

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Разработка ПО информационных систем»
Тема: Статистика футбольных матчей

Студенты гр. 6303

Архипцев Е.Д.

Матюшина М.Е.

Эвергрин П.

Преподаватель

Заславский М.М.

Санкт-Петербург

2019

ЗАДАНИЕ

Студенты Архипцев Е.Д., Матюшина М.Е., Эвергрин П.

Группа 6303.

Тема проекта: Разработка приложения «Статистика футбольных матчей».

Исходные данные:

Необходимо реализовать приложение, позволяющее

- 1)осуществлять сбор статистики футбольных матчей
- 2)осуществлять поиск матчей по выбранным параметрам
- 3)производить сравнение двух команд, в том числе графически

Содержание пояснительной записки:

«Содержание», «Введение», «Качественные требования к решению»,
«Сценарии использования», «Модель данных», «Разработанное
приложение», «Выводы», «Приложение», «Список использованных
источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания:

Дата сдачи задания:

Дата защиты задания:

Студенты гр. 6303

Архипцев Е.Д.

Матюшина М.Е.

Эвергрин П.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

В рамках данного курса предполагалось разработать какое-либо приложение в команде на одну из поставленных тем. Была выбрана тема создания приложения «Статистика футбольных матчей», ориентированного на хранение, анализ и использование данных футбольных матчей. Было проведено исследование возможностей **нереляционных** баз данных на примере **NoSQL** решения *MongoDB*. Найти исходный код и всю дополнительную информацию можно по ссылке: <https://github.com/moevm/nosql1h19-football-stats>

SUMMARY

As part of this course, it was planned to develop an application as a team from one of the given topics. The topic of creation of the application "Statistics of football matches" focusing on storage, analysis and use of data of football matches was chosen. A study of the possibilities of non-relational databases on the example of NoSQL solutions MongoDB was conducted. You can find the source code and all additional information here through the link: <https://github.com/moevm/nosql1h19-football-stats>

Содержание.

1. Введение.....	5
2. Качественные требования к решению.....	5
3. Сценарии использования.....	5
3.1. Макет UI.....	5
3.2. Описание сценариев использования.....	8
4. Модель данных.....	10
4.1. NoSQL модель данных.....	10
4.1.1. Описание и расчет объема.....	10
4.1.2. Запросы.....	11
4.2. SQL модель данных.....	12
4.2.1. Описание и расчет объема.....	13
4.2.2. Запросы.....	14
4.3. Сравнение SQL и NoSQL.....	15
5. Разработанное приложение.....	15
5.1. Краткое описание.....	15
5.2. Схема экранов приложения.....	16
5.3. Используемые технологии.....	17
5.4. Ссылки на Приложение.....	17
6. Выводы.....	17
6.1. Результаты.....	17
6.2. Недостатки и пути для улучшения полученного решения.....	17
7. Приложение.....	18
7.1. Документация по сборке и разворачиванию приложения.....	18
7.2. Снимки экранов приложения.....	19
8. Список литературы.....	22

1. Введение.

Цель работы – создать удобное решение для хранения, анализа и использования информации о футбольных матчах.

Было решено разработать веб-приложение, которое позволит хранить в электронном виде всю информацию о матчах, осуществлять сбор статистики футбольных матчей, осуществлять поиск матчей по выбранным параметрам матча, а также производить сравнение двух команд, в том числе графически.

2. Качественные требования к решению.

Требуется разработать приложение с использованием СУБД MongoDB.

3. Сценарии использования.

3.1.Макет UI.

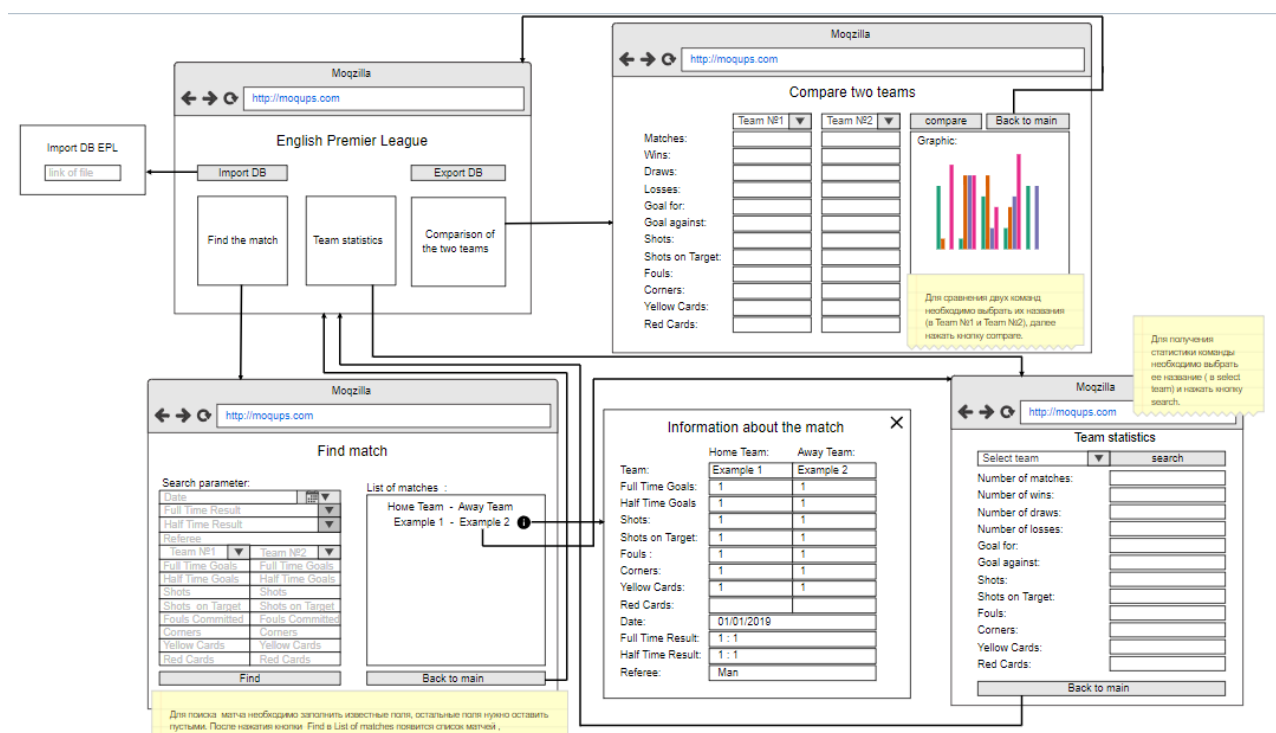


Рисунок 1 – Общий вид.

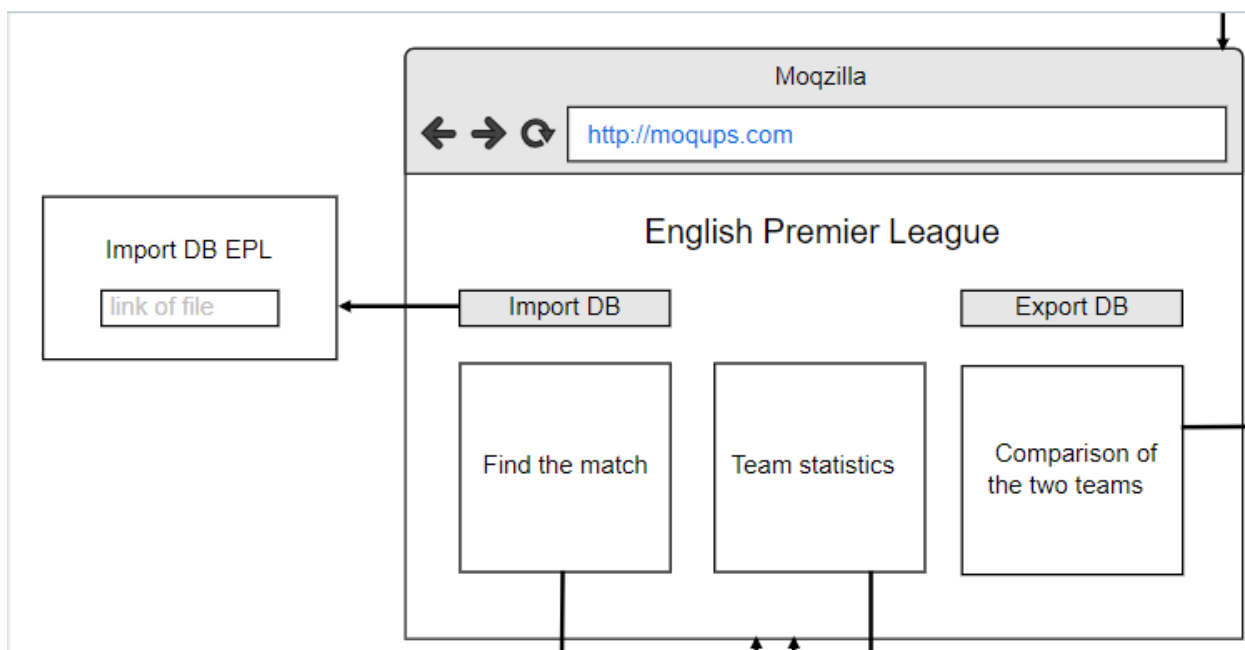


Рисунок 2 – Главная страница

The diagram shows a web browser window titled 'Mozilla' with the address bar set to 'http://moqups.com'. The main content area is titled 'Find match'. On the left, there is a 'Search parameter:' section with a form containing fields for 'Date', 'Full Time Result', 'Half Time Result', 'Referee', 'Team №1', 'Team №2', 'Full Time Goals', 'Half Time Goals', 'Shots', 'Shots on Target', 'Fouls Committed', 'Corners', 'Yellow Cards', and 'Red Cards'. Below the form is a 'Find' button. On the right, there is a 'List of matches :' section with a table showing 'Home Team - Away Team' and 'Example 1 - Example 2'. An information icon (i) is next to the second row. Below the table is a 'Back to main' button. A yellow box at the bottom contains the following text:

Для поиска матча необходимо заполнить известные поля, остальные поля нужно оставить пустыми. После нажатия кнопки Find в List of matches появится список матчей, удовлетворяющих условиям поиска. Получить полную информацию о матче из списка можно нажатием значка info, расположенного рядом с матчем. При нажатии на название команды в списке, пользователь переходит к статистике команды.

Рисунок 3 – Поиск матча

X

Information about the match

	Home Team:	Away Team:
Team:	Example 1	Example 2
Full Time Goals:	1	1
Half Time Goals:	1	1
Shots:	1	1
Shots on Target:	1	1
Fouls :	1	1
Corners:	1	1
Yellow Cards:	1	1
Red Cards:		
Date:	01/01/2019	
Full Time Result:	1 : 1	
Half Time Result:	1 : 1	
Referee:	Man	

Рисунок 4 – Информация о матче

Moqzilla

<http://moqups.com>

Team statistics

Select team
▼
search

Number of matches:

Number of wins:

Number of draws:

Number of losses:

Goal for:

Goal against:

Shots:

Shots on Target:

Fouls:

Corners:

Yellow Cards:

Red Cards:

Back to main

статистики команды
необходимо выбрать
ее название (в select
team) и нажать кнопку
search.

Рисунок 5 – Статистика команды

7

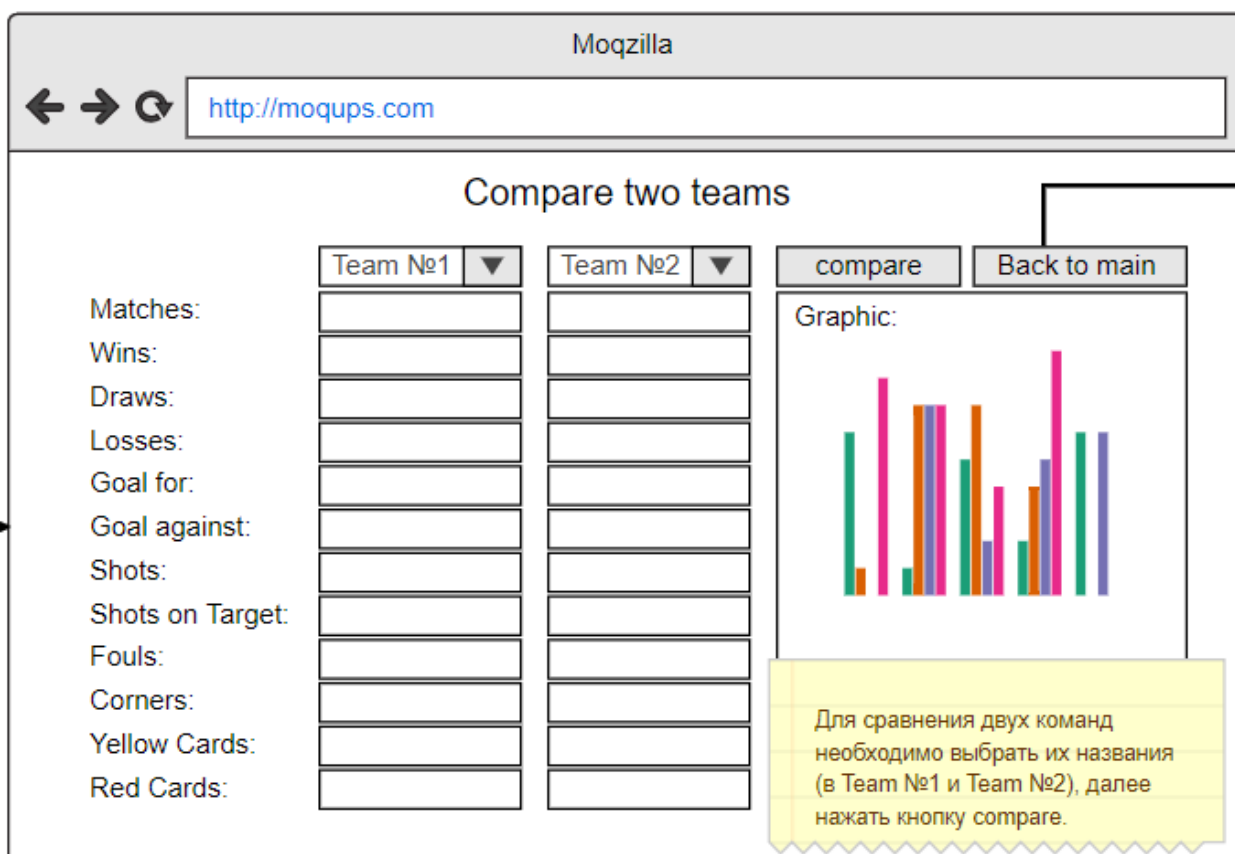


Рисунок 6 – Сравнение двух команд

3.2. Описание сценариев использования.

Пользователь заходит на главную страницу сайта, который предназначен для получения информации о матчах Английской Премьер-лиги. Он видит меню навигации, состоящее из трех кнопок:

- Find match - поиск матча,
- Team Statistics – статистика команды,
- Compare two teams – сравнение двух команд.

Также на главной странице расположены кнопки для экспорта и импорта базы данных матчей. Нажимая на кнопку импорта, пользователь получает возможность ввести путь до импортируемого файла. Нажимая на кнопку экспорта, база данных матчей экспортируется в файл, происходит скачивание.

Пользователь переходит в интересующий его раздел меню навигации сайта.

Поиск матчей.

Пользователь вбивает известные ему параметры матча в соответствующие поля, остальные поля не заполняются. После

нажатия кнопки *Find* в *List of matches* появляется список матчей, удовлетворяющих условиям поиска. Список представляет собой пары *Домашняя Команда (Home Team)* – *Гостевая Команда (Away Team)*. Чтобы получить подробную информацию о матче, пользователь нажимает на значок *info* рядом с матчем. При нажатии на название какой-либо команды, пользователь переходит к ее статистике в раздел *Статистика команды (Team statistics)*, где ее название уже подставлено в *select*. Чтобы вернуться в меню, следует нажать кнопку *Back to main*.

Общие параметры матча:

- Date
- Full Time Result
- Half Time Result
- Referee
- Team №1
- Team №2

Параметры для каждой команды:

- Full Time Goals
- Half Time Goals
- Shots
- Shots on Target
- Fouls Committed
- Corners
- Yellow Cards
- Red Cards

Статистика команды.

В разделе *Team statistics* пользователь в *select team* выбирает название команды, статистику которой хочет получить и нажимает кнопку *search*. Пользователь получает **следующие данные:**

- Number of matches – общее количество матчей команды
- Number of wins – количество побед
- Number of draws – количество игр, сыгранных в ничью
- Number of losses – количество проигрышей
- Goal for – количество голов
- Goal against – количество пропущенных голов
- Shots – количество ударов
- Shots on Target – количество ударов в цель

- Fouls – количество фолов
- Corners – количество угловых
- Yellow Cards – количество желтых карточек
- Red Cards – количество красных карточек

Чтобы вернуться в главное меню, следует нажать кнопку *Back to main*.

Сравнение двух команд

Для сравнения двух команд, в разделе *Compare two teams*, пользователь выбирает названия команд *Team1* и *Team2* и нажимает на кнопку *compare*. После этого появляются статистики двух команд, а также график сравнения. Чтобы вернуться в главное меню, следует нажать кнопку *Back to main*.

4. Модель данных.

4.1. NoSQL модель данных.

4.1.1. Описание и расчет объема.

Данные хранятся в **MongoDB**. В БД одна коллекция – *Matches*. В коллекции – документы, содержащие описание матча.

Структура документа

Документы имеют следующие поля:

- **_id** - идентификатор документа, генерируется автоматически. **Тип** – *ObjectID*. **V** = 12b
- **Date** – дата матча. **Тип** – *date* **V** = 8b
- **HomeTeam** – домашняя команда. **Тип** - *string*. $V = 1b * N$, где N - средняя длина строки, пусть $N = 12$. Тогда **V** = 12b.
- **AwayTeam** – гостевая команда. **Тип** - *string*. $V = 1b * N$, где N - средняя длина строки, пусть $N = 12$. Тогда **V** = 12b.
- **FTHG** – голы домашней команды за Full Time. **Тип** – *integer*. **V**= 4b.
- **FTAG** – голы гостевой команды за Full Time. **Тип** – *integer*. **V**= 4b.
- **FTR** – результат за Full Time. **Тип** - *string*. **V** = 1b, т.к. строка состоит из одной буквы ($H=Home\ Win$, $D=Draw$, $A=Away\ Win$).
- **HTHG** – голы домашней команды за Half Time. **Тип** – *integer*. **V**= 4b.

- **HTAG** – голы гостевой команды за Half Time. **Тип** – *integer*. **V**= 4b.
- **HTR** – результат за Full Time. **Тип** - *string*. **V** = 1b, т.к. строка состоит из одной буквы (*H=Home Win, D=Draw, A=Away Win*).
- **Referee** – судья матча. **Тип** - *string*. **V** = 1b * N, где N - средняя длина строки, пусть N = 10. Тогда **V** = 10b.
- **HS** – удары домашней команды. **Тип** – *integer*. **V**= 4b.
- **AS** – удары гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HST** – удары в цель домашней команды. **Тип** – *integer*. **V**= 4b.
- **AST** – удары в цель гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HF** – фолы домашней команды. **Тип** – *integer*. **V**= 4b.
- **AF** – фолы гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HC** – угловые домашней команды. **Тип** – *integer*. **V**= 4b.
- **AC** – угловые гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HY** – желтые карточки домашней команды. **Тип** – *integer*. **V**= 4b.
- **AY** – желтые карточки гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HR** – красные карточки домашней команды. **Тип** – *integer*. **V**= 4b.
- **AR** – красные карточки гостевой команды. **Тип** – *integer*. **V**= 4b.

Объем документа в среднем: 120b.

Объем всей БД: $V = 120b * N$, где N количество документов в БД (N=4028).

$V = 483360b = 472kb$.

Избыточность данных за счет того, что такие поля, как название команды 1, название команды 2, имя рефери, хранятся несколько раз. Пусть средняя длина строки названия команды - 12, количество команд - 20, длина строки, содержащей имя судьи - 10, количество судей - 7. Тогда объем, необходимый для хранения этих данных $V = 12 * 20 + 10 * 7 = 310b$.

4.1.2. Запросы.

Запрос на добавление матча

```
db.matches.insert_one({ "AC": 5, "AF": 11, "AR": 0, "AS": 10, "AST": 9, "AY": 2, "AwayTeam": "Crystal Palace", "Date": "2018-08-11", "FTAG": 2, "FTHG": 0, "FTR": "A", "HC": 5, "HF": 9, "HR": 0, "HS": 15, "HST": 6, "HTAG": 1, "HTHG": 0, "HTR": "A", "HY": 1, "HomeTeam": "Fulham", "Referee": "M Dean" },)
```

Для добавления матча нужен один запрос.

Запрос на поиск матча

Для поиска матча нужен один запрос.

```
db.matches.find( {"AC": 9, "AF": 11, "AR": 0, "AS": 17, "AST": 8, "AY": 3,
$or:[{"AwayTeam" : "West Ham","HomeTeam" : "North Ham"}, {"AwayTeam" :
"North Ham","HomeTeam" : "West Ham"}], "Date": "2014-08-23", "FTAG": 3,
"FTHG": 1, "FTR": "A", "HC": 6, "HF": 14, "HR": 0} )
```

Запрос на получение статистики команды

```
db.matches.aggregate([ {"$match": {"$or": [{"AwayTeam":"Chelsea"},
{"HomeTeam": "Chelsea"}]}}, {"$group": {"_id": "null","Matches": {"$sum":
1},"FTHG": {"$sum": "$FTHG"},"HS": {"$sum": "$HS"},"AS": {"$sum":
"$AS"},"AC": {"$sum": "$AC"},"AF": {"$sum": "$AF"},"AR": {"$sum": "$AR"},
"AS": {"$sum": "$AS"},"AST": {"$sum": "$AST"},"AY": {"$sum": "$AY"},
"FTAG": {"$sum": "$FTAG"},"FTHG": {"$sum": "$FTHG"},"FTR": {"$sum":
"$FTR"},"HC": {"$sum": "$HC"},"HF": {"$sum": "$HF"},"HR": {"$sum":
"$HR"},"HS": {"$sum": "$HS"},"HST": {"$sum": "$HST"},"HTAG": {"$sum":
"$HTAG"},"HTHG": {"$sum": "$HTHG"},"HTR": {"$sum": "$HTR"},"HY": {"$sum":
"$HY"}},}} ])
```

Для получения статистики команды нужен один запрос.

Запрос на сравнение двух команд

```
db.matches.find({"$or": [ {"$and": [{"AwayTeam": teamA}, {"HomeTeam":
teamB}]}, {"$and": [{"AwayTeam": teamB}, {"HomeTeam": teamA}]} ], {"_id":
0}).sort("Date")
```

Для сравнения двух команд нужен один запрос.

4.2. SQL модель данных.

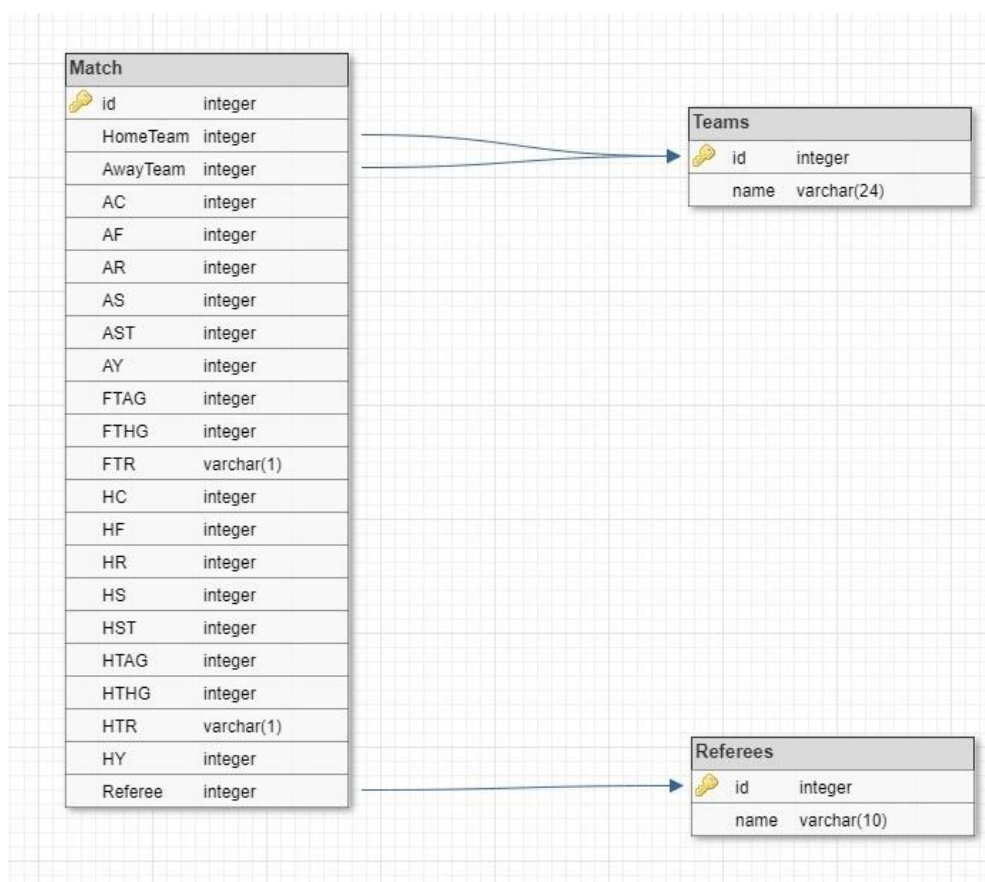


Рисунок 7 – SQL модель данных.

4.2.1. Описание и расчет объема.

База данных состоит из 3 таблиц:

1) **Match** – таблица, содержащая информацию о матче. Включает следующие поля:

- **Id** - порядковый номер матча (*является Primary Key*). **Тип** – *integer*. **V**= 4b.
- **HomeTeam** и **AwayTeam** – идентификаторы команд (*являются Foreign Key*) **Тип** каждого – *integer*. **V**= 4b.
- **FTHG** – голы домашней команды за Full Time. **Тип** – *integer*. **V**= 4b.
- **FTAG** – голы гостевой команды за Full Time. **Тип** – *integer*. **V**= 4b.
- **FTR** - результат за Full Time. **Тип** - *string (varchar(N))*. **V** = 1b, т.к. N=1 - строка состоит из одной буквы (*H=Home Win, D=Draw, A=Away Win*).
- **HTHG** - голы домашней команды за Half Time. **Тип** – *integer*. **V**= 4b.
- **HTAG** - голы гостевой команды за Half Time. **Тип** – *integer*. **V**= 4b.
- **HTR** -результат за Half Time. **Тип** - *string (varchar(N))*. **V** = 1b, т.к. N=1 - строка состоит из одной буквы (*H=Home Win, D=Draw, A=Away Win*).
- **HS** - удары домашней команды. **Тип** – *integer*. **V**= 4b.
- **AS** - удары гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HST** - удары в цель домашней команды. **Тип** – *integer*. **V**= 4b.
- **AST** - удары в цель гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HF** - фолы домашней команды. **Тип** – *integer*. **V**= 4b.
- **AF** - фолы гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HC** - угловые домашней команды. **Тип** – *integer*. **V**= 4b.
- **AC** - угловые гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HY** - желтые карточки домашней команды. **Тип** – *integer*. **V**= 4b.
- **AY** - желтые карточки гостевой команды. **Тип** – *integer*. **V**= 4b.
- **HR** - красные карточки домашней команды. **Тип** – *integer*. **V**= 4b.
- **AR** - красные карточки гостевой команды. **Тип** – *integer*. **V**= 4b.
- **Referee** – идентификатор судьи матча. (*является Foreign Key*). **Тип** – *integer*. **V**= 4b.

Итого: V = 82b.

2) **Teams** – таблица, содержащая названия команд.

Включает следующие поля:

- **Id** - порядковый номер команды (*является Primary Key*) – 4b
- **Name** – название команды. **Тип** - *string (varchar(N))*. **V** = *varchar(24)* = 24b.

Итого: $V = 28b$.

3) **Referees** – таблица, содержащая имена судей.

Включает следующие поля:

- **Id** - идентификатор судьи (*является Primary Key*). **Тип** – *integer*. **V** = 4b.
- **Name** – имя судьи. **Тип** - *string (varchar(N))*. **V** = *varchar(10)* = 10b.

Итого: $V = 14b$.

Рассчитаем суммарный объем для хранения записей: $V = N * 88 + N_t * 28 + N_r * 14$, где $N = 4028$ - количество матчей, $N_t = 20$ - среднее количество команд, $N_r = 7$ - среднее количество судей.

Таким образом, объем для хранения N записей:

$$V = N * 88 + 658$$

Результат: $V = 4028 * 88 + 658 = 355122b = 346,8kb$

4.2.2. Запросы.

Для добавления записи в таблицу используется один запрос.

Примеры:

- Запрос на добавление команд:

```
INSERT INTO Teams ("Man United - Leicester")
```

- Запрос на добавление судьи:

```
INSERT INTO Referees("A Marriner")
```

- Запрос на добавление матча:

```
INSERT INTO Match(1,7,2,5,3,2,5,2,4,1,5,5,1,1,0,4,1,0,1)
```

При поиске создается вложенность запросов, поэтому их количество увеличивается.

Пример:

- Запрос на получение всех матчей домашней команды по названию команды:

```
SELECT * FROM Matches(WHERE HomeTeam = (SELECT id FROM Teams WHERE name = "Man United - Leicester") )
```

Также количество запросов будет расти за счет подготовленных запросов, которые используются для предотвращения возникновения sql- инъекций.

4.3. Сравнение SQL и NoSQL.

1. SQL модель выигрывает по количеству занимаемого объема (*346,8kb - SQL vs 472kb - NoSQL*).
2. В NoSQL происходит избыточность данных, за счёт того, что мы повторно храним такие поля, как название команды 1, название команды 2, имя рефери.
3. В SQL потребовалось создать целых 3 таблицы. В NoSQL используется всего одна коллекция. Выигрывает NoSQL.
4. Количество запросов для NoSQL модели данных гораздо меньше, чем для SQL. Однако запросы SQL модели более читабельны, в них проще вносить изменения.
5. NoSQL реализация будет выигрывать в структуре и скорости сложных запросов.

Выбор сделан в пользу NoSQL модели.

5. Разработанное приложение.

5.1. Краткое описание.

Веб-приложение. В БД хранятся матчи Английской Премьер-лиги. Пользователь приложения может осуществлять поиск матча по выбранным параметрам, производить сравнение двух команд, а также получить информацию о статистике каждой команды.

5.2. Схема экранов приложения.

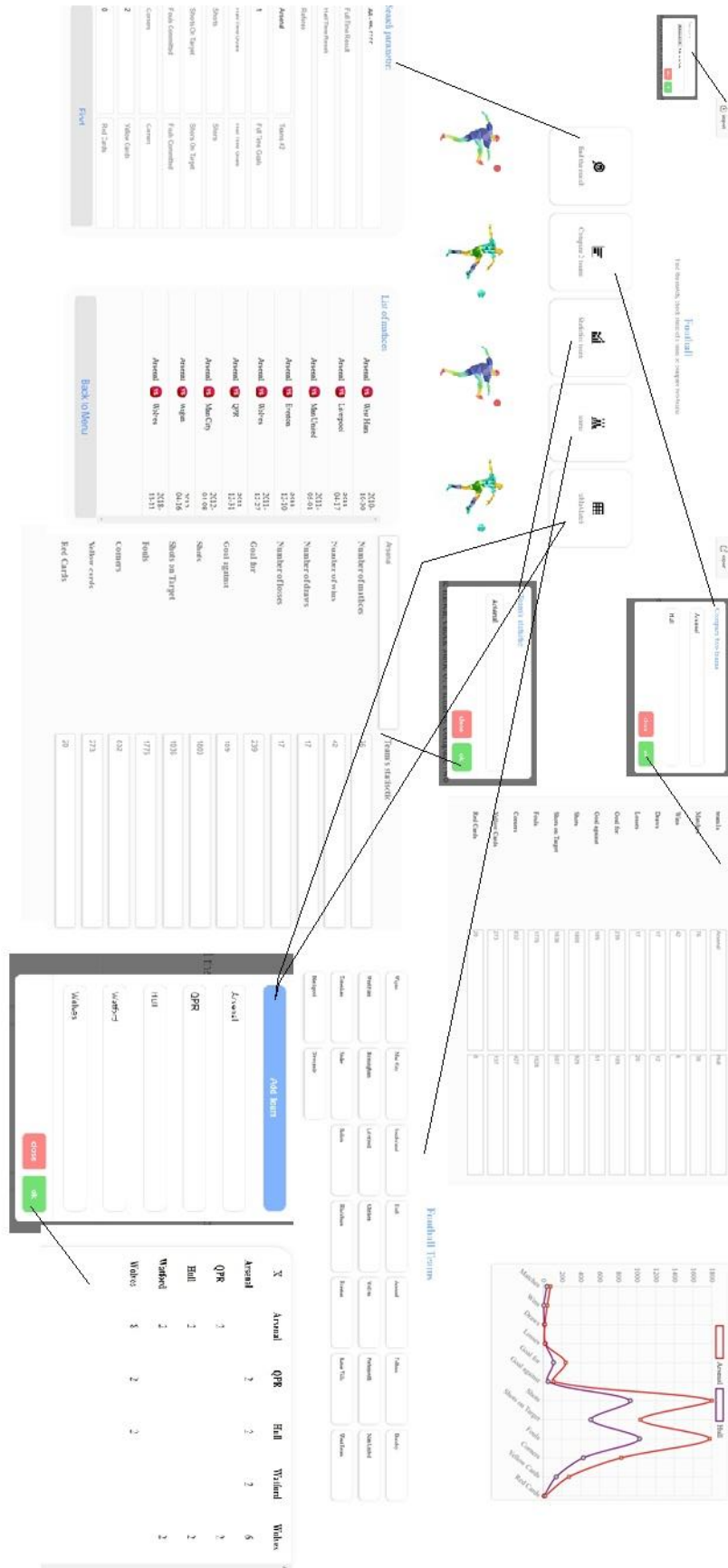


Рисунок 8 – Схема экранов приложения

5.3. Используемые технологии.

- Python 3.7.
- MongoDB
- Flask- Python microframework based on Werkzeug, Jinja 2 and good intentions
- Html
- Css
- Java script

5.4. Ссылки на приложение.

Ссылка на приложение:

<https://github.com/moevm/nosql1h19-football-stats>

6. Вывод.

6.1. Результаты.

В ходе работы было разработано приложение для хранения, анализа и использования данных футбольных матчей. В учебных целях были добавлены импорт и экспорт датасетов. Также была добавлена сводная таблица с результатами матчей выбранных команд, а также список со всеми футбольными командами, участвующими в Английской Премьер-лиге. Была предусмотрена возможность добавлять новые команды.

6.2. Недостатки и пути для улучшения полученного решения.

На данный момент импорт больших датасетов может занимать длительное время. Одним из способов решения может быть использование другого алгоритма импорта, с написанием python скриптов для обработки датасета.

ПРИЛОЖЕНИЕ.

Документация по сборке и разворачиванию приложения.

Инструкция по локальному запуску:

1. установить виртуальную среду
`python -m venv venv`
2. создать виртуальную среду
`virtualenv venv`
3. активировать виртуальную среду
`venv\Scripts\activate`
4. установка нужных пакетов
`pip install flask`
`pip install pymongo`
`pip install numpy`
5. импортирование flask
`python`
`import flask`
`exit()`
6. установка переменной среды FLASK_APP
`set FLASK_APP=nosql.py`
7. запуск приложения
`flask run`
8. Перейти по адресу <http://localhost:5000/>

Снимки экранов приложения.

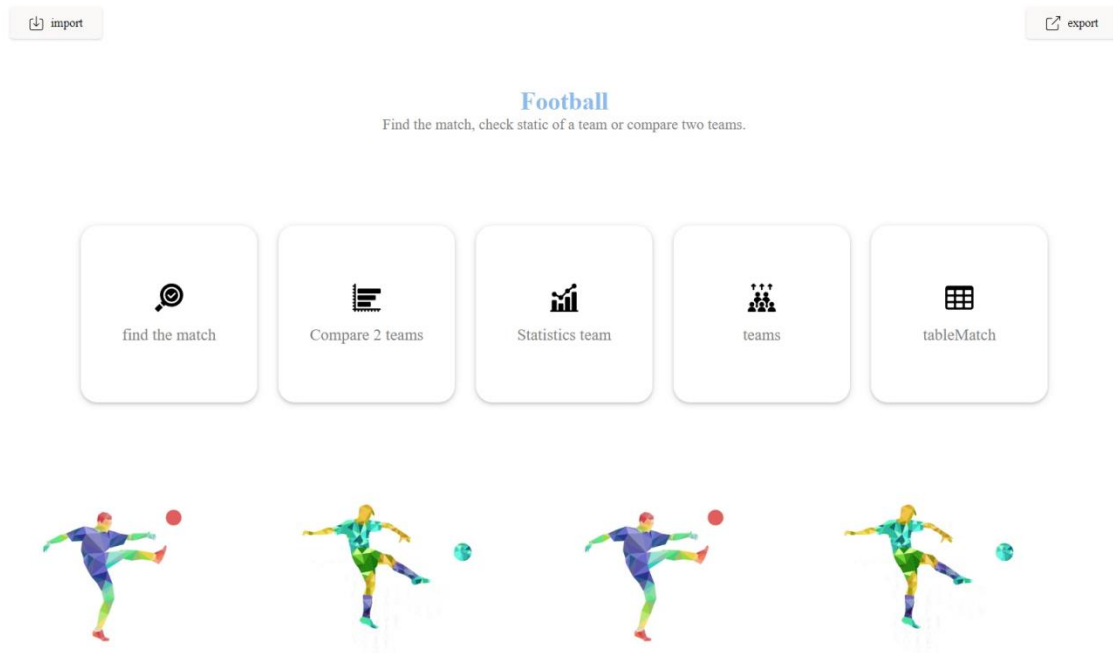


Рисунок 9 – Главная страница.

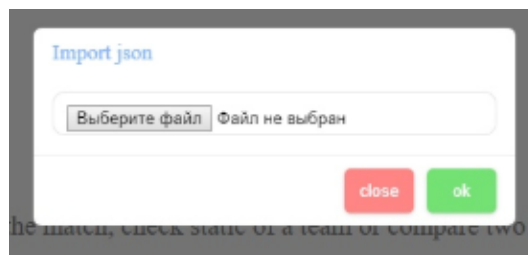


Рисунок 10 – Импорт базы данных.

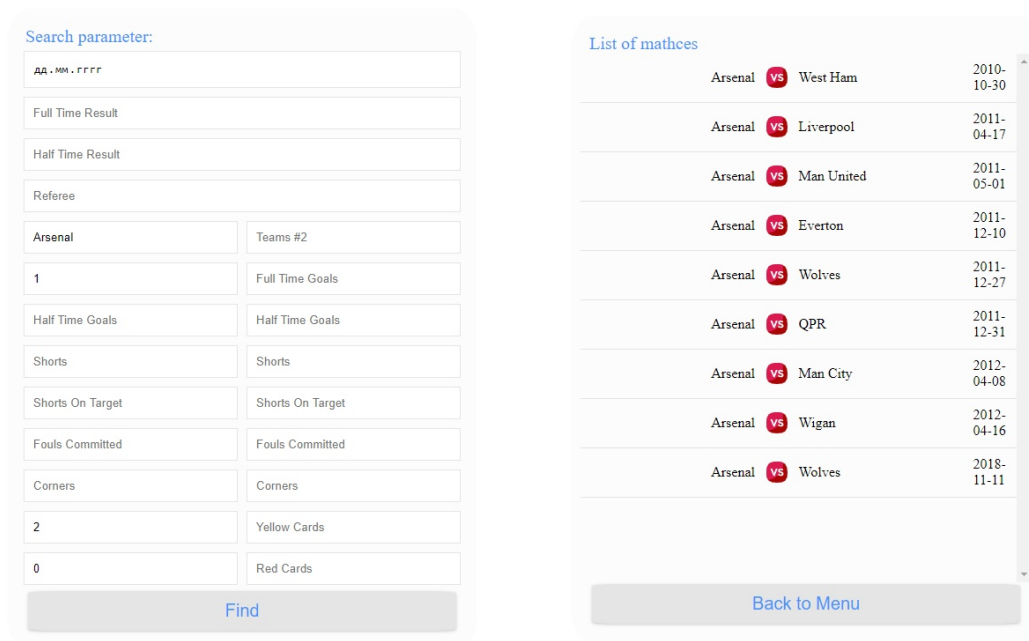


Рисунок 11-12 – Поиск матча.

Compare two teams

Arsenal

Hull

close ok

teams	Arsenal	Hull
Matches	76	38
Wins	42	6
Draws	17	12
Losses	17	20
Goal for:	239	109
Goal against	109	51
Shots	1800	929
Shots on Target	1036	507
Fouls	1776	1028
Corners	832	427
Yellow Cards	273	137
Red Cards	20	8

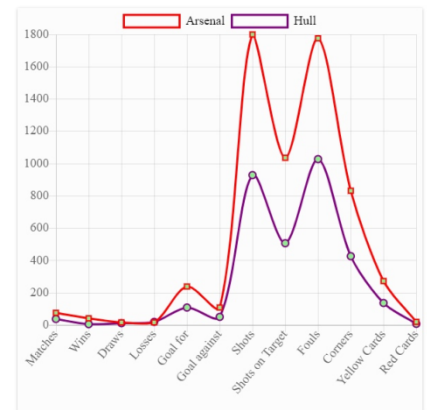


Рисунок 13-14 – Сравнение двух команд.

Team's statistic

Arsenal

close ok

Arsenal	Team's statistic
Number of mathces	76
Number of wins	42
Number of draws	17
Number of losses	17
Goal for	239
Goal against	109
Shots	1800
Shots on Target	1036
Fouls	1776
Corners	832
Yellow cards	273
Red Cards	20

Рисунок 15-16 – Статистика команды.

Football Teams

Wigan	Man City	Sunderland	Hull	Arsenal	Fulham	Burnley
West Ham	Birmingham	Liverpool	Chelsea	Wolves	Portsmouth	Man United
Tottenham	Stoke	Bolton	Blackburn	Everton	Aston Villa	West Brom
Blackpool	Newcastle					

Рисунок 17 – Список команд.

Add Team

Arsenal

QPR

Hull

Watford

Wolves

close

ok

X	Arsenal	QPR	Hull	Watford	Wolves
Arsenal		2	2	2	6
QPR	2				2
Hull	2				2
Watford	2				2
Wolves	8	2	2		

Рисунок 18-19 – Таблица с результатами матчей выбранных команд.

Список литературы.

1. Макет UI: <https://app.moqups.com/jNMJTa8zST/view/page/ad64222d5>
2. Репозиторий проекта: <https://github.com/moevm/nosql1h19-football-stats>
3. Документация MongoDB: <https://docs.mongodb.com>