*PROJEKT ZESPOŁOWY*

Projekt i implementacja aplikacji wspomagającej ligowe rozgrywki piłkarzyków.

*Artur Kucybała, Radosław Jagiełło, Mateusz Gwiazda*

*Grupa Z709*

**ETAP I**

**Analiza systemu**

1. Przedstawienie koncepcji systemu

1.1. Ogólny opis produktu

Aplikacja wspomagająca ligowe rozgrywki piłkarzyków powstała na potrzeby pracy inżynierskiej. Głównym celem wdrożenia oprogramowania jest udostępnienie światu funkcjonalnej aplikacji dzięki której każda grupa użytkowników będzie mogła w prostszy i bardziej wydajny sposób zarządzać prowadzonymi rozgrywkami w utworzonej przez tą grupę lidze. Aplikacja ta będzie informować każdego użytkownika o najważniejszych informacjach danej ligi, m.in. tabeli z punktacją, meczach odbytych oraz nadchodzących, profilach drużyn itp.

Aplikacja będzie przeznaczona przede wszystkim dla firm oraz innych zamkniętych grup. Z przeprowadzonych wstępnych analiz wynika, że w dużej części firm pracownicy w wolnym od pracy czasie prowadzą między sobą rozgrywki piłkarzyków lecz nie posiadają odpowiedniego narzędzia, które mogło by im pomóc stworzyć zamkniętą ligę dzięki, której mecze są z

Czas poświęcony na realizację projektu będzie także czasem edukowania i poznawania nowych możliwości platformy Visual Studio oraz wzorca MVC.

Diagram kontekstowy aplikacji wspomagającej zarządzanie ligą piłkarzyków.

***Administrator***

***Zarejestrowany użytkownik***

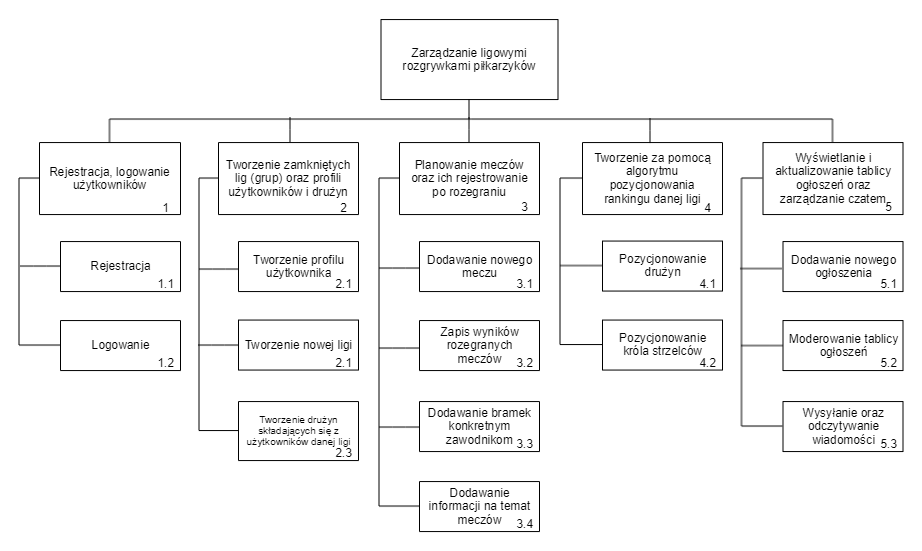
***Baza danych***

1.2. Charakterystyka użytkowników

* Użytkownik - jest to osoba która korzysta z aplikacji. Użytkownik posiada prywatne konto w systemie, ma możliwość zalogowania się oraz korzystania z wszystkich funkcjonalności systemu.
* Administrator jest to osoba administrująca i zarządzająca systemem, której zadania to przede wszystkim weryfikacja użytkowników, dokonywanie wszelkich aktualizacji oraz dbanie o bezpieczeństwo danych użytkowników.

2. Specyfikacja funkcjonalna

2.1. Diagram hierarchii funkcji (FHD)



2.2. Główne funkcje systemu

Użytkownik w kontekście aplikacji może:

* Tworzyć indywidualne konto użytkownika.
* Tworzyć nowe grupy w postaci własnej ligi.
* Aktualizować oraz sprawdzać status swojej ligi (będąc jej administratorem).
* Tworzyć zamknięte grupy użytkowników (nowe ligi).
* Administrować własną grupą użytkowników.
* Prowadzić rozmowę z innymi użytkownikami w postaci zamkniętego czatu.

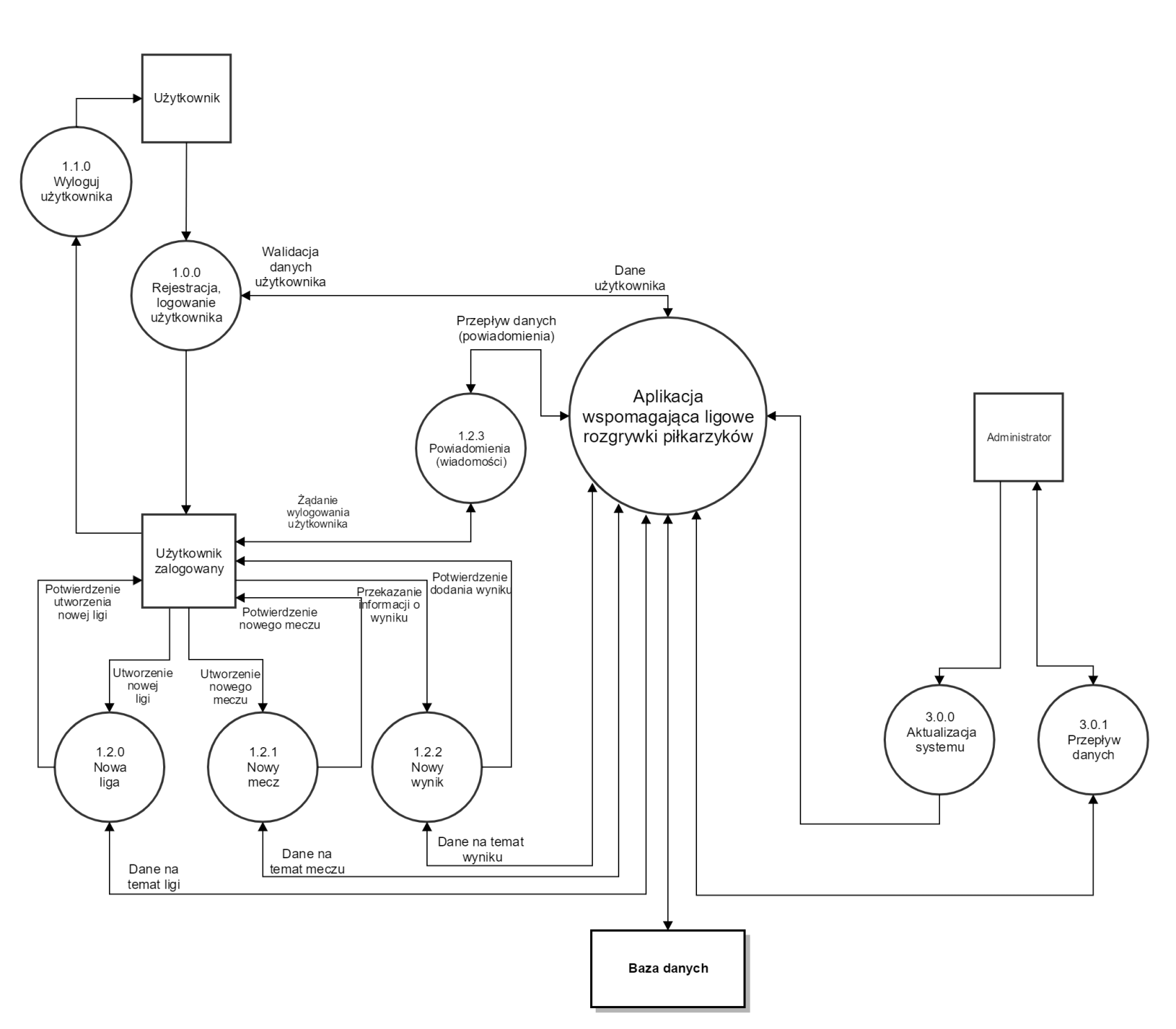
Administrator w kontekście aplikacji może:

* Dokonywać aktualizacji systemu.
* Zarządzać kontami użytkowników.
* Zarządzać bazą danych systemu.

Baza danych w kontekście aplikacji może:

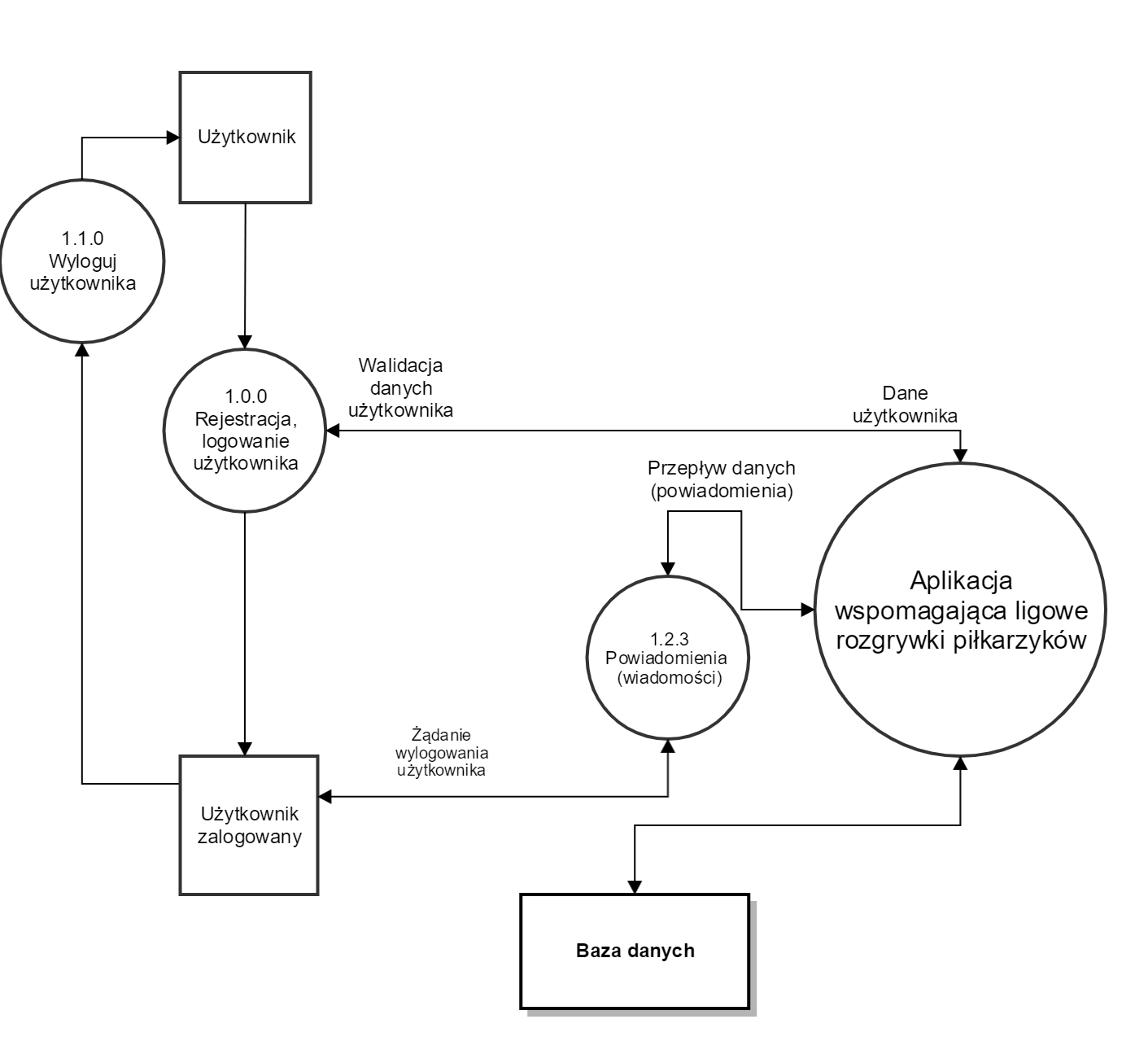
* Przyjmować oraz przechowywać dane użytkowników.
* Udostępniać przechowywane dane na żądanie systemu.

2.3 Diagram poziomu zerowego aplikacji wspomagającej ligowe rozgrywki piłkarzyków



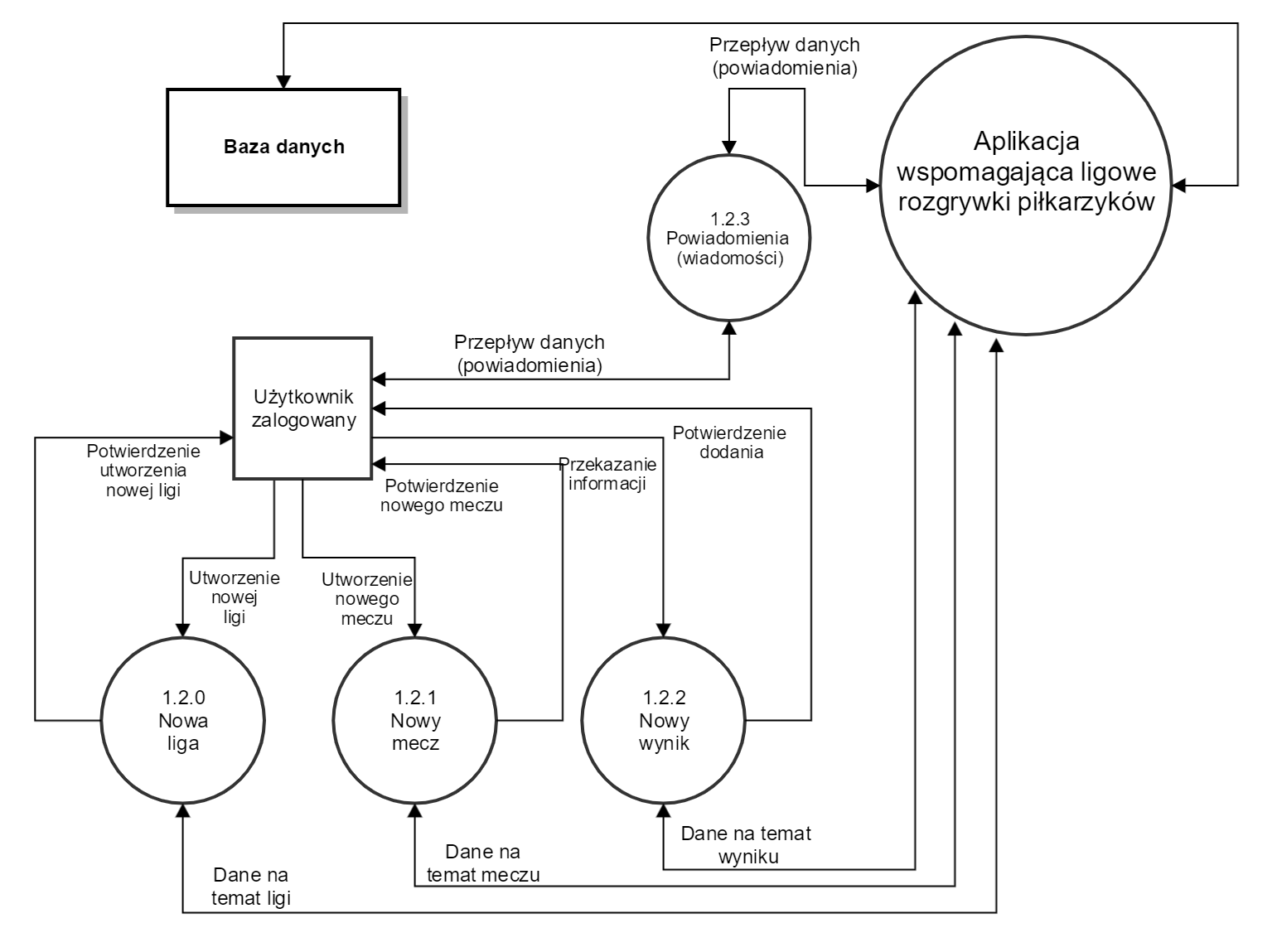
|  |  |  |
| --- | --- | --- |
| **Nazwa elementu** | **Typ elementu** | **Opis** |
| Użytkownik | Terminator | Osoba, która używa aplikacji wspomagającej ligowe rozgrywki piłkarzyków |
| Administrator | Terminator | Osoba zarządzająca całym systemem |
| 1.0.0  Rejestracja, logowanie użytkownika | Proces | Proces, w którym użytkownik rejestruje się i loguje w aplikacji |
| 2.0.0  Rejestracja, logowanie administratora | Proces | Proces w którym administrator rejestruje się i loguje w aplikacji |
| 1.1.0  Wyloguj użytkownika | Proces | Proces, w którym użytkownik zostaje wylogowany z aplikacji |
| 2.1.0  Wyloguj administratora | Proces | Proces w którym administrator zostaje wylogowany z aplikacji |
| 1.2.0  Nowa liga | Proces | Proces w którym zalogowany użytkownik tworzy na swoim profilu nową ligę piłkarzyków  (staje się on automatycznie administratorem danej grupy) |
| 1.2.1  Nowy mecz | Proces | Proces w którym zalogowany użytkownik tworzy na swoim profilu nowy mecz |
| 1.2.2  Nowy wynik | Proces | Proces, w którym użytkownik dodaje wynik odbytego meczu w lidze, której jest administratorem |
| 1.2.3  Powiadomienia  (wiadomości) | Proces | Proces, w którym użytkownik wysyła wiadomość tekstową do innego użytkownika będącego członkiem tej samej grupy. Jest to również proces, w którym system wysyła powiadomienia do użytkownika w postaci tekstowej oraz w formie tablicy ogłoszeń |
| 3.0.0  Aktualizacja  systemu | Proces | Proces, w którym terminator "Administrator" dokonuje zmian w aplikacji |
| 3.0.1  Przepływ danych | Proces | Proces, w którym terminator "Administrator" uzyskuje żądane dane z systemu oraz takie dane wysyła w postaci komunikatów. |
| Dane użytkownika | Przepływ | Przepływ danych zarejestrowanego użytkownika |
| Żądanie wylogowania użytkownika | Przepływ | Przepływ danych z żądaniem wylogowania użytkownika |
| Utworzenie nowej ligi | Przepływ | Przepływ zawierający informacje o nowej lidze stworzonej przez użytkownika |
| Potwierdzenie nowej ligi | Przepływ | Przepływ danych zawierający informację o dodaniu w aplikacji nowej ligi |
| Utworzenie nowego meczu | Przepływ | Przepływ danych zawierający informację o nowo utworzonym meczu |
| Potwierdzenie nowego meczu | Przepływ | Przepływ danych zawierający informację o przyjęciu przez aplikację nowego meczu |
| Przekazanie informacji o wyniku | Przepływ | Przepływ danych zawierający informacje o wyniku odbytego meczu (oraz innych informacji na temat odbytego meczu) |
| Potwierdzenie dodania wyniku | Przepływ | Przepływ danych zawierający informację o przyjęciu przez aplikację danych na temat odbytego meczu |
| Przepływ danych (powiadomienia) | Przepływ | Przepływ danych służący do komunikacji pomiędzy użytkownikami tej samej grupy oraz pomiędzy aplikacją a użytkownikiem |
| Dane na temat ligi | Przepływ | Przepływ danych zawierający informację o lidze |
| Dane na temat meczu | Przepływ | Przepływ danych zawierający informacje na temat meczu |
| Dane na temat wyniku | Przepływ | Przepływ danych zawierający informacje na temat wyniku oraz innych informacji, o których wiadomo po jego odbyciu |
| Baza danych | Magazyn danych | Magazyn danych na temat całości aplikacji |

3.2.1. Diagram rejestracji, logowania i wylogowania użytkownika.



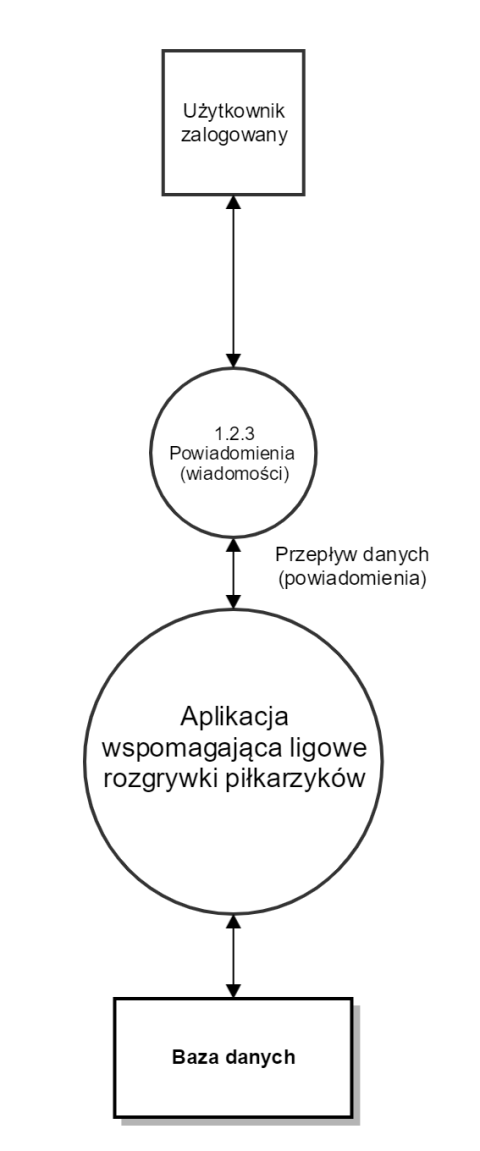
|  |  |  |
| --- | --- | --- |
| **Nazwa elementu** | **Typ elementu** | **Opis** |
| Użytkownik | Terminator | Osoba, która używa aplikacji wspomagającej ligowe rozgrywki piłkarzyków |
| 1.0.0  Rejestracja, logowanie użytkownika | Proces | Proces, w którym użytkownik rejestruje się i loguje w aplikacji |
| 1.1.0  Wyloguj użytkownika | Proces | Proces, w którym użytkownik zostaje wylogowany z aplikacji |
| Walidacja danych użytkownika | Przepływ | Przepływ danych potwierdzających możliwość rejestracji/logowania użytkownika |
| Dane użytkownika | Przepływ | Przepływ danych zarejestrowanego użytkownika |
| Walidacja danych użytkownika | Przepływ | Przepływ danych potwierdzających możliwość rejestracji/logowania użytkownika |

3.2.3. Diagram tworzenia nowej ligi, meczu oraz wyników



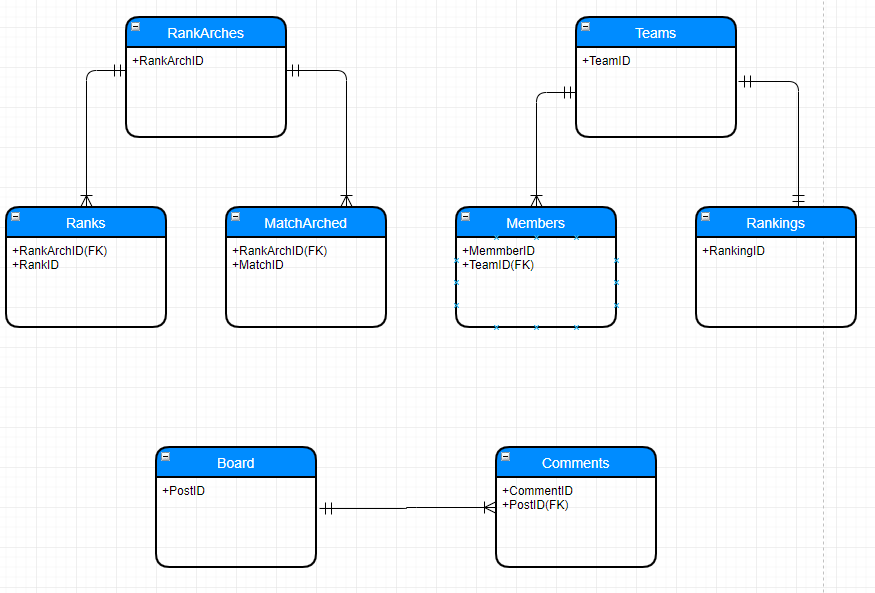
|  |  |  |
| --- | --- | --- |
| **Nazwa elementu** | **Typ elementu** | **Opis** |
| 1.2.0  Nowa liga | Proces | Proces w którym zalogowany użytkownik tworzy na swoim profilu nową ligę piłkarzyków  (staje się on automatycznie administratorem danej grupy) |
| 1.2.1  Nowy mecz | Proces | Proces w którym zalogowany użytkownik tworzy na swoim profilu nowy mecz |
| 1.2.2  Nowy wynik | Proces | Proces, w którym użytkownik dodaje wynik odbytego meczu w lidze, której jest administratorem |
| Baza Danych | Terminator | Baza przechowująca wszystkie informacje o użytkownikach i przekazująca te dane na żądanie systemu lub administratora. |
| Utworzenie nowej ligi | Przepływ | Przepływ zawierający informacje o nowej lidze stworzonej przez użytkownika |
| Potwierdzenie nowej ligi | Przepływ | Przepływ danych zawierający informację o dodaniu w aplikacji nowej ligi |
| Utworzenie nowego meczu | Przepływ | Przepływ danych zawierający informację o nowo utworzonym meczu |
| Potwierdzenie nowego meczu | Przepływ | Przepływ danych zawierający informację o przyjęciu przez aplikację nowego meczu |
| Przekazanie informacji o wyniku | Przepływ | Przepływ danych zawierający informacje o wyniku odbytego meczu (oraz innych informacji na temat odbytego meczu) |
| Potwierdzenie dodania wyniku | Przepływ | Przepływ danych zawierający informację o przyjęciu przez aplikację danych na temat odbytego meczu |
| Przepływ danych (powiadomienia) | Przepływ | Przepływ danych służący do komunikacji pomiędzy użytkownikami tej samej grupy oraz pomiędzy aplikacją a użytkownikiem |
| Dane na temat ligi | Przepływ | Przepływ danych zawierający informację o lidze |
| Dane na temat meczu | Przepływ | Przepływ danych zawierający informacje na temat meczu |
| Dane na temat wyniku | Przepływ | Przepływ danych zawierający informacje na temat wyniku oraz innych informacji, o których wiadomo po jego odbyciu |
| Baza danych | Magazyn danych | Magazyn danych na temat całości aplikacji |

3.2.4. Diagram przepływu powiadomień (wiadomości)

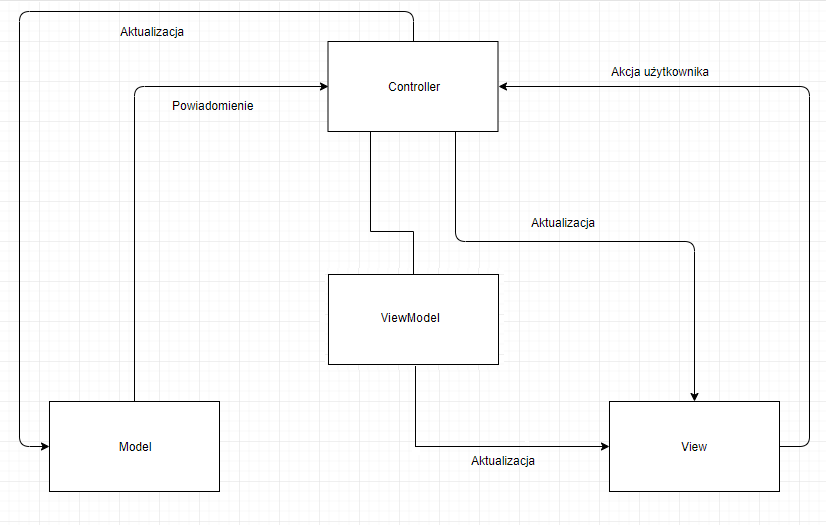


|  |  |  |
| --- | --- | --- |
| **Nazwa elementu** | **Typ elementu** | **Opis** |
| Użytkownik | Terminator | Osoba, która używa aplikacji wspomagającej ligowe rozgrywki piłkarzyków |
| 1.2.3  Powiadomienia  (wiadomości) | Proces | Proces, w którym użytkownik wysyła wiadomość tekstową do innego użytkownika będącego członkiem tej samej grupy. Jest to również proces, w którym system wysyła powiadomienia do użytkownika w postaci tekstowej oraz w formie tablicy ogłoszeń |
| Baza danych | Magazyn danych | Magazyn danych na temat całości aplikacji |

3. Diagram ERD (Entity Relationship Diagrams).



4. Model architektury systemu



**Architektura MVC**

MVC (Model-view-controller) jest tzw. wzorcem architektonicznym, stosowanym przy tworzeniu nowoczesnych systemów informatycznych.

Główną koncepcją MVC jest wymuszenie podziału aplikacji na 3 niezależne warstwy reprezentujące kolejno:

* (Model) Model - reprezentuje naszą logikę biznesową. Tutaj znajdują się wszelkie obiekty, które służą do wykonywania wszelkich operacji związanych z implementacją funkcjonalności naszej aplikacji
* (View) Widok - warstwa prezentacji. Widok odpowiedzialny jest za prezentację użytkownikowi wyników działania logiki biznesowej
* (Controller) Kontroler - obsługuje żądania użytkownika. Wszelkie żądania deleguje do odpowiednich metod Modelu.

Podział na warstwy służy uporządkowaniu architektury systemu. Dzięki temu, że każda logiczna część jest od siebie oddzielona, zmiana w jednym miejscu, nie powoduje konieczności wykonywania lawinowej ilości zmian w innych miejscach systemu.

Standardowy wzorzec MVC został rozszerzony o ViewModel – klasę modelu na potrzeby widoku.  
ViewModel zawiera:

* Kluczowe dla warstwy prezentacji dane
* Listę elementów zależnych - np. kontekstów które możemy wykorzystać w Tasku
* Informacje/dane dodatkowe przydatne w generowaniu widoku - np. informacje o stronicowaniu

Tworzony system jest aplikacją internetową. Do tego typu aplikacji, wzorzec MVC jest najkorzystniejszym wyborem. Posiada takie zalety jak:

* Podział na moduły porządkujące kod aplikacji,
* Oddzielenie logiki biznesowej od widoku,
* Brak zależności modelu od widoku,
* Ułatwia odnalezienie konkretnej części kodu,
* Łatwiejsza rozbudowa poprzez modułową budowę

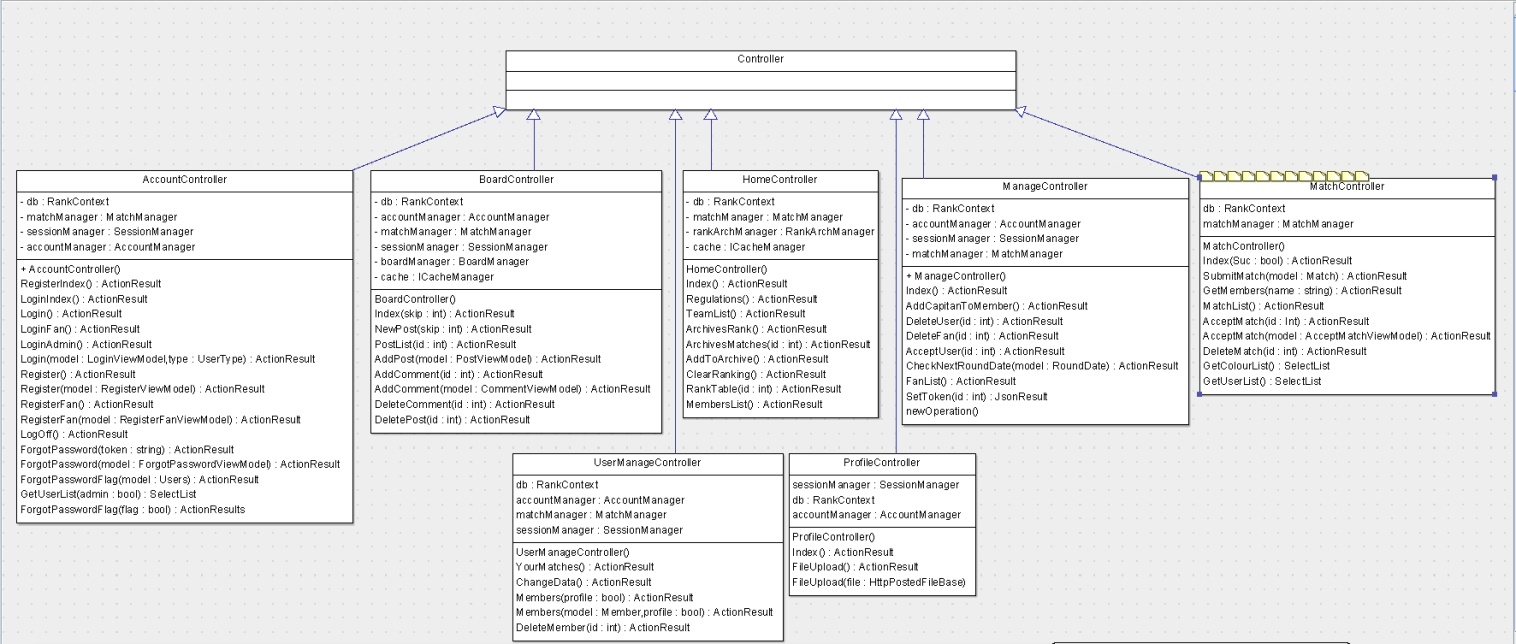
**ETAP II**

**Projekt systemu**

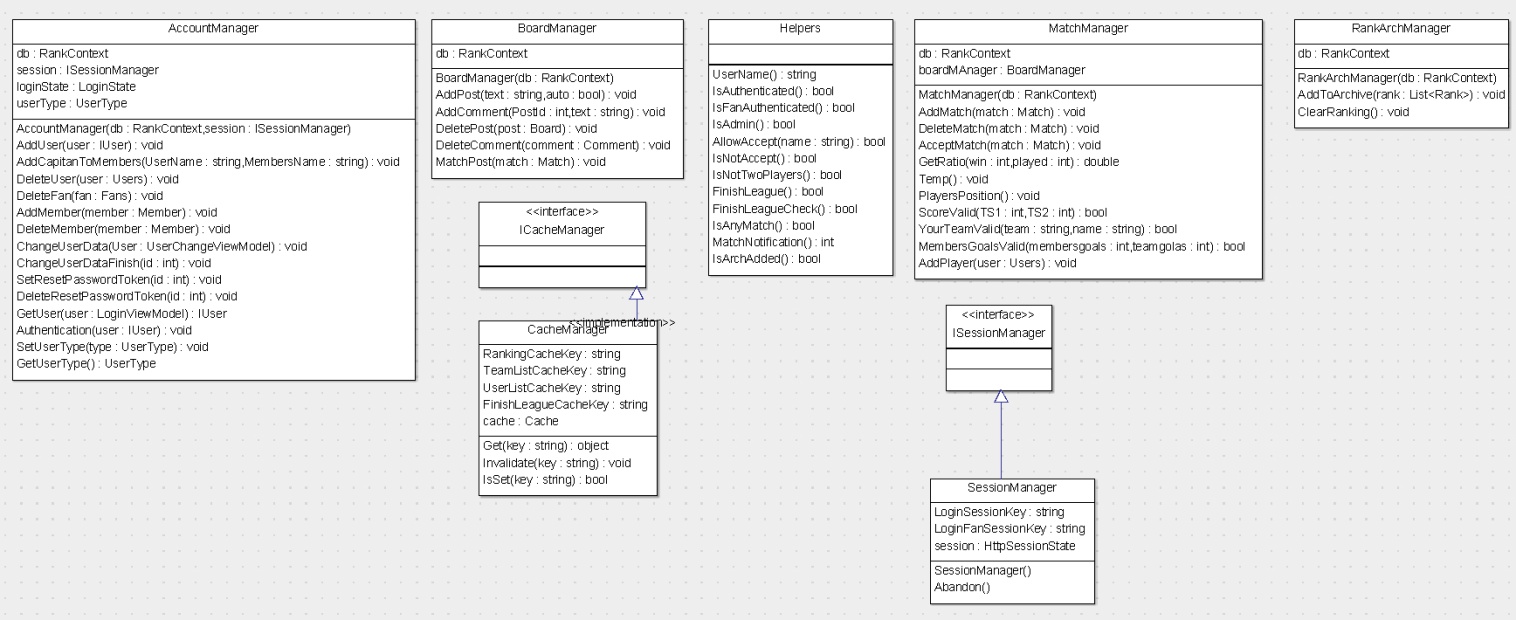
1. Projekt architektury systemu

1.1. Projekt diagramu klas

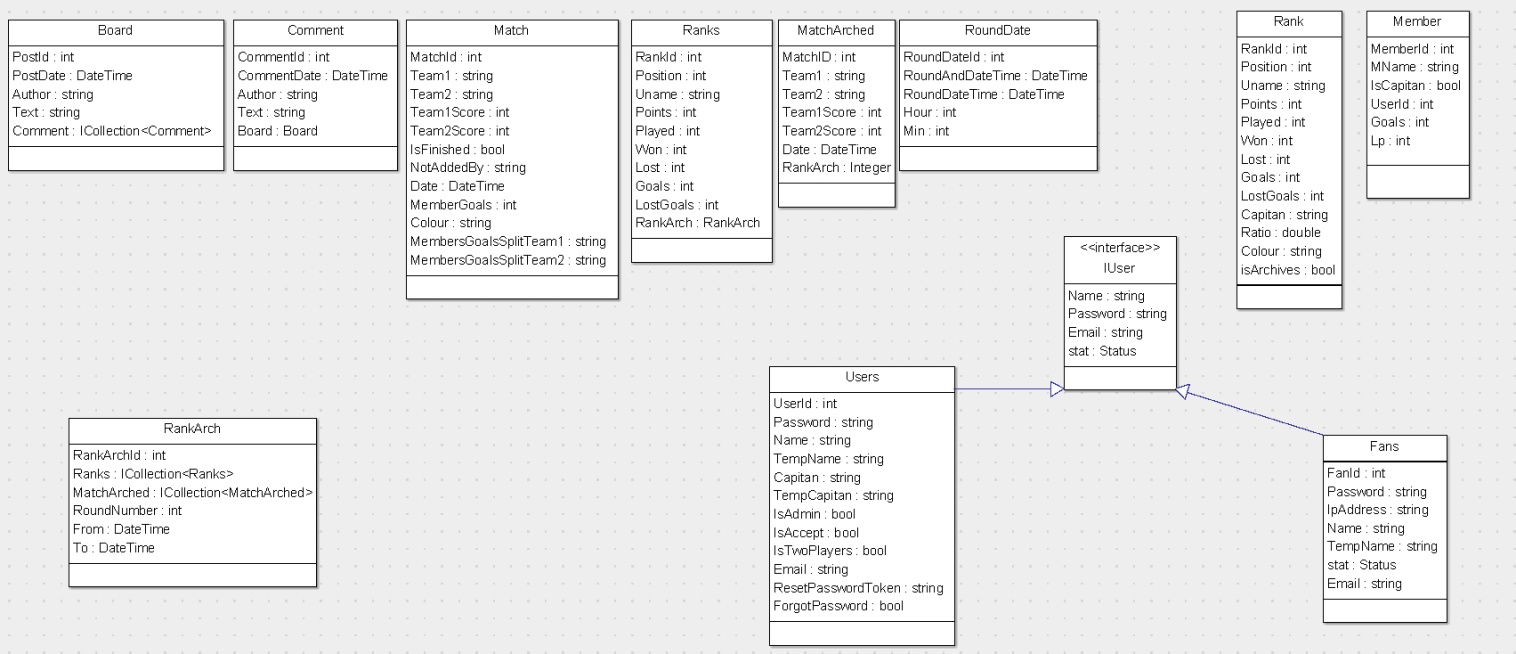
1.1.1. Warstwa konrolera



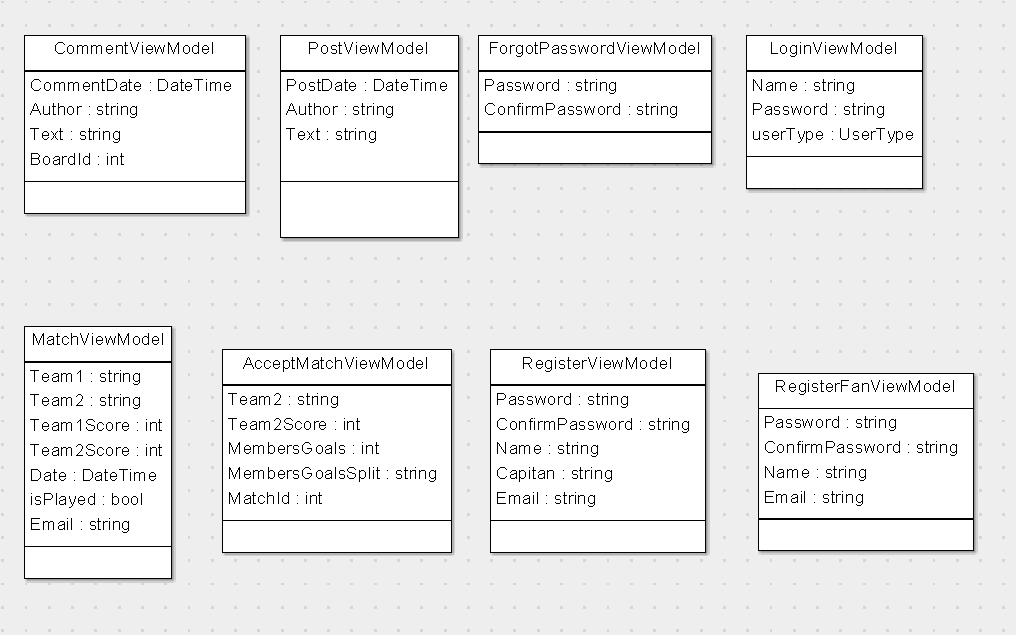
1.1.2. Warstwa infrastruktury



1.1.3. Warstwa modelu

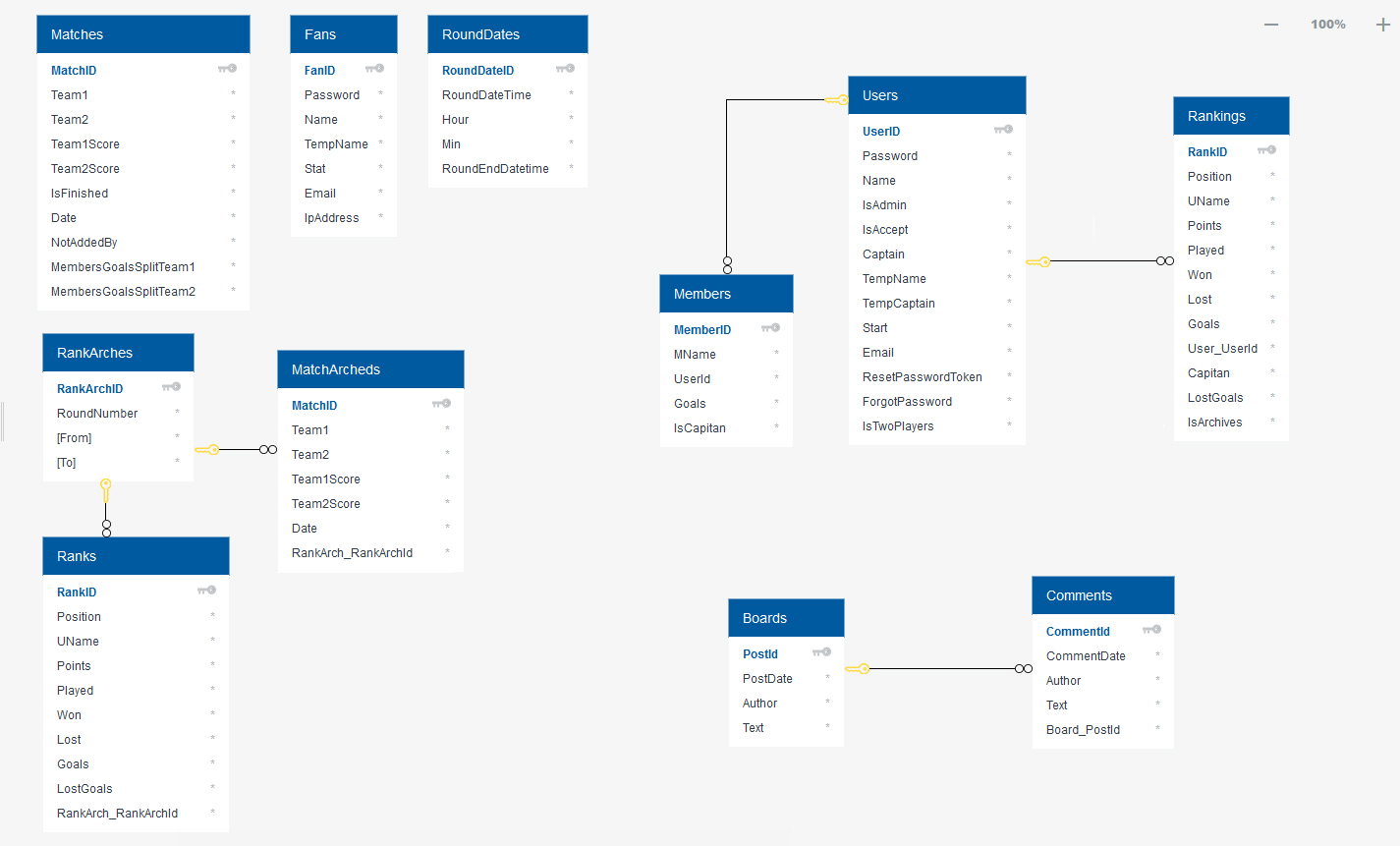


1.1.3. Warstwa widok-model



2. Projekt bazy danych

2.1. Diagram tabel i relacji



2.2. Opis pól i ich atrybutów

Tabela : Matches

|  |  |
| --- | --- |
| Nazwa pola: | Typ: |
| MatchId | int |
| Team1 | string |
| Team2 | string |
| Team1Score | int |
| Team2Score | int |
| IsFinished | bool |
| Date | DateTime |
| NotAddedBy | string |
| MembersGoalsSplitTeam1 | string |
| MembersGoalsSplitTeam2 | string |

Tabela : Fans

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| FanId | int |
| Password | string |
| Name | string |
| TempName | string |
| Stat | Status |
| Email | string |
| IpAddress | string |

Tabela : RoundDates

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| RoundDateId | int |
| RoundDateTime | DateTime |
| Hour | int |
| Min | int |
| RoundEndDateTime | DateTime |

Tabela : RankArches

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| RankArchId | int |
| RoundNumber | int |
| [From] | DateTime |
| [To] | DateTime |

Tabela : MatchArcheds

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| MatchID | Int |
| Team1 | String |
| Team2 | String |
| Team1Score | Int |
| Team2Score | Int |
| Date | DateTime |
| RankArch\_RankArchId | RankArch |

Tabela : Ranks

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| RankID | Int |
| Position | Int |
| UName | String |
| Points | int |
| Played | int |
| Won | int |
| Loas | int |
| Goals | int |
| LostGoals | int |
| RankArch\_RankArchId | RankArch |

Tabela : Members

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| MemberID | Int |
| MName | String |
| UserId | Int |
| Goals | Int |
| IsCapitan | bool |

Tabela : Users

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| UserID | Int |
| Password | String |
| Name | String |
| IsAdmin | bool |
| IsAccept | bool |
| Capitan | String |
| TempName | String |
| TempCapitan | String |
| Stat | Status |
| Email | String |
| ResetPasswordToken | String |
| ForgotPassword | Bool |
| IsTwoPlayers | bool |

Tabela : Rankings

|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| RankID | Int |
| Position | Int |
| UName | string |
| Points | int |
| Played | int |
| Won | int |
| Lost | int |
| Goals | int |
| User\_UserId | int |
| Capitan | string |
| LostGoals | int |
| IsArchives | bool |

Tabela : Boards

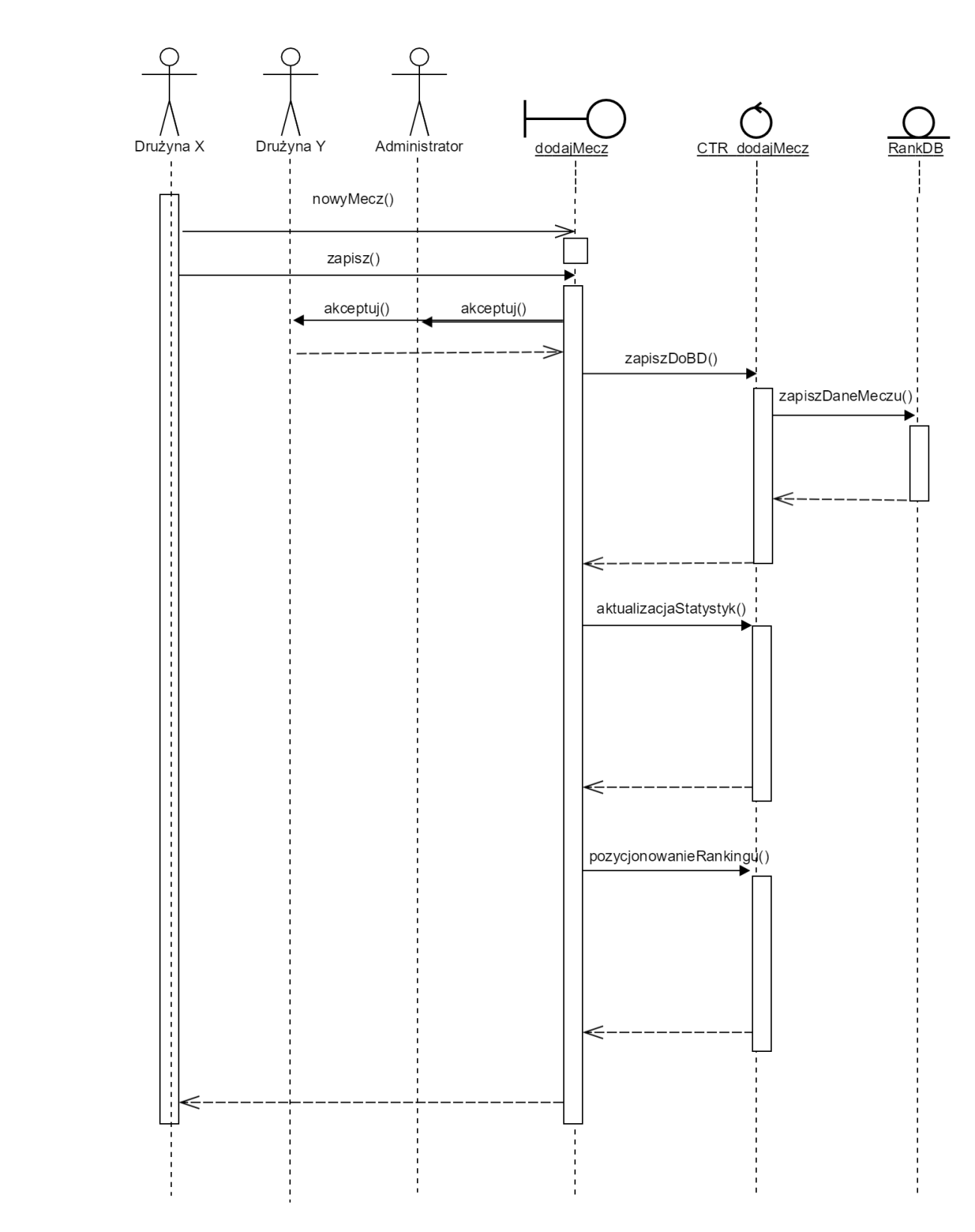
|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| PostId | Int |
| PostDate | DateTime |
| Author | String |
| Text | string |

Tabela : Comments

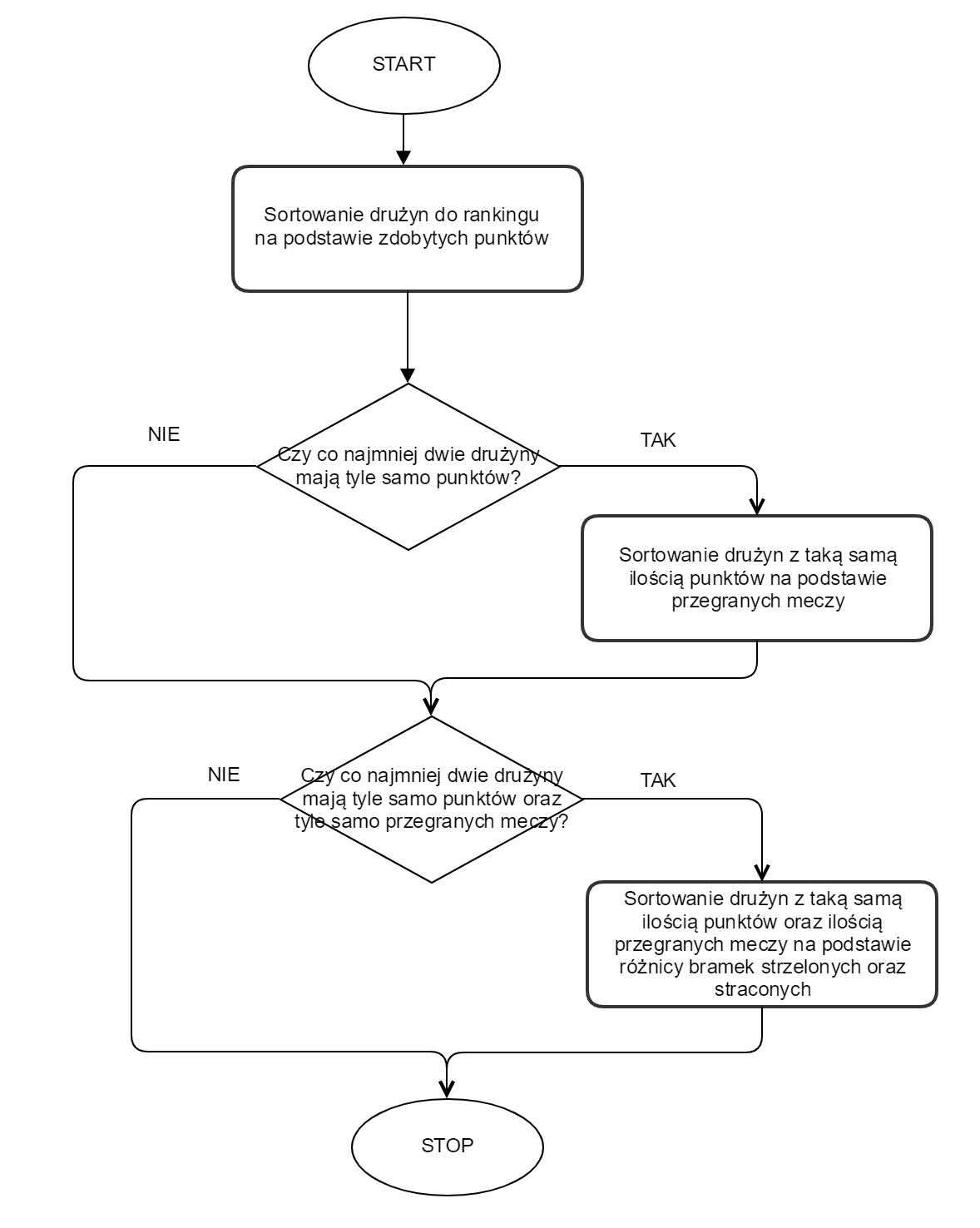
|  |  |
| --- | --- |
| Nazwa pola : | Typ: |
| CommentId | Int |
| CommentDate | DateTime |
| Author | String |
| Text | String |
| Board\_PostId | Board |

3. Projekt algorytmów

3.1. Zapis meczu - diagram sekwencji



3.2. Sortowanie drużyn do rankingu (diagram blokowy)



4. Projekt interfejsu użytkownika

Interfejs użytkownika definiuje przestrzeń, dzięki której użytkownik może podejmować interakcję z systemem. Z perspektywy użytkownika jest to bardzo ważna część systemu odpowiadająca przede wszystkim za jego postrzeganie oraz bezproblemowe korzystanie.

4.1. Makiety okien aplikacji

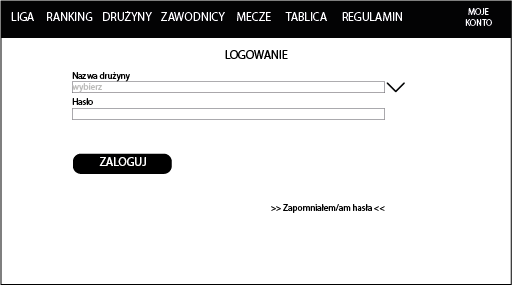
4.1.1. Rejestracja drużyny

Proces rejestracji drużyny jest pierwszym krokiem, który powinien wykonać nowy użytkownik chcący wziąć udział w rozgrywkach.



4.1.2. Formularz logowania drużyny

Następnie użytkownik po przejściu procesu rejestracji powinien się zalogować aby w pełni korzystać z możliwości aplikacji.



4.1.3. Drużyny

Poniżej widzimy makietę przedstawiającą zarejestrowane drużyny w systemie. Widok ten jest przypisany tylko do konta administratora, który może zarządzać wszystkimi drużynami oraz rundami rozgrywek.



4.1.4. Mecze

Makieta ta przedstawia widok dodawania odbytego meczu. Mecz może dodać zarówno administrator jak i użytkownik należący do drużyny w nim uczestniczącej.



4.1.5. Ranking



4.1.6. Tablica informacyjna dla wszystkich użytkowników

