



Dokumentacja projektu wykonanego
w ramach zajęć
Bazy Danych I

Temat: "NBA Fantasy League"

Autor:
Piotr Woś
WFILS
290536

I. Projekt koncepcji, założenia

1. Zdefiniowanie tematu projektu

Baza danych do serwisu internetowego umożliwiającego użytkownikom udział w grze online, która polega na tworzeniu przez nich własnych drużyn złożonych z zawodników występujących na codzień w NBA.

Drużyny te następnie rywalizują między sobą na podstawie statystyk z występów w rzeczywistych meczach NBA zawodników znajdujących się w ich składach.

2. Analiza wymagań użytkownika

Funkcjonalności bazy danych:

- przechowywanie danych zawodników występujących w NBA
- przechowywanie danych oficjalnych zespołów NBA
- przechowywanie danych o meczach NBA
- przechowywanie statystyk zawodników z każdego meczu NBA
- obsługa danych użytkowników serwisu
- obsługa danych dotyczących zespołów użytkowników (fantasy teams)

3. Zaprojektowanie funkcji

Podstawowe funkcje realizowane w bazie danych:

- możliwość uzyskania informacji o danym zawodniku występującym w NBA - zarówno szczegółowe informacje na

jego temat (np. pozycja, wzrost, numer koszulki), jak i jego statystyki wyrażone jako średnie na mecz

- możliwość uzyskania informacji o danym zespole NBA - szczegółowe dane (np. konferencja, dywizja) oraz jego bilans wygranych oraz przegranych meczy
- możliwość stworzenia użytkownika, dostępu do jego danych, modyfikacji ich, a także usunięcia użytkownika
- możliwość stworzenia drużyny użytkownika (jeśli użytkownik nie posiada jeszcze drużyny) i zapewnienie jej niepowtarzalności
- możliwość zarządzania składem drużyny użytkownika wraz z weryfikacją warunków jakie każda drużyna użytkownika powinna spełniać:
 - suma płac zawodników w składzie wirtualnej drużyny nie może przekraczać budżetu drużyny,
 - w drużynie powinna składać się z dokładnie 5 zawodników,
 - zawodnicy w danej drużynie nie mogą się powtarzać,
 - w drużynie musi być conajmniej dwóch zawodników, którzy mogą grać jako obrońcy (guards),
 - w drużynie musi być conajmniej dwóch zawodników, którzy mogą grać jako skrzydłowi (forwards),
 - w drużynie musi być conajmniej jeden zawodnik, który może grać jako center
- możliwość uzyskania danych swojej wirtualnej drużyny (np. wysokości kontraktów zawodników w niej grających, wysokość budżetu)
- możliwość uzyskania rankingu najlepszych drużyn użytkowników wraz z ich wynikami

II. Projekt diagramów (konceptualny)

4. Data Flow Diagram (DFD)

Diagramy załączone są również w oryginalnej rozdzielczości w folderze "Diagramy" w tym katalogu.

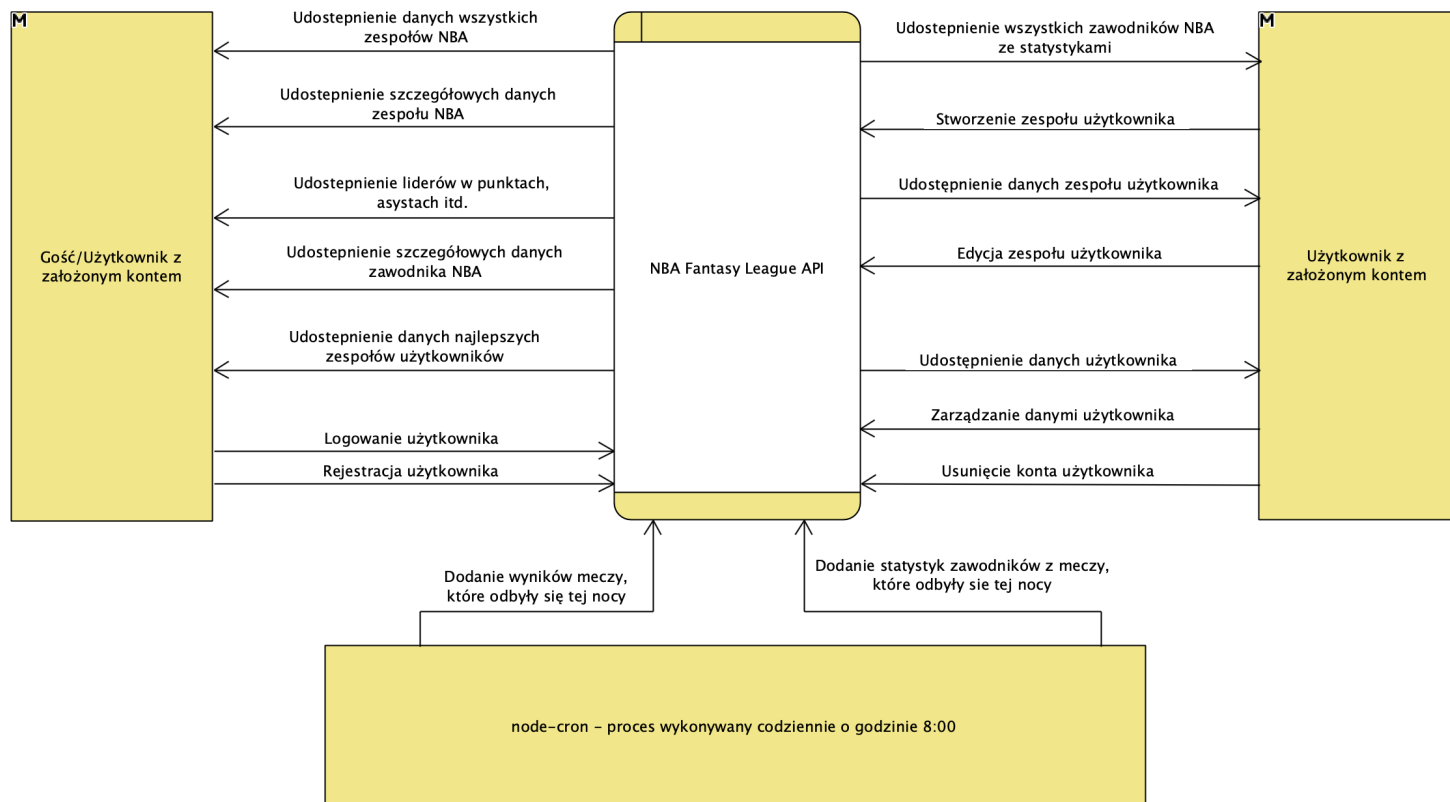


Diagram DFD - kontekst

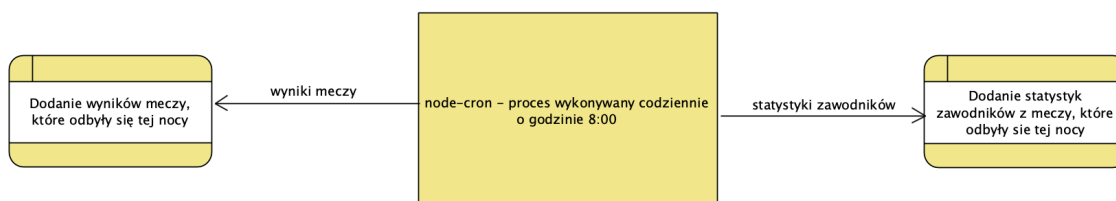


Diagram DFD poziom 0 - node-cron - proces wykonywany automatycznie, codziennie o godzinie 8:00

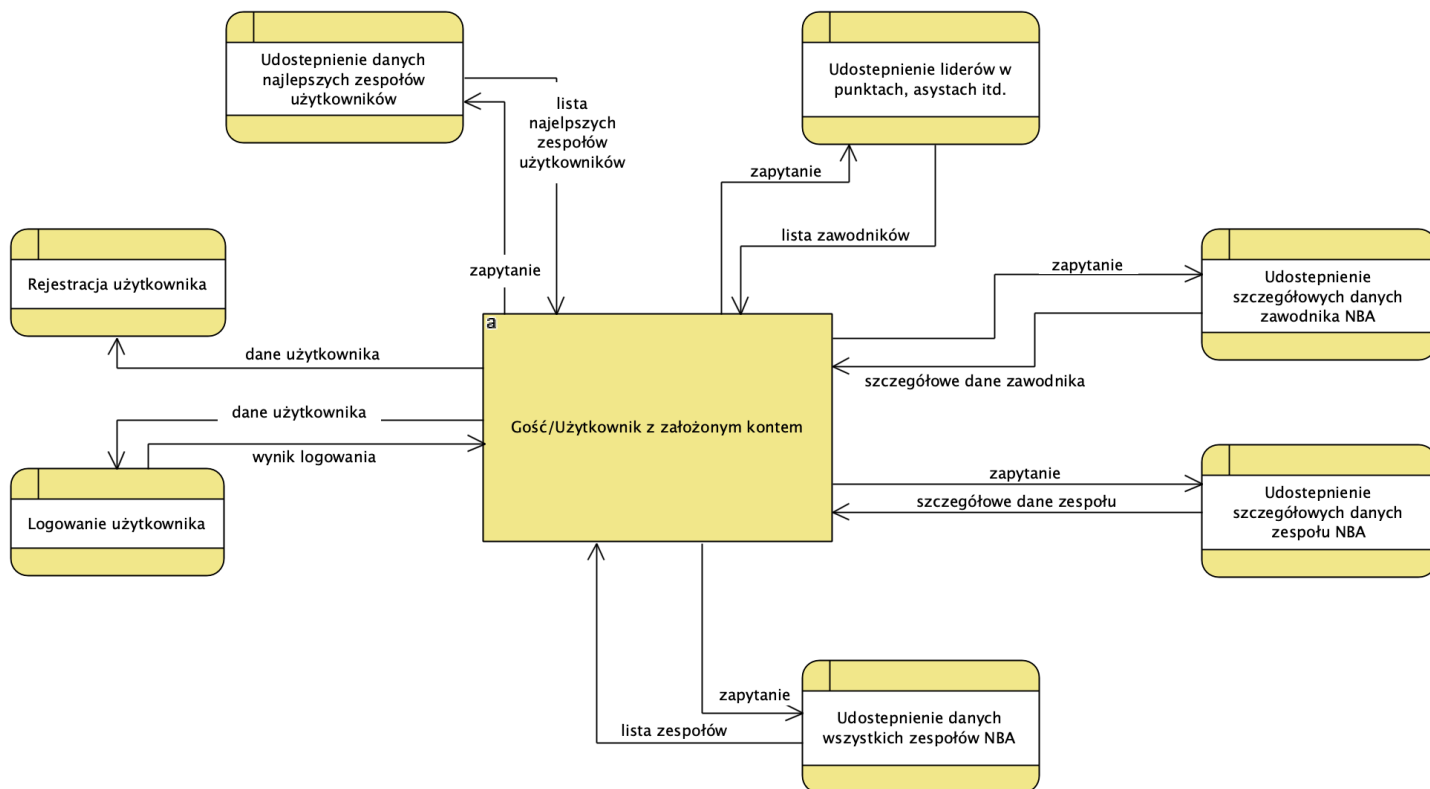


Diagram DFD poziom 0 - Gość/Użytkownik z założonym kontem

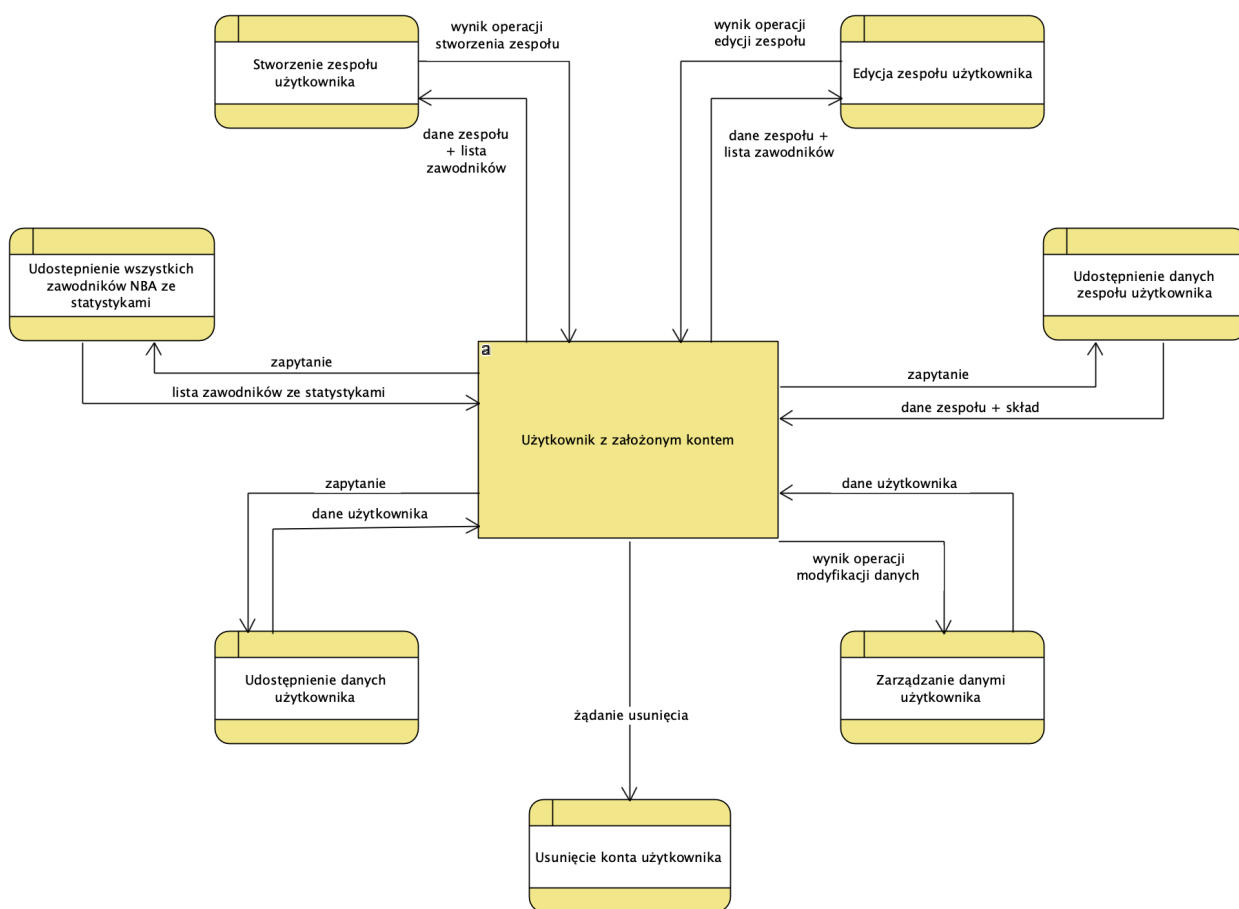


Diagram DFD poziom 0 - Użytkownik z założonym kontem

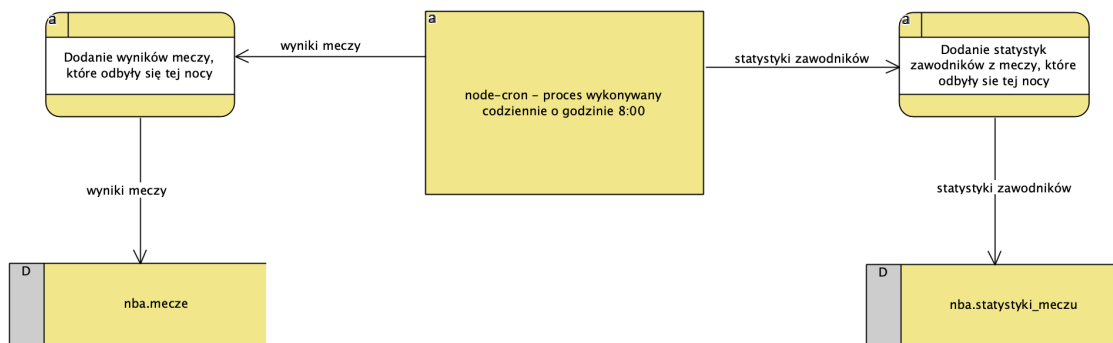


Diagram DFD poziom 1 - node-cron - proces wykonywany automatycznie, codziennie o godzinie 8:00

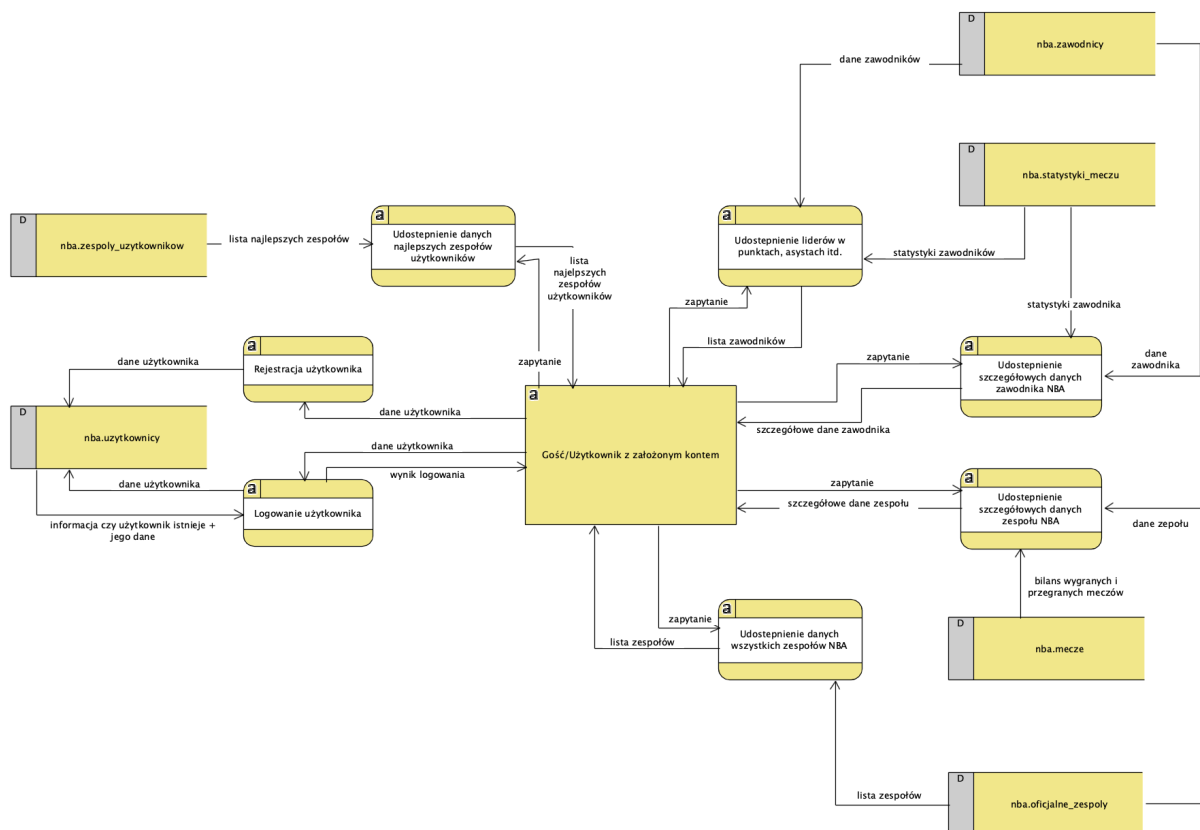


Diagram DFD poziom 1 - Gość/Użytkownik z założonym kontem

5. Entity-Relationship Diagram (ERD)



Diagram ERD

III. Projekt logiczny

6. Projektowanie tabel, kluczy, indeksów

Elementy, których obecności nie widać na diagramie ERD:

- domeny mające na celu zapewnienie poprawności danych w bazie

- sekwencje dla atrybutów `id_uzytkownika` oraz `id_zespolu_uzytkownika`
- indeksy stworzone automatycznie przez bazę PostgreSQL dla wszystkich atrybutów, które pełnią rolę klucza głównego encji, a także dla wszystkich atrybutów w tabeli, które oznaczone są słowem kluczowym `UNIQUE` (zgodnie z oficjalną dokumentacją PostgreSQL)
- procedury oraz wyzwalacze służące weryfikacji wprowadzanych danych przez użytkowników (szerzej opisane w ostatnim punkcie tego paragrafu)
- procedury oraz wyzwalacze mające na celu aktualizację danych zawodnika (`zarobki_zawodnika`) po dodaniu do bazy nowych meczy z udziałem tegoż zawodnika, a także danych zespołu użytkownika (`wynik_zespolu_uzytkownika`) (szerzej opisane w ostatnim punkcie tego paragrafu)

Kod źródłowy znajduje się w folderze `baza_danych`.

7. Słowniki danych

- **nba.oficjalne_zespoly**

Domeny:

- `nba.triCode` - 3 literowy skrót nazwy oficjalnego zespołu NBA (np. "MIA" dla Miami Heat)

- **nba.zawodnicy**

Domeny:

- nba.position - skrót reprezentujący pozycje na jakich zawodnik może grać (np. G-F = guard-forward czyli obronca-skrzydłowy)
- nba.jerseyNo - numer koszulki zawodnika (w NBA dopuszczany jest numer 00)
- nba.salary - zarobki zawodnika
- nba.weight - waga zawodnika
- nba.height - wzrost zawodnika

- **nba.mecze**

Domeny:

- nba.game_day - reprezentacja daty w formacie RRRRMMDD

- **nba.statystyki_meczu**

Domeny:

- nba.game_day - j.w.

- **nba.uzytkownicy**

- **nba.zespoly_uzytkownikow**

- **nba.zawodnicy_zespoly_uzytkownikow** - tabela asocjacyjna

Wszystkie słowniki danych oraz wykorzystywane w nich domeny znajdują się w pliku tabele.sql w folderze baza_danych.

8. Analiza zależności funkcyjnych i normalizacja tabel

Baza została znormalizowana do postaci 3NF, a zatem dane w niej są atomowe, wiersz przechowuje informacje o pojedynczego obiektu, każda tabela odpowiada jednemu rodzajowi obiektów, a kolumny nienależące do klucza głównego nie są zależne jedna od drugiej.

9. Zaprojektowanie operacji na danych

W bazie danych zostały zdefiniowane procedury oraz wyzwalacze służące weryfikacji wprowadzanych danych dotyczących wirtualnych zespołów:

- Procedura nba.zapewnijZgodnoscBudzetu() oraz wyzwalacz _02_zapewnijZgodnoscBudzetu - ma na celu zapewnienie, aby suma płac zawodników w składzie wirtualnej drużyny nie przekraczała budżetu drużyny,
- Procedura nba.zapewnijPoprawnalloscZawodnikow() oraz wyzwalacz _03_zapewnijPoprawnalloscZawodnikow - weryfikuje czy drużyna składa się z dokładnie 5 zawodników,
- Procedura nba.zapewnijNiepowtarzalnoscZawodnikow() oraz wyzwalacz _04_zapewnijNiepowtarzalnoscZawodnikow - gwarantuje, iż zawodnicy w danej drużynie nie mogą się powtarzać,
- Procedury nba.zapewnijPoprawnosc[Obroncow/Skrzydlowych/Centrow]() oraz wyzwalacze _0[5/6/7]_zapewnijPoprawnosc[Obroncow/Skrzydlowych/

Centrow] - gwarantują, że w drużynie musi być conajmniej dwóch zawodników, którzy mogą grać jako obrońcy (guards), conajmniej dwóch zawodników, którzy mogą grać jako skrzydłowi (forwards) oraz conajmniej jeden zawodnik, który może grać jako center

Powyżej wymienione procedury znajdują się w pliku `druzyny_uzytkownikow_procedury.sql` w katalogu `baza_danych`.

Ponadto w pliku `mecze_statystyki_procedury.sql` w katalogu `baza_danych` można znaleźć procedury oraz wyzwalacze które wykonują się po wstawieniu nowego rekordu do tabeli `nba.statystyki_meczu`:

- Procedura `nba.aktualizujZarobkiZawodnikow()` oraz wyzwalacz `_01_aktualizujZarobkiZawodnikow` - po wstawieniu nowego rekordu reprezentującego zdobycze (punktowe i nie tylko) zawodnika zmianie ulegają jego średnie z całego sezonu. Na podstawie tychże średnich procedura wylicza oraz aktualizuje zarobki zawodnika, które są swojego rodzaju wartością zawodnika w mojej grze.
- Procedura `nba.aktualizujWynikiZespolow()` oraz wyzwalacz `_02_aktualizujWynikiZespolow` - punkty służące do klasyfikacji zespołów użytkowników obliczane są na podstawie zdobyczy z rozgrywek zawodników znajdujących się w składzie tychże zespołów. Procedura sumuje pozytywne zdobycze (punkty, asysty, zbiórki, przechwyty) oraz odejmuje od nich zdobycze

negatywne (straty). Następnie dodaje je do ogólnego wyniku zespołu użytkownika.

IV. Projekt funkcjonalny

10. Interfejsy do prezentacji, edycji i obsługi danych, instrukcja obsługi części klienckiej projektu

Aplikacja kliencka w stylu "Single Page Application" została wdrożona na wydziałowym serwerze Pascal. Posiada intuicyjny interfejs użytkownika, analogiczny ze znanymi serwisami internetowymi.

Na stronie głównej w górnej części ekranu znajduje się pasek nawigacyjny. Dla niezalogowanego użytkownika widoczna jest opcja rejestracji oraz zalogowania.

Pod banerem oraz nagłówkiem "NBA Fantasy League" znajdują się 4 sekcje:

- NBA Teams - sekcja z wszystkimi drużynami należącymi do ligi NBA. Po kliknięciu w daną drużynę zostaną wyświetlone jej dane szczegółowe oraz bilans rozegranych meczy.
- League Leaders - sekcja z najlepszymi zawodnikami w lidze w danej statystyce - punktach, asystach i zbiórkach. Po kliknięciu w danego zawodnika zostaną wyświetlone jego dane oraz indywidualne statystyki.
- Fantasy Teams - sekcja z najlepszymi wirtualnymi drużynami zawodników oraz ich wynikami
- About Author - sekcja z informacjami o autorze aplikacji

Po zalogowaniu bądź rejestracji do których służą dedykowane formularze, następuje przeniesienie do panelu użytkownika. Jeśli użytkownik nie posiada jeszcze własnej drużyny udostępniana jest mu możliwość jej utworzenia. Dzieje się to poprzez interaktywny formularz składający się z komponentu zawierającego zawodników wybranych przez użytkownika, panelu sterowania, za pomocą którego można dodawać lub usuwać zawodników z listy oraz dolny komponent wyświetlający wszystkich zawodników NBA posortowanych po średnich punktowych na mecz. Po nawigacji po wszystkich zawodnikach służą przyciski umieszczone na samym dole realizujące funkcjonalność paginacji. Aby usunąć lub dodać zawodnika należy najpierw zaznaczyć go klikając na wiersz w którym się znajduje. Następnie za pomocą panelu sterowania można wykonywać zdefiniowane akcje. Tak samo wygląda edycja składu drużyny użytkownika. Po stworzeniu drużyny w panelu użytkownika wyświetlane są komponenty dotyczące danych drużyny oraz jej składu. Możliwa jest również edycja profilu użytkownika - danych użytkownika, zmiany hasła, a także usunięcia konta. Wylogowanie następuje po kliknięciu w ikonkę wylogowania na pasku użytkownika.

11. Wizualizacja danych

W celu uproszczenia zapytań do bazy danych zostały stworzone widoki umieszczone w pliku widoki.sql w katalogu baza_danych.

W pliku tym każdy widok poprzedzony jest opisem swojej funkcjonalności.

V. Dokumentacja

12. Wprowadzanie danych

Dane dotyczące oficjalnych drużyn NBA, zawodników w nich występujących, meczy oraz zdobywczy zawodników w tychże meczach zostały zaimportowane z oficjalnego API NBA (data.nba.com) poprzez skrypty napisane w języku JavaScript. Ponadto dane dotyczące meczy oraz zdobywczy zawodników z poprzedniego wieczoru importowane są automatycznie każdego dnia o godzinie 8:00 z wykorzystaniem modułu node-cron (ze względu na różnicę stref czasowych mecze w NBA zazwyczaj zaczynają się około godziny 1:00 "naszego" czasu).

13. Dokumentacja techniczna

W celu umożliwienia dostępu do bazy danych został stworzony sytem warstwowy na który składają się dwie aplikacje.

Backend - aplikacja webowa w stylu RESTful napisana w środowisku Node.js z wykorzystaniem standardu języka ECMAScript 2018, korzystająca z framework'a Express w wersji 4.

Aplikacja w celu połączenia z bazą danych na serwerze Pascal wykorzystuje pakiet node-postgres - zbiór modułów

zapewniających dostęp do bazy danych na akceptowalnym poziomie abstrakcji.

Do testów jednostkowych oraz behawioralnych aplikacji użyto modułów "mocha" oraz "chai".

W celu automatycznego pobierania danych z oficjalnego API NBA wykorzystany został moduł "node-cron".

Moduły "bcrypt" oraz "jsonwebtoken" są wykorzystywane do autentykacji i uwierzytelnienia za pomocą token'ów.

Struktura kodu aplikacji:

W głównym katalogu "backend" znajdują się:

- index.js - główny plik aplikacji, w którym tworzona i konfigurowana jest instancja serwera. Tworzony jest również "router", który dopasowuje przychodzące zapytania do obsługujących je kontrolerów
- package.json - plik konfiguracyjny
- controllers/ - katalog zawierający kontrolery - znajduje się tutaj cała logika aplikacji
- db/ - katalog w którym stworzona została warstwa abstrakcji w dostępie do bazy danych (stworzenie puli połączeń)
- lib/ - katalog ze skryptami służącymi do pobierania danych z API NBA
- middleware/ - katalog z narzędziami pomocniczymi w autentykacji i autoryzacji
- tests/ - katalog z testami jednostkowymi oraz behawioralnymi

Frontend - aplikacja kliencka w stylu Single Page Application stworzona z wykorzystaniem języka ECMAScript 2018 oraz framework'a Vue.js.

Aplikacja udostępnia interfejs użytkownika w dostępie do funkcjonalności API.

Struktura kodu aplikacji:

W głównym katalogu "frontend" znajdują się:

- package.json - plik konfiguracyjny
- src/ - folder z kodem źródłowym aplikacji
 - router/ - "router" do nawigowania pomiędzy komponentami aplikacji
 - components/ - komponenty, z których zbudowana jest aplikacja

Aplikacje zostały wdrożone na wydziałowym serwerze Pascal. Można znaleźć je pod linkami:

backend: <http://pascal.fis.agh.edu.pl:13250/api>

frontend: <http://pascal.fis.agh.edu.pl:14400/#/>

14. Wykaz literatury

- postgresql.org/docs/
- "Web Development with Node and Express" by Ethan Brown, O'Reilly Media, Inc.

- "Practical Node.js - Building Real-World Scalable Web Apps" by Azat Mardan, Apress
- expressjs.com/en/4x/api.html
- "Getting to Know Vue.js: Learn to Build Single Page Applications in Vue from Scratch" by Brett Nelson, Apress
- "Hands-On Web Development with Vue.js" by Roman Kuba, Packt Publishing
- vuejs.org/v2/guide/
- semantic-ui.com/introduction/getting-started.html