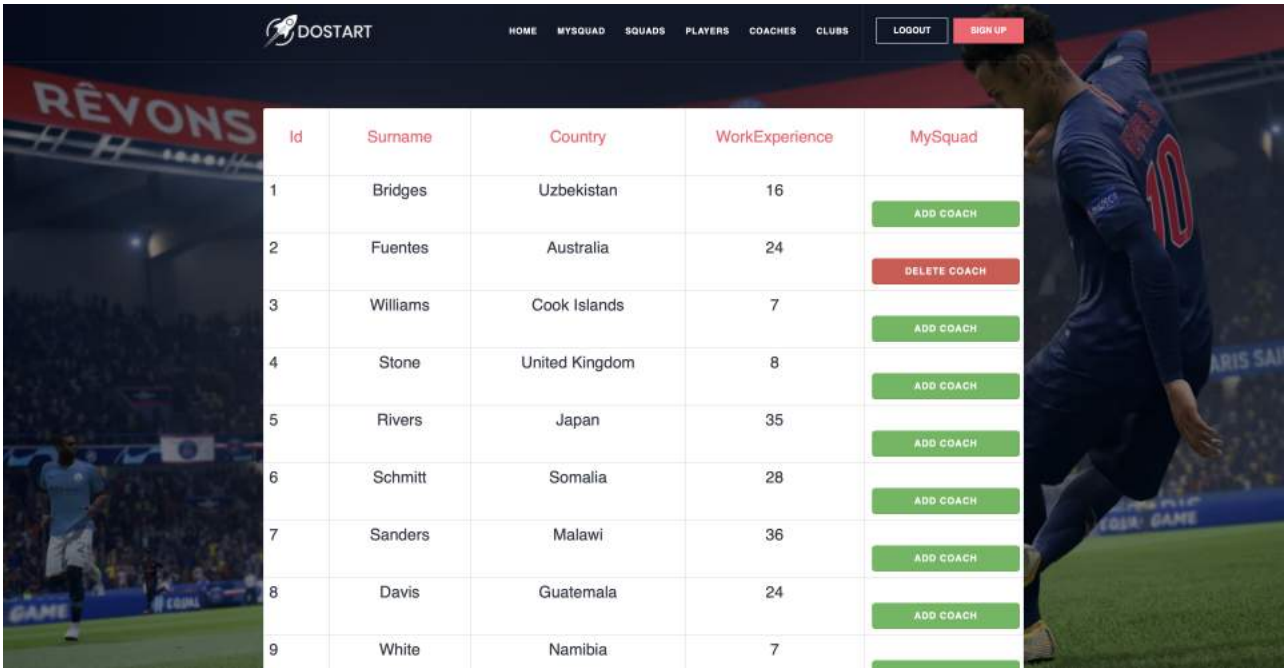


Авторизованный пользователь в качестве дополнительного функционала получает возможность просматривать актуальную цену футболистов на рынке, а также организовывать свою команду путем добавления понравившихся игроков и тренера в свой состав. Списки всех футболистов и тренеров с новыми возможностями представлены на рисунках 3.11 – 3.12.



Id	Surname	Rating	Country	Club	Price	MySquad
1	Messi	94	Argentina	Paris Saint-Germain	180000\$	DELETE PLAYER
2	Ronaldo	94	Portugal	Manchester United	168000\$	ADD PLAYER
3	Neymar Jr	92	Brazil	Paris Saint-Germain	220000\$	DELETE PLAYER
4	Hazard	91	Belgium	Real Madrid	147000\$	ADD PLAYER
5	De Bruyne	91	Belgium	Manchester City	136500\$	DELETE PLAYER
6	Modric	91	Croatia	Real Madrid	189000\$	ADD PLAYER
7	Suarez	91	Uruguay	Atletico Madrid	126000\$	DELETE PLAYER
8	Chiellini	90	Italy	Juventus	160000\$	ADD PLAYER
9	Oblak	90	Slovenia	Atletico Madrid	130000\$	

Рисунок 3.11 – Список всех футболистов (для пользователя)



Id	Surname	Country	WorkExperience	MySquad
1	Bridges	Uzbekistan	16	ADD COACH
2	Fuentes	Australia	24	DELETE COACH
3	Williams	Cook Islands	7	ADD COACH
4	Stone	United Kingdom	8	ADD COACH
5	Rivers	Japan	35	ADD COACH
6	Schmitt	Somalia	28	ADD COACH
7	Sanders	Malawi	36	ADD COACH
8	Davis	Guatemala	24	ADD COACH
9	White	Namibia	7	

Рисунок 3.12 – Список всех тренеров (для пользователя)

На рисунках 3.13 – 3.14 показано добавление и удаление футболистов из состава. Добавление и удаление тренера происходит аналогично.

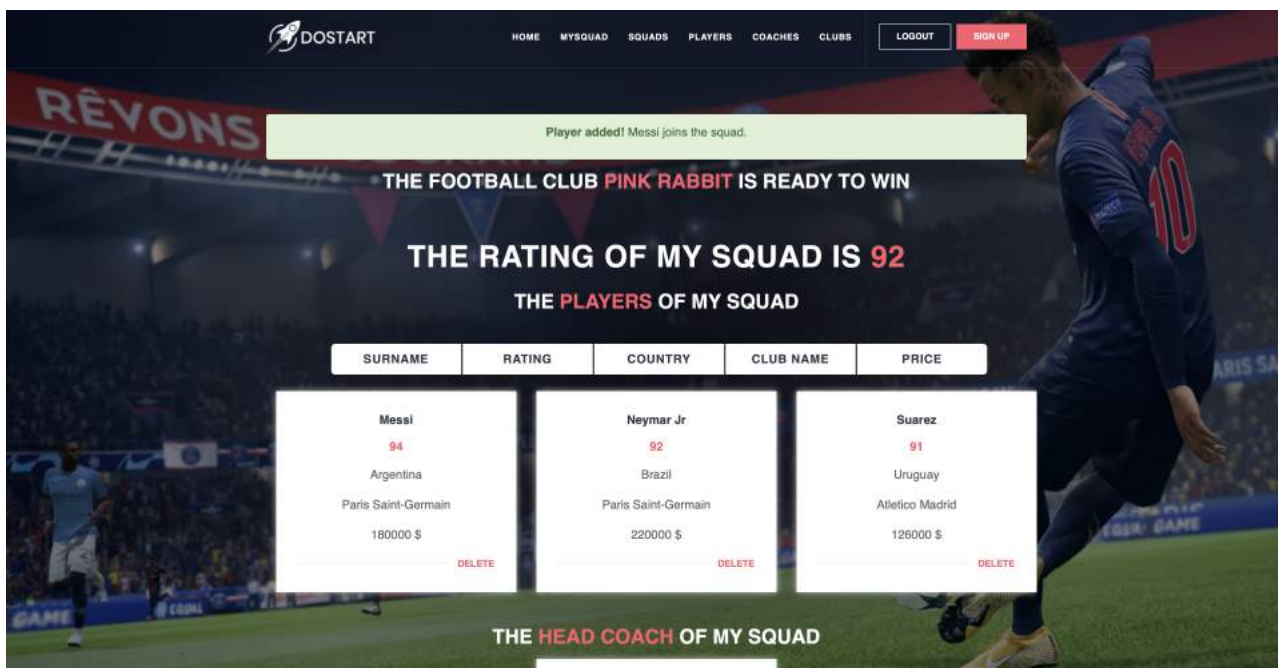


Рисунок 3.13 – Добавление футболиста в состав

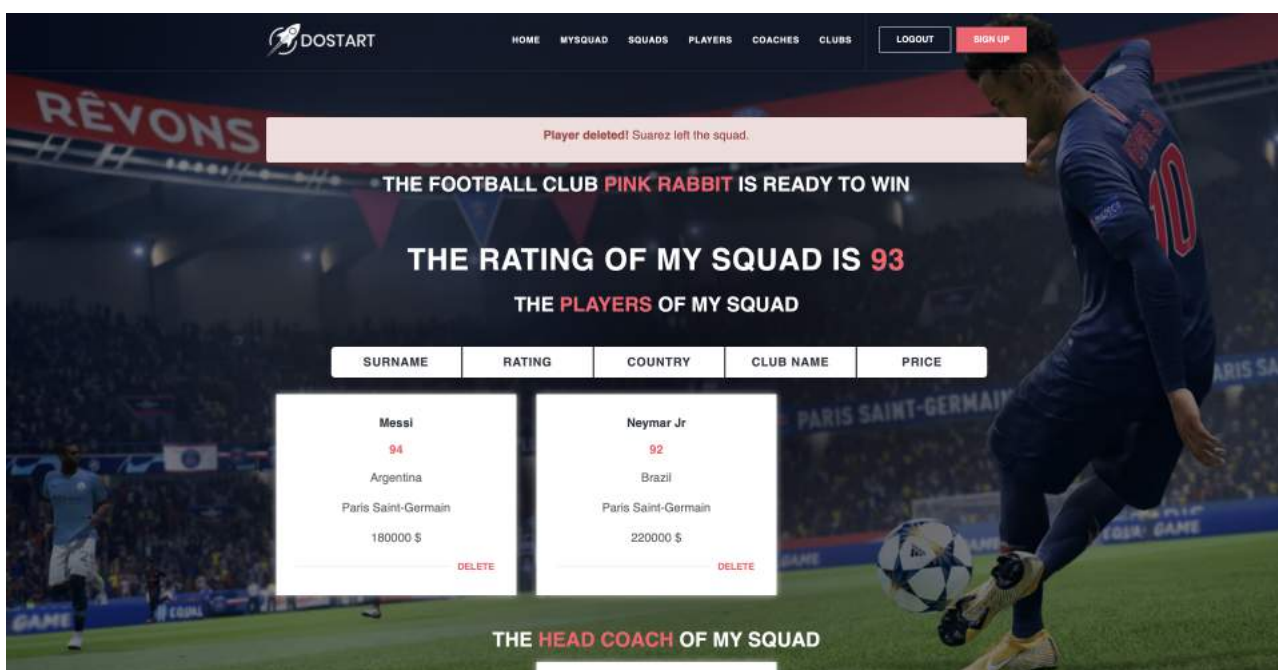


Рисунок 3.14 – Удаление футболиста из состава

Добавленных в состав футболистов можно отсортировать по фамилии, рейтингу, стране, названию команды или цене (как по возрастанию, так и по убыванию). В качестве примера на рисунках 3.15 – 3.16 показан состав с отсортированными по фамилии в алфавитном порядке футболистами, а также по убыванию их цены.

**THE RATING OF MY SQUAD IS 86**  
**THE PLAYERS OF MY SQUAD**

SURNAME	RATING	COUNTRY	CLUB NAME	PRICE
Aguero	89	Argentina	FC Barcelona	142500 \$
Dzhikiya	77	Russia	Spartak Moscow	49000 \$
Golovin	78	Russia	AS Monaco	76000 \$
Messi	94	Argentina	Paris Saint-Germain	180000 \$
Neymar Jr	92	Brazil	Paris Saint-Germain	220000 \$
Suarez	91	Uruguay	Atletico Madrid	126000 \$

Рисунок 3.15 – Футболисты, отсортированными по фамилии в алфавитном порядке

**THE RATING OF MY SQUAD IS 86**  
**THE PLAYERS OF MY SQUAD**

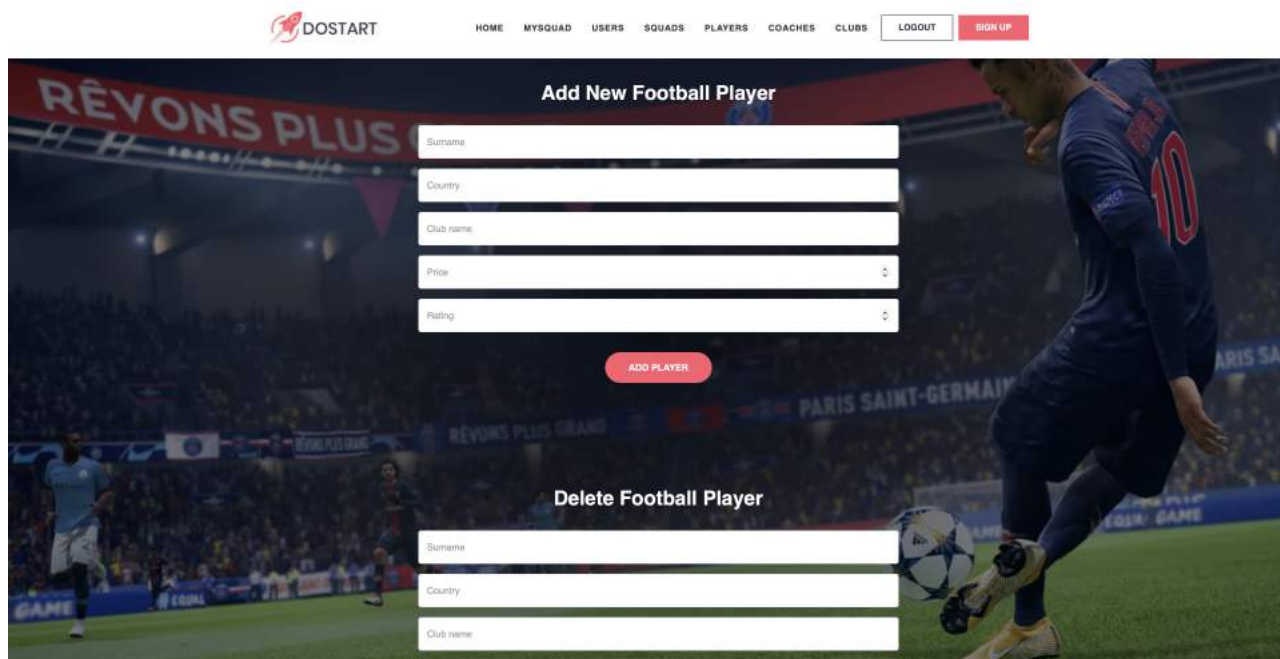
SURNAME	RATING	COUNTRY	CLUB NAME	PRICE
Neymar Jr	92	Brazil	Paris Saint-Germain	220000 \$
Messi	94	Argentina	Paris Saint-Germain	180000 \$
Aguero	89	Argentina	FC Barcelona	142500 \$
Suarez	91	Uruguay	Atletico Madrid	126000 \$
Golovin	78	Russia	AS Monaco	76000 \$
Dzhikiya	77	Russia	Spartak Moscow	49000 \$

Рисунок 3.16 – Футболисты, отсортированными по убыванию их цены



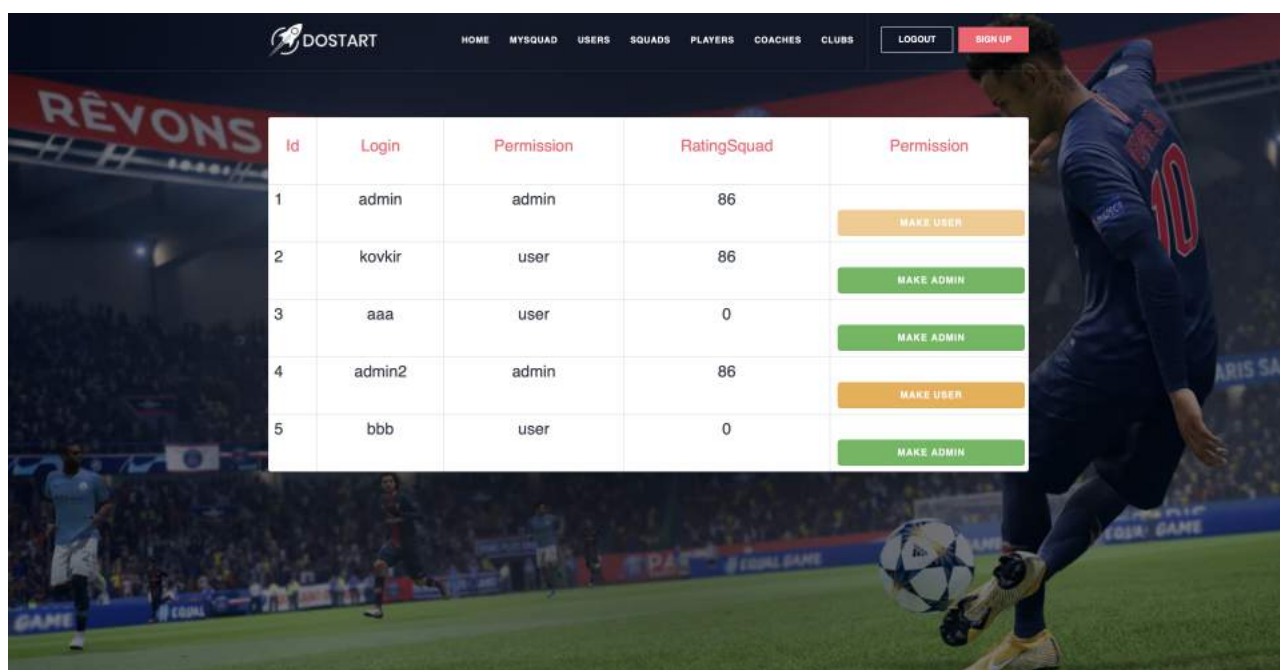
### 3.7.3 Демонстрация возможностей администратора

На домашней странице администратор получил возможность добавлять или удалять футболистов из базы данных (рисунок 3.17). Он также может просматривать список зарегистрированных пользователей и изменять их права доступа (рисунок 3.18).



The screenshot shows the DOSTART website interface. At the top, there is a navigation bar with links: HOME, MYSQUAD, USERS, SQUADS, PLAYERS, COACHES, CLUBS, LOGOUT, and SIGN UP. Below the navigation bar, there are two main sections: 'Add New Football Player' and 'Delete Football Player'. The 'Add New Football Player' section has a form with fields for Surname, Country, Club name, Price, and Rating, and an 'ADD PLAYER' button. The 'Delete Football Player' section has a form with fields for Surname, Country, and Club name.

Рисунок 3.17 – Поля для добавления или удаления футболистов из БД



The screenshot shows the DOSTART website interface. At the top, there is a navigation bar with links: HOME, MYSQUAD, USERS, SQUADS, PLAYERS, COACHES, CLUBS, LOGOUT, and SIGN UP. Below the navigation bar, there is a table listing registered users. The table has columns: Id, Login, Permission, RatingSquad, and Permission. The table contains 5 rows of data. Each row has a 'MAKE USER' or 'MAKE ADMIN' button next to it.

Id	Login	Permission	RatingSquad	Permission
1	admin	admin	86	MAKE USER
2	kovkir	user	86	MAKE ADMIN
3	aaa	user	0	MAKE ADMIN
4	admin2	admin	86	MAKE USER
5	bbb	user	0	MAKE ADMIN

Рисунок 3.18 – Список зарегистрированных пользователей

В отличие от авторизованного пользователя, который может только просматривать список составов других игроков, администратор имеет права на просмотр футболистов других клубов (рисунки 3.19 – 3.20).

Id	Name	CoachSurname	Rating	Players
1	Legend 17	Bridges	86	<a href="#">VIEW PLAYERS</a>
2	Pink Rabbit	Fuentes	86	<a href="#">VIEW PLAYERS</a>
3	aaaSquad	Null	0	<a href="#">VIEW PLAYERS</a>
4	admin2Squad	Williams	86	<a href="#">VIEW PLAYERS</a>
5	bbbSquad	Null	0	<a href="#">VIEW PLAYERS</a>

Рисунок 3.19 – Список составов других игроков (для администратора)

Id	Surname	Rating	Country	Club	Price	MySquad
1	Messi	94	Argentina	Paris Saint-Germain	180000\$	<a href="#">DELETE PLAYER</a>
3	Neymar Jr	92	Brazil	Paris Saint-Germain	220000\$	<a href="#">ADD PLAYER</a>
7	Suarez	91	Uruguay	Atletico Madrid	126000\$	<a href="#">ADD PLAYER</a>
22	Aguero	89	Argentina	FC Barcelona	142500\$	<a href="#">ADD PLAYER</a>
537	Golovin	78	Russia	AS Monaco	76000\$	<a href="#">DELETE PLAYER</a>
764	Dzhikiya	77	Russia	Spartak Moscow	49000\$	<a href="#">ADD PLAYER</a>

Рисунок 3.20 – Футболисты состава Pink Rabbit



### 3.8 Вывод

В данном разделе были описаны средства реализации ПО, архитектура приложения, был произведен выбор СУБД для реляционной модели БД. После сравнения самых популярных СУБД по выделенным критериям был выбран PostgreSQL, так как он единственный подошел по всем требованиям. Также были описаны реализации триггеров, функций, а также ролевые модели базы данных. Была продемонстрирована работа Web-приложения для каждой категории пользователя.

## 4 Исследовательская часть

### 4.1 Описание эксперимента

В качестве эксперимента было решено провести нагрузочное тестирование разработанного Web-приложения с помощью программы Apache JMeter [10], которая используется для имитации большой нагрузки на сервер, чтобы проверить его на прочность или проанализировать общую производительность при различных типах нагрузки.

В ходе эксперимента в течение 100 секунд к Web-приложению информационной системы футбольного симулятора подключалось 10 пользователей, которые отправляли три вида запросов:

- получить список всех футболистов;
- получить список всех тренеров;
- получить список всех клубов.

Суть эксперимента заключалась в том, чтобы сравнить среднее время выполнения этих запросов с учетом увеличения количества активных пользователей. Во время эксперимента таблицы футболистов, тренеров и клубов базы данных содержали 3000, 1000 и 445 строк соответственно.

## 4.2 Результаты эксперимента

Графики зависимости времени ответа на запросы от времени тестирования и от кол-ва активных пользователей представлены на рисунках 4.1 – 4.2.

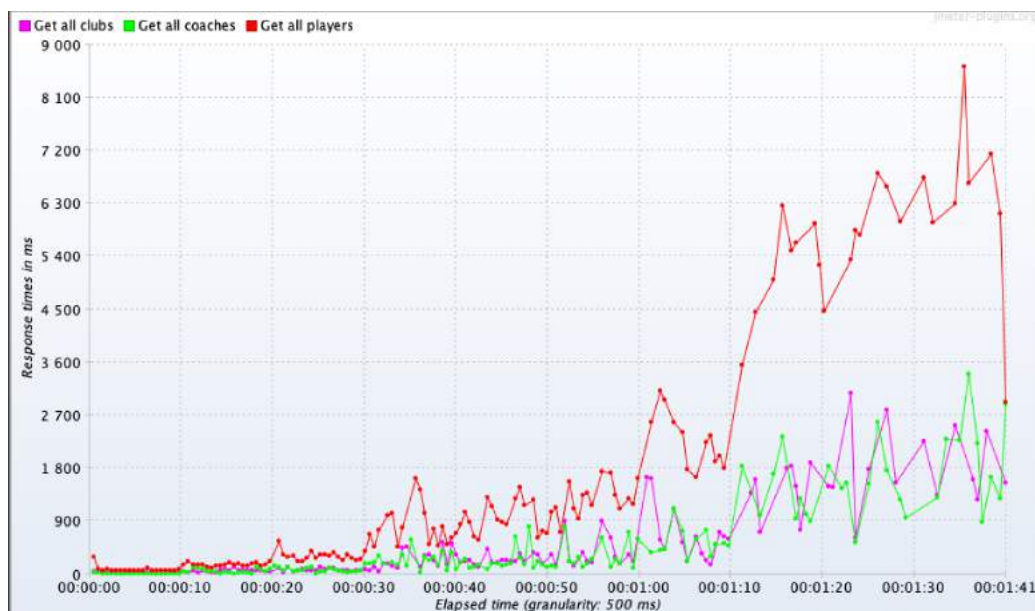


Рисунок 4.1 – График зависимости времени ответа на запросы от времени тестирования



Рисунок 4.2 – График зависимости времени ответа на запросы от кол-ва активных пользователей

На рисунке 4.3 представлена таблица с результатами эксперимента, столбцы которой отвечают за:

- количество разных типов запросов;
- среднее время ответа на запрос;
- минимальное время ответа на запрос;
- максимальное время ответа на запрос.

Время в таблице измеряется в миллисекундах.

Label	# Samples	Average	Min	Max
Get all players	377	982	66	9066
Get all clubs	371	257	18	3081
Get all coaches	369	239	14	3402
TOTAL	1117	496	14	9066

Рисунок 4.3 – Таблица с результатами эксперимента

### 4.3 Вывод

В результате эксперимента было установлено, что среднее время запроса на получение списка всех футболистов (0.98 секунды) в 3.82 раза больше, чем время получения списка всех клубов и в 4.11 раз больше времени получения списка всех тренеров. В свое время как время запросов на получения списков тренеров практически не отличается от времени получения списков клубов (быстрее всего в 1.08 раз).

Стоит отметить, что разработанное Web-приложение плохо себя показало при относительно небольшом кол-ве пользователей. Всего при 10 одновременно активных пользователях, время запроса на получение списка всех футболистов увеличилось до 7 секунд (для клубов и тренеров почти до 2 секунд).

## ЗАКЛЮЧЕНИЕ

Во время выполнения курсовой работы была разработана база данных, а также создано Web-приложение для информационной системы футбольного симулятора. Были рассмотрены модели баз данных и произведен выбор СУБД для реляционной модели БД. После сравнения самых популярных СУБД по выделенным критериям для поставленной задачи был выбран PostgreSQL.

В ходе эксперимента выяснилось, что время запроса на получение списка всех клубов в разы больше, чем время, требуемое на получение списка тренеров (в 4.11 раза) или клубов (в 3.82 раза). Кроме этого эксперимент показал, что Web-приложение плохо себя показывает при относительно небольшом кол-ве активных пользователей (10 человек), так как в этот случае время ожидания на запрос получения всех футболистов возрастает до 7 секунд.

Также в ходе выполнения курсовой работы были выполнены следующие задачи:

- проанализированы существующие решения;
- формализована задача и определен необходимый функционал;
- рассмотрены модели баз данных и выбрана подходящая;
- проанализированы существующие СУБД и выбрана нужная;
- спроектирована и разработана БД;
- спроектировано и разработано WEB-приложение.

Поставленная цель была достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аналитическое агентство АВТОСТАТ [Электронный ресурс]. — URL: <https://www.autostat.ru> (дата обращения: 18.07.2022).
2. What is a Relational Database (RDBMS)? [Электронный ресурс]. — URL: <https://www.oracle.com/database/what-is-a-relational-database/> (дата обращения: 18.07.2022).
3. Нереляционные данные и базы данных NoSQL [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/azure/architecture/data-guide/big-data/non-relational-data> (дата обращения: 18.07.2022).
4. Документация по C Sharp [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 18.07.2022).
5. Visual Studio [Электронный ресурс]. — URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 18.07.2022).
6. ASP.NET [Электронный ресурс]. — URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата обращения: 18.07.2022).
7. Общие сведения об ASP.NET Core [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0> (дата обращения: 18.07.2022).
8. Общие сведения ASP.NET Core MVC [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/aspnet/core/mvc/overview?view=aspnetcore-6.0> (дата обращения: 18.07.2022).
9. PostgreSQL [Электронный ресурс]. — URL: <https://www.postgresql.org/docs/> (дата обращения: 18.07.2022).
10. Apache JMeter [Электронный ресурс]. — URL: <https://jmeter.apache.org> (дата обращения: 18.07.2022).



**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

# **Создание информационной системы для футбольного симулятора**

**Студент: Ковалец Кирилл Эдуардович ИУ7-63Б**

**Научный руководитель: Романова Татьяна Николаевна**

# Цель и задачи

**Целью** курсовой работы является разработка базы данных для информационной системы футбольного симулятора.

## **Задачи:**

- проанализировать существующие решения;
- формализовать задачу и определить необходимый функционал;
- рассмотреть модели баз данных и выбрать подходящую;
- проанализировать существующие СУБД и выбрать нужную;
- спроектировать и разработать БД;
- спроектировать и разработать Web-приложение.



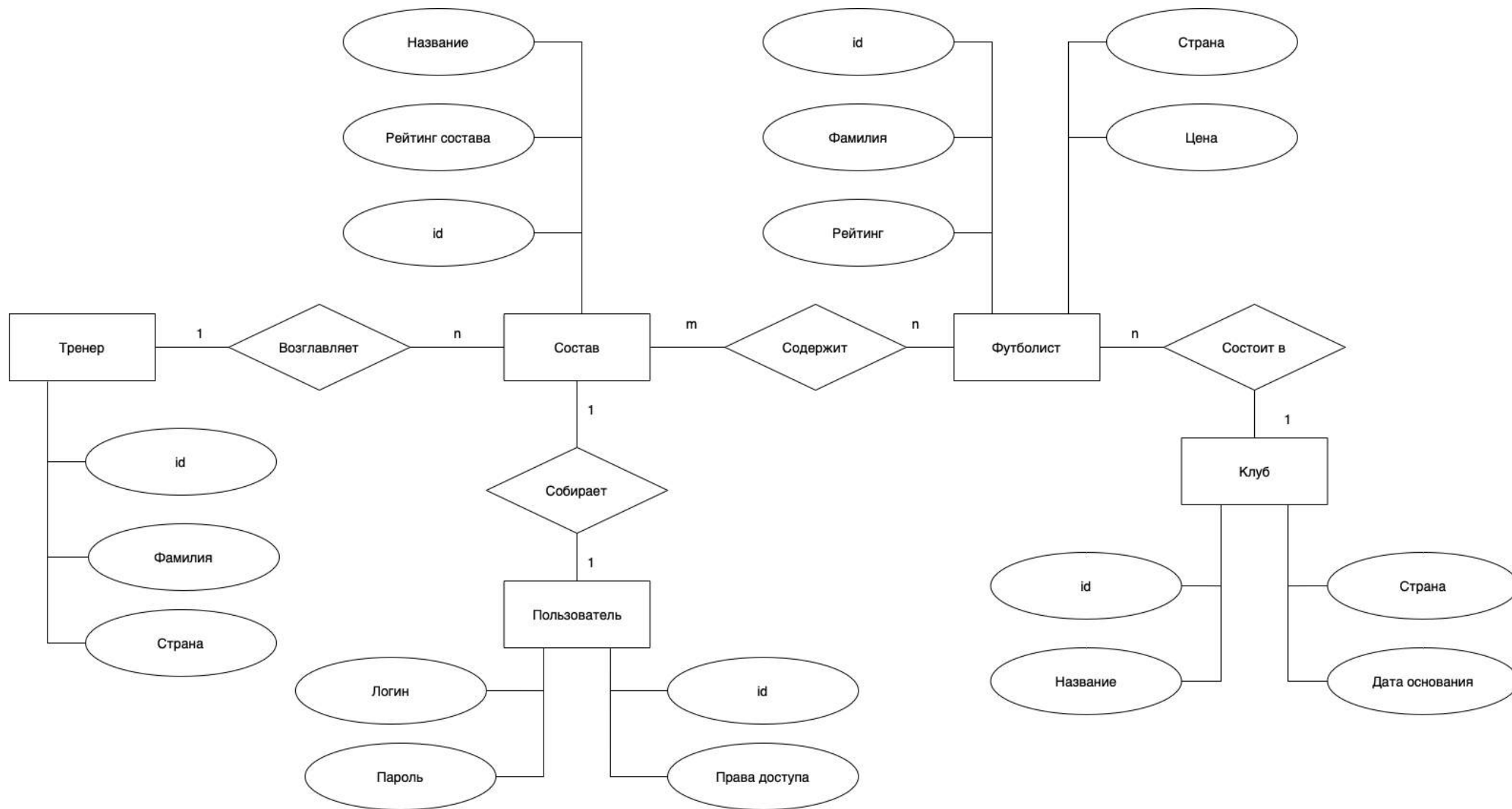
# Анализ существующих решений

## Критерии сравнения:

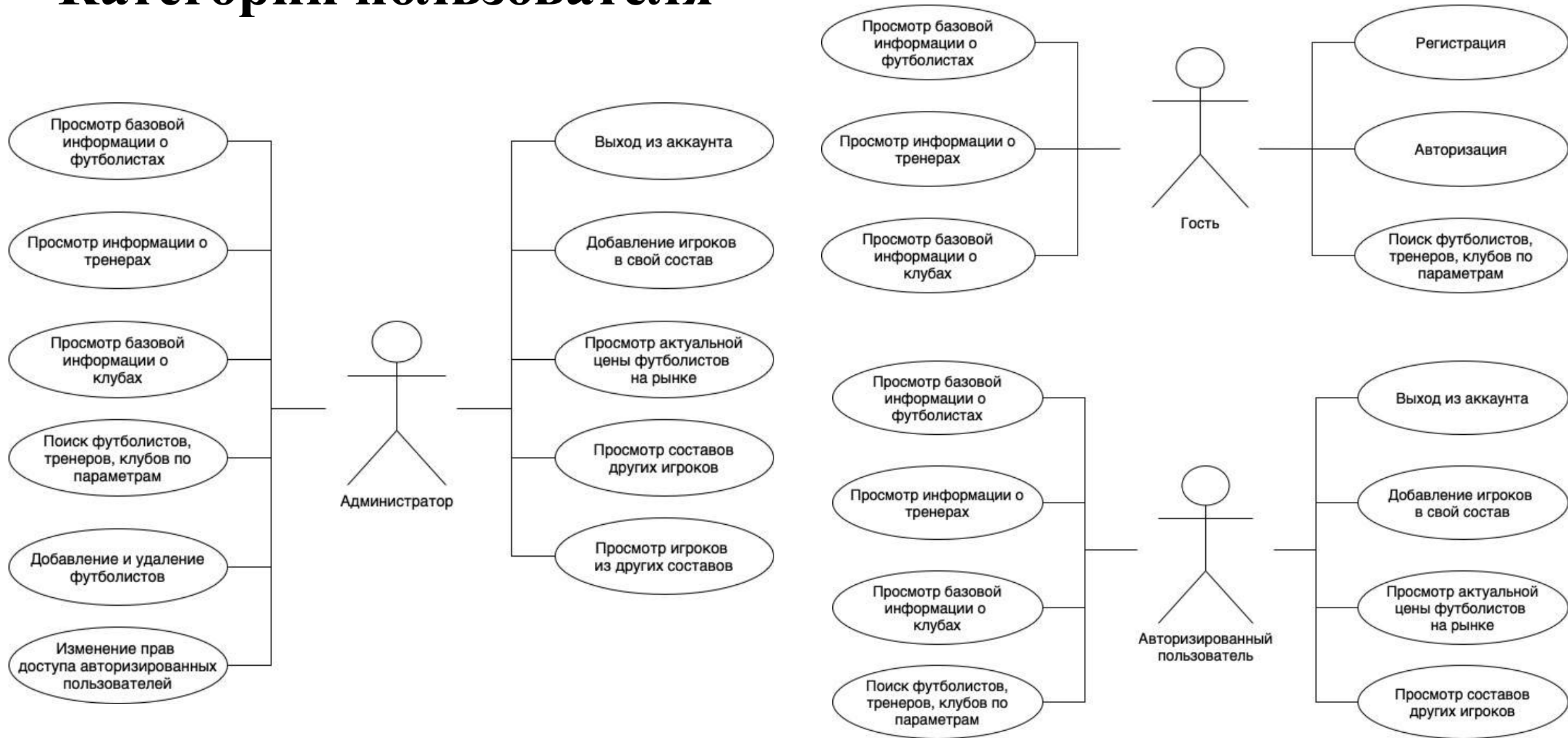
1. поиск игроков по заданным параметрам;
2. просмотр текущей цены футболистов;
3. сборка собственного состава футболистов;
4. просмотр рейтинга составов других игроков;
5. наличие информации об игроках;
6. наличие информации о тренерах;
7. наличие информации о клубах.

Критерий	Futbin	Futwiz	Futhead	FIFA Companion
1	+	+	+	+
2	+	+	—	—
3	+	+	+	+
4	—	—	—	+
5	+	+	+	+
6	—	—	—	+
7	—	—	+	—

# ER-диаграмма в нотации Чена



# Категории пользователя



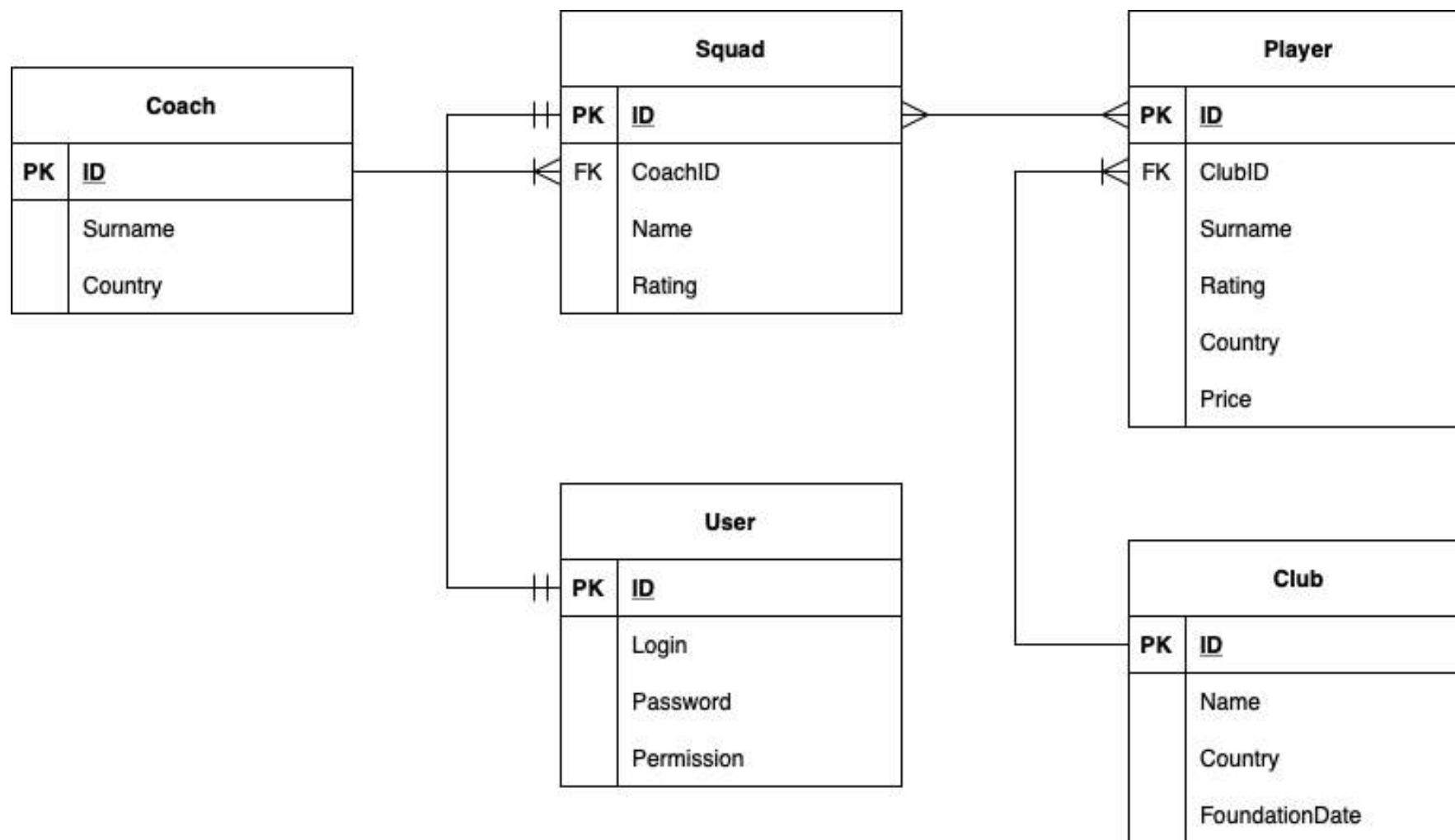
# Выбор реляционной СУБД

## Критерии сравнения:

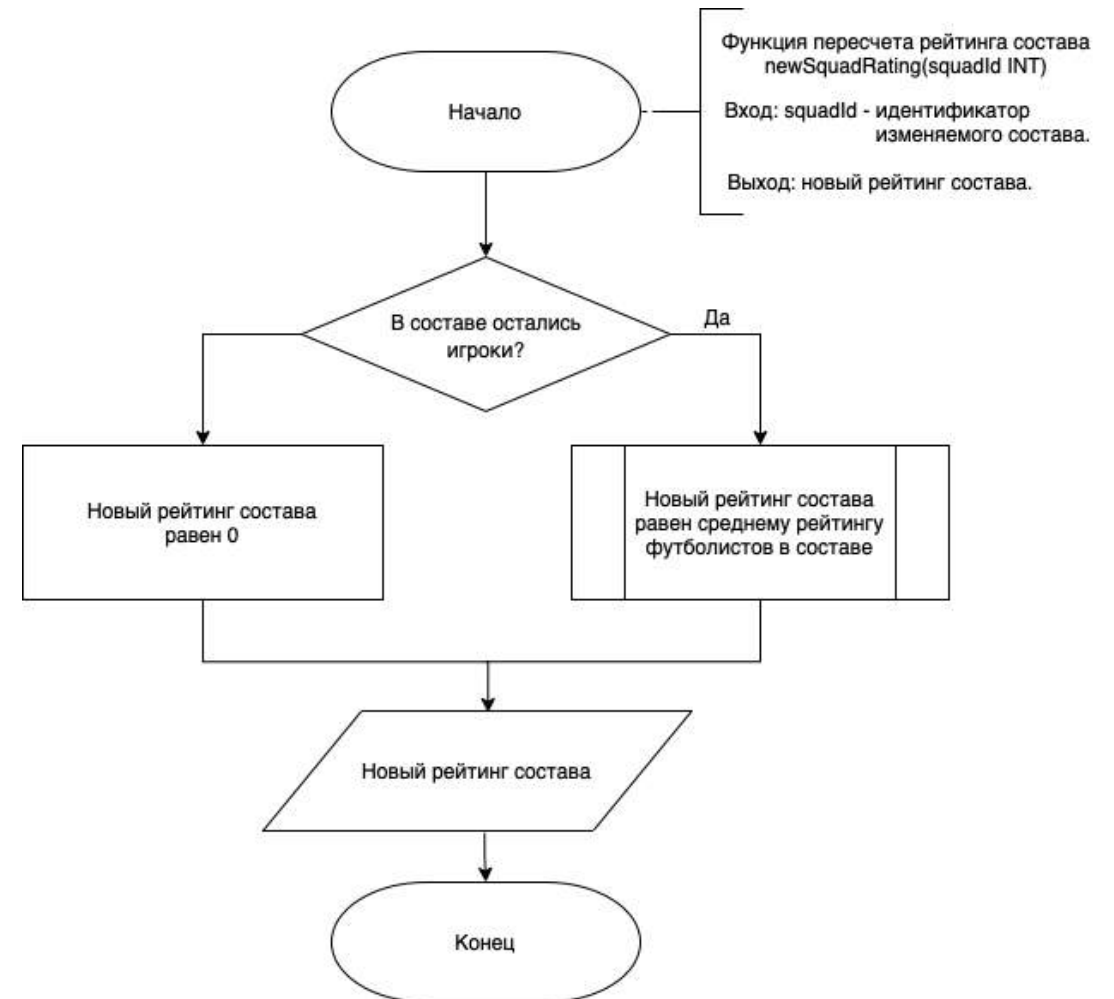
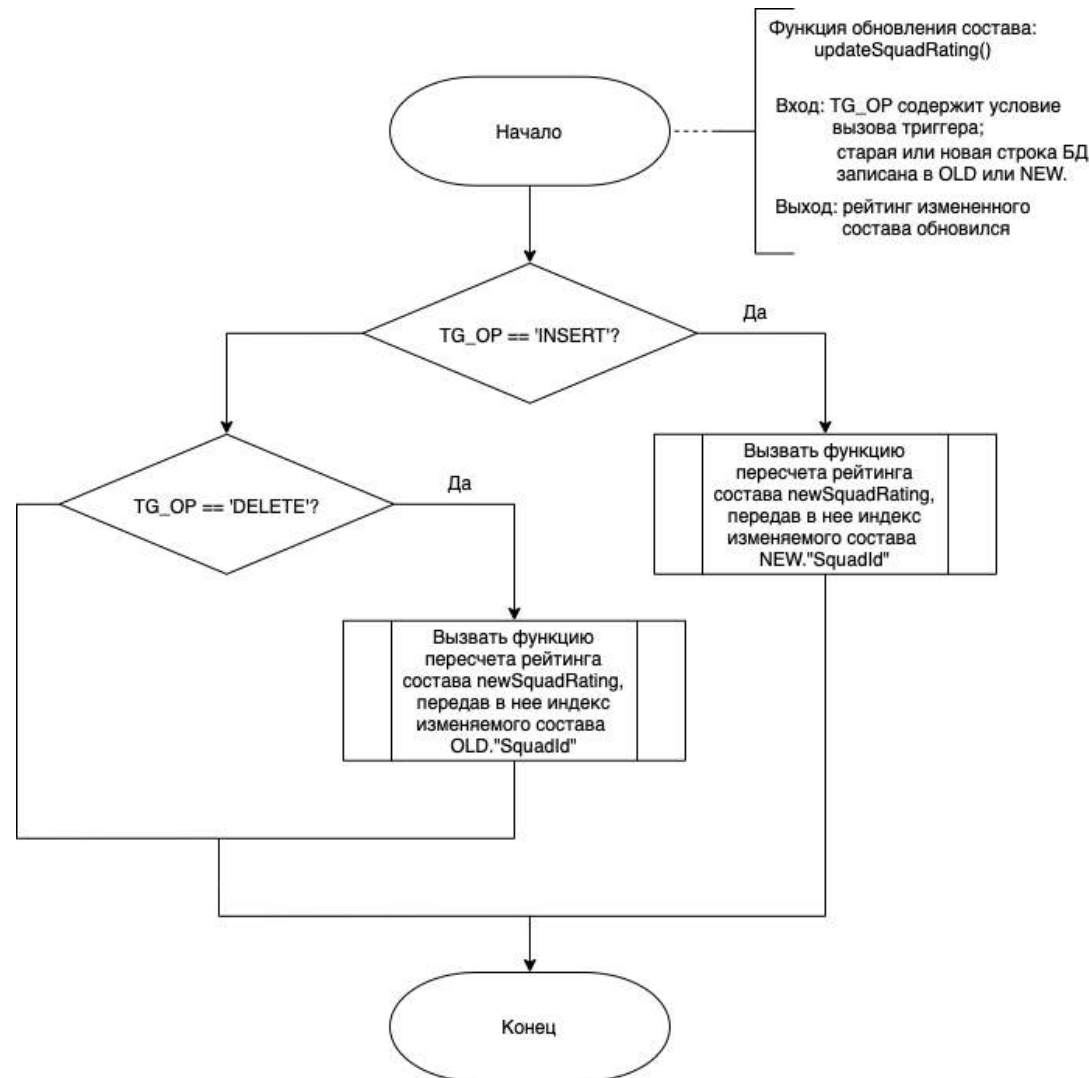
1. бесплатное распространение;
2. наличие подробной документации;
3. наличие высокого уровня оптимизации;
4. опыт работы с данной СУБД.

Критерий	MySQL	Oracle	PostgreSQL	Microsoft SQL Server
1	+	–	+	–
2	+	+	+	+
3	+	–	+	–
4	–	–	+	–

# Диаграмма базы данных



# Схема функции обновления рейтинга (вызывается триггерами)



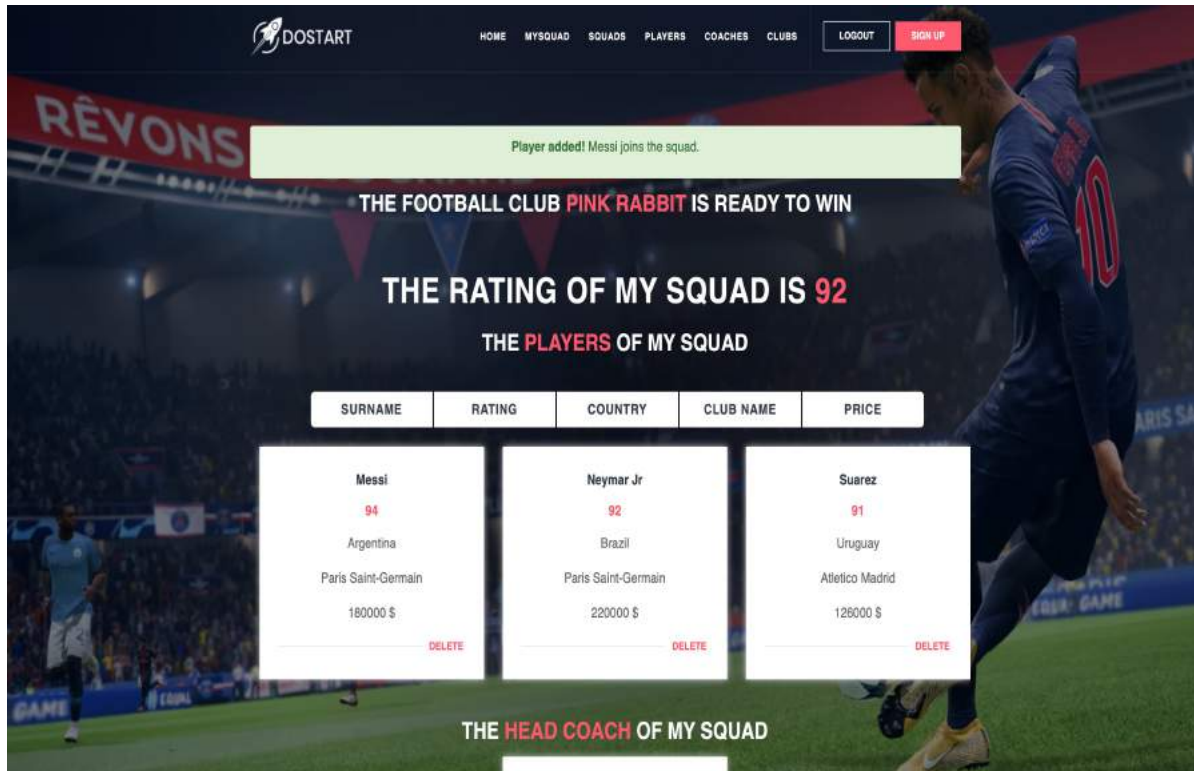
# Архитектура приложения и средства реализации

За основу архитектуры разработанного Web-приложения была взята модель MVC (MODEL, VIEW, CONTROLLER).

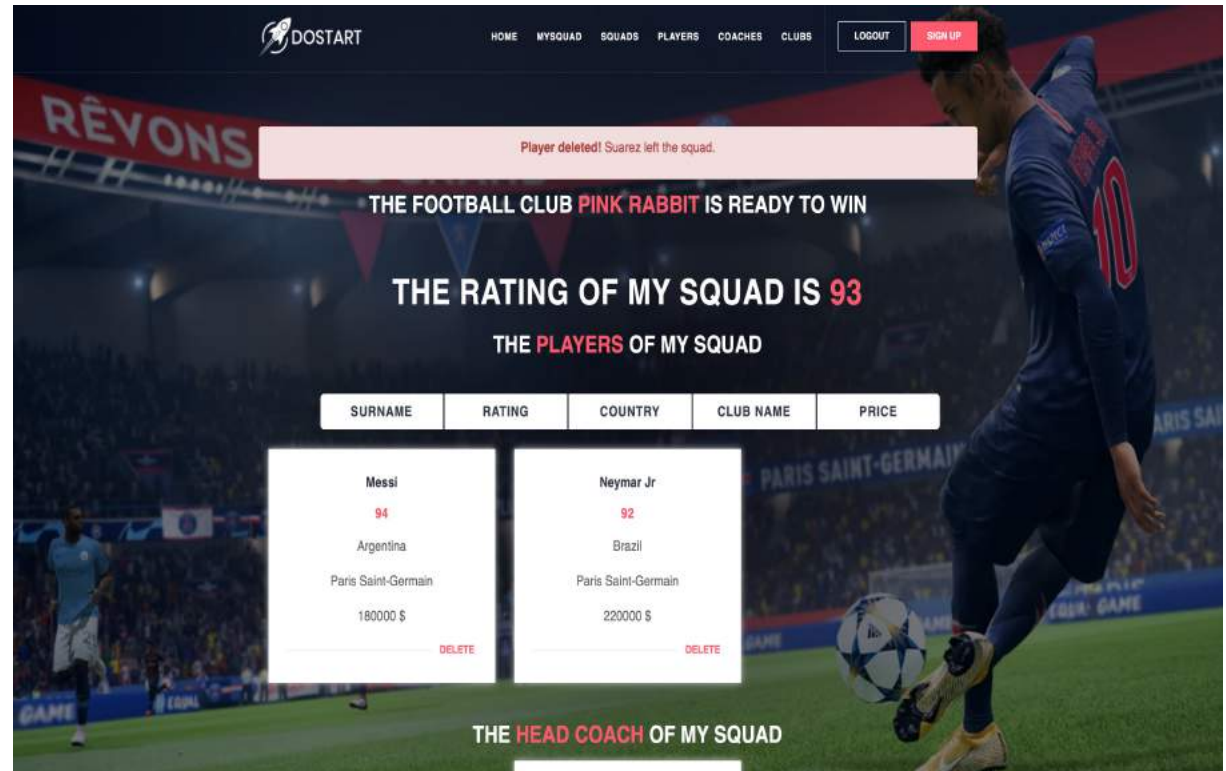
## Средства реализации ПО:

- Язык программирования: C#;
- Среда разработки: Visual Studio;
- Фреймворк для Web-разработки: ASP.NET Core;
- Фреймворк представления: ASP.NET Core MVC.

# Демонстрация работы Web-приложения (изменение состава)



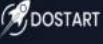
Добавление футболиста в состав



Удаление футболиста из состава



# Демонстрация работы Web-приложения (поиск футболистов)

 HOME MYSQUAD SQUADS PLAYERS COACHES CLUBS LOGOUT SIGN UP

**KOVKIRI! IT'S TIME TO ASSEMBLE YOUR SQUAD!**

**Selection Of Football Players**

Surname:

Country:

Club name:

Minimal price:  Maximum price:

Minimal rating:  Maximum rating:


Which players to look for

☒ All players

☐ Players from my squad

**FIND PLAYERS**

Поиск футболистов по заданным параметрам

 HOME MYSQUAD SQUADS PLAYERS COACHES CLUBS LOGOUT SIGN UP

Id	Surname	Rating	Country	Club	Price	MySquad
1	Messi	94	Argentina	Paris Saint-Germain	180000\$	<b>DELETE PLAYER</b>
2	Ronaldo	94	Portugal	Manchester United	168000\$	<b>ADD PLAYER</b>
3	Neymar Jr	92	Brazil	Paris Saint-Germain	220000\$	<b>DELETE PLAYER</b>
4	Hazard	91	Belgium	Real Madrid	147000\$	<b>ADD PLAYER</b>
5	De Bruyne	91	Belgium	Manchester City	136500\$	<b>DELETE PLAYER</b>
6	Modric	91	Croatia	Real Madrid	189000\$	<b>ADD PLAYER</b>
7	Suarez	91	Uruguay	Atletico Madrid	126000\$	<b>DELETE PLAYER</b>
8	Chiellini	90	Italy	Juventus	160000\$	<b>ADD PLAYER</b>
9	Oblak	90	Slovenia	Atletico Madrid	130000\$	<b>ADD PLAYER</b>

Список всех доступных футболистов

# Исследовательская часть

В ходе нагрузочного тестирования в течение 100 секунд к Web-приложению подключалось 10 пользователей, которые отправляли три вида запросов:

- получить список всех футболистов;
- получить список всех тренеров;
- получить список всех клубов.

В результате эксперимента было установлено, что:

- среднее время запроса на получение списка всех футболистов (0.98 секунды) в 3.82 раза больше, чем время получения списка всех клубов и в 4.11 раз больше времени получения списка всех тренеров;
- при 10 одновременно активных пользователях, время запроса на получение списка всех футболистов увеличилось с 0.66 секунд до 7 (для клубов и тренеров с 0.14 почти до 2 секунд).

# Заключение

В ходе выполнения курсовой работы были выполнены следующие задачи:

- проанализированы существующие решения;
- формализована задача и определен необходимый функционал;
- рассмотрены модели баз данных и выбрана подходящая;
- проанализированы существующие СУБД и выбрано нужное;
- спроектирована и разработана БД;
- спроектировано и разработано Web-приложение.

Поставленная цель была достигнута.

# Направление дальнейшего развития

В ходе дальнейшего развития возможны следующие варианты:

- Добавление большей информации об игроках (фотографии, кол-во голов, средний рейтинг за игру);
- Добавление таблицы агентов игроков;
- Добавление возможности пользователям выбирать футбольный стадион для своего состава, для этого будет необходимо добавить таблицу стадионов.