

Łączy nas Scala — sprawozdanie

1.Opis projektu

Projekt składa się aplikacji internetowej umożliwiającej organizację turniejów koszykówki. API udostępnia narzędzia do tworzenia, edycji oraz usuwania zespołów, zawodników oraz samych turniejów i rozgrywanych na nich spotkań. Serwis ma również zaimplementowane uwierzytelnianie i autoryzację użytkowników .

Do części frontendowej użyliśmy frameworka Scala Play, korzystającego z dobrodziejstw paradygmatu funkcyjnego. Baza danych została utworzona na klastrze dokumentowej bazy MongoDB. Aplikacja łączy się z bazą za pomocą serwera napisanego w środowisku Node.js za pomocą frameworka Express.

W ramach udogodnienia używania projektu zarówno aplikacja webowa jak i serwer mają przygotowane pliki Dockerfile, najlepiej je uruchomić na osobnych terminalach:

```
$ cd scalaProject-main/server
$ docker build -t server .
$ docker run -p 3001:3001 server
```

oraz

```
$ cd ../webapp/target/docker/stage
$ docker build -t webapp .
$ docker run --network=host -p 9000:9000 webapp
```

Konieczne jest uruchomienie serwera w pierwszej kolejności, gdyż aplikacja webowa posiada autoryzację łączącą się od razu z bazą.

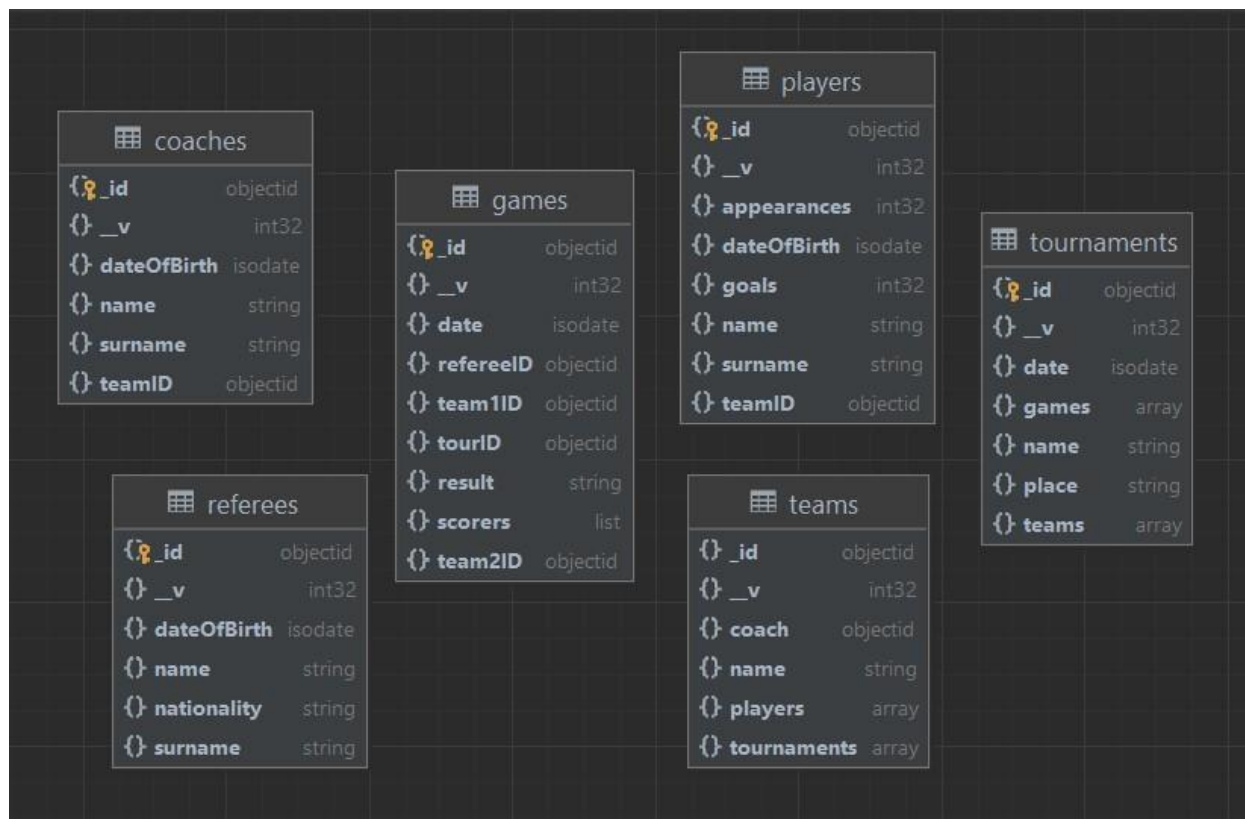
2.Struktura bazy danych

Ze względu na potrzebę zapewnienia wszystkich informacji i statystyk dotyczących rozgrywek zdecydowaliśmy, że w bazie powinny znaleźć się następujące dane:

- turniej - kto w nim uczestniczył, jakie mecze zostały rozegrane, gdzie i kiedy się odbył
- mecz - kto z kim grał, o której godzinie dokładnie, jaki był wynik, kto sędziował i kto zdobywał punkty (zrezygnowaliśmy z kartek i asyst, na naszych turniejach wszyscy grają przepisowo :))))
- drużyna - w jakich turniejach uczestniczyła, jakich posiada zawodników, jakiego trenera
- trener - czy jest zatrudniony; jeśli tak to w jakim klubie
- zawodnik - w jakim klubie gra, ile ma występów a ile punktów zdobył
- sędzia - jakiej jest narodowości (organizatorzy turniejów stawiają na multikulturowość)

Oczywiście stosowne dokumenty powinny mieć podstawowe informacje typu imię i nazwisko, data urodzenia lub nazwa ogólna.

Początkowo wytypowaliśmy poniższą strukturę bazy dokumentowej do naszego projektu:



Dokładny opis kolekcji z przykładami zawieramy poniżej:

1. **players** — kolekcja zawierająca dane oraz statystyki zawodników (występy i bramki), każdy zawodnik może należeć do maksymalnie 1 klubu, może też być wolnym zawodnikiem
 - a. *name* - imię zawodnika (String)
 - b. *surname* - nazwisko zawodnika (String)
 - c. *dateOfBirth* - data urodzenia (Date)
 - d. *appearances* - liczba występów (Int)
 - e. *goals* - liczba zdobytych punktów (Int)
 - f. *teamID* - identyfikator klubu zawodnika (ObjectId)

```
{
  "_id": {"$oid": "628198719f6938c26998f69d"},
  "__v": 0,
  "appearances": 0,
  "dateOfBirth": {"$date": "2000-02-28T23:00:00.000Z"},
  "goals": 0,
  "name": "Bartosz",
  "surname": "Wiśniewski",
  "teamID": {"$oid": "62829c5030c158c42a071d3c"}
},
```

Zawodnicy (ze względu na dodanie informacji o zdobywcach punktów w poszczególnych meczach) nie mogli być wyłącznie zagnieżdżonymi dokumentami wewnątrz zespołów.

2. **coaches** - kolekcja zawierająca informacje o trenerach

- name* - imię trenera (String)
- surname* - nazwisko trenera (String)
- dateOfBirth* - data urodzenia (String)
- teamID* - identyfikator prowadzonej drużyny (ObjectId)

```
{
  "_id": {"$oid": "6282ea5396509a3a077f4eac"},
  "__v": 0,
  "dateOfBirth": {"$date": "1972-03-14T23:00:00.000Z"},
  "name": "Józef",
  "surname": "Lustgarten",
  "teamID": {"$oid": "6284ee916498af0b0f7e36c9"}
},
```

Wstępnie rozważaliśmy trenerów jako zagnieżdżone dokumenty w drużynach, jednakże w trakcie pracy wyszło, że przydałaby się grupa trenerów bezrobotnych, których można przydzielać i wymieniać między klubami w związku z czym uczyniliśmy z ich osobną kolekcją.

3. **teams** — kolekcja zawierająca dane o klubach, w tym identyfikatory ich zawodników oraz turnieje, w których uczestniczył zespół

- name* - nazwa klubu (String)
- coachID* - identyfikator trenera (ObjectId)
- players* - tablica identyfikatorów zawodników (Array[ObjectId])
- tournaments* - tablica identyfikatorów turniejów (Array[ObjectId])

```
{
  "_id": {"$oid": "62829a662a6012913c319c66"},
  "__v": 0,
  "coach": {"$oid": "62828bf50533c8dcc9473016"},
  "name": "Real Madryt",
  "players": [
    {"$oid": "628198319f6938c26998f699"},
    {"$oid": "628198609f6938c26998f69b"},
    {"$oid": "628198719f6938c26998f69d"},
    {"$oid": "628198f19f6938c26998f69f"}
  ],
  "tournaments": [
    {"$oid": "62848e28c644bcc7aa5b89bf"}
  ]
},
```

4. **referees** - kolekcja zawierająca dane sędziów

- name* - imię sędziego (String)
- surname* - nazwisko sędziego (String)
- dateOfBirth* - data urodzenia sędziego (Date)
- nationality* - narodowość sędziego (String)

```
{
  "_id": {"$oid": "627e2e67210dfa6ba18f0893"},
  "__v": 0,
  "dateOfBirth": {"$date": "1980-03-31T00:00:00.000Z"},
  "name": "Leszek",
  "nationality": "Polska",
  "surname": "Siwik"
},
```

Sędziowie musieli zostać wyeksportowani w związku z możliwością prowadzenia wielu spotkań.

5. **tournaments** - kolekcja zawierająca informacje o rozgrywkach

- name* - nazwa turnieju (String)
- place* - miejsce rozgrywek (String)
- date* - termin rozpoczęcia rozgrywek (Date)
- teams* - tablica identyfikatorów zespołów biorących udział w turnieju (Array[ObjectId])
- games* - tablica identyfikatorów spotkań rozegranych w ramach turnieju (Array[ObjectId])

```
{
  "_id": {"$oid": "62848e28c644bcc7aa5b89bf"},
  "__v": 0,
  "date": {"$date": "2022-05-12T22:00:00.000Z"},
  "games": [
    {"$oid": "62848e28c644bcc7aa5b89c8"},
    {"$oid": "62848e28c644bcc7aa5b89cb"},
    {"$oid": "62848e28c644bcc7aa5b89ce"}
  ],
  "name": "Liga Mistrzów",
  "place": "Wilamowice",
  "teams": [
    {"$oid": "62829a662a6012913c319c66"},
    {"$oid": "62829b412a6012913c319c87"},
    {"$oid": "62829c5030c158c42a071d3c"}
  ]
},
```

6. **games** - kolekcja zawierająca dane dotyczące meczów między zespołami
- tournamentID* - identyfikator turnieju w ramach którego rozgrywany był mecz (ObjectId)
 - team1ID* - identyfikator pierwszej drużyny (ObjectId)
 - team2ID* - identyfikator drugiej drużyny (ObjectId)
 - refereeID* - identyfikator sędziego (ObjectId)
 - result* - końcowy wynik meczu (String)
 - date* - termin spotkania (DateTime)
 - scorers* - tablica identyfikatorów strzelców bramek (Array[ObjectId])

```
{
  "_id": {"$oid": "62848e28c644bcc7aa5b89c8"},
  "__v": 0,
  "date": {"$date": "2022-05-20T15:15:00.000Z"},
  "refereeID": {"$oid": "6282e9e296509a3a077f4eab"},
  "result": "3-4",
  "scorers": [
    {"$oid": "628199089f6938c26998f6a1"},
    {"$oid": "628198319f6938c26998f699"},
    {"$oid": "628198719f6938c26998f69d"}
  ],
  "team1ID": {"$oid": "62829a662a6012913c319c66"},
  "team2ID": {"$oid": "62829b412a6012913c319c87"},
  "tourID": {"$oid": "62848e28c644bcc7aa5b89bf"}
},
```

Po analizie modelu uznaliśmy, że kolekcja gier niekoniecznie jest potrzebna osobno (sprawia tylko dodatkowe problemy z związane z pipeline'ami po stronie frontendu), w związku z czym wzięliśmy pod uwagę zagnieżdżenie dokumentów z grami wewnątrz kolekcji turniejów.

Tak wyglądałoby to po zmianie:

```
{
  "_id": {"$oid": "62848e28c644bcc7aa5b89bf"},
  "__v": 0,
  "date": {"$date": "2022-05-12T22:00:00.000Z"},
  "games": [
    {
      "date": {"$date": "2022-05-20T15:15:00.000Z"},
      "refereeID": {"$oid": "6282e9e296509a3a077f4eab"},
      "result": "3-4",
      "scorers": [
        {"$oid": "628199089f6938c26998f6a1"},
        {"$oid": "628198319f6938c26998f699"},
        {"$oid": "628198719f6938c26998f69d"}
      ],
      "team1ID": {"$oid": "62829a662a6012913c319c66"},
      "team2ID": {"$oid": "62829b412a6012913c319c87"}
    },
    {
      "date": {"$date": "2022-05-17T15:30:00.000Z"},
      "refereeID": {"$oid": "6282e9c396509a3a077f4eaa"},
      "result": "0-0",
      "scorers": [],
      "team1ID": {"$oid": "62829b412a6012913c319c87"},
      "team2ID": {"$oid": "62829c5030c158c42a071d3c"}
    }
  ],
  "name": "Liga Mistrzów",
  "place": "Wilamowice",
  "teams": [
    {"$oid": "62829a662a6012913c319c66"},
    {"$oid": "62829b412a6012913c319c87"},
    {"$oid": "62829c5030c158c42a071d3c"} ]
}
```

Jednakże, okazało się, że dostęp do wszystkich gier jest potrzebny w widoku głównym, gdzie wyświetlają się zbliżające się mecze. Zrezygnowaliśmy zatem z tego pomysłu.

Na głównej stronie aplikacji wyświetlamy dokumenty uszeregowane według pewnych pól. Nie ma chyba lepszej możliwości do skorzystania z indeksów:

a.) indeks porządkujący graczy według strzelonych bramek i rozegranych spotkań


```
db.players.getIndexes()
```

Output Result 3

2 rows

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2
2	{"goals": new NumberInt("1"), "appearances": new NumberInt("-1")}	goals_1_appearances_-1	2

b.) indeks porządkujący turnieje według daty rozpoczęcia

```
db.tournaments.getIndexes()
```

Output Result 4

2 rows

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2
2	{"date": new NumberInt("1")}	date_1	2

c.) indeks porządkujący mecze według daty rozegrania

```
db.games.getIndexes()
```

Output Result 5

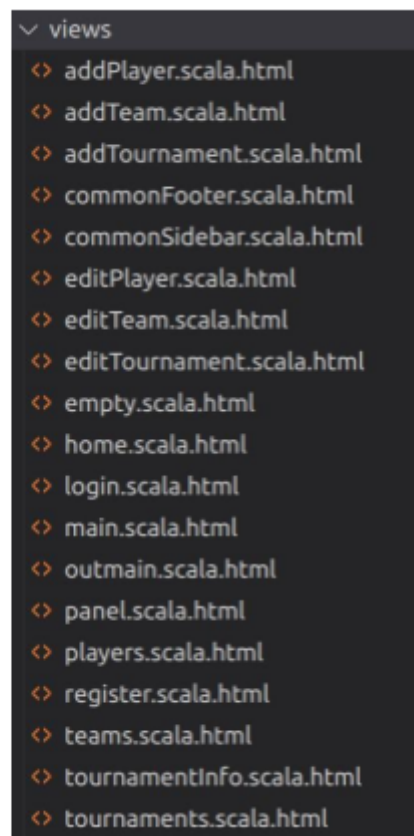
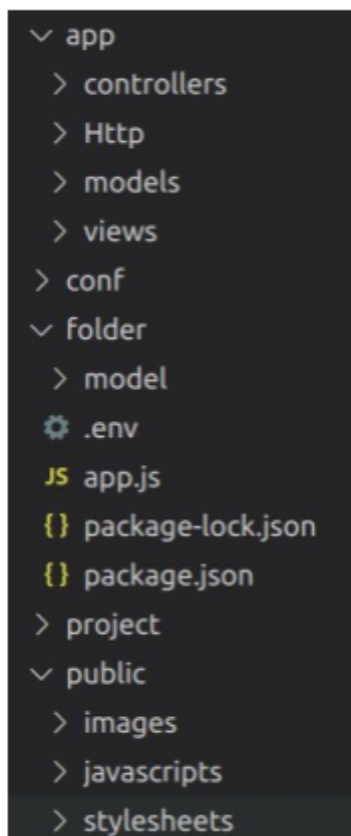
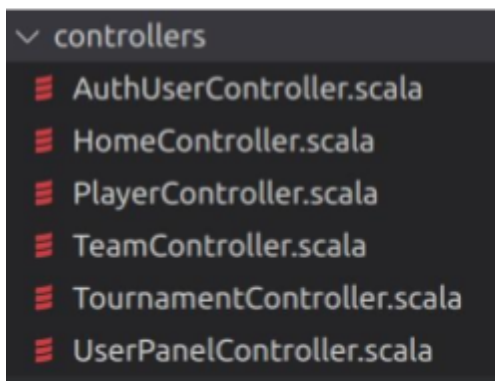
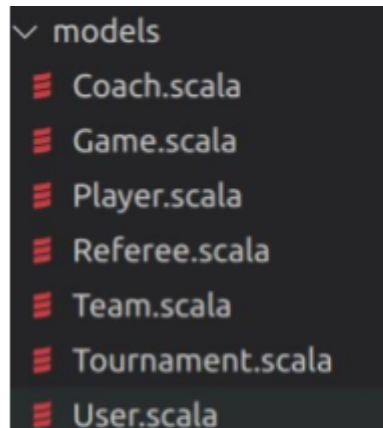
2 rows

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2
2	{"date": new NumberInt("1")}	date_1	2

3. Aplikacja Frontendowa

Głównym problem jaki napotkaliśmy w pracy z frameworkiem Scala Play była bardzo uboga dokumentacja oraz brak możliwości odwoływania się do globalnych serwisów lub zmiennych.

Struktura projektu (uwzględniając pliki, w które istotnie zawierały efekt naszej pracy) wyglądała następująco:



Każdy widok aplikacji odpowiadający kolekcji z bazy danych (Tournaments, Players, Teams) musiał mieć osobno wstrzykiwane kolekcje obiektów

```
Ok((views.html.teams("addTeam"))(views.html.addTeam(teamForm)(players)(availableCoaches))(teams)(coaches))
```

Użyliśmy do tego mechanizmów case classes w celu mapowania pobranych z bazy kolekcji dokumentów na kolekcję konkretnych obiektów oraz pipeline'ów dzięki którym bardzo łatwe okazało się lawirowanie między danymi i wyszukiwanie tych pożądaných.

```
case class Player (val _id: String, var name : String,
                  var surname : String, val dateOfBirth : Date,
                  var goals : Int, var appearances : Int, var teamID : String){
  def getData(): String = {
    name + " " + surname
  }
}
```

```
val tournaments = Http.HttpRequestHandler.getTournaments.filter(el => el.teams.contains(index))
val players = Http.HttpRequestHandler.getPlayers.filter(el => el.teamID == index)
```

Ciekawym mechanizmem okazały się argumenty typu implicit, które w przypadku nie określenia jawnie argumentów są automatycznie dostarczane przez kompilator choćby przy obsłudze requestów i wyświetlaniu wiadomości:

```
@(playerForm: Form[playerData])(teams:Seq[Team])
|(implicit request: RequestHeader, messagesProvider: MessagesProvider)
```


Po uruchomieniu aplikacji powita nas widok zalogowania (można również się zarejestrować):

ŁĄCZY NAS SCALA

LOGIN

Username:

mateusz

Password:

.....

Submit

[REGISTER](#)

SM Group - All rights reserved

Na głównej stronie aplikacji wyświetlają się najlepsi zawodnicy (liczba punktów) oraz zbliżające się spotkania i turnieje:

ŁĄCZY NAS SCALA

HomeTournamentsTeamsPlayersPanelLogout

Top scorers

Min number of points scored

1

Show

Jimmy Butler	Miami Heat	62
Adam Nowak	Miami Heat	60
Mateusz Skowron	Chicago Bulls	30
Jayson Tatum	Boston Celtics	6
Krzysztof Ibisz	Free player	2

Upcoming games

Chicago Bulls	Miami Heat	2022-06-29 15:50
Chicago Bulls	Boston Celtics	2022-06-30 06:00
Chicago Bulls	Miami Heat	2022-07-09 12:55
Miami Heat	Boston Celtics	2022-07-09 14:50
Chicago Bulls	Boston Celtics	2022-07-10 10:50

Upcoming tournaments

Second Tournament	2022-06-29	Warsaw
Third Tournament	2022-07-09	New York
Fourth Tournament	2022-10-27	Chicago

W kolejnych zakładkach znajdują się narzędzia do dodawania, edytowania i usuwania dokumentów z konkretnych kolekcji:

a.) Turnieje

ŁĄCZY NAS SCALA

HomeTournamentsTeamsPlayersPanelLogout

Name

Place

Data

dd.mm.rrrr

Teams

Chicago Bulls

Miami Heat

Boston Celtics

Add tournament

Tournament name	Place	Date		
First Tournament	Cracow	2022-06-09	Show	Edit
Second Tournament	Warsaw	2022-06-29	Show	Edit
Third Tournament	New York	2022-07-09	Show	Edit
Fourth Tournament	Chicago	2022-10-27	Show	Edit

SM Group - All rights reserved

b.) Szczegółowy widok wewnątrz turnieju

ŁĄCZY NAS SCALA

HomeTournamentsTeamsPlayersPanelLogout

Second Tournament

Game

Chicago Bulls Miami Heat

Result

Date

dd.mm.rrrr --:--

Referee

Łukasz Andrzejewski

Scorers

Lonzo Ball

Jaylen Brown

Jimmy Butler

DeMar DeRozan

Update game info

Team nr.1	Result	Team nr.2	Referee	Date of game	Scorers of goals
Chicago Bulls		Miami Heat	Łukasz Andrzejewski	2022-06-29 15:50	
Chicago Bulls		Boston Celtics	Janusz Calik	2022-06-30 06:00	
Miami Heat		Boston Celtics	Łukasz Andrzejewski	2022-07-09 14:50	

SM Group - All rights reserved

c.) Drużyny

ŁĄCZY NAS SCALA

Home

Tournaments

Teams

Players

Panel

Logout

Name

Coach

John LEvis

Players

Nikola Jokic

Donovan Mitchel

Luka Doncic

Krzysztof Ibisz

Andrzej Duda

Add team

Team name	Coach		
Chicago Bulls	Franciszek Smuda	Delete	Edit
Miami Heat	Krystian Nowak	Delete	Edit
Boston Celtics	Anthony Killos	Delete	Edit

SM Group - All rights reserved

d.) Zawodnicy

ŁĄCZY NAS SCALA

Home

Tournaments

Teams

Players

Panel

Logout

Name

Surname

Date of birth

dd.mm.rrrr

Team

Chicago Bulls

Points

Appearances

Add player

Name and surname	Team	Date of birth	Appearances	Points		
Mateusz Skowron	Chicago Bulls	2006-01-10	2	30	Delete	Edit
Adam Nowak	Miami Heat	1995-01-10	6	60	Delete	Edit
Jimmy Butler	Miami Heat	1981-07-13	8	62	Delete	Edit
Tyler Herro	Miami Heat	1958-06-14	0	0	Delete	Edit
Kyle Lowry	Miami Heat	1975-03-05	0	0	Delete	Edit
Zach LaVine	Chicago Bulls	1965-07-15	0	0	Delete	Edit
DeMar DeRozan	Chicago Bulls	1985-08-16	0	0	Delete	Edit
Lonzo Ball	Chicago Bulls	1955-07-14	0	0	Delete	Edit
Nikola Sucevic	Chicago Bulls	1974-06-11	0	0	Delete	Edit
Jayson Tatum	Boston Celtics	1990-07-05	2	6	Delete	Edit

e.) Panel z danymi dotyczącymi konta oraz możliwością dodania sędziów.

ŁĄCZY NAS SCALA

Home

Tournaments

Teams

Players

Panel

Logout

Hello test!

Your roles:

User

Your account is OK!

Name

Surname

Date of birth

dd.mm.rrrr

Nationality

England

Add referee

Name and surname	Date of birth	Nationality	
Łukasz Andrzejewski	1992-06-14	Poland	Delete
Grzegorz Czajka	1995-02-14	Poland	Delete
Piotr Pastrusiak	1985-12-13	Poland	Delete
Filip Marek	1943-01-04	Poland	Delete
Janusz Calik	1971-01-12	Poland	Delete
Anthony Edwards	1960-02-16	England	Delete
Dick Bavetta	1986-06-14	England	Delete
Carlos Flor	1975-07-17	Spain	Delete

SM Group - All rights reserved

Kliknięcie Edit w dowolnej kolekcji uaktywnia ukryty formularz edycji istniejących dokumentów w bazie, natomiast Delete usuwa takowe.

Zbieżność imion i nazwisk wśród sędziów i zawodników zupełnie przypadkowa ;)

Mateusz Skowron
Kamil Miśkowiec