# Projekt aplikacji „FootballFun”

**Cel:**

Stworzenie aplikacji korzystającej z bazy danych która umożliwi dostęp do informacji o stanie najlepszych lig piłkarskich w Europie oraz danie możliwości rozrywki związanej z piłką nożną po przez tworzenie własnej fikcyjnej ligi.

**Analiza zagadnienia:**

Piłka nożna jest najpopularniejszym sportem na świecie, również w Polsce. Wielu kibiców chce posiadać dostęp do informacji odnośnie ich ulubionych drużyn z najlepszych lig Europy na bieżąco. Istnieje też grupa ludzi poszukującej rozrywki związanej ze światem piłki nożnej.

**Zagadnienia funkcjonalne:**

Użytkownik uruchamiając aplikacje będzie mógł:

* Wyświetlić stan wybranej przez siebie ligi
* Stworzenie konta użytkownika umożliwiające stworzenie własnej ligi
* Stworzyć własną ligę składającą się z wybranych przez użytkownika drużyn oraz symulowanie wyniku meczy pomiędzy wybranymi przez użytkownika drużynami pod warunkiem zalogowania
* Wyświetlić informację o wybranej drużynie
* Wyświetlenie informacji na temat wybranego stadionu

**Zagadnienia niefunkcjonalne:**

Aplikacja będzie miała również funkcję:

* Edycji i usuwania kont użytkowników
* Edycji i usuwania lig użytkowników
* Tworzenia widoków lig
* Sortowania tabeli ligowych wyświetlonych przez użytkownika

**Opis tabel:**

* Zespoły – zawiera informacje o wszystkich zespołach które znajdują się w bazie danych, takie jak kraj i miejscowość pochodzenia itp.
* Stadiony – zawiera informacje na temat stadionów
* Ekstraklasa/La-Liga/Serie A/Premier Leauge/Ligue 1/Bundesliga – są to tabele zawierające aktualny stan w lidze czyli punktu każdej drużyny, ilość rozegranych meczy itp.
* Mecze - przechowuje wyniki meczów wraz z datą rozegrania i nazwą rozgrywek w ramach której był rozegrany
* Użytkownicy – zawiera informacje o kontach użytkowników takie jak poziom dostępu i nazwę ulubionej drużyny oraz nazwę ligi abstrakcyjnej
* Liga\_abstrakcyjna – każdy użytkownik zalogowany będzie mógł stworzyć jedną którą będzie mógł edytować, tablica będzie zawierała informacje o aktualnym stanie ligi
* Mecze\_abstrakcyjne – będzie przechowywała wyniki meczów ze wszystkich lig abstrakcyjnych

**Triggery:**

* aktualizujące dane w tabelach ligowych po każdym meczu
* do sprawdzania poprawności loginu dla nowego użytkownika
* do sprawdzenia poprawności tworzonych lig przez użytkowników
* sprawdzający czy dodawany zespół do puli zespołów nie został dodany wcześniej

**Widoki:**

Będą to widoki utworzone z połączenia tabel danej ligi i tabeli zespoły by przyśpieszyć wyświetlanie stanu tabeli ligowej.

**Procedury/Funkcje:**

* wyświetlające stan danej ligi
* wyświetlająca informacje o stadionie
* wyświetlająca informacje o danym zespole
* sortujące odpowiednio wybraną tabele ligową
* tworzącą nowych użytkowników
* edytująca konto użytkownika
* usuwająca konto użytkownika
* tworzące ligi użytkowników
* generujące wynik fikcyjnego spotkania
* usuwające ligi użytkowników
* edytujące ligi użytkowników

**Dostęp użytkowników:**

1. Użytkownik niezalogowany :

* Przeglądanie wyników meczów, informacji o stadionach
* Przeglądanie stanu tabel ligowych tylko w ligach prawdziwych, nie będzie miał dostępu do lig stworzonych przez użytkowników

1. Użytkownik zalogowany:

* Przeglądanie wyników meczów, informacji o stadionach
* Przeglądanie stanu tabel ligowych tylko w ligach prawdziwych
* Tworzenie własnej ligi
* Wybranie ulubionej drużyny
* Symulowanie wyników meczów

1. Admin:

* Przeglądanie wyników meczów, informacji o stadionach
* Przeglądanie stanu tabel ligowych tylko w ligach prawdziwych
* Tworzenie własnej ligi
* Wybranie ulubionej drużyny
* Symulowanie wyników meczów
  + Wprowadzanie wyników meczów z prawdziwych lig
  + Usuwanie kont użytkowników
  + Aktualizowanie tabel lig prawdziwych, oraz edycja lig użytkowników

**Scenariusz użycia:**

Aktorzy:

* Aplikacja
* Baza danych
* Użytkownik

Warunek wstępny – aplikacja jest uruchomiona i ma dostęp do bazy danych

Warunek końcowy – użytkownik zamyka aplikacje

Scenariusz główny:

1. Aplikacja prosi o zalogowanie się
2. Użytkownik loguje się

WHILE(zalogowany=true)

1. System wyświetla dostępne opcje
2. Użytkownik wybiera jedną z opcji
3. Zostaje wywołana odpowiednia procedura w bazie danych związana z opcją wybraną przez użytkownika

END WHILE

1. Użytkownik wylogowuje się

Scenariusz alternatywny:

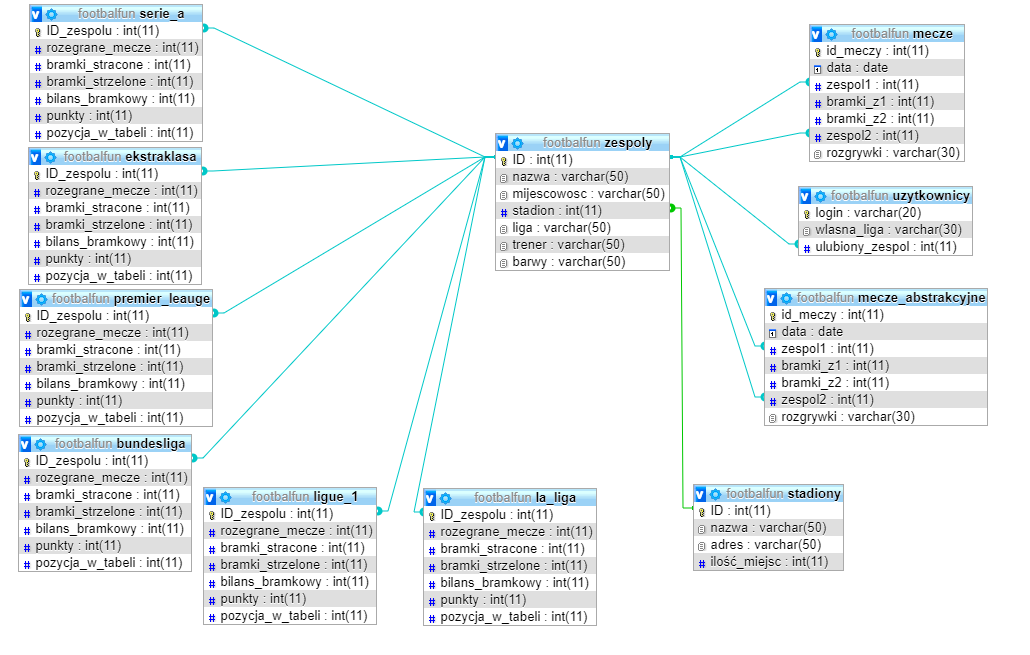
2a. Użytkownik wybiera opcję bez logowania

WHILE(zalogowany=true)

1. System wyświetla dostępne opcje
2. Użytkownik wybiera jedną z opcji
3. Zostaje wywołana odpowiednia procedura w bazie danych związana z opcją wybraną przez użytkownika

END WHILE

**UML Bazy Danych:**



**Autorzy:**

Piotr Klepczyk 236408

Jan Sieradzki 236441