Gliwice, 12.09.2018

Menager piłkarski z bazą danych

Prowadzący: Krzysztof Pasterak

Autor:

Krzysztof Werner

# Temat

Tematem projektu jest stworzenie menagera piłkarskiego opartego o bazę danych.

# Analiza tematu

Podczas analizy tematu musiałem podjąć decyzję jakiej bazy danych użyję, oraz jaką strukturę będzie miał mój program.

Pierwszą decyzja był… wybór bazy. Zastanawiałem się miedzy relacyjną i nierelacyjną bazą danych. Nierelacyjna baza danych pozwoliłaby mi na trzymanie informacji o różnych typach piłkarzy w jednej tabeli bez wielu niepotrzebnych kolumn. Problemem okazał się brak bibliotek, oraz mojego doświadczenia w bazach danych tego typu. Ze względu na niewielkie skomplikowanie modelu zdecydowałem się na sqlite – co oznacza po prostu trzymanie danych w jednym pliku i wyciąganie ich za pomocą sqlowych zapytań.

Co do struktury programu – chciałem, aby mój program był przejrzysty i skalowalny. Postanowiłem go więc podzielić na model – w którym trzymam informację o klasach bazodanowych, Controller – kontroler zarządzający programem, Helpery – czyli wydzielone minikontrolery odpowiedzialne za różne funkcje w programie oraz Dao – czyli warstwę komunikacji z bazą danych. Strukture programu wziąłem z projektów w których pracowałem – dlatego struktura jak i nazewnictwo są bardzo zbliżone do projektów w Javie.

# Specyfikacja zewnętrzna

Instrukcja obsługi jest dostępna w programie pod opcja menu Instructions.

Program posiada interfejs tekstowy dostępny w konsoli. Na razie program oferuje tylko jeden tryb gry. Aby rozpocząć działanie z programem – gdy pojawi się menu należy wybrać opcję 3, gdy program był już wcześniej uruchamiany – jak nie, ten krok można pominąć. Program wtedy czyści bazę danych. Następnie należy wybrać opcję 2 – setup database, która tworzy odpowiednie tabele w bazie danych i wypełnia je początkowymi danymi.

Gdy dane zostaną dodane do bazy można wybrać opcję 1 – new game. Na początku tworzona jest lista meczów na cały sezon. Następnie użytkownik dostaję do wyboru klub. Gdy poprawnie do wybierze – można rozpocząć grę.

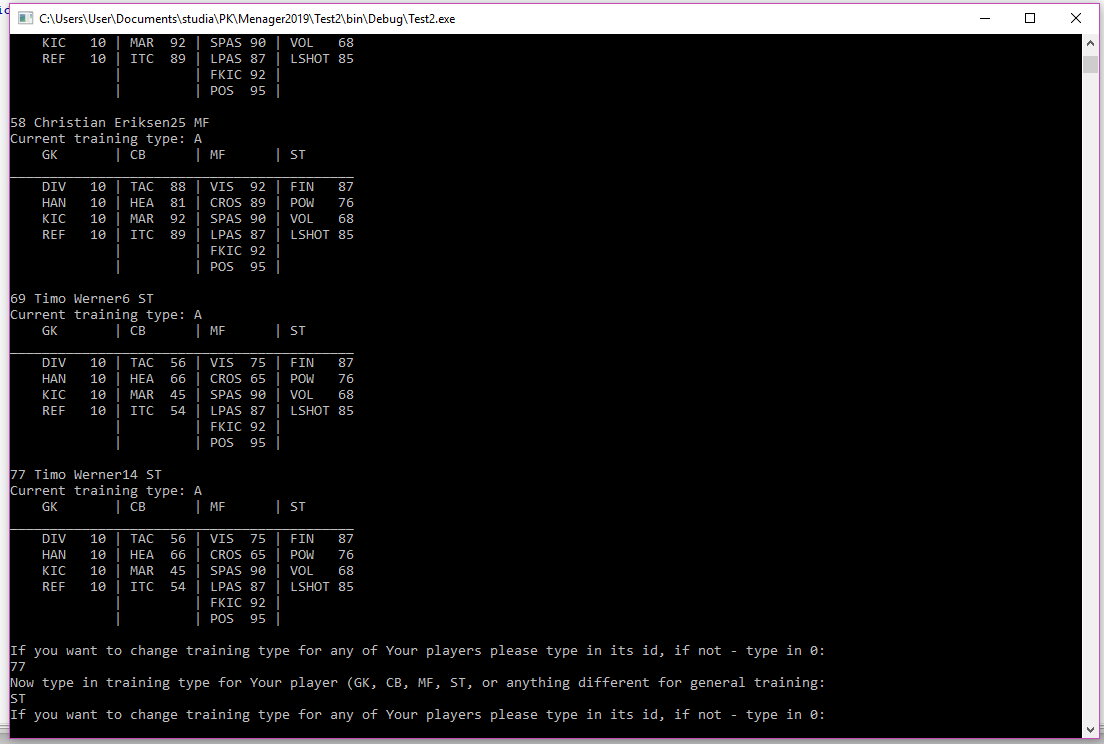
Przed każdym meczem użytkownik jest pytany o taktykę zespołu na dany mecz. Do wyboru jest: ultradefensywna, defensywna, zbalansowana, ofensywna i ultraofensywna. Wybór taktyki zmienia algorytm wyliczający szanse zespołu w danym meczu, oraz strzelone i stracone gole.

Po rozegranym meczu wyświetlane są wyniki w danej kolejce. Następnie użytkownik może ustawić plan treningowy dla każdego z zawodników w swoim zespole. Domyślnie jest wybrany trening zbalansowany. W ramach tego treningu zawodnik MOŻE otrzymać +1 do 3 różnych statystyk. Niestety przy takim treningu – użytkownik nie ma wpływu jakie statystyki zostaną poprawione – może się okazać, że napastnikowi podniosły się umiejętności bramkarskie. Użytkownik może wybrać trening pozycyjny - GK dla bramkarza, CB dla obrońcy, MF dla pomocnika czy ST dla napastnika. Oczywiście można przypisać terning dla napastnika nawet obrońcy. W ramach takiego treningu zawodnikowi MOŻE się podnieść jedna umiejętność w ramach danej gupy umiejętności o +2.

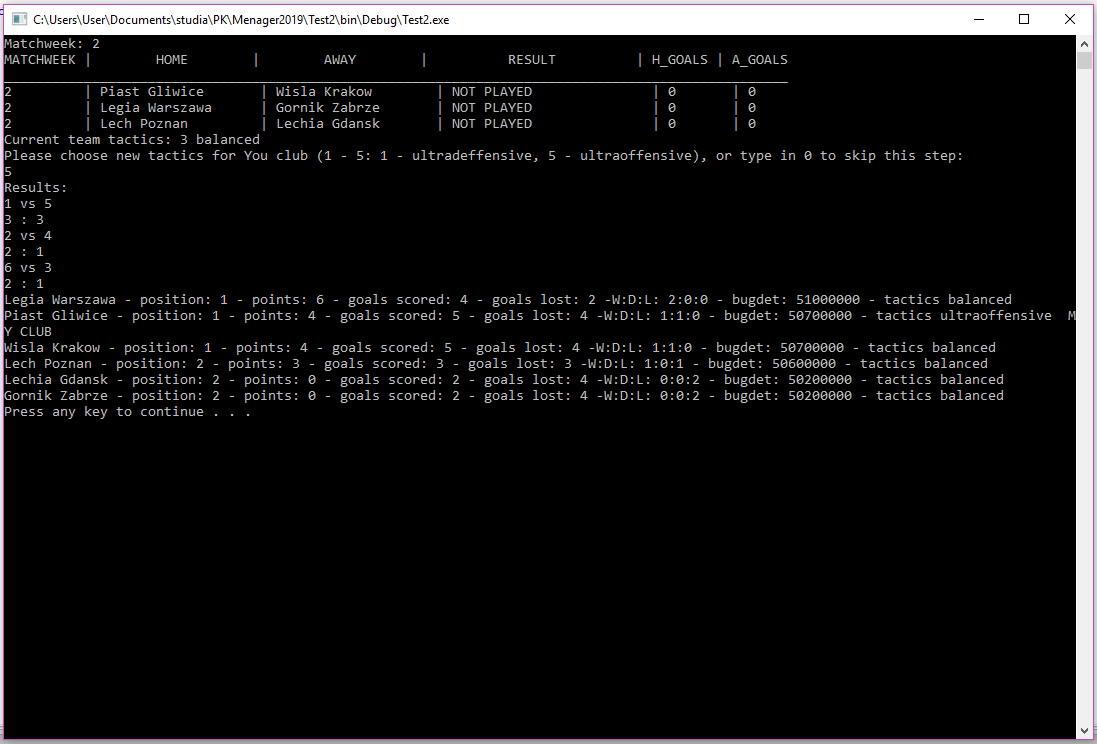
Użytkownik rozgrywa mecze, oraz ustala treningi i taktykę aż do połowy sezonu. Wtedy pojawia się okienko transferowe. W ramach okienka użytkownik ma możliwość zamiany jednego ze swoich zawodników na takiego dostępnego na liście transferowej. W zależności czy transferowany zawodnik będzie miał mniejsze czy większe umiejętności od wytransferowanego zawodnika z konta klubu ubędzie lub przybędzie pieniędzy w budżecie.

Następnie następuje kolejna połowa sezonu zakończona okienkiem transferowym i oficjalnymi wynikami. Użytkownik jest pytany czy chce rozpocząć następny sezon i gdy tak się stanie – kolejny sezon jest tworzony.

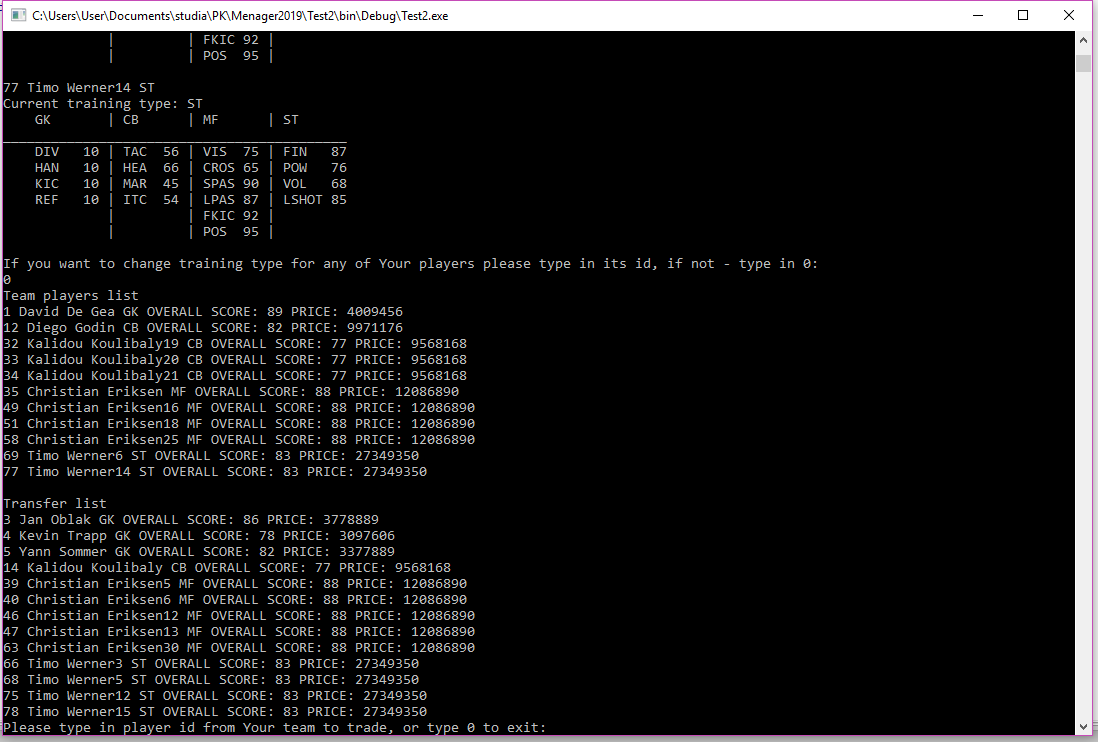
Poniżej – klika zrzutów z działania aplikacji:



Wybór treningu



Mecz



Lista transferowa

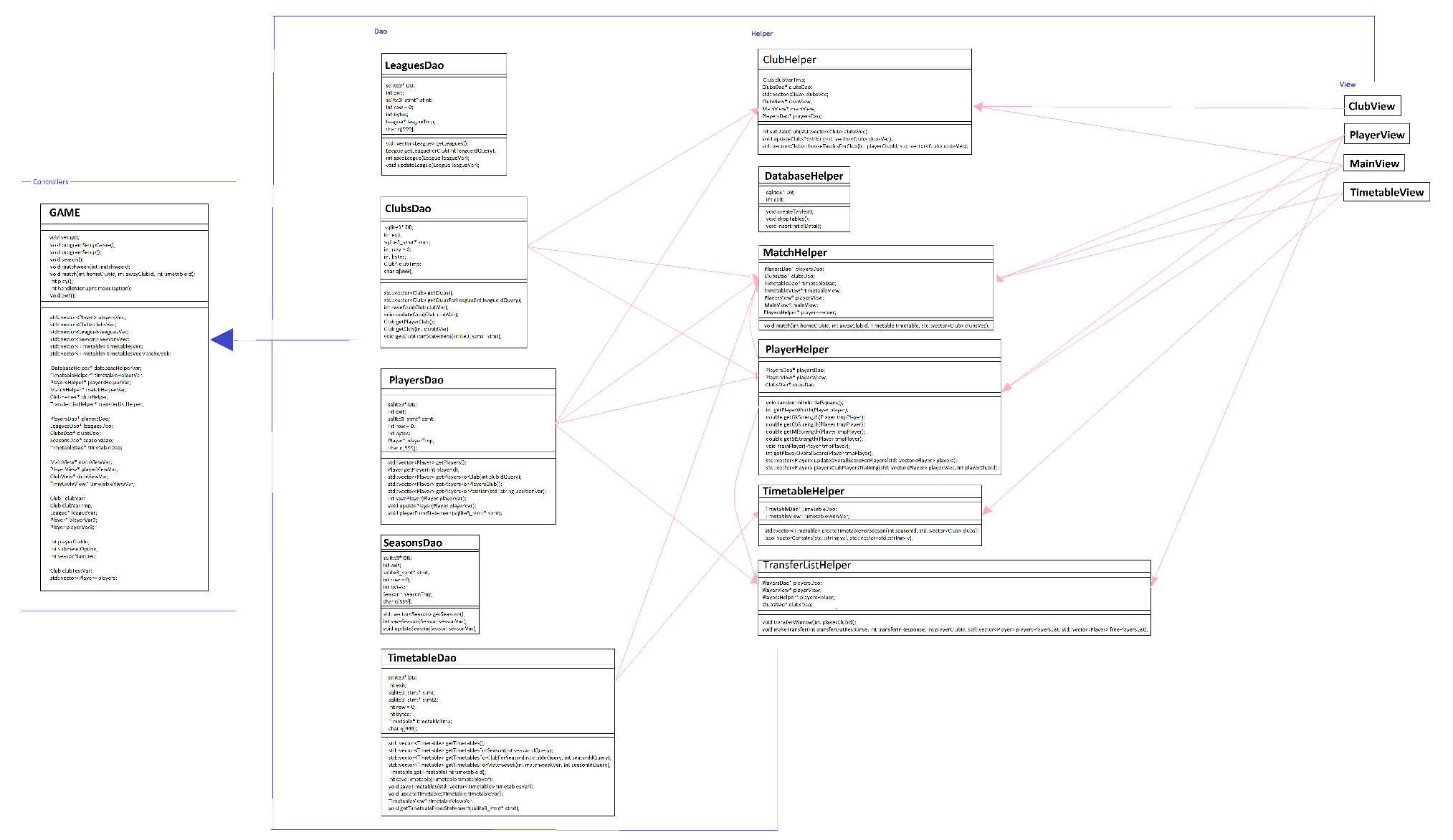
# Specyfikacja wewnętrzna

Ze względu na duża ilość klas osobno prezentuje model i osobno pozostałą część aplikacji (przepraszam za jakość, ale nie potrafiłem użyć programu do tworzenia UMLa).

Model:



Reszta klas:



W ramach tematów zajęć wykorzystano kontenery STL, algorytmy i iteratory STL, wątki (niestety nie std::thread, tylko boost::thread i mutex, ponieważ kompilator gcc nie współpracuje za dobrze z std::thread), oraz wyjątki. Kontenery STL, oraz algorytmy i iteratory zostały wykorzystane do trzymania danych – takich jak lista zawodników, lista klubów, lista meczów oraz wiele, wiele innych. Są obecne niemal w każdej funkcji w programie. Wątki wykorzystano do tego, aby zrównoleglić obliczenia dotyczące na przykład meczów – gdzie zrównoleglono oblicznie siły zawodników dla całego zespołu. Zastosowałem także wyjątki do obsługi nieprawidłowych danych wejściowych użytkownika, oraz obsługi błędów bazy danych.

# Testowanie i uruchamianie

Testy programu były prawdziwą próbą charakteru – zwłaszcza pod koniec, gdy podczas testowania, czy kolejne sezony generują się poprawnie trzeba było rozegrać poprzednie. Przede wszystkim zmuszenie do działania bibliotek zewnętrznych – sqlite3 oraz boost wymagało wielu testów, ze względu na brak Visual Studio i używania kompilatora gcc, który nie zawsze chciał współpracować. A i ja nie miałem z nim doświadczenia. Bardzo dużo początkowych błędów w programie dotyczyło tego, że przy wielokrotnym odczycie danych z bazy czasami przy kopiowaniu obiektów, czy ich pokazywaniu – podczas zapisu pojawiały się dziwne obiekty o id idącym w miliony, podczas gdy powinny mieć mniej niż 100. Na szczęście dzięki inteligentnemu zarządzaniu połączeniami z bazą udało się te błędy pokonać. Dużo pracy kosztowało także zbalansowanie algorytmu meczowego, aby dawał prawdopodobne wyni, uwzględniając szczęście czy przewagę własnego stadionu. W tym pomogły funkcje randomizujące z nowego C++, które są oparte o rozkład normalny. W obecnej wersji program działa stabilnie (piszę obecnej, bo może go będę rozbudowywał – to fajny projekt).

# Uwagi i wnioski

Przy tworzeniu tego projektu patrzyłem z perspektywy programisty Javy i zdziwiło mnie, że C++ to język, w którym można tworzyć duże, skalowalne i czytelne aplikacje. Gdybym jeszcze znał obowiązujące wzorce projektowe w C++, myślę że mógłbym w nim tworzyć naprawdę fajne aplikacje, nie tylko projekty na studia. C++ kojarzył mi się z niezrozumiałą składnią i wskaźników na referencje czy innych rzeczy, których nie rozumie chyba nawet ich twórca, ale ze zdziwieniem odkryłem, że dzięki odpowiednim bibliotekom i uporządkowanej strukturze jest to naprawdę czytelne. Podczas pisania projektu bardzo mnie smucił brak szerokiego wsparcia dla baz danych (zwłaszcza tych sieciowych – np. PostgreSQL, czy MySQLa), oraz brak ORMa, ale zdaję sobie sprawę, że C++ nie jest zwykle wykorzystywany w ten sposób, aby powstanie ORMa było kluczowe.