

Polsko-Japońska Akademia Techniki Komputerowych



Tournament Helper

czyli system wspomagający organizowanie
turniejów Magic: the Gathering

Dokumentacja dla projektu

Sporządzono: czerwiec 2020

Spis treści

| | |
|---|----|
| Dziedzina biznesowa i cel | 2 |
| Zakres odpowiedzialności systemu | 3 |
| Użytkownicy systemu | 3 |
| Wymagania użytkownika..... | 3 |
| System powinien umożliwiać następujące funkcjonalności: | 5 |
| Diagram kontekstowy..... | 6 |
| Diagram przypadków użycia..... | 7 |
| Analityczny diagram klas | 8 |
| Projektowy diagram klas | 9 |
| Wymagania niefunkcjonalne | 10 |
| 1. Łatwość użytkowania | 10 |
| 2. niezawodność | 10 |
| 3. Bezpieczeństwo danych | 10 |
| 4. Łatwa modyfikowalność..... | 10 |
| Opis przyszłej ewolucji systemu | 10 |
| Słownik pojęć z dziedziny problemowej..... | 10 |
| Scenariusz przypadku użycia "Zorganizuj Turniej" | 11 |
| Diagram stanów dla klasy Turniej | 13 |
| Diagram aktywności dla przypadku użycia "Zorganizuj Turniej" | 14 |
| Diagram sekwencyjny dla przypadku użycia "Zorganizuj Turniej" | 15 |
| Projekt GUI dla przypadku użycia "Zorganizuj Turniej" | 16 |

Dziedzina biznesowa i cel

Organizacje i sklepy organizujące turnieje gry Magic: the Gathering komunikują, że potrzebują lepszego, unowocześnionego systemu do prowadzenia i przechowywania informacji o organizowanych przez nich (i nie tylko przez nich) turniejach, ponieważ obecnie dostarczone, oficjalne narzędzie, czyli WER (Wizards Events Reporter) nie spełnia wymagań na wielu poziomach, między innymi:

- Interfejs jest nieintuicyjny i nieprzejrzysty (program jest przestarzały);
- Bardzo podstawowe działania są trudne do wyegzekwowania, a błędy ludzkie bardzo trudno jest naprawić (lub czasem w ogóle nie można edytować / cofnąć akcji);
- Brakuje wielu funkcjonalności, które w przeciągu kilkunastu ostatnich lat okazały się być kluczowe, a nie były dodane (i wciąż nie są) do oficjalnej dystrybucji WER;
- Brakuje operowania na historii turniejów i szczegółach odbytych gier, jak i szczegółów dotyczących graczy oraz ich decków (talii kart, którymi grają w turniejach);
- Program jest bardzo wolny, mało responsywny i często się zawiesza – powstają błędy bez udziału człowieka.

Organizacja turniejów to nie tylko rejestrowanie uczestników, zapisywanie zgłoszonych wyników i parowanie (dobieranie w pary w celu rozegrania meczu) graczy w każdej rundzie. To też dbanie o poprawną strukturę turnieju (parowanie nie odbywa się losowo – zależy od zdobytych punktów każdego gracza), o pilnowanie ram czasowych rund, o weryfikowanie i edycję błędnie zgłoszonych wyników i cofanie niepoprawnie wprowadzonych akcji.

Należy też zaznaczyć, że takie narzędzie winno również służyć jako baza danych do odbytych / zarejestrowanych w systemie turniejów. Choćby dlatego, aby ułatwić interwencje sędziów za pomocą ułatwienia dostępu do historii zachowań danego gracza. Biorąc też pod uwagę potężne narzędzia do analizy danych, warto też zapamiętywać szczegóły dotyczące decków, którymi grali uczestnicy. Pozwoli to też na zautomatyzowanie części rejestracji w postaci automatycznego skanowania listy kart, które rejestruje do turnieju gracz. Zarówno organizator jak i uczestnik otrzymają informacje rzetelnie i natychmiastowo, jeżeli pojawi się problem z rejestracją, a wykluczanie możliwych błędów ludzkich jest kluczowe przy organizacji turniejów, które potrafią liczyć nawet kilka tysięcy graczy.

Zakres odpowiedzialności systemu

System ma umożliwiać użytkownikowi (czyli organizatorowi lub sędziemu) sprawne tworzenie instancji turnieju, przechowywanie niezbędnych informacji służących poprawnemu przebiegowi turnieju, rejestrowanie użytkowników i monitorowanie odpowiednich akcji sędziów. Dodatkowo chcemy pamiętać dane zarejestrowanych użytkowników (wszystkich – organizatorów, sędziów lub graczy) i ich relacje związane z danymi turniejami. System ma pozwolić na modyfikacje wprowadzonych danych w celu naniesienia poprawek do odpowiednich danych. Dane winny być zapisywane w trwałej formie, która pozwoli na ich odtworzenie przy następnym uruchomieniu programu.

Dodatkowo system powinien dbać o zachowanie inwariantów turnieju – winne być wykrywane niezgodności dotyczące decków, limitów czasowych lub parowania graczy. Dokładniejsza specyfikacja jest opisana w Comprehensive Rules gry MtG z dodatkiem opisującym specyfikacje turniejowe.

Użytkownicy systemu

System winien być w stanie przełączać się między widokami dla **organizatora** i dla **sędziego**, czyli jedynych użytkowników systemu. Widok organizatora i sędziego są wspólne – można ich w teorii traktować jak jednego użytkownika, lecz klient rozróżnia akcje organizatora i sędziego. Mimo tego, klienci nie chcą izolacji funkcjonalności tych aktorów – zakłada się, że obaj aktorzy korzystają z tej samej aplikacji na tej samej maszynie – wymiennie.

Wymagania użytkownika

System powinien przechowywać informacje o:

- **Osobach**, które są związane z turniejem. Wyszczególniając:
 - **Gracza**
 - **Sędziego**
 - **Organizatora** (ponieważ system lokalny danego organizatora może przechowywać [za pomocą importowania i eksportowania] informacje na temat innych, nieprowadzonych przez niego turniejów. Warto też mieć w nim informacje o samym sobie)
- **Turniejach**, które składają się z:
 - **Rund**, które składają się z
 - **Meczów**, które składają się z:

- **Gier**, w których gra dwóch **Graczy**
- **Interwencji** prowadzonych przez **Sędziów**
- **Decków**, którymi dany **Gracz** grał (poszczególne w każdym **Turnieju**).
- **Serii Turniejowych**, do których należą niektóre **Turnieje**
- **Sponsorów**, którzy sponsorują odpowiednie **Turnieje**
- **Nagród**, które są przyznawane **Graczom** w **Turniejach**

Z następującymi ograniczeniami, których sam powinien pilnować:

1. Dana Osoba może być zarówno Graczem, Organizatorem jak i Sędzią, lecz w kontekście **tego samego Turnieju**, jeżeli jest Graczem, to nie może być też Organizatorem lub Sędzią. Dana Osoba będąca dowolną z tych specjalizacji może stać się później dodatkowo inną z tych specjalizacji.
2. Dany Gracz nie może zostać przypisany do danego Turnieju, jeżeli jego deck jest nielegalny w danym formacie Turnieju. System powinien korzystać z odpowiedniego, udostępnionego przez Wizards of the Coast API, które dynamicznie aktualizuje listę nielegalnych kart w danych formatach. Ważnym jest jednak, aby system nie zgłaszał błędów, jeżeli w zakończonym już Turnieju, jakiś deck stał się nielegalny – takie sytuacje się zdarzają. System **nie** powinien zgłaszać żadnych komunikatów w takich przypadkach.
3. Dana Gra może się zakończyć wygraną jednego z Graczy lub Remisem, zapisując wynik w postaci trzech liczb – wygranych, przegranych i zremisowanych gier w kontekście danego gracza. Kontekst drugiego gracza powinien być automatycznie wypełniany lustrzanym odbiciem wpisanego wyniku, w celu wyłączenia organizatora w powielaniu wpisywania wyników – mogą powstać błędy przy ręcznym wprowadzaniu takich wyników. Należy też udostępnić przyciski, które będą reprezentowały automatyczne wypełnianie tych pól najpopularniejszymi wynikami, ale trzeba też dodać ręczne ich wprowadzanie, ponieważ zdarzają się bardzo specyficzne przypadki.
4. Zakończony Mecz może mieć co najwyżej dwie wygrane po stronie tego samego Gracza (format BO3 – Best of Three). Zwycięzca otrzymuje 3 punkty w kontekście Turnieju, przegrany 0, a w przypadku remisu każdy Gracz otrzymuje po 1 punkcie.
5. Parowanie Graczy w kontekście Rundy odbywa się za pomocą porównania punktów. W pierwszej kolejności parujemy graczy z taką samą liczbą punktów. Jeżeli to niemożliwe, to losujemy Gracza z większą liczbą punktów, który zagra z losowym Graczem z mniejszą liczbą punktów. Dwójka Graczy **nie** może zagrać ze sobą więcej niż raz w jednym Turnieju. Jeżeli liczba graczy jest nieparzysta, to losowy Gracz wśród tych z najmniejszą liczbą punktów dostaje tak zwanego bye'a, czyli darmową wygraną i nie uczestniczy w żadnej Grze podczas tej Rundy.
6. Sędziowie mogą Interweniować w dane Gry z różnych powodów. Każda Interwencja winna być zarejestrowana w Turnieju z podaniem przyczyny i werdyktu. Należy zapamiętywać potencjalne kary (penalties) wystosowane przez Sędziego w kierunku Gracza(y). Gracz, który osiągnie **trzy** takie same ostrzeżenia (warnings) podczas tego samego Turnieju zostaje zdyskwalifikowany.

Konkretne dane, które należy pamiętać dla każdej grupy informacji to:

1. Osoba:
 - a. *Imię*
 - b. *Nazwisko*
 - c. *Przynależność do organizacji* (opcjonalna – sama nazwa)
2. Gracz:
 - a. *Numer DCI* (unikalny identyfikator gracza)

Diagram kontekstowy

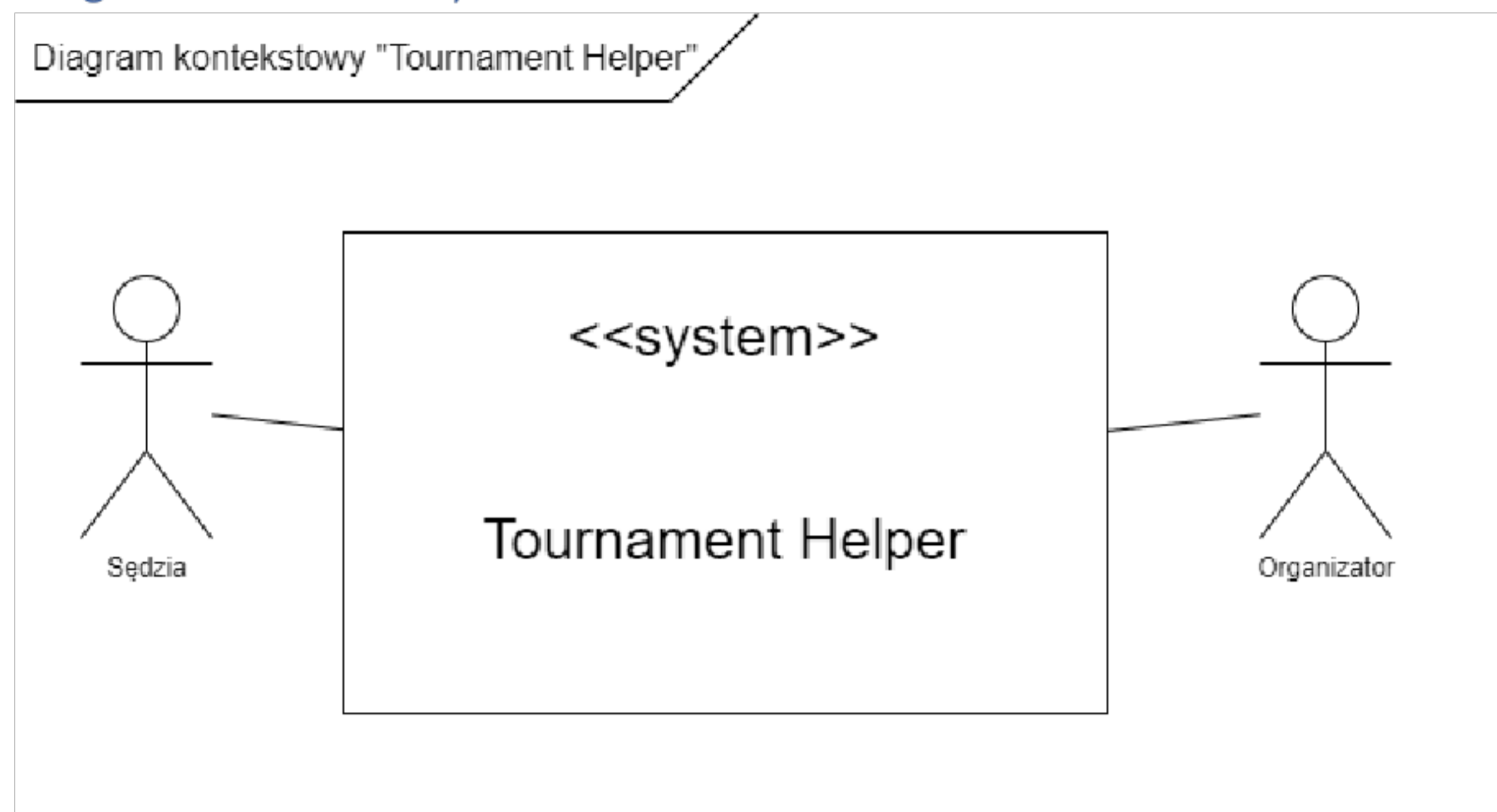
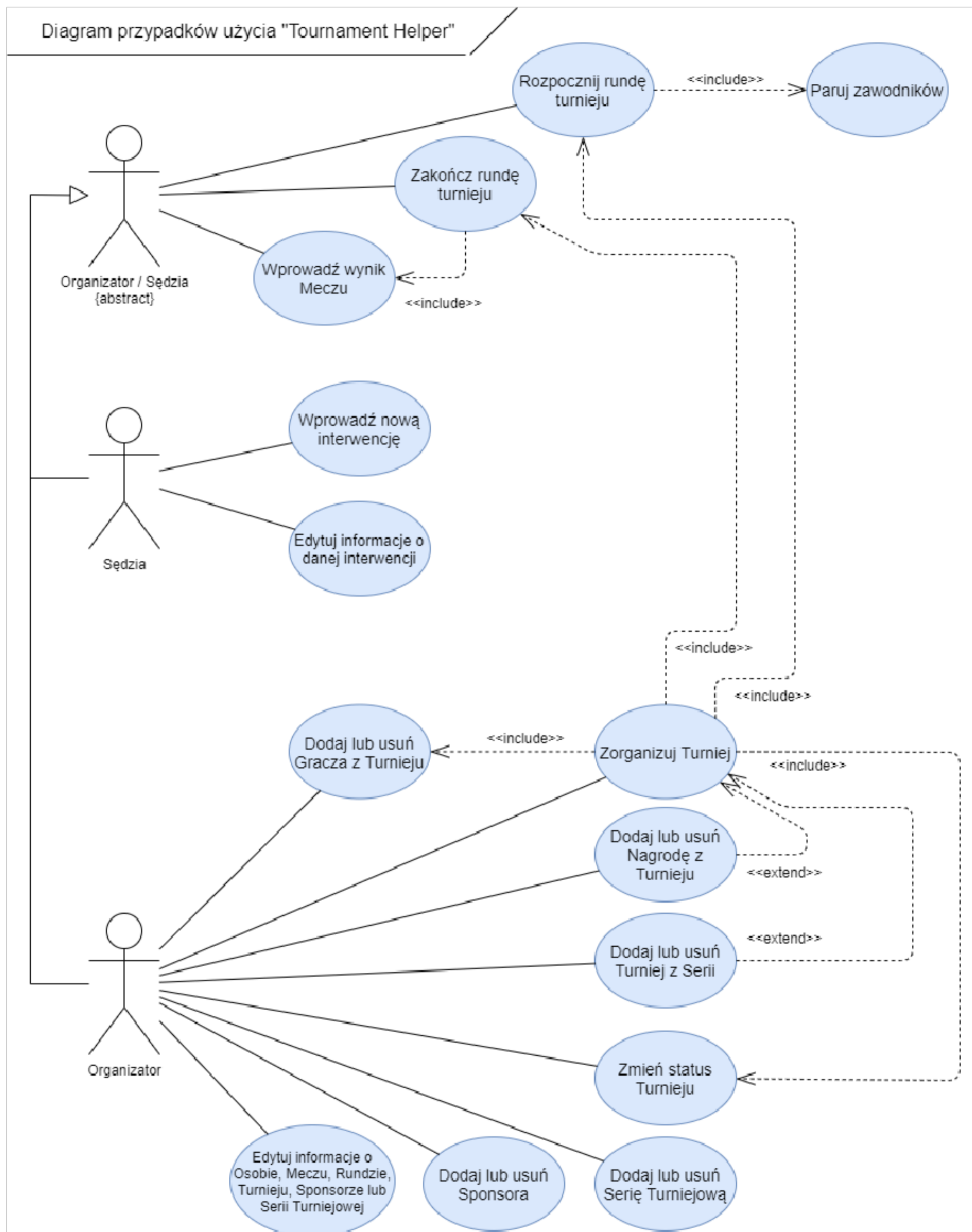
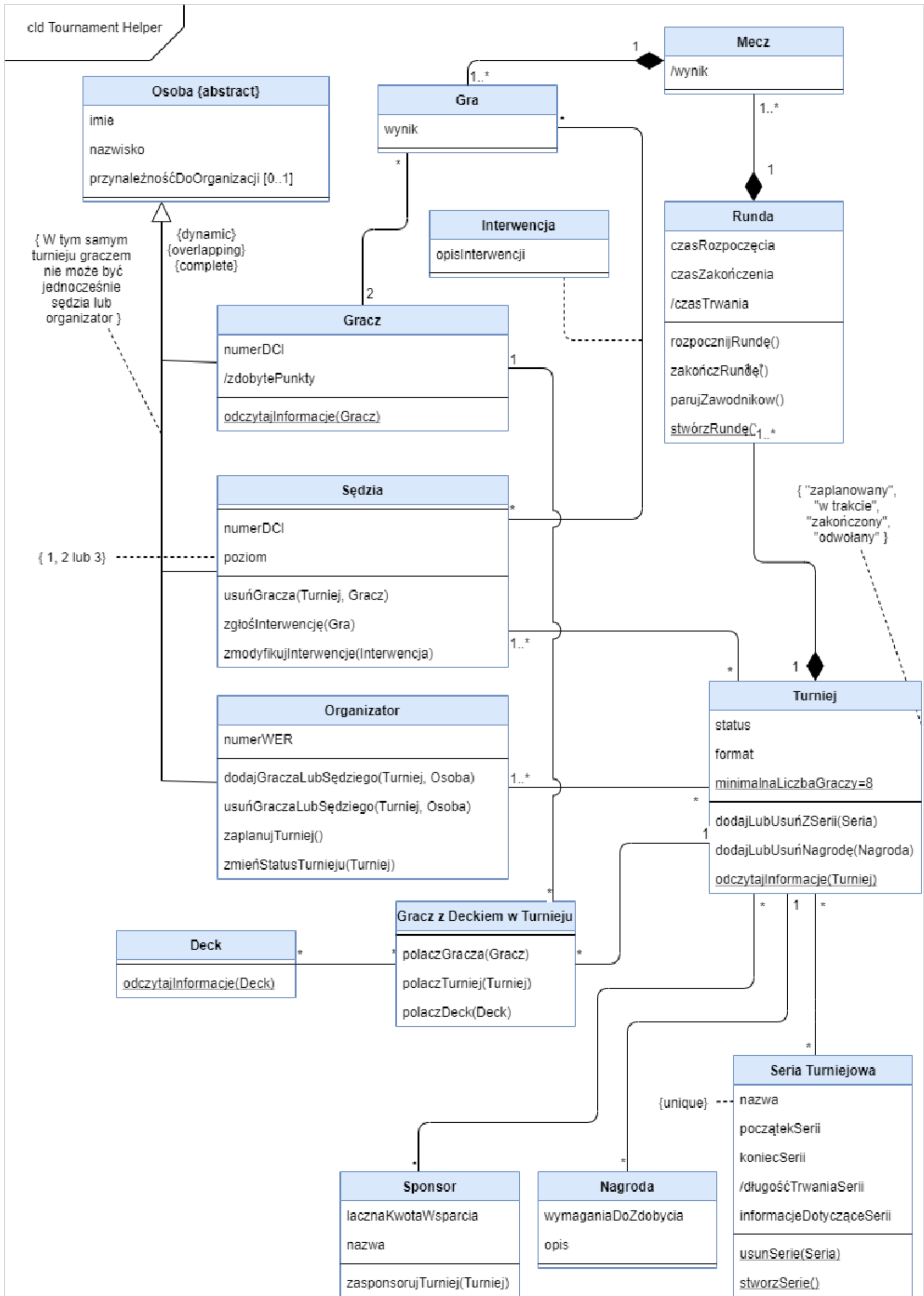


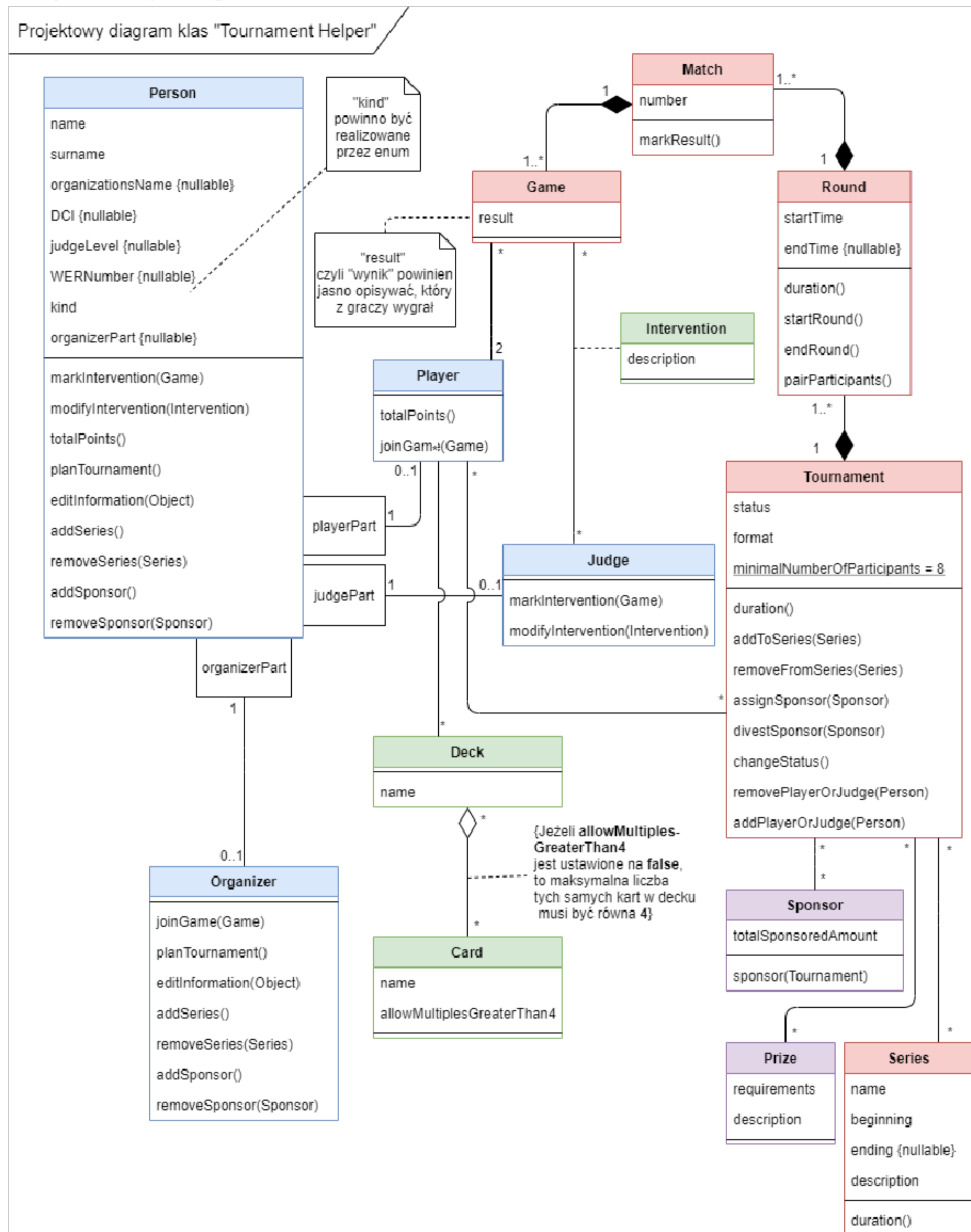
Diagram przypadków użycia



Analityczny diagram klas



Projektowy diagram klas



Wymagania niefunkcjonalne

1. Łatwość użytkowania

System winien być prosty i intuicyjny w użyciu. Należy odpowiednio pogrupować i opisać funkcjonalności, aby niedoświadczony w użytkowaniu komputera klient mógł z łatwością opanować funkcje systemu. Należy zadbać o to, aby kluczowe w użyteczności funkcjonalności były podobnie pogrupowane i podobnie działały jak te z oficjalnej aplikacji WER, aby użytkownicy mogli stosunkowo łatwo przenieść się na nasz system.

Czas szkolenia **nie** powinien wynieść **więcej niż godzinę**. Koniecznością winno być jedynie przedstawienie różnic i dodatkowych możliwości między użytkowaniem Tournament Helpera a WERa.

Należy dodatkowo stworzyć okienka z podpowiedziami dla funkcjonalności, których w WERze nie było.

2. niezawodność

System winien działać niezawodnie przez co najmniej 24 miesiące od uruchomienia, zakładając wystarczającą ilość miejsca w pamięci komputera, na którym system jest zainstalowany (w celu przechowywania historii jak i samych danych). W przypadku awarii nie z winy użytkownika naprawa systemu nie powinna zająć dłużej niż jeden dzień.

3. Bezpieczeństwo danych

System winien przechowywać dane w sposób bezpieczny, czyli taki, który uniemożliwia ich wyciek w rozumieniu odpowiadającym uznanym standardom bezpieczeństwa na dzień sporządzania dokumentu.

Wstępną implementacją ekstensji danych będzie silnik MySQL Server w najnowszej wersji, lecz zakłada się, że w przyszłości silnik może zostać zmieniony. Dlatego podczas implementacji należy skorzystać z uniwersalnego narzędzia, dzięki któremu będzie można w przyszłości z łatwością zmienić silnik bazy. Obecnym rozwiązaniem jest framework Hibernate w wersji najnowszej.

4. Łatwa modyfikowalność

System winien wspierać łatwą modyfikowalność działania, aby poprawnie odwzorowywać naturę dynamicznie zmieniających się zasad (czasem fundamentalnych) opisujących cechy abstrakcji, z którymi pracuje (między innymi Turnieje czy wymagania odnośnie Decków).

Opis przyszłej ewolucji systemu

W przyszłości planowane jest dodanie do systemu dodatkowych funkcjonalności, które pozwolą na interakcje z Graczami. System może być wzbogacony o licznik czasu, aby Gracze mogli sprawdzać, ile czasu zostało w Rundzie, o aplikację mobilną, aby Gracze mogli zgłaszać wyniki bez potrzeby informowania Organizatora / Sędziego werbalnie lub za pomocą specjalnych kartek do wypełnienia i o dodatkowy system rejestracji Graczy, co pozwoli na zautomatyzowanie rejestracji na Turnieje i walidacji spisów Decków.

Słownik pojęć z dziedziny problemowej

- *Gracz*: osoba, która uczestniczy w Grach – gra w nich za pomocą własnego Decku.
- *Deck*: talia kart zawierająca **co najmniej 60** z nich. Dodatkową restrykcją jest to, że żadna z kart nie może się powtarzać w decku więcej niż **4** razy. Istnieją wyjątki, które są określone zasadami lub tekstem na samej karcie.

- *Gra*: część Meczu, gdzie dwaj Gracze grają swoimi Deckami. Wynikiem może być wygrana jednego z Graczy lub remis.
- *Mecz*: składa się z Gier – ich liczba jest zależna od struktury Turnieju i Rundy. Tradycyjnie gra się do dwóch wygranych jednego gracza (tak zwane *Best Of Three*) lub do zakończenia czasu Rundy. Wynik Meczu jest zgłaszany w formie dwóch, lustrzanych trójek liczb naturalnych: **X-Y-Z**, gdzie pierwsza liczba oznacza liczbę wygranych Gier danego Gracza, druga oznacza liczbę przegranych Gier danego Gracza, a trzecia liczbę remisów. Z perspektywy drugiego Gracza wynik wygląda tak samo, ale z odwróconymi pierwszymi dwoma wartościami.
- *Runda*: określa czas, podczas którego rozgrywane są *Mecze*. Wszystkie startują jednocześnie i muszą skończyć się najpóźniej wraz z zakończeniem Rundy (szczegóły dotyczące przekraczania tego czasu zostały pominięte – nie są potrzebne. W razie wątpliwości patrz termin "*Terminacja*" w zewnętrznych źródłach). Podczas Rundy każdy Gracz musi zostać przydzielony do pary i rozegrać Mecz. W przypadku nieparzystej liczby Graczy, ten z najniższą liczbą punktów otrzymuje darmowe zwycięstwo (liczone jako **2-0-0**).
- *Turniej*: określa wydarzenie, w którym biorą udział Gracze. Podzielony na Rundy, podczas których rozgrywane są *Mecze*, reprezentuje zawody między tymi Graczami. Zasady Turnieju obowiązują każdego i są z góry narzucone. Minimalna liczba graczy to **8**. Wymagane jest, aby każdy z Turniejów miał co najmniej jednego Sędziego. Organizator Turnieju winien jest zapewnić sędziowskie zasoby.
- *Sędzia*: osoba, która sędziuje dany Turniej. Jego zadaniem jest pilnowanie porządku i pomaganie Graczom w udzielaniu informacji i w rozstrzyganiu sporów. Optymalnym jest mieć co najmniej jednego Sędziego na 20-30 Graczy, lecz nie jest to wymagane. Sędzia winien również reagować samodzielnie na zauważone łamanie zasad i udzielać stosownych werdyktów, których szczegóły nie są ważne w tym projekcie.
- *Organizator*: reprezentuje osobę, która organizuje Turniej. Dba o jego poprawne prowadzenie i nadzoruje pracę Sędziów. Jest też odpowiedzialny za egzekwowanie kar przez nich wymierzonych.
- *Seria Turniejowa*: grupa, do której mogą należeć Turnieje. Wiążą się z nimi szczegóły, które nie mają znaczenia dla systemu. Należy jednak być w stanie zwrócić informacje o graczach bazując je na asocjacjach pośrednich z danymi Seriami Turniejowymi (na przykład Gracz, który zdobył największą liczbę punktów we wszystkich Turniejach w danej Serii).

Scenariusz przypadku użycia "Zorganizuj Turniej"

| Sekcja | Treść |
|-----------------|--|
| Aktorzy | Organizator, Sędzia |
| Warunki wstępne | Mamy co najmniej 8 Graczy, którzy chcą się zapisać na Turniej; Mamy co najmniej 1 Sędziego na 20-30 Graczy; |
| Warunki końcowe | W bazie danych został zapisany Turniej, który odbył się zgodnie z procedurami. |

| Rezultat | Wspomaganie odpowiedniego przeprowadzenia Turnieju |
|-------------------------|---|
| Scenariusz główny | <ol style="list-style-type: none"> 1. System wyświetla listę graczy z możliwością dodania Gracza. 2. Organizator rozpoczyna Turniej. 3. System wyświetla listę graczy zapisanych do Turnieju. 4. Organizator / Sędzia rozpoczyna Rundę z opcją automatycznego parowania Graczy. 5. System paruje Graczy i wyświetla ich listę (w postaci pary Graczy na Mecz). 6. Organizator / Sędzia wpisuje kolejno zgłoszone przez Graczy wyniki. 7. System czeka na wpisanie wszystkich wyników (powtarzany jest wielokrotnie punkt 6.). 8. Organizator / Sędzia wpisuje ostatni wynik. 9. System aktywuje przycisk z możliwością rozpoczęcia następnej Rundy. 10. Punkty od 4. do 9. są powtarzane $\lceil \log_2 n \rceil$ razy¹ wraz z numerowaniem Rund. 11. System wyświetla informacje o zakończeniu Turnieju i liście Graczy posortowanej po zdobytych punktach. 12. Organizator / Sędzia zamyka okno z prowadzeniem Turnieju. |
| Scenariusz alternatywny | <ol style="list-style-type: none"> 2.1. Organizator / Sędzia odwołuje Turniej. 4.1. Organizator / Sędzia wywołuje parowanie Graczy ręcznie i rozpoczyna Rundę z ręcznie sparowanymi Graczami. 6.1. Sędzia przechodzi na widok wprowadzenia Interwencji do danej Gry, wprowadza odpowiednie dane i powraca do punktu 6. 9.1. System aktywuje przycisk zakończenia Turnieju (ponieważ ostatnia Runda dobiegła końca) i przechodzimy od razu do punktu 11. |

¹ $\lceil \log_2 n \rceil$ reprezentuje optymalną liczbę Rund tak, aby można było drabinkowo (system pucharowy) rozstrzygnąć, kto jest zwycięzcą, gdzie n jest liczbą Graczy biorących udział w Turnieju.

Diagram stanów dla klasy Turniej

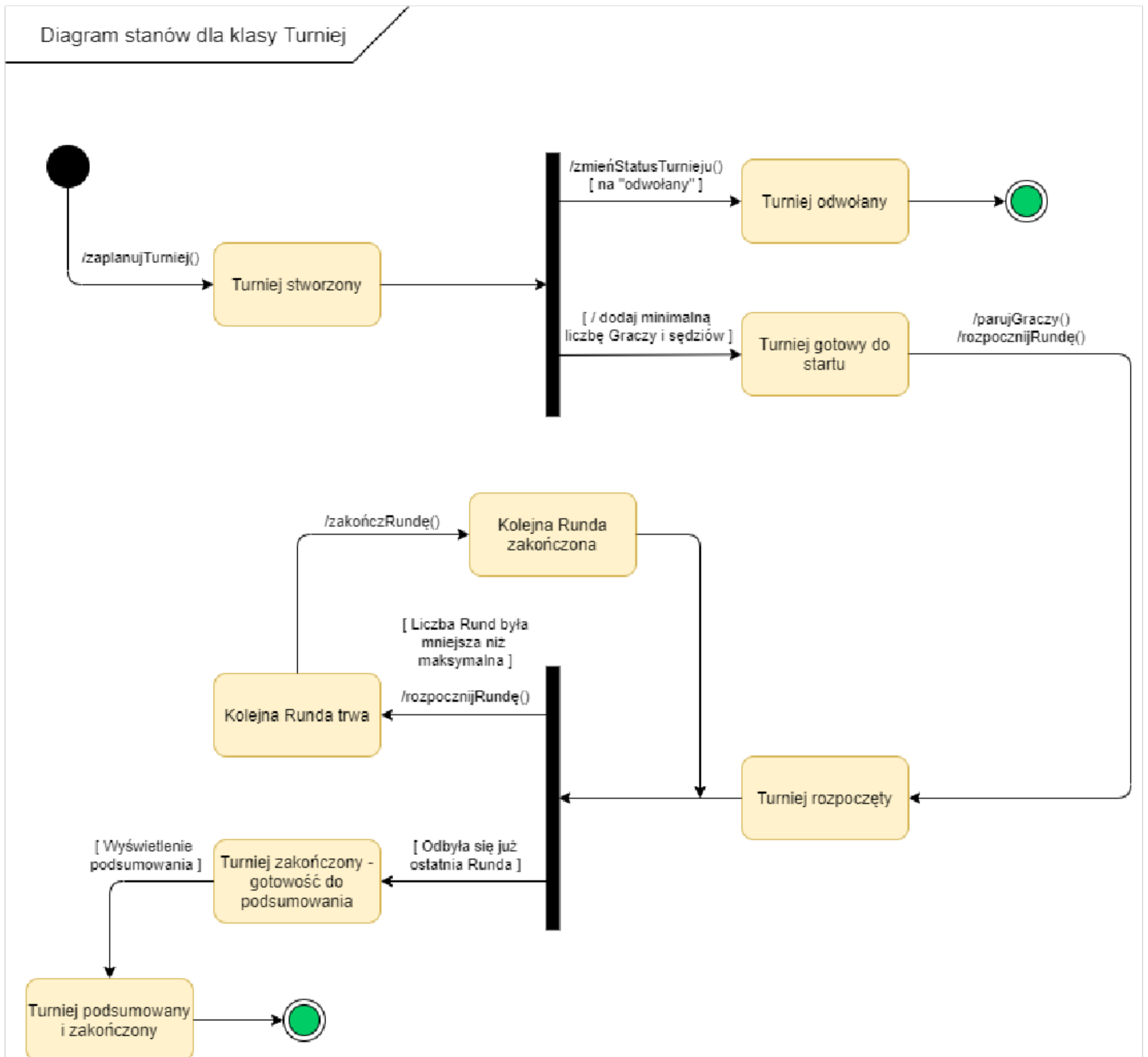


Diagram aktywności dla przypadku użycia "Zorganizuj Turniej"

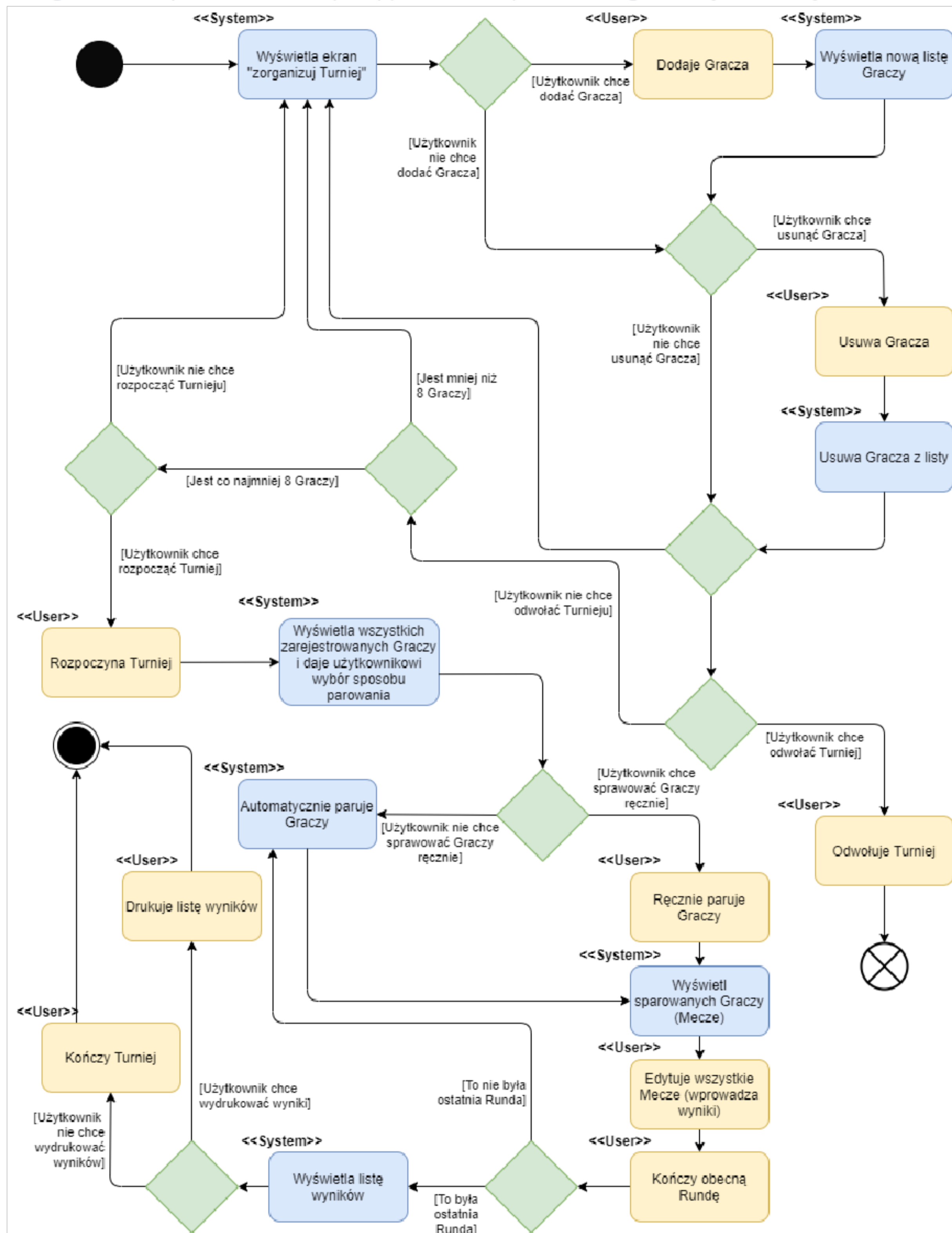
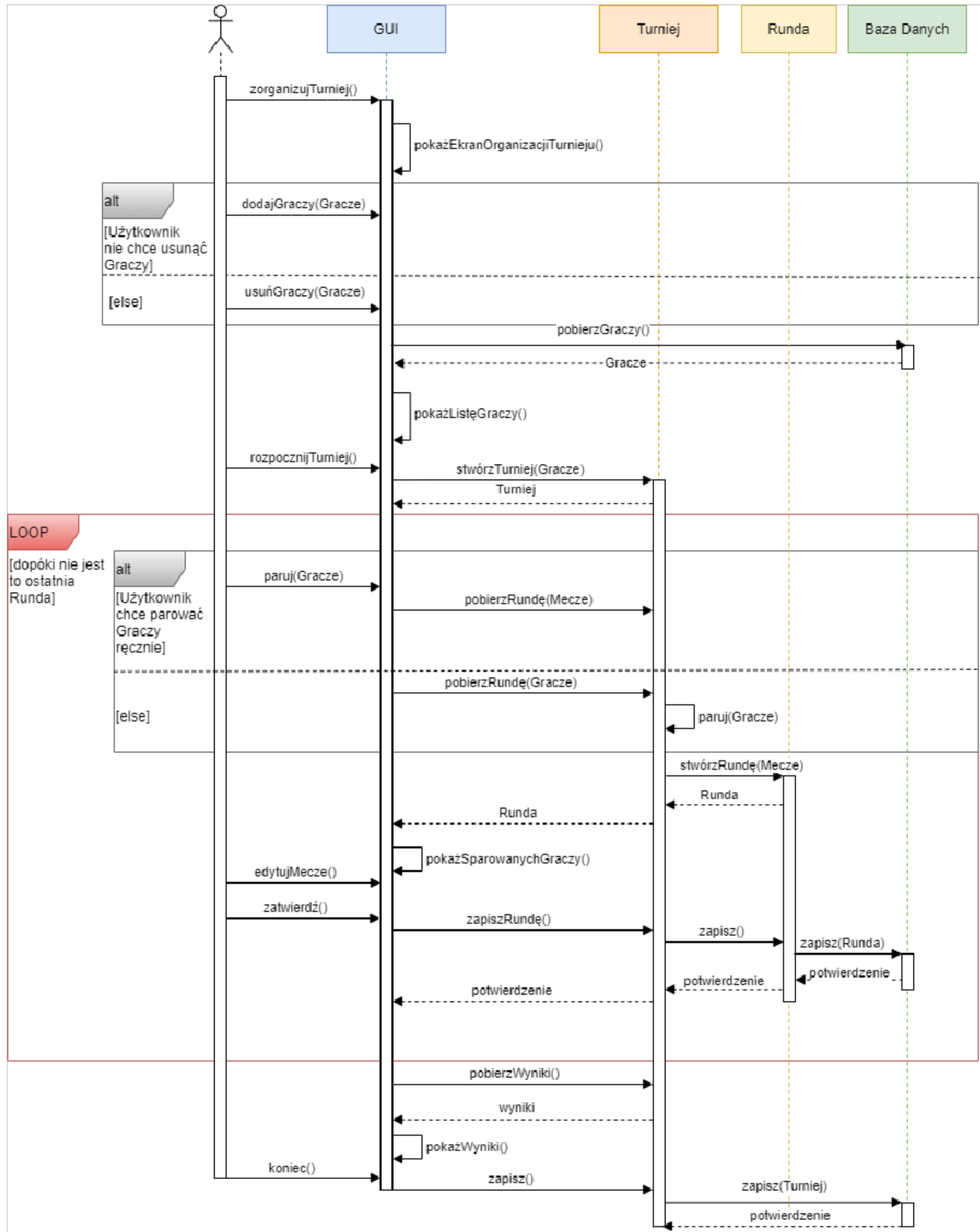
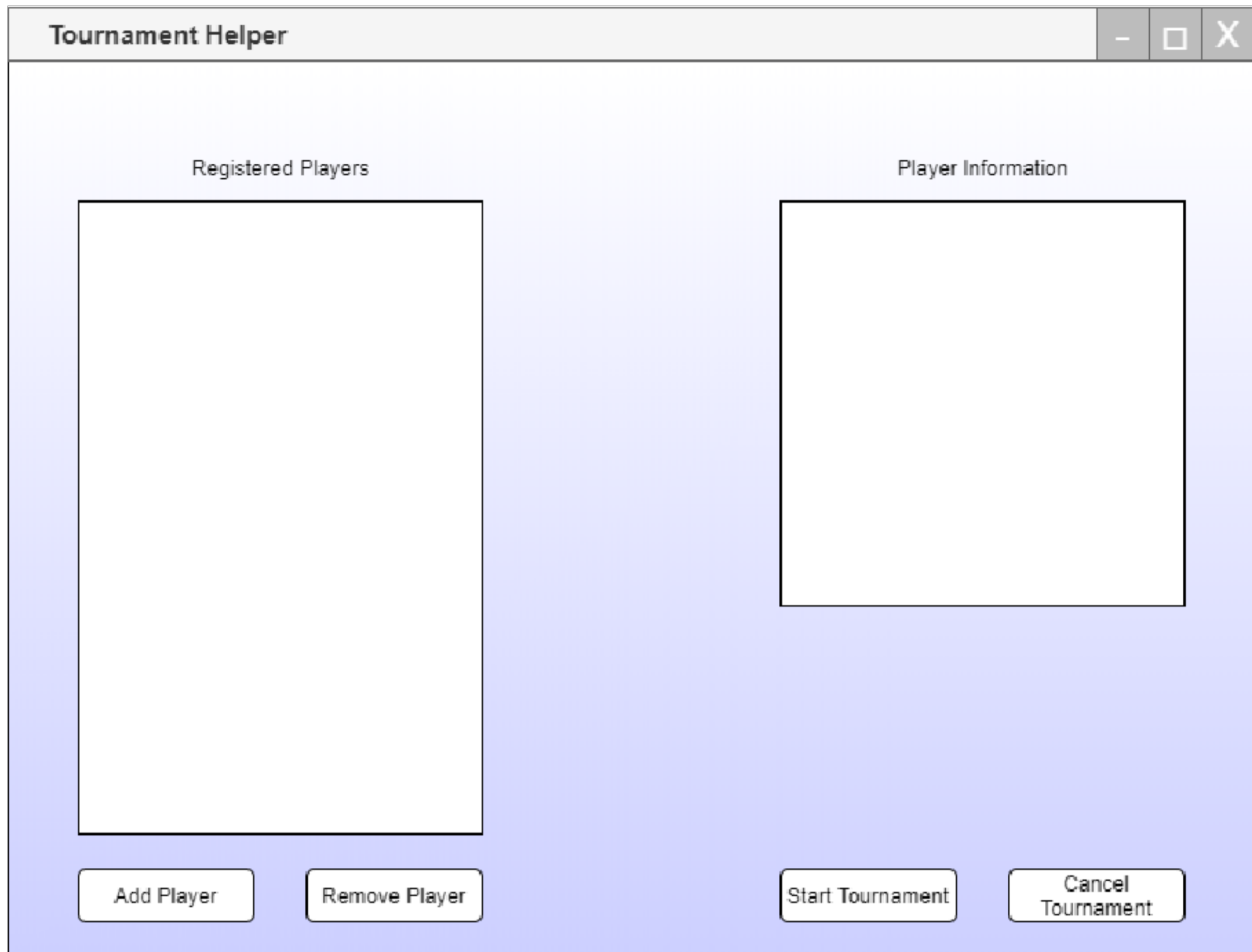


Diagram sekwencyjny dla przypadku użycia "Zorganizuj Turniej"



Projekt GUI dla przypadku użycia "Zorganizuj Turniej"

- Ekran początkowy organizowania Turnieju:



Jest to pierwszy ekran, który pojawia się po wybraniu z menu opcji z organizacją Turnieju. Z tego poziomu widzimy Graczy obecnie zarejestrowanych na Turniej z możliwością dodania nowego Gracza lub usunięcia go z kolejki. Po wybraniu Gracza możemy wyświetlić jego informacje po prawej stronie lub go usunąć.

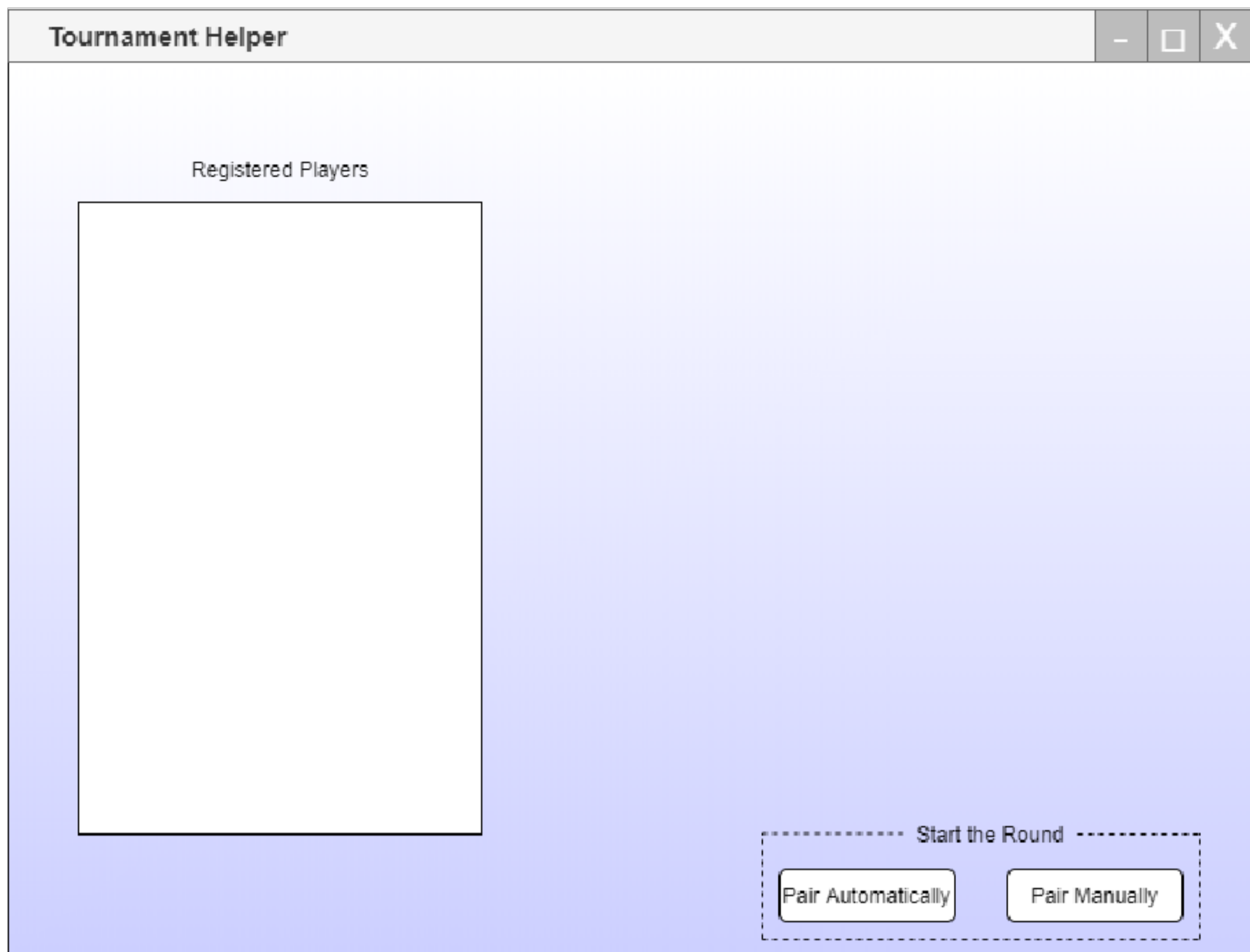
Możemy Turniej rozpocząć (jeżeli mam co najmniej **8** Graczy) lub anulować.

- Ekran dodania nowego Gracza do Turnieju:

The screenshot shows a window titled "Tournament Helper" with standard window controls (minimize, maximize, close). The main content area is titled "Add a player" and is split into two columns. The left column is titled "Select a Player from the database" and contains a large, empty white rectangular area. Below this area is a grey "Add" button. The right column is titled "Input a new Player" and is enclosed in a dashed border. It contains three input fields: "Name" and "Surname" are side-by-side at the top, and "DCI" is centered below them. Below the "DCI" field is a white "Add" button.

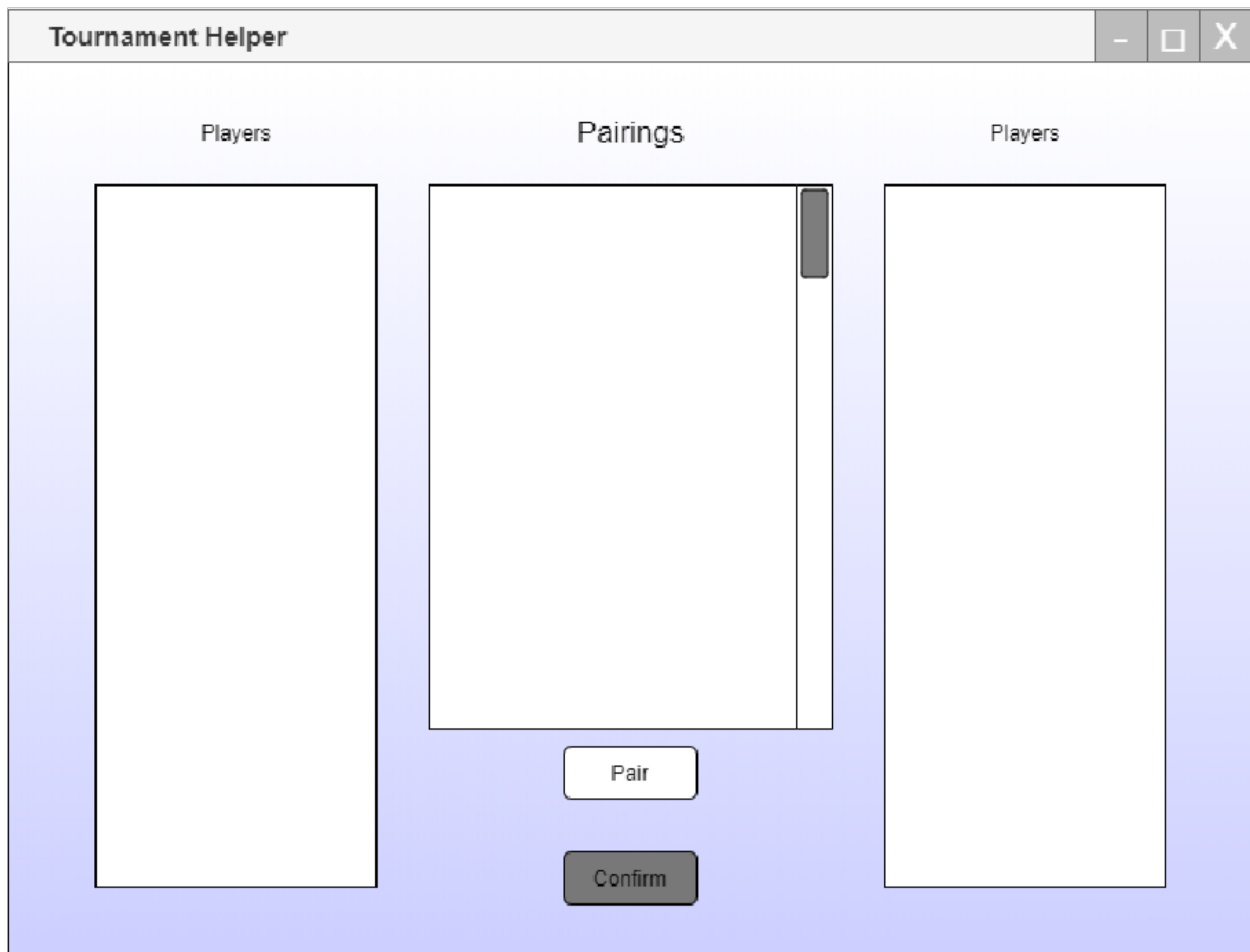
Po wybraniu opcji z dodaniem nowego Gracza do Turnieju mamy możliwość dodać Gracza z Graczy już zapisanych w naszej bazie danych lub wprowadzenie nowego Gracza, który pojawi się potem w naszej bazie niezależnie od tego, czy rozpoczniemy potem Turniej.

- Ekran podsumowujący zarejestrowanych Graczy z możliwością rozpoczęcia Turnieju:



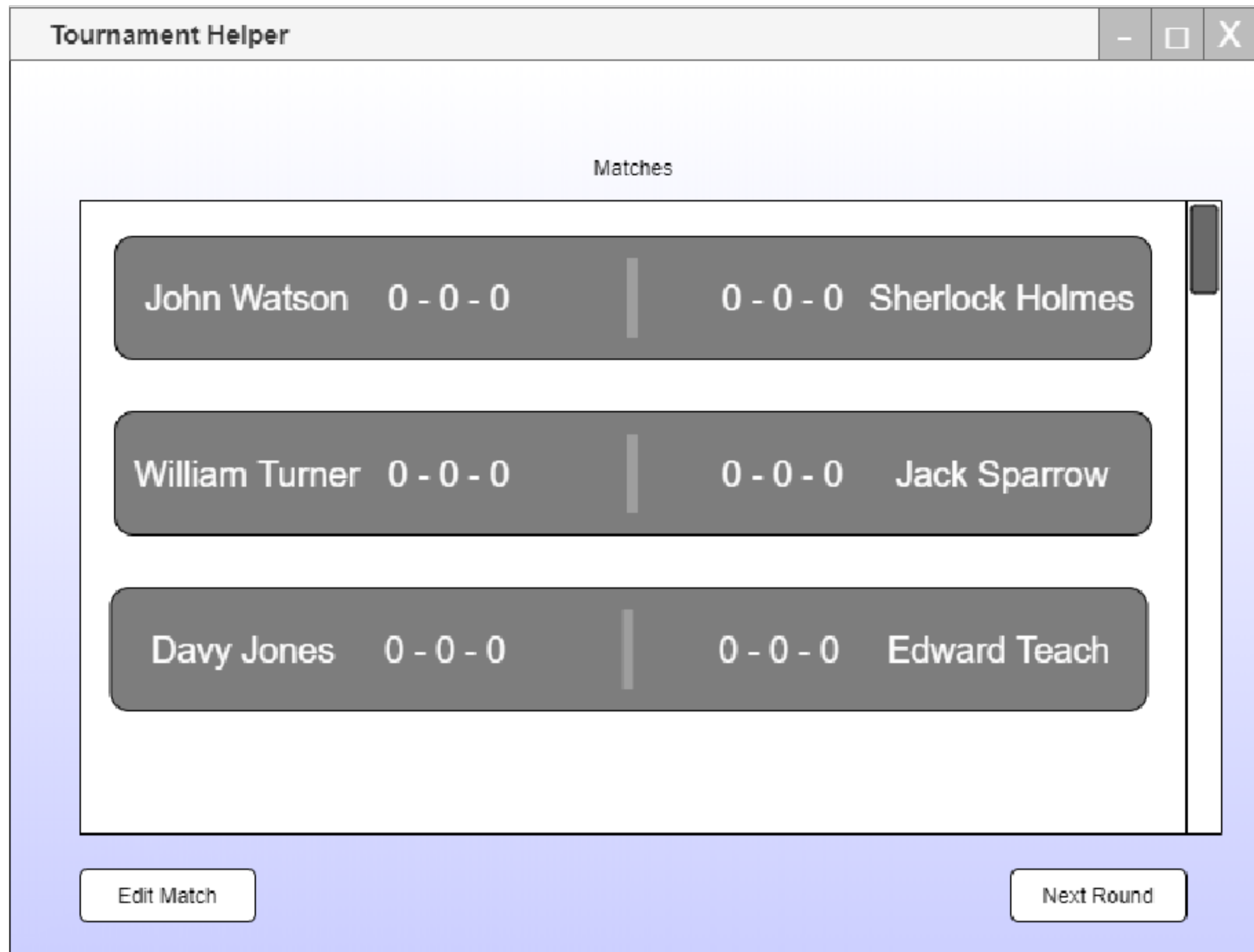
W tym ekranie widzimy wszystkich zarejestrowanych Graczy w postaci prostej listy i mamy możliwość ich sparowania (dobrania ich w pary). Możemy to zrobić ręcznie lub automatycznie – system rozlosuje te pary.

- Ekran umożliwiający ręczne parowanie Graczy:



Widzimy tutaj dwie listy Graczy (identyczne jak na poprzednim ekranie) – każdy Gracz znajduje się na każdej z list. Aby połączyć w pary dwóch Graczy należy ich zaznaczyć (system zadba o to, aby nie móc zaznaczyć 2 razy tego samego Gracza) na obydwu listach i kliknąć przycisk parujący. Gracze ci zostają usunięci z list po bokach i dodani (już sparowani) do listy pośrodku.

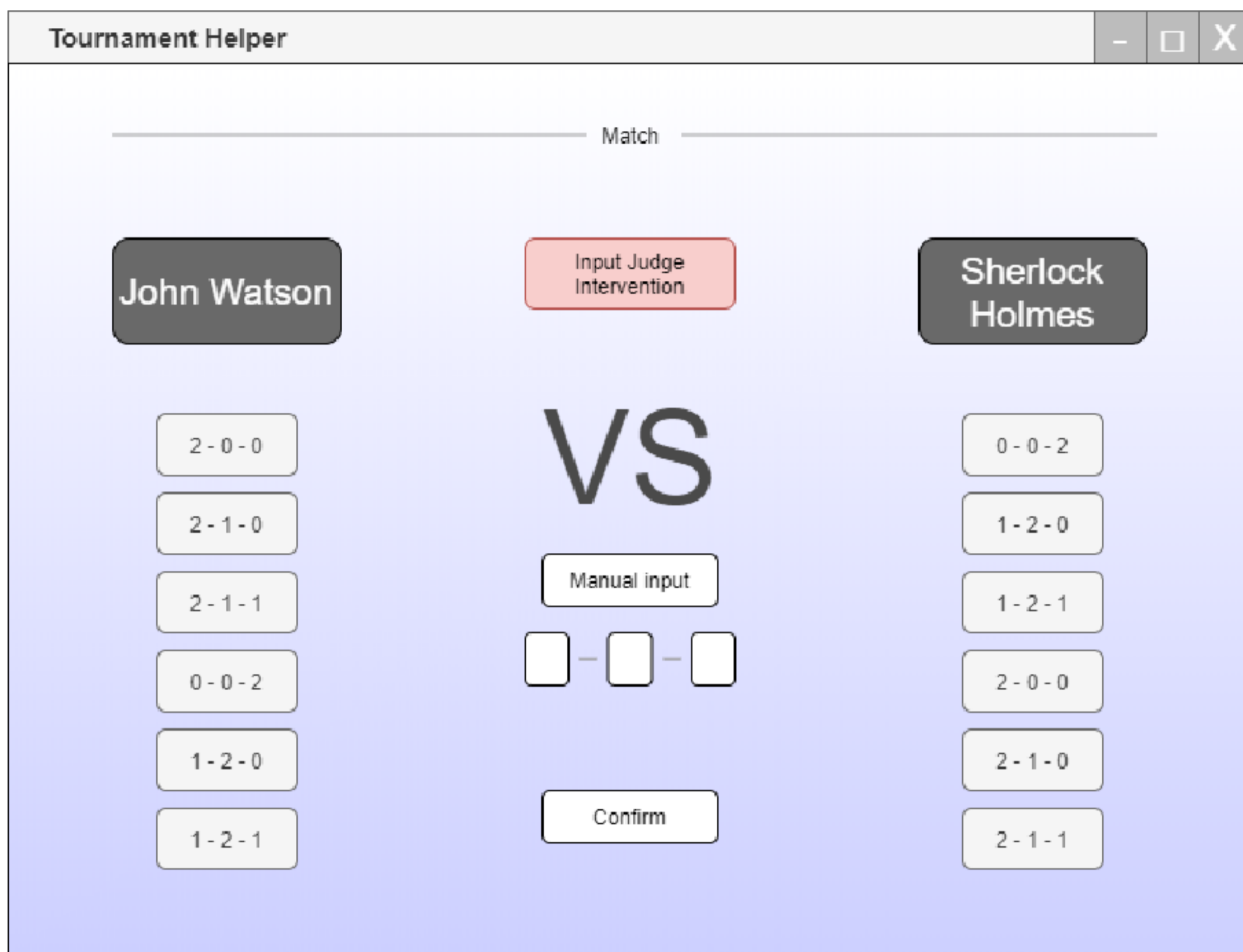
- Ekran pokazujący obecnie rozgrywane mecze (w postaci listy) między sparowanymi Graczami – wyświetla się on po rozpoczęciu Rundy (niezależnie czy parowaliśmy Graczy ręcznie czy automatycznie):



W tym ekranie możemy zaznaczyć Mecz i go edytować (wprowadzić zgłoszony przez Graczy wynik lub wprowadzić zapis z interakcji sędziego). Kiedy wszystkie Mecze zostaną już zakończone, zgłoszone i ich stany zapisane, możemy przejść do następnej Rundy (implikuje to już **koniecznie automatyczne** parowanie następnych Graczy –

zależnie od wyników). Jeżeli była to ostatnia Runda to nie pojawi się znowu ekran Rundy.

- Ekran z edycją Meczu:



Warto zwrócić uwagę na wygodne opcje wprowadzenia wyniku poprzez kliknięcie w przycisk (zgodnie z wymaganiami wprowadziliśmy wygodne wybieranie

najpopularniejszych wyników gier za pomocą przycisków) lub, w przypadku mniej typowych wyników, możemy wprowadzić je ręcznie.

- Ekran wprowadzenia interwencji przez sędziego (po wciśnięciu odpowiedniego przycisku z poprzedniego ekranu):



The screenshot shows a window titled "Tournament Helper" with standard window controls (minimize, maximize, close). The main content area has a light blue gradient background. At the top, the text "Match Intervention" is centered between two horizontal lines. Below this, the names "John Watson" and "Sherlock Holmes" are displayed in dark grey rounded rectangular buttons, separated by a large "VS" in the center. Underneath, the text "Input Intervention details below:" is centered between two horizontal lines. A large, empty white rectangular box is provided for entering details. At the bottom center, there is a white button with the text "Confirm".

- Ekran podsumowujący wyniki Turnieju:



Widzimy tutaj ekran z podsumowaniem całego Turnieju. Użytkownik ma teraz możliwość po prostu zakończyć Turniej lub wpierw wydrukować listę zawodników wraz z wynikami. Kończy to również sam Turniej.

Decyzje projektowe i skutki analizy dynamicznej

- Trwałość programu jest uzyskiwana poprzez wykorzystanie bazy danych z silnikiem MySQL Server oraz Frameworkiem Hibernate dla języka programowania Java, która jest wyborem narzędzia do implementacji całego projektu.
- Każda klasa w drodze implementacji otrzyma niezbędne metody *get()* i *set()* dla parametrów. Nie zostały one uwzględnione w diagramie z racji ograniczonego miejsca.
- Niektóre klasy zostały wzbogacone o metody, które są konieczne do prawidłowego odwzorowania działania aplikacji (zgodnie z życzeniami klienta) – zarówno funkcjonalnie jak i implementacyjnie (przykładowo, framework Hibernate wymagał odpowiednich metod i adnotacji).
- Diagram przypadków użycia został wzbogacony i wiele dodatkowych przypadków, które powstały przez rozdzielenie bardziej skomplikowanych i złożonych przypadków. Ich relacje zostały uwzględnione odpowiednimi notacjami `<<include>>` oraz `<<extend>>`.
- Dziedziczenie typu *overlapping* i *dynamic* zostało zrealizowane przez agregację. Implikuje to usunięcie abstrakcyjności z cech klasy Osoby (Person).

Wykorzystane narzędzia i edytory

- Wszystkie diagramy i projekty GUI: portal **draw.io**.
- Edytor dokumentu: **Microfost Word 2013**