Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: Program do analizy turniejów szachowych

Autorzy: Patryk Walczyński , Mateusz Dziuba

Grupa: I1-222A

Kierunek: informatyka

Rok akademicki: …

Poziom i semestr: I/4

Tryb studiów: stacjonarne

Należy pozostawić wszelkie nagłówki tego dokumentu, a umieszczać treść w odpowiednich miejscach zamiast obecnych objaśnień.

Stronę tytułową można sformatować w dowolny sposób, ale należy pozostawić zawartość informacyjną w układzie pokazanym powyżej.

Praca powinna zostać złożona wyłącznie w formacie pdf. Przed wygenerowaniem ostatecznej wersji należy zaktualizować spis treści – wyświetlane dwa poziomy.

Niniejszą informację należy również usunąć z wersji końcowej.

Spis treści

[2 Odnośniki do innych źródeł 4](#_Toc1976793)

[3 Słownik pojęć 5](#_Toc1976794)

[4 Wprowadzenie 6](#_Toc1976795)

[4.1 Cel dokumentacji 6](#_Toc1976796)

[4.2 Przeznaczenie dokumentacji 6](#_Toc1976797)

[4.3 Opis organizacji lub analiza rynku 6](#_Toc1976798)

[4.4 Analiza SWOT organizacji 6](#_Toc1976799)

[5 Specyfikacja wymagań 7](#_Toc1976800)

[5.1 Charakterystyka ogólna 7](#_Toc1976801)

[5.2 Wymagania funkcjonalne 7](#_Toc1976802)

[5.3 Wymagania niefunkcjonalne 8](#_Toc1976803)

[6 Zarządzanie projektem 9](#_Toc1976804)

[6.1 Zasoby ludzkie 9](#_Toc1976805)

[6.2 Harmonogram prac 9](#_Toc1976806)

[6.3 Etapy/kamienie milowe projektu 9](#_Toc1976807)

[7 Zarządzanie ryzykiem 10](#_Toc1976808)

[7.1 Lista czynników ryzyka 10](#_Toc1976809)

[7.2 Ocena ryzyka 10](#_Toc1976810)

[7.3 Plan reakcji na ryzyko 10](#_Toc1976811)

[8 Zarządzanie jakością 11](#_Toc1976812)

[8.1 Scenariusze i przypadki testowe 11](#_Toc1976813)

[9 Projekt techniczny 12](#_Toc1976814)

[9.1 Opis architektury systemu 12](#_Toc1976815)

[9.2 Technologie implementacji systemu 12](#_Toc1976816)

[9.3 Diagramy UML 12](#_Toc1976817)

[9.4 Charakterystyka zastosowanych wzorców projektowych 12](#_Toc1976818)

[9.5 Projekt bazy danych 12](#_Toc1976819)

[9.6 Projekt interfejsu użytkownika 12](#_Toc1976820)

[9.7 Procedura wdrożenia 13](#_Toc1976821)

[10 Dokumentacja dla użytkownika 14](#_Toc1976822)

[11 Podsumowanie 15](#_Toc1976823)

[11.1 Szczegółowe nakłady projektowe członków zespołu 15](#_Toc1976824)

[12 Inne informacje 16](#_Toc1976825)

# Odnośniki do innych źródeł

* + Zarządzania projektem – sugerowane JazzHub
  + Wersjonowanie kodu – Github
  + System obsługi defektów – np. Bitbucket, JazzHub

# Słownik pojęć

# Wprowadzenie

## Cel dokumentacji

Celem dokumentacji jest przekazanie dokładnych informacji na temat oprogramowania.

Głównie jego celów, zastosowania, pracy nad nim i wykonania.

## Przeznaczenie dokumentacji

Dokumentacja jest przeznaczona dla osoby nadzorującej projekt, użytkownika oprogramowania lub osób zajmujących się implementacją.

## Opis organizacji lub analiza rynku

Oprogramowanie będzie przydatne dla szachistów i trenerów szachowych chcących przygotować się do turnieju. Z programu mogą korzystać pojedynczy gracze lub kluby szachowe wraz z trenerami .W przypadku graczy z tytułem mistrzowskim jest ich ponad 15 tysięcy więc każdy jest potencjalnym klientem natomiast liczba trenerów czy amatorów turniejów szachowych jest o wiele większa. Mimo dużej liczby graczy przygotowanie do turnieju wygląda podobnie u każdego, co powoduję że wszyscy potencjalni klienci mają ze sobą dużo wspólnego jeśli chodzi o wymagania odnośnie oprogramowania. Ze względu na obecną sytuację na świecie liczba zainteresowanych szachami mocno wzrosła i cały czas rośnie, co powoduję że liczba potencjalnych klientów rośnie.

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

Aplikacja pomagająca w przygotowaniu się do turnieju szachowego.

### Podstawowe założenia

System będzie służył do analizy gier pojedynczego gracza. Za pomocą oprogramowania będziemy wstanie zobaczyć jakie otwarcia wybiera nasz przeciwnik i czym odpowiada na otwarcia, a także z jakim rezultatem mu to wychodzi. Program przeanalizuje grę przeciwnika i poda jego silne i słabe strony. W przypadku analizy całych turniejów w celu nauki lub przygotowania na kolejne, program pokaże dokładną statystykę wraz z trendami i jak one ulegały zmianą.

### Cel biznesowy

Celem projektu jest dostarczenie narzędzia umożliwiającego przygotowanie się do turnieju szachowego. Najważniejszymi jego cechami które mają przekonać użytkownika do jego użytku są łatwość w użyciu i ogólnodostępność. Oprogramowanie będzie ogólnodostępne, a dostęp do jego bardziej zaawansowanych funkcji płatny.

### Użytkownicy

Lista użytkowników:

- Trener szachowy

- Profesjonalny gracz

- zwykły gracz

### Korzyści z systemu

(1) Przygotowanie planu treningowego

(2) Możliwość przygotowania się pod znanego przeciwnika, poznanie jego dobrych i słabych stron.

(3) Analiza gier zawodowych graczy w celach naukowych

### Ograniczenia projektowe i wdrożeniowe

Przepisy prawne:

Specyficzne technologie: Brak

Narzędzia: Język programowania Python wraz z licznymi bibliotekami

Protokoły komunikacyjne:

Aspekty zabezpieczeń: Zabezpieczony dostęp do serwera, kopia zapasowa danych,

dla kont użytkowników: dwustopniowa weryfikacja przy logowaniu.

Zgodność ze standardami:

Powiązania z innymi aplikacjami: Brak

Platforma sprzętowa: Komputer

System operacyjny: Windows

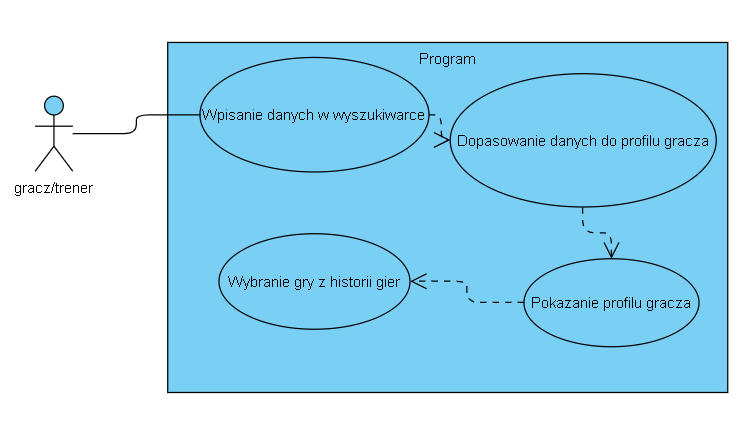
## Wymagania funkcjonalne

### Lista wymagań

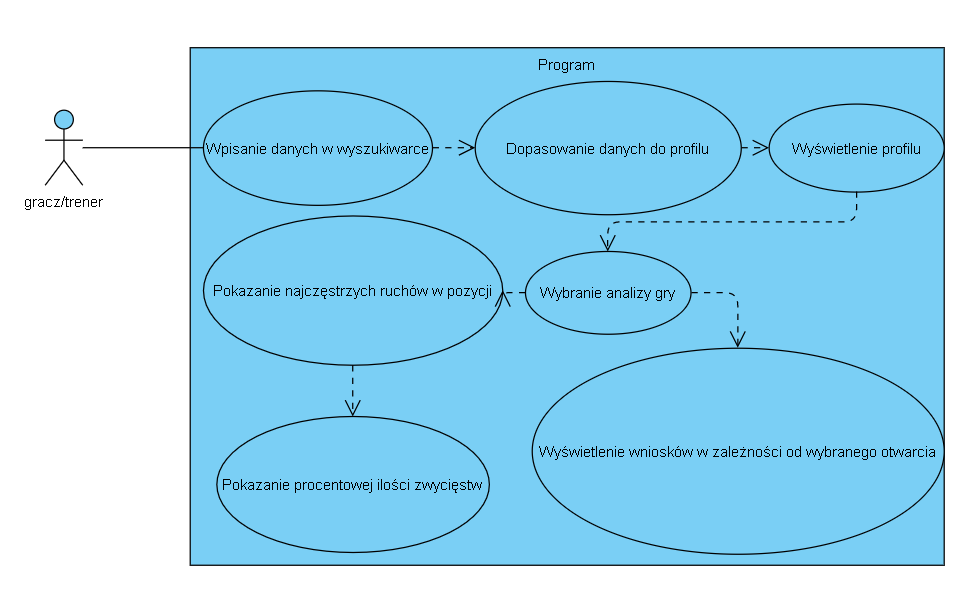
1. Przegląd wybranych partii danego gracza przez trenera/gracza
2. Analiza partii danego gracza przez trenera/gracza
3. Przegląd statystyk danego gracza
4. Przegląd statystyk danego turnieju
5. Logowanie

### Diagramy przypadków użycia

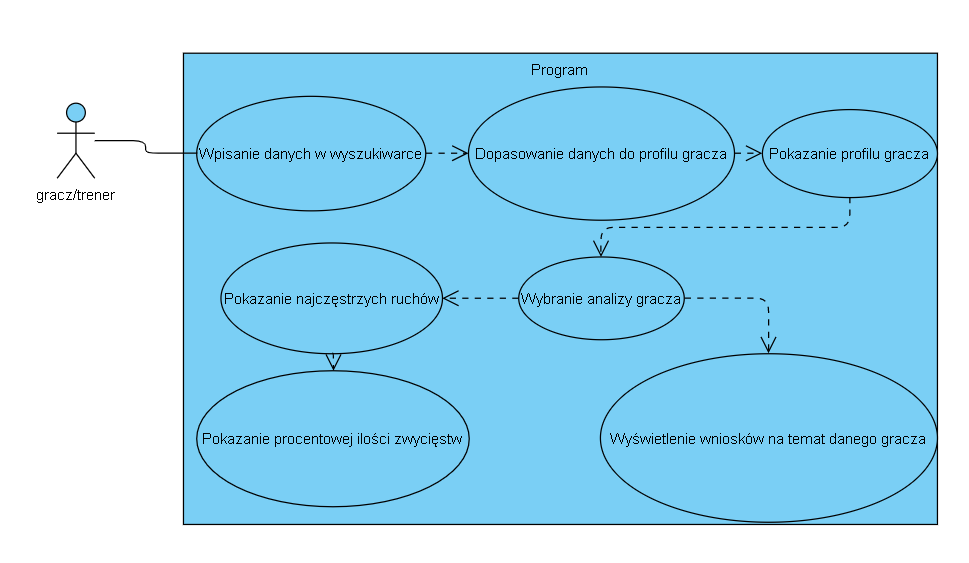
Przegląd wybranych partii danego gracza przez trenera/gracza



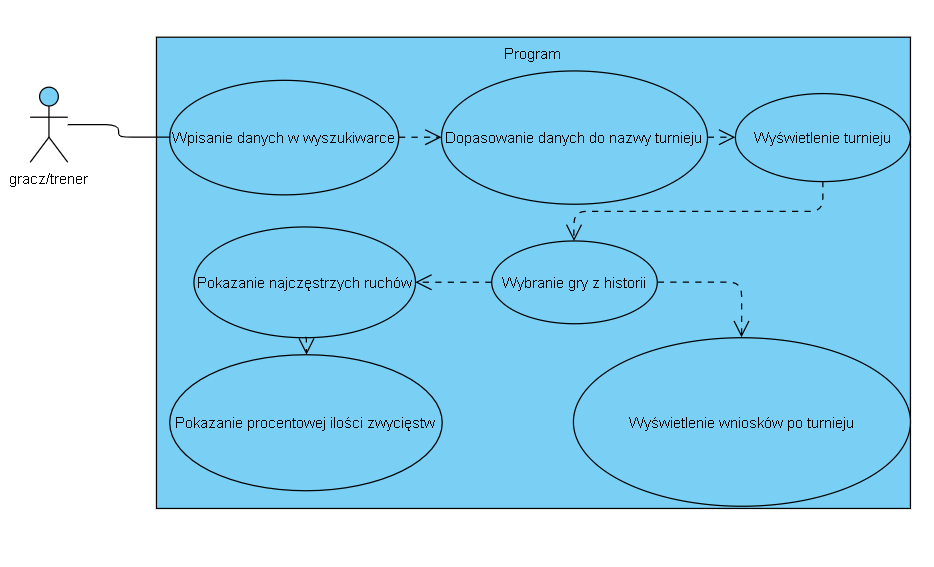
Analiza partii danego gracza przez trenera/gracza



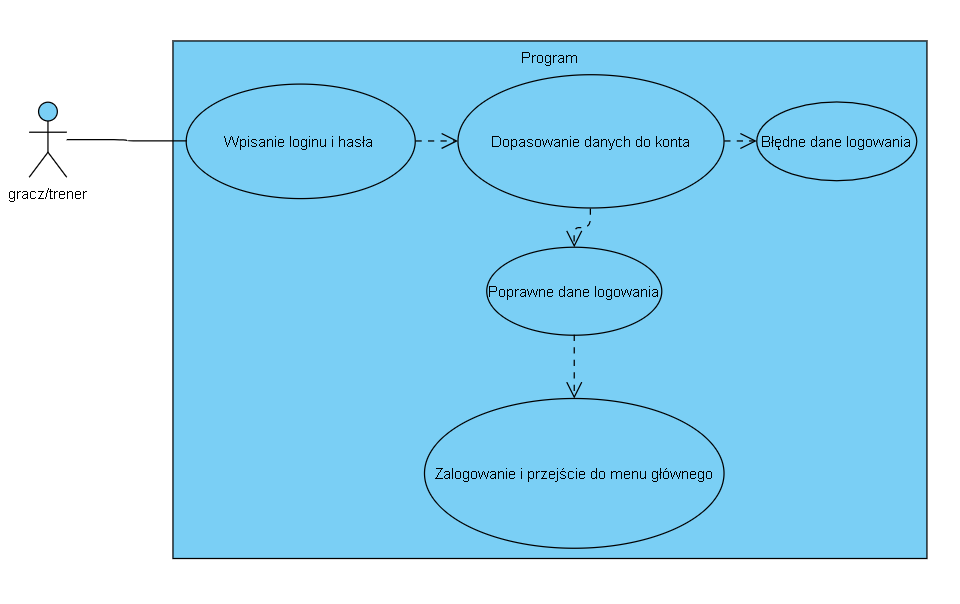
Przegląd statystyk danego gracza



Przegląd statystyk danego turnieju



Logowanie do programu



### Szczegółowy opis wymagań

|  |  |
| --- | --- |
| Numer | (1) |
| Nazwa | Przeglądnie wybranych partii |
| Cel | Możliwość obejrzenia partii z zwykłych gier lub turnieju |
| Użytkownicy | Zwykły/Profesjonalny gracz, Trener |
| Scenariusze | Warunki początkowe: Użytkownik chce obejrzeć wybraną partie.  Przebieg działania:  Użytkownik wpisuje danę zawodnika lub nazwę turnieju.  (1)  Wpisanie danych zawodnika:  Program wyświetla historie gier.  Użytkownik wybiera którą program ma wyświetlić.  Program wyświetla grę.  (2)  Wpisanie nazwy turnieju:  Program wyświetla listę uczestników.  Użytkownik wybiera gracza.  Program wyświetla historie gier z danego turnieju.  Użytkownik wybiera którą program ma wyświetlić.  Program wyświetla grę. |
| Efekty | Program wyświetla partie wraz ze wszystkimi potrzebnymi informacjami. |
| Częstotliwość | 3 |
| Istotność | 5 |

|  |  |
| --- | --- |
| Numer | (2) |
| Nazwa | Analiza partii danego gracza |
| Cel | Analiza swoich gier.  Pomoc w przygotowaniu treningu w oparciu o popełniane błędy/tendencje podopiecznego. |
| Użytkownicy | Zwykły/Profesjonalny gracz, Trener |
| Scenariusze | Warunki początkowe: Gracz/Trener chce obejrzeć wybraną partie.  Przebieg działania:  Gracz/Trener wpisuje danę zawodnika.  Użytkownik wybiera wyświetlenie historii gier.  Program wyświetla historie gier .  Gracz/Trener wybiera którą program ma wyświetlić.  Program wyświetla statystyki związane z daną partią. |
| Efekty | Gracz uczy się najczęstszych posunięć, a także tego czego przeciwnik nie będzie się spodziewał.  Trener jest w stanie przeanalizować gry podopiecznego i przygotować plan treningowy. |
| Częstotliwość | 4 |
| Istotność | 4 |

|  |  |
| --- | --- |
| Numer | (3) |
| Nazwa | Przegląd statystyk danego gracza |
| Cel | Nauka umiejętności i nawyków danego gracza |
| Użytkownicy | Zwykły/Profesjonalny gracz, Trener |
| Scenariusze | Warunki początkowe: Gracz/Trener chce przejrzeć statystyki danego gracza.  Przebieg działania:  Gracz/Trener wpisuje danę zawodnika.  Użytkownik wybiera wyświetlenie statystyk gracza.  Program wyświetla menu gdzie dostępne są różne informację i statystyki, które użytkownik może wybrać do wyświetlenia. |
| Efekty | Użytkownik jest w stanie poznać jak gra inny zawodnik, co pozwala w przygotowaniu się na turniej lub w przypadku trenera ten jest w stanie przygotować odpowiedni plan treningowy na danego przeciwnika. |
| Częstotliwość | 5 |
| Istotność | 5 |

|  |  |
| --- | --- |
| Numer | (4) |
| Nazwa | Przegląd statystyk danego turnieju |
| Cel | Poznanie statystyk danego turnieju szachowego |
| Użytkownicy | Zwykły/Profesjonalny gracz, Trener |
| Scenariusze | Warunki początkowe: Gracz/Trener chce przejrzeć statystyki danego turnieju.  Przebieg działania: Gracz/Trener wpisuje nazwę turnieju.  Program wyświetla menu gdzie dostępne są różne informację i statystyki, które użytkownik może wybrać do wyświetlenia. |
| Efekty | Efekty: Gracz jest w stanie zapoznać się z trendami turnieju i zaobserwować jak się one zmieniały wraz z trwaniem turnieju. |
| Częstotliwość | 3 |
| Istotność | 2 |

|  |  |
| --- | --- |
| Numer | (5) |
| Nazwa | Logowanie |
| Cel | Zalogowanie się do konta użytkownika |
| Użytkownicy | Zwykły/Profesjonalny gracz, Trener |
| Scenariusze | Warunki początkowe: Użytkownik chce zalogować się do programu  Przebieg działania: Użytkownik wpisuje dane   1. Dane logowania są poprawne   Program łączy podane dane z kontem i loguje użytkownika   1. Dane logowania są niepoprawne   Program wyświetla komunikat o błędnym loginie lub haśle  (2.1) Przy dalszych błędnych próbach logowania program przed zalogowaniem daje do wypełnienia CAPTCHA’e  . |
| Efekty | Użytkownik posiada unikalne konto z zapisanymi ustawieniami i preferencjami, a także w przypadku wykupionego konta premium także z dodatkowymi możliwościami |
| Częstotliwość | 5 |
| Istotność | 5 |

## Wymagania niefunkcjonalne

-Konto każdego użytkownika chronione loginem i hasłem

-Przy rejestracji użytkownik musi uzupełnić CAPTCHA

-Przy podaniu niepoprawnych danych logowania użytkownik musi uzupełnić CAPTCHA

-Kopia zapasowa serwera wykonywana co 7 dni

-Aplikacja będzie kompatybilna z innymi aplikacjami systemowymi

-Duża łatwość w korzystaniu przez użytkownika, proste i przejrzyste menu

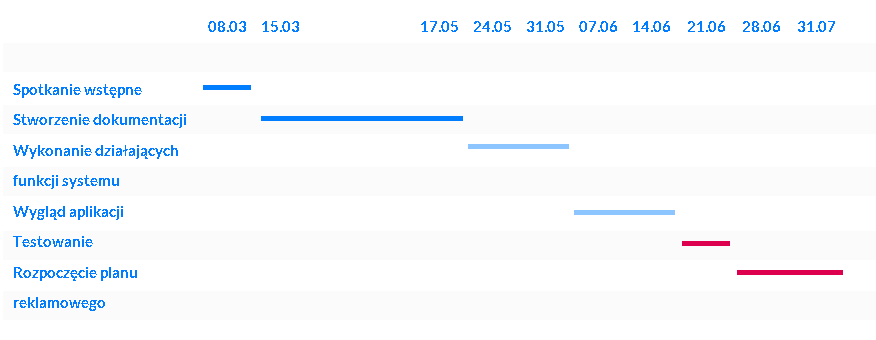
-Aplikacja będzie zajmować 100mb

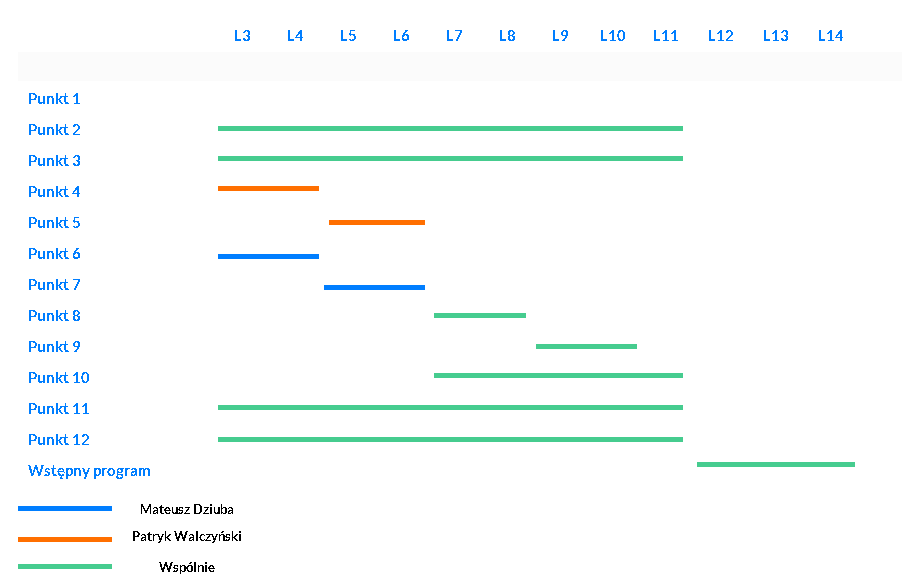
# Zarządzanie projektem(uzasadnienie)

## Zasoby ludzkie

* Kierownik Projektu **(1 osoba)** – Odpowiedzialny za całościowy przebieg projektu, podział zadań dla poszczególnych osób biorących udział w projekcie.
* Programiści **(2 osoby)** – Odpowiedzialni za wykonanie funkcjonalnego oraz wydajnego programu.
* UI Designer **(2 osoby)** – Odpowiedzialni za programowanie pod względem wyglądu aplikacji. Dba o wizualny wygląd aplikacji dla komfortu pracy użytkowników.
* Testerzy **(2 osoby)** – Odpowiedzialni za testowanie programu , wszystkich zaimplementowanych funkcji programu, przetestowanie działania programu z różnymi danymi, testowanie pod względem komfortu pracy z daną aplikacją.

## Harmonogram prac





## Etapy/kamienie milowe projektu

1. Ustalenie tematu projektu
2. Ustalenie wstępnych założeń i funkcjonalności projektu
3. Ustalenie harmonogramu pracy
4. Utworzenie dokumentacji projektu
5. Utworzenie funkcjonalnego programu
6. Zadbanie o wizualną część aplikacji.
7. Sprawdzenie poprawności programu
8. Testowanie
9. Ewentualna poprawa programu
10. Podsumowanie

# Zarządzanie ryzykiem

## Lista czynników ryzyka

Wypełniona lista kontrolna

* Błędy w kompilacji programu
* Problemy z integracją
* Zła organizacja pracy
* Mała ilość darmowych danych do przetworzenia
* Błędy w obliczaniu przez wykonywalne funkcje
* Dane na podstawie rozegranych gier (nie zawsze najlepsze wybory)
* Błędne dane
* Ograniczony czas na wykonanie projektu
* Wysokie obciążenie podzespołów poprzez wykonywalne funkcje

## Ocena ryzyka

prawdopodobieństwo i wpływ

|  |  |  |  |
| --- | --- | --- | --- |
| Lp. | Czynnik ryzyka | Wpływ | Prawdopodobieństwo |
| 1 | Ograniczony czas na wykonanie projektu | Nieukończenie projektu  na czas | 40% |
| 2 | Błędy w kompilacji Programu | Błędne działanie programu | 20% |
| 3 | Błędy w obliczaniu przez wykonywalne funkcje | Błędne działanie programu | 30% |
| 4 | Zła organizacja pracy | Niepoprawna organizacja pracy | 20% |
| 5 | Mała ilość darmowych danych do przetworzenia | Niska wiarygodność po przetworzeniu danych | 20% |
| 6 | Dane na podstawie rozegranych gier (nie zawsze najlepsze wybory) | Niska wiarygodność po przetworzeniu danych | 10% |
| 7 | Błędne dane | Niepoprawna analiza przetworzonych danych | 10% |
| 8 | Wysokie obciążenie podzespołów poprzez wykonywalne funkcje | Niska wydajność aplikacji | 20% |

## Plan reakcji na ryzyko

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

|  |  |  |
| --- | --- | --- |
| Lp. | Czynnik ryzyka | Plan reakcji |
| 1 | Ograniczony czas na wykonanie projektu | Przeniesienie ostatecznego terminu zakończenia projektu |
| 2 | Błędy w kompilacji Programu | Częste testowanie aplikacji podczas tworzenia. |
| 3 | Błędy w obliczaniu przez wykonywalne funkcje | Testowanie aplikacji pod względem wiarygodności przetworzenia danych |
| 4 | Zła organizacja pracy | Poprawa planu organizacji pracy |
| 5 | Mała ilość darmowych danych do przetworzenia | Większe przeszukanie darmowych danych lub inwestycja w zakupienie większych ilości danych |
| 6 | Dane na podstawie rozegranych gier (nie zawsze najlepsze wybory) | Podczas analizy danych branie pod uwagę różnych możliwości. |
| 7 | Błędne dane | Sprawdzenie baz danych względem ich poprawności |
| 8 | Wysokie obciążenie podzespołów poprzez wykonywalne funkcje | Usprawnienie wykonywalnych funkcji pod względem wydajności |

# Zarządzanie jakością

## Scenariusze i przypadki testowe

szczegółowy plan testowania systemu – głównie testowanie funkcjonalności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

|  |  |  |
| --- | --- | --- |
| **ID** | B1 | |
| **Nazwa Scenariusza** | Poprawność wyliczonych statystyk | |
| **Kategoria** | Poprawność obliczeń | |
| **Opis** | Weryfikacja zwróconych danych przez funkcje | |
| **Tester** |  | |
| **Termin** |  | |
| **Przebieg działań** | 1.Pobranie danych do przetworzenia  2.Użycie aplikacji do wygenerowania statystyk turnieju  3.Własne obliczenie statystyk na podstawie danych  4.Weryfikacja poprawności wygenerowanych statystyk w porównaniu do własnych obliczeń | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Pobrane po danym turnieju | Statystyka danego turnieju | Poprawność wyliczeń przez aplikacje |

|  |  |  |
| --- | --- | --- |
| **ID** | B2 | |
| **Nazwa Scenariusza** | Poprawność działania wizualnego aplikacji | |
| **Kategoria** | Testowanie pod względem wizualnym | |
| **Opis** | Testowanie Wizualnego działania aplikacji, poprawności używanych przycisków, działania dodawania danych. | |
| **Tester** |  | |
| **Termin** |  | |
| **Przebieg działań** | 1.Testowanie możliwości dodawania danych do wyliczeń  2.Testowanie możliwych przycisków w aplikacji | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Losowe dane z turnieju | Obliczone statystyki | Poprawna możliwość dodania danych do testowania, działające przyciski do zmian w statystykach |

|  |  |  |
| --- | --- | --- |
| **ID** | B3 | |
| **Nazwa Scenariusza** | Testowanie wydajności systemu | |
| **Kategoria** | Test wydajności, optymalizacji | |
| **Opis** | Testowanie pod względem szybkości obliczanych statystyk | |
| **Tester** |  | |
| **Termin** |  | |
| **Przebieg działań** | 1.Użycie losowych danych po turnieju  2.Testowanie pod względem szybkości wyliczanych statystyk  3. Zwracanie uwagi które czynności mogą się wykonywać najdłużej  4. Próba optymalizacji systemu do większej wydajności, zwiększenia szybkości wyliczenia statystyk | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Losowe dane po turnieju | Statystyki z turnieju | Zwrócone dane, odpowiednia wydajność systemu |

|  |  |  |
| --- | --- | --- |
| **ID** | B4 | |
| **Nazwa Scenariusza** | Testowanie wyszukiwania zawodnika za pomocą jego danych | |
| **Kategoria** | Wyszukiwanie | |
| **Opis** | Wpisanie danych różnych zawodników w wyszukiwarce graczy za pomocą poprawnych i niepoprawnych danych | |
| **Tester** |  | |
| **Termin** |  | |
| **Przebieg działań** | 1.Wpisanie nazwy zawodników z poprawnymi danymi  2. Wpisanie nazwy zawodników z niepoprawnymi danymi | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Imiona i Nazwiska graczy, niektóre z celowymi błędami | Wyszukanie danego zawodnika | Zwrócone dane, Brak problemów przy lekko błędnym danych wejściowych |

|  |  |  |
| --- | --- | --- |
| **ID** | B5 | |
| **Nazwa Scenariusza** | Przesuwanie pionków po planszy | |
| **Kategoria** | Używanie planszy | |
| **Opis** | Sprawdzanie działania poprawności przesuwania pionków | |
| **Tester** |  | |
| **Termin** |  | |
| **Przebieg działań** | 1.Tester przesuwa pionki na poprawne pola i sprawdza wyświetlanie się danych  2. Tester przesuwa pionki na niepoprawne pola w celu sprawdzenia zachowania programu | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Lista poprawnych i nie poprawnych ruchów w pozycji | 1. Wyświetlenie statystyk po wykonaniu ruchu 2. Program nie pozwala na wykonanie niepoprawnego ruchu w sytuacji powtórzenia błędnego ruchu system informuje o błędzie | Zwrócone dane, wyświetlenie poprawnych statystyk |

|  |  |  |
| --- | --- | --- |
| **ID** | B6 | |
| **Nazwa Scenariusza** | Sprawdzenie poprawności danych w bazie danych | |
| **Kategoria** | Poprawność wprowadzonych danych | |
| **Opis** | Sprawdzanie danych w systemie | |
| **Tester** | Zwykły użytkownik | |
| **Termin** | Cały czas | |
| **Przebieg działań** | 1.Tester sprawdza poprawność informacji zawartych w programie | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Dane zawarte w programie | 1. Poprawność danych 2. Zgłoszenie nieprawidłowości | Zwrócone dane |

|  |  |  |
| --- | --- | --- |
| **ID** | B7 | |
| **Nazwa Scenariusza** | Poprawność działania programu przy zmianie ustawień | |
| **Kategoria** | Poprawność programu | |
| **Opis** | Zmiana ustawień rozdzielczości, języka, motywu w celu sprawdzenia działalności programu po ich zmianie | |
| **Tester** | Zwykły użytkownik | |
| **Termin** | Cały czas | |
| **Przebieg działań** | 1.Tester testuje różne funkcje następnie zmienia ustawienia i testuje je ponownie | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Podstawowe funkcje programu | 1. Brak zmian w działalności programu | Zwrócone dane |

|  |  |  |
| --- | --- | --- |
| **ID** | B8 | |
| **Nazwa Scenariusza** | Poprawne poruszanie się po turnieju | |
| **Kategoria** | Poprawność wyświetlania turnieju | |
| **Opis** | Program poprawnie będzie pokazywał kolejne mecze w dalszych częściach turnieju | |
| **Tester** |  | |
| **Termin** |  | |
| **Przebieg działań** | 1.Tester wchodzi na różne turnieje i się porusza między meczami w nich | |
| **Użyte dane** | **Zwrócone dane** | **Warunek zaliczenia** |
| Losowe turnieje | Program odpowiednio będzie pokazywał kolejne mecze przy różnych formatach turniejowych | Zwrócone dane |

# Projekt techniczny

## Opis architektury systemu

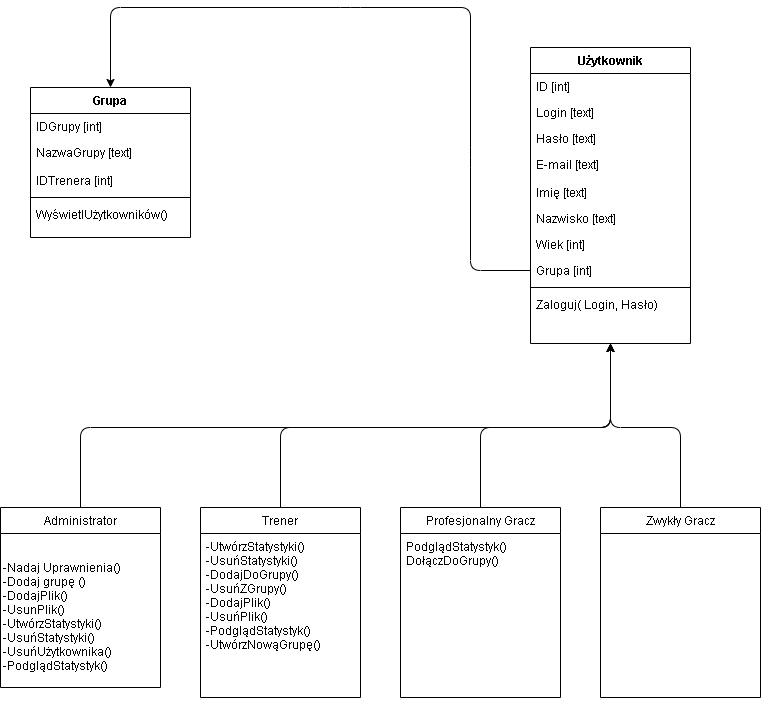
System będzie pobierał danę z bazy danych następnie analizował je i pokazywał wnioski w formie analizy przedstawionej w programie

## Technologie implementacji systemu

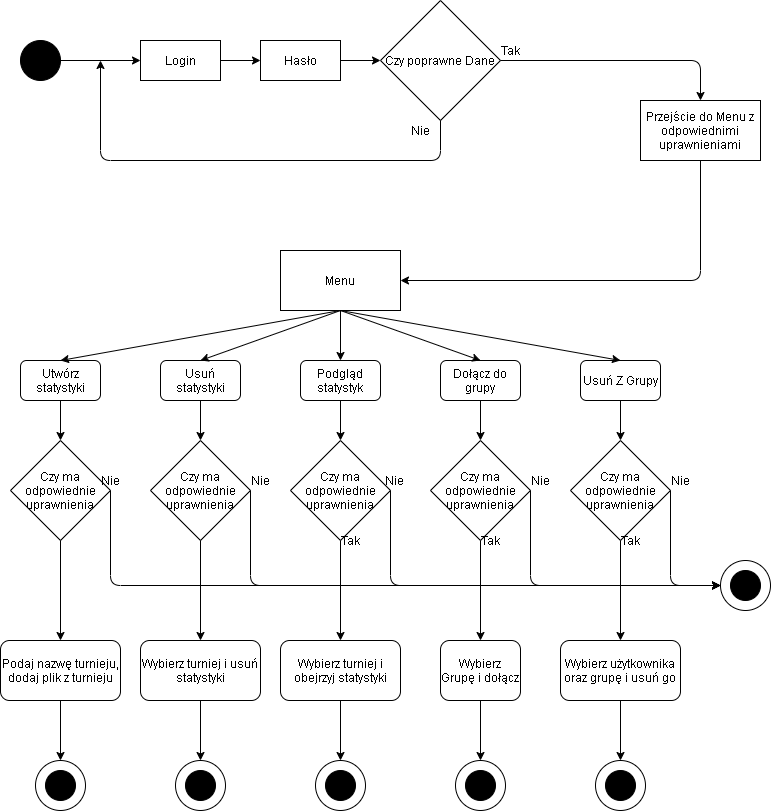
|  |  |
| --- | --- |
| Technologia | Uzasadnienie |
| Python | Używany do ogólnej interpretacji aplikacji, wszystkich jego funkcjonalności, łatwy w obsłudze, duża ilość bibliotek |
| MySQL | Użyta do przetrzymywania dużych ilości danych w bazie danych, np. używana do przetrzymywania danych na temat użytkowników, trenerów, statystyk turniejowych. |

## Diagramy UML

### Diagram(-y) klas

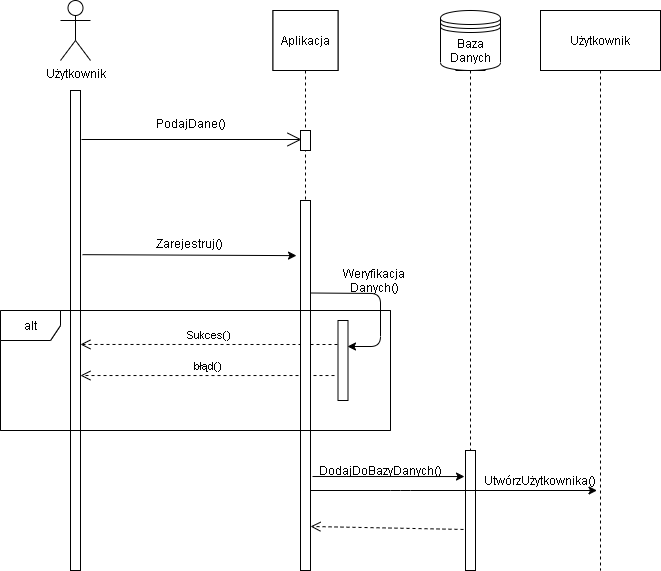


### Diagram(-y) czynności

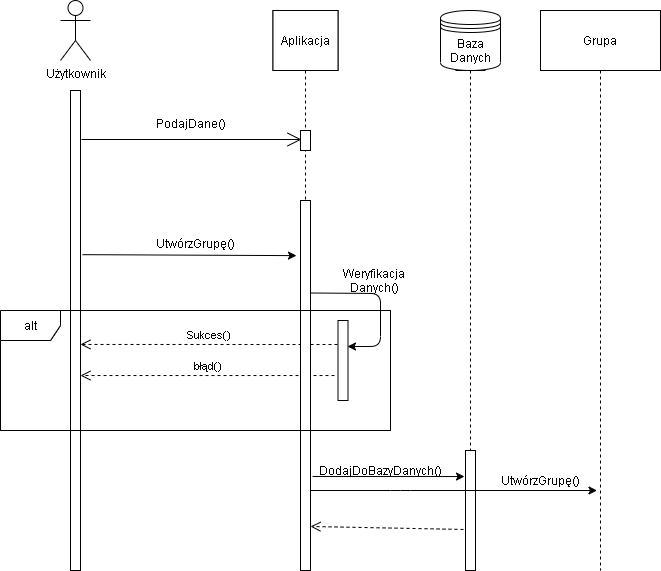


### Diagramy sekwencji

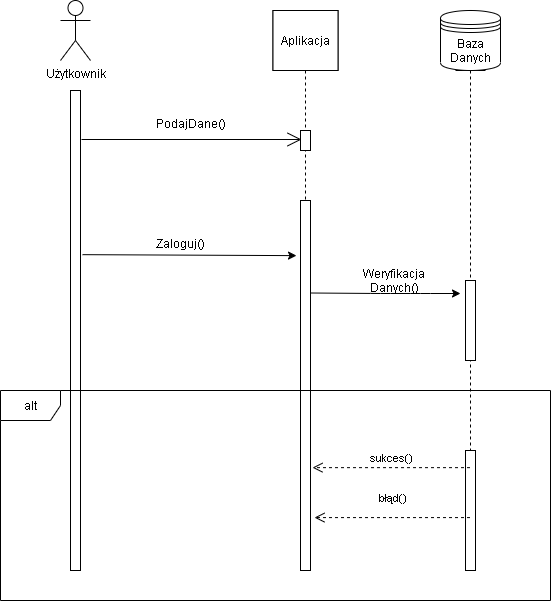
**Rejestracja**



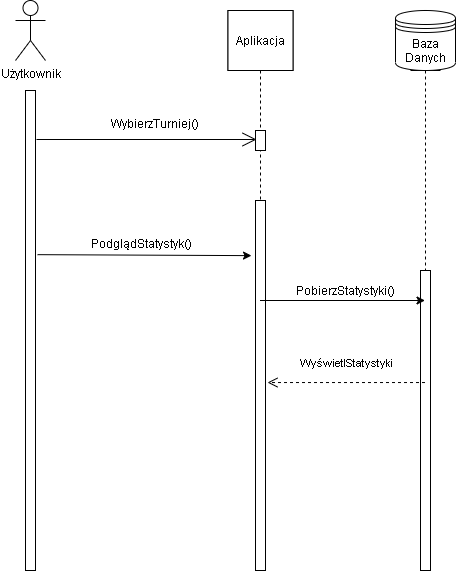
**Tworzenie Nowej Grupy (Dla trenera i jego graczy)**



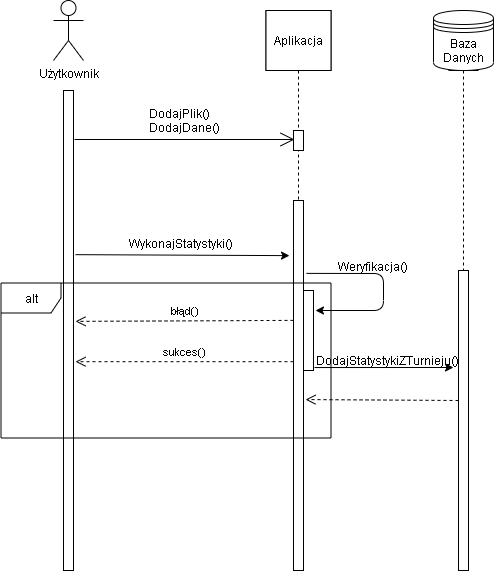
**Logowanie**



**Podgląd Statystyk (Podgląd stworzonych przez trenerów statystyk)**

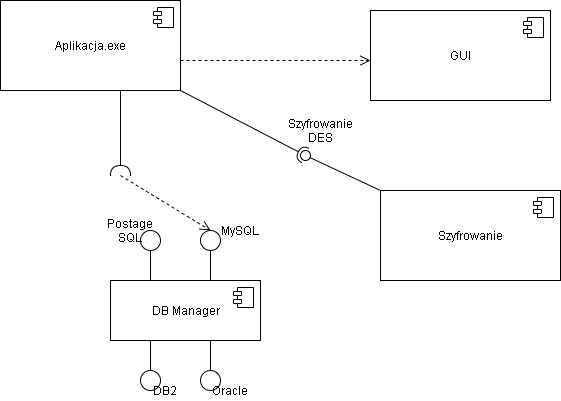


**Tworzenie Nowych Statystyk Dla Danego Turnieju**

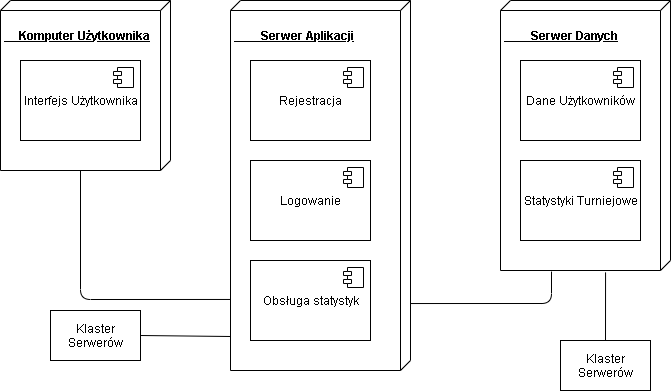


### Inne diagramy

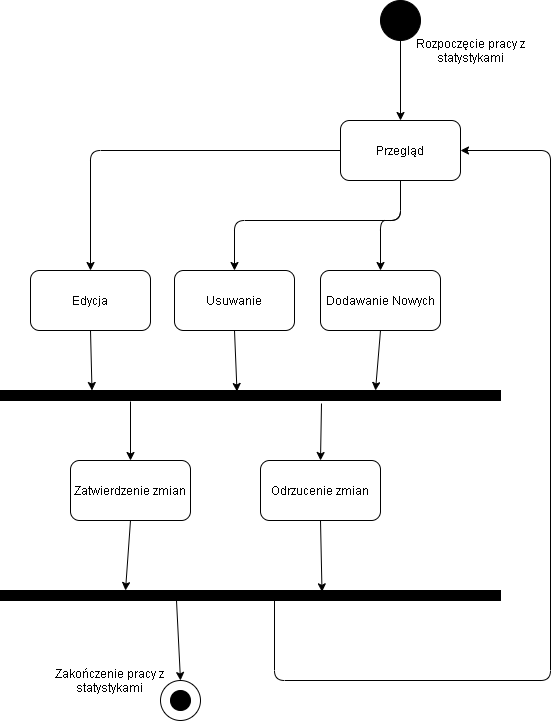
**Diagram Komponentów**



**Diagram Rozmieszczenia**

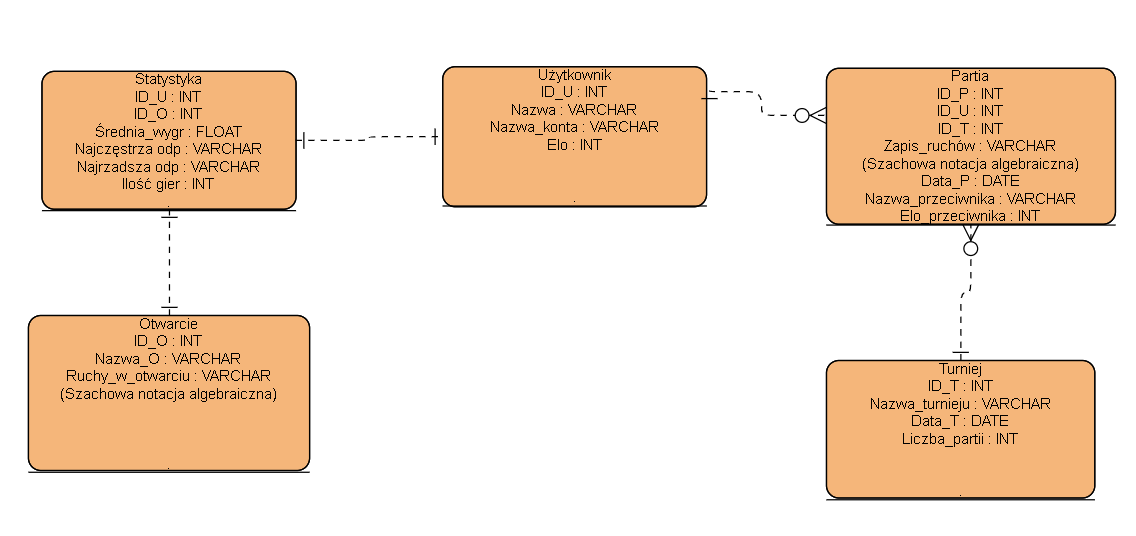


**Diagram maszyny stanowej do pracy z statystykami turniejowymi**



## Projekt bazy danych

### Schemat



### Projekty szczegółowe tabel

Użytkownik

|  |  |
| --- | --- |
| ID\_U | Identyfikator gracza |
| Nazwa | Dane gracza |
| Nazwa\_konta | Dane konta |
| Elo | Punkty rankingowe oficjalnego rankingu FIDE |

Statystyka

|  |  |
| --- | --- |
| ID\_U | Identyfikator gracza |
| ID\_O | Identyfikator danego otwarcia |
| Średnia\_wygr | Liczba wygranych gier przy danym otwarciu |
| Najczęstrza odp | Najczęstrza odpowiedz na dane otwarcie |
| Najrzadsza odp | Najrzadsza odpowiedz na dane otwarcie |
| Ilość gier | Ilość gier danym otwarciem w grach turniejowych |

Otwarcie

|  |  |
| --- | --- |
| ID\_O | Identyfikator otwarcia |
| Nazwa\_O | Nazwa otwarcia i jego wariant |
| Ruchy\_w\_otwarciu | Zapis otwarcia w szachowej notacji algebraicznej |

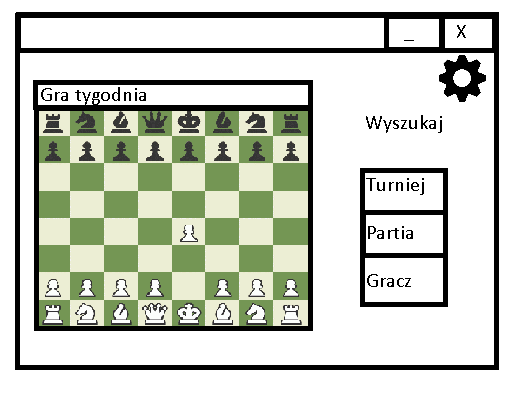
Partia

|  |  |
| --- | --- |
| ID\_P | Identyfikator partii |
| ID\_U | Identyfikator gracza |
| ID\_T | Identyfikator turnieju (NULL w przypadku gdy gra nie była rozgrywana w turnieju) |
| Zapis\_ruchów | Zapis partii w szachowej notacji algebraicznej |
| Data\_P | Data rozegrania partii |
| Nazwa\_przeciwnika | Dane przeciwnika |
| Elo\_przeciwnika | Punkty rankingowe oficjalnego rankingu FIDE przeciwnika |

Turniej

|  |  |
| --- | --- |
| ID\_T | Identyfikator danego turnieju |
| Nazwa\_turnieju | Nazwa turnieju |
| Data\_T | Data rozpoczęcia turnieju |
| Liczba\_partii | Liczba rozegranych gier w turnieju |

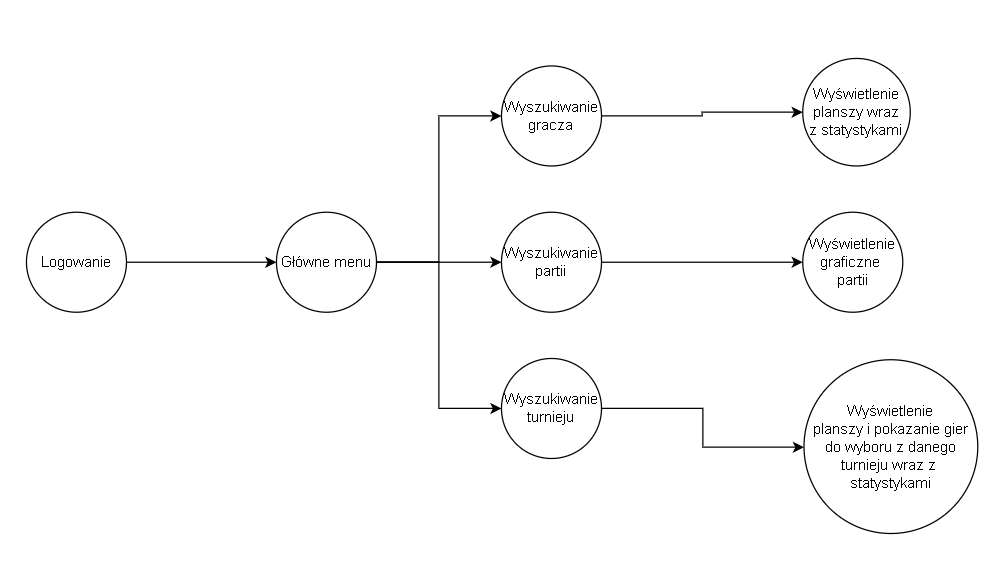
## Projekt interfejsu użytkownika



### Lista głównych elementów interfejsu

1. Główne menu
2. Ekran Logowania
3. Wyszukiwanie
4. Plansza z grą wraz z pokazanymi statystykami

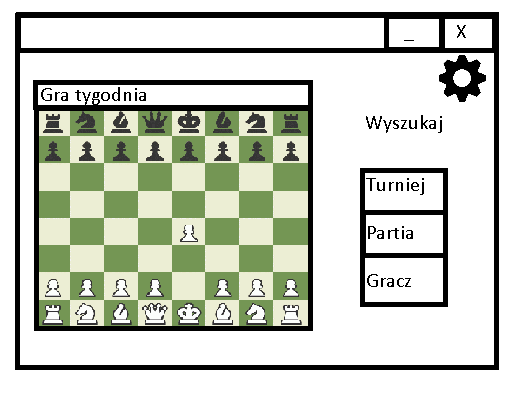
### Przejścia między głównymi elementami



### Projekty szczegółowe poszczególnych elementów

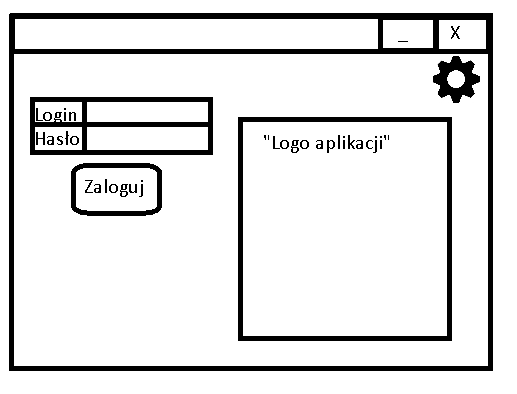
(1)

Główne menu



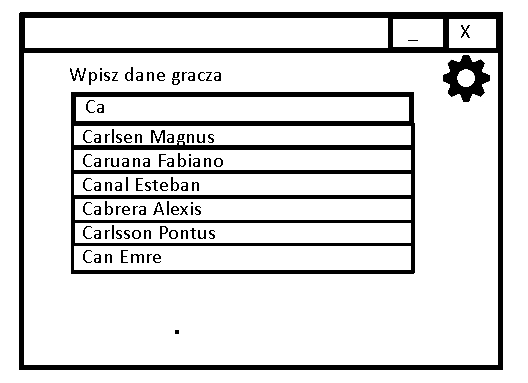
(2)

Logowanie



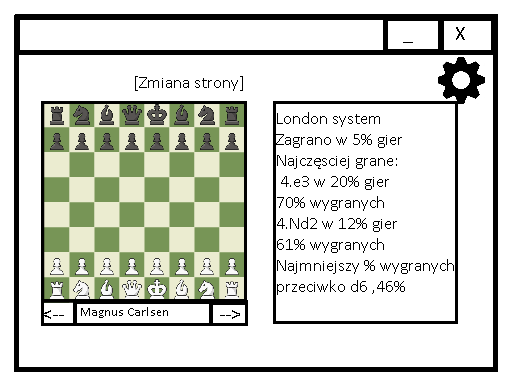
(3)

Wyszukiwanie



(4)

Plansza z grą wraz z pokazanymi statystykami



## Procedura wdrożenia

# Dokumentacja dla użytkownika

# Podsumowanie

## Szczegółowe nakłady projektowe członków zespołu

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – udział każdej osoby w skali całego projektu

|  |  |
| --- | --- |
| Patryk Walczyński | Mateusz Dziuba |
| 50% | 50% |
|  |  |

Nakład pracy taki sam jak przy harmonogramie.

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach