

**Dokumentacja**  
**BD1**  
**Projekt**  
**Football Manager**

**Daniel Tomala - 311145**  
**Marcin Kowalczyk - 318677**  
**Stanisław Sikorski - 318720**

**Politechnika Warszawska**  
**Wydział Elektroniki i Technik Informacyjnych**  
**Warszawa 25.01.2023 r.**

### 1. Link do repozytorium na GitLabie

<https://gitlab-stud.elka.pw.edu.pl/dtomala/pap22z-z07>

Skrypty SQL znajdują się pod ścieżką **src/main/sql**

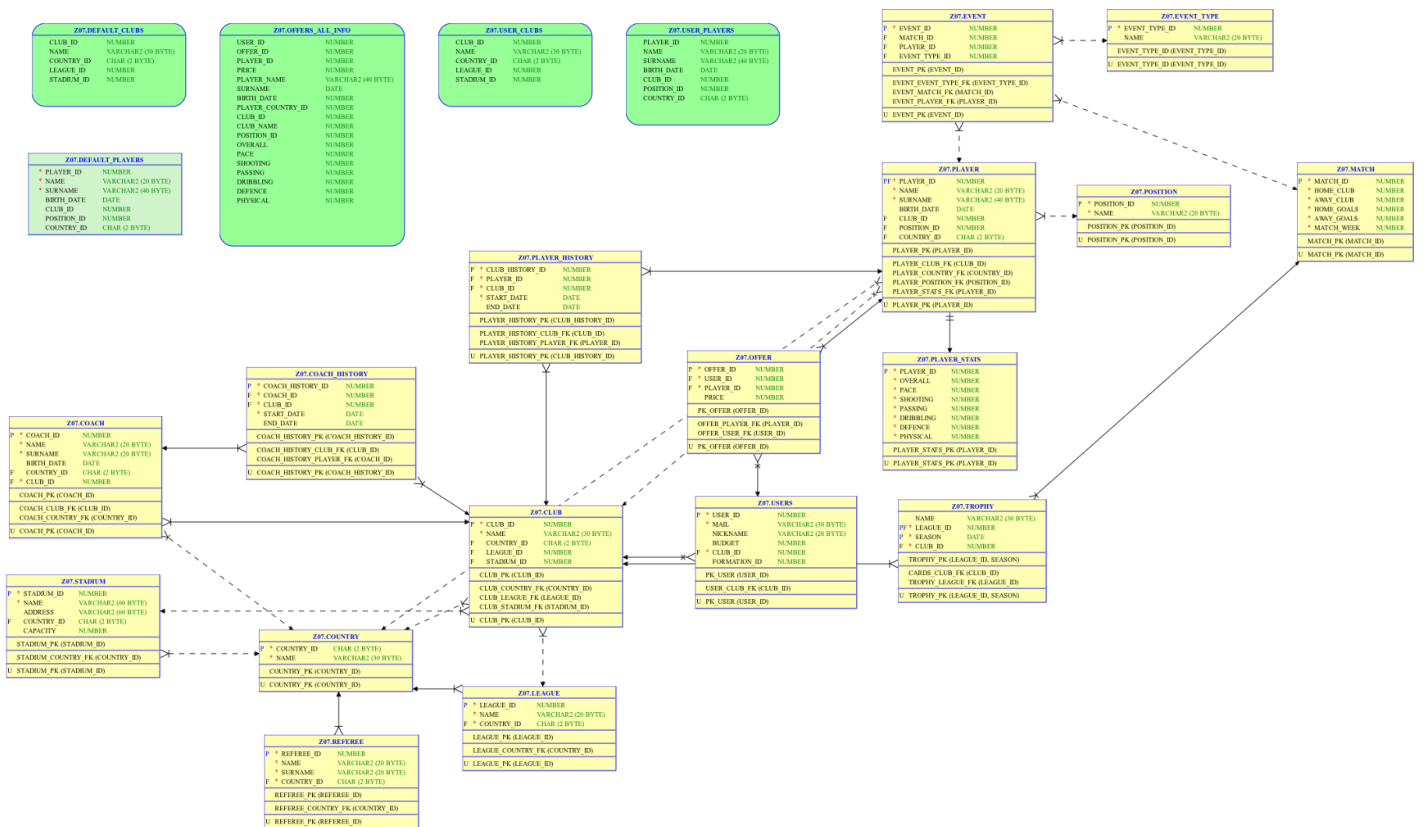
### 2. Tematyka projektu i opis rozwiązania

Celem naszego projektu było zaprojektowanie i przygotowanie bazy danych wykorzystywanej w aplikacji "Football Manager". Aplikacja ta pozwala na założenie przez użytkownika własnego konta oraz zarządzanie własnym klubem piłkarskim. Posiada funkcjonalności tworzenia własnego składu, a co za tym idzie kupowania i sprzedawania zawodników, rozgrywania meczów z komputerowymi rywalami, śledzenie statystyk zawodników, ostatnich meczów itd. Przygotowywana baza danych musiała spełniać wymagania konieczne do prawidłowego funkcjonowania aplikacji.

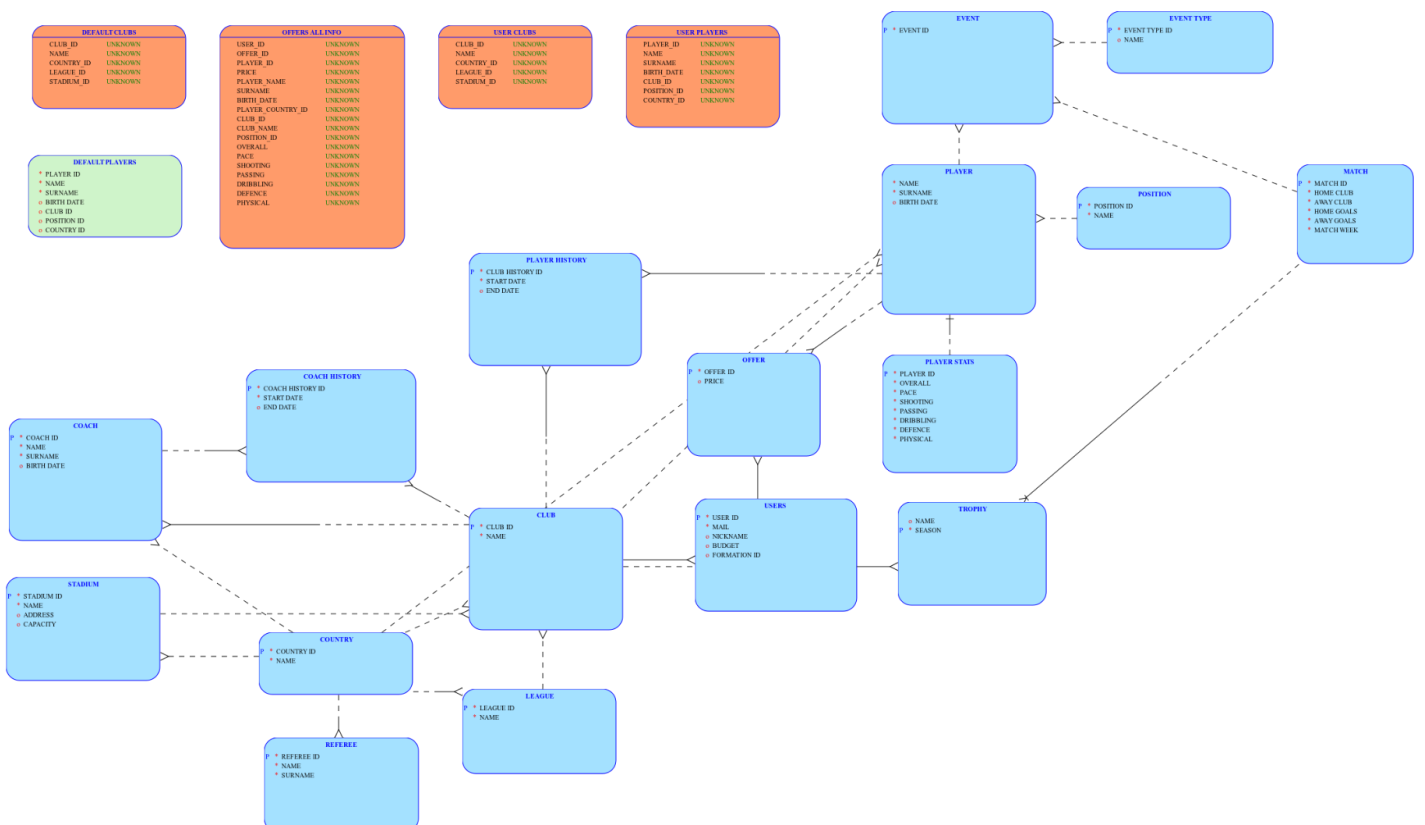
### 3. Tabele

- a. **Club** - tabela przechowująca informacje o klubach zarządzanych przez komputer w rozgrywkach oraz o klubach użytkownika.
- b. **Coach** - tabela przechowująca informacje o trenerach
- c. **Coach\_history** - tabela zawierająca informacje o historii pracy trenera
- d. **Country** - tabela zawierająca nazwy i aliasy krajów
- e. **League** - tabela zawierająca informacje o ligach
- f. **Match** - tabela zawierająca informacje o wszystkich rozegranych meczach
- g. **Offer** - tabela zawierająca informacje o obecnie możliwych do zakupu zawodnikach
- h. **Player** - tabela zawierająca informacje o zawodnikach
- i. **Player\_history** - tabela zawierająca o historii klubów zawodnika
- j. **Player\_stats** - tabela zawierająca informacje na temat statystyk zawodnika
- k. **Position** - tabela zawierająca informacje o pozycji zawodnika
- l. **Referee** - tabela zawierająca informacje o sędziach
- m. **Stadium** - tabela zawierająca informacje o stadionach klubów
- n. **Users** - tabela zawierająca informacje o użytkownikach, ich klubie oraz stosowanej formacji

## 4. Model ER



## 5. Model Relacyjny



## 6. Skrypty

- a. **db-create.sql** - skrypt odpowiedzialny za utworzenie tabel w bazie danych, wraz z wymaganymi powiązaniem.
- b. **db-load.sql** - skrypt odpowiedzialny za wgranie danych do odpowiednich tabel.
- c. **db-functions.sql** - skrypt zawierający funkcje i procedury wykorzystywane w aplikacji.
- d. **db-triggers.sql** - skrypt zawierający triggerzy wywoływane w różnych momentach działania aplikacji.
- e. **db-test-script.sql** - skrypt zawierający testy.

## 7. Aplikacja

W celu włączenia aplikacji, należy z poziomu IDE (najlepiej IntelliJ) uruchomić klasę *Application*. Po chwili konfiguracji aplikacja powinna uruchomić się automatycznie w przeglądarce - jeżeli jednak tak się nie stanie, należy ręcznie wejść w przeglądarce pod adres <http://localhost:8080/>. Ze względu na wymagania projektu na przedmiocie PAP, aby skorzystać z aplikacji konieczne, jest zalogowanie się do niej przy pomocy konta Google (pobierany jest jedynie adres email oraz imię i nazwisko w celu identyfikacji użytkownika).

W aplikacji znajdują się trzy zakładki: *Club*, *Squad* oraz *Market*.

- a. **Club** - pozwala na podgląd tabeli aktualnych rozgrywek (kluby oraz ich punkty/bramki wyświetlane w tabeli wczytywane są z bazy danych), przycisk *Play Match*, pozwala na "rozegranie" (symulowanie) meczu. Wynik jest losowany, a następnie jest on zapisywany do tabeli *Match* w bazie danych. Po rozegranym meczu można zauważyć, że na ekranie głównym zakładki *Club* wyświetla się teraz wynik odbytego spotkania.
- b. **Squad** - pozwala na zarządzanie klubem, wybór zawodników na danej pozycji, formacji oraz podglądu statystyk zawodników. Informacje o zawodnikach oraz ich statystykach są wczytywane z bazy danych. Każdy użytkownik, identyfikowany właśnie na podstawie konta Google, posiada własnych zawodników, których informacje są przetrzymywane w bazie danych.
- c. **Market** - pozwala na zakup oraz sprzedaż zawodników. Zakupieni zawodnicy są dodawani do klubu, natomiast sprzedani z niego usuwani. Początkowe oferty, są automatycznie wygenerowane i zapisane w bazie danych.

## **8. Analiza rozwiązania**

Podczas tworzenia aplikacji, pomimo początkowo ustalonych założeń, kilkakrotnie lekko zmienialiśmy to jak wyglądała nasza baza danych. Wiązało się to z koniecznością modyfikowania, dodawania, usuwania tabel lub danych pól ze względu na sposób implementacji logiki aplikacyjnej. Często przez to musieliśmy całkowicie wyczyścić bazę danych i ładować wszystko od nowa. Na szczęście w bazie wówczas znajdowały się tylko dane, które w łatwy sposób mogliśmy odtworzyć, jednak gdyby w bazie znajdowałyby się dane z aplikacji np. informacje o użytkownikach lub ich dotychczasowym postępie rozgrywki, to takie zmiany w bazie danych byłyby zdecydowanie bardziej problematyczne. Patrząc z perspektywy czasu, mogliśmy lepiej i bardziej szczegółowo opracować początkowy projekt stworzenia bazy danych. Podczas tworzenia naszej aplikacji stosowaliśmy podejście w którym tworzymy bazę danych na potrzeby naszej aplikacji, czego konsekwencją była konieczność dostosowywania bazy danych aby spełniała oczekiwania aplikacji. Gdybyśmy zastosowali odwrotne podejście, tzn. najpierw stworzyli bazę danych i dopiero później tworzyli aplikację do jej obsługi, to z pewnością nie byłoby wymagane wprowadzanie tylu zmian w bazie danych, natomiast wówczas te zmiany byłyby zapewne konieczne w aplikacji.