

Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat:	Oprogramowanie zarządzające turniejami szachowymi
Autorzy:	Oskar Wiszowaty, Wojciech Osiak
Grupa:	I1-222A
Kierunek:	informatyka
Rok akademicki:	2020/2021
Poziom i semestr:	I/4
Tryb studiów:	stacjonarne



1 Spis treści

2	Odnosniki do innych źródeł.....	3
3	Słownik pojęć	4
4	Wprowadzenie	5
4.1	Cel dokumentacji.....	5
4.2	Przeznaczenie dokumentacji	5
4.3	Opis organizacji lub analiza rynku.....	5
4.4	Analiza SWOT organizacji	6
5	Specyfikacja wymagań	7
5.1	Charakterystyka ogólna.....	7
5.3	Wymagania нефunkcjonalne.....	22
6	Zarządzanie projektem	23
6.1	Zasoby ludzkie	23
6.2	Harmonogram prac.....	23
6.3	Etapy/kamienie milowe projektu	23
7	Zarządzanie ryzykiem.....	24
7.1	Lista czynników ryzyka	24
7.3	Plan reakcji na ryzyko	24
8	Zarządzanie jakością.....	25
8.1	Scenariusze i przypadki testowe	25
9	Projekt techniczny	30
9.1	Opis architektury systemu.....	30
9.2	Technologie implementacji systemu.....	31
9.3	Diagramy UML	31
9.5	Projekt bazy danych	41
9.6	Projekt interfejsu użytkownika.....	42
9.7	Procedura wdrożenia	52
10	Dokumentacja dla użytkownika.....	53
11	Podsumowanie	54
11.1	Szczegółowe nakłady projektowe członków zespołu	54
12	Inne informacje	55

2 Odnośniki do innych źródeł

- Dokumentacja projektu – https://github.com/osuchtk/io_praca_zaliczeniowa.git
- Fragment aplikacji – <https://github.com/wishuuu/Skoczek-g6>

3 Słownik pojęć

- Organizator – założyciel turnieju
- Użytkownik – każda osoba posiadająca konto w aplikacji
- Gracz – osoba uczestnicząca w turnieju
- Trener – osoba wspierająca gracza w przygotowaniu do turnieju
- System – tworzona aplikacja na telefony z systemem Android/ iOS
- Token – indywidualny numer pozwalający na weryfikację użytkownika
- Widget – funkcjonalny element aplikacji

4 Wprowadzenie

4.1 Cel dokumentacji

Celem dokumentacji jest przedstawianie projektu oraz jego wymagań i sposobu wdrażania na rynek. Zawiera szczegółowe informacje na temat sposobu działania oraz jego zastosowania praktycznego (informacje dla kogo oraz w jakim celu). W dokumentacji są również zawarte informacje o wymaganiach projektu w zakresie zasobów ludzkich i infrastruktury oraz szczegółowy projekt techniczny.

4.2 Przeznaczenie dokumentacji

Dokumentacja jest przeznaczona dla osób odpowiedzialnych za produkcję i wdrożenie projektu oraz za kontrolę jakości wykonania, przejrzystość pracy i czasowość oddawania poszczególnych etapów pracy, a także dla organizacji zlecającej utworzenie produktu.

4.3 Opis organizacji lub analiza rynku

Aplikacja jest stworzona na potrzeby organizacji turniejów szachowych. Została stworzona na wniosek firmy organizującej takie turnieje w województwie zachodniopomorskim. Jeśli zostanie pozytywnie zaopiniowana będzie miała szansę stać się oficjalną aplikacją wspomagającą krajowe turnieje szachowe na terytorium Polski.

Aby móc korzystać z systemu potrzebne będzie pobranie oprogramowania oraz zgłoszenie turnieju do odpowiednich jednostek organizacyjnych.

Na początku istnienia aplikacji prowadzone będą akcje promocyjne organizowane przez wydawców, czyli organizacja własnych turniejów w wykorzystaniem oprogramowania. Będzie to miało na celu przekonanie inwestorów do systemu oraz będzie szansą na pozytywną opinię wojewódzkiego związku szachowego.

4.4 Analiza SWOT organizacji

Strenghts <ul style="list-style-type: none">• Przejrzystość projektu• Łatwość wdrażania• Łatwość zarządzania	Weaknesses <ul style="list-style-type: none">• Krótki czas na wdrożenie• Konieczność szkoleń pracowników• Konieczność szkoleń kadry zarządzającej projektem
Opportunities <ul style="list-style-type: none">• Szansa na promocję szachów• Podniesienie poziomu rozgrywek turniejowych	Threats <ul style="list-style-type: none">• Konkurencyjne rozwiązania• Użytkownik końcowy nie będzie umiał posługiwać się produktem

5 Specyfikacja wymagań

5.1 Charakterystyka ogólna

5.1.1.1 Definicja produktu

System jest aplikacją służącą do dobierania graczy w turniejach szachowych i śledzenia historii odbytych partii. Przyjęta nazwa robocza aplikacji to Skoczek g6.

5.1.2 Podstawowe założenia

System będzie pozwalał na automatyczne generowanie partii szachowych przy organizacji turniejów. Dobieranie graczy do partii oparte będzie na systemie rankingowym FIDE, a także na dotychczasowych wynikach ich partii w rozgrywanym aktualnie turnieju. Przyspieszy to przygotowanie par turniejowych, a więc ułatwi organizację turnieju.

System będzie przydzielał graczy rozgrywających partie do konkretnych stanowisk w określonych turach rozgrywek.

System będzie też umożliwiał graczom przejrzanie historii partii rozegranych przez dobranych im oponentów. Pozwala to graczom na wcześniejsze przygotowanie się do partii.

5.1.3 Cel biznesowy

- Ułatwienie organizacji turniejów szachowych
- Podniesienie poziomu rozgrywek
- Promocja szachów jako sportu
- Rozwijanie umiejętności szachistów
- Wykształcenie zdolności myślenia strategicznego wśród użytkowników
- Podniesienie umiejętności zespołu tworzącego projekt

5.1.4 Użytkownicy

1. Organizator
2. Gracz
3. Trener (osoba wspierająca gracza)

5.1.5 Korzyści z systemu

1. Organizator
 - a. Ułatwienie i przyspieszenie procesu organizacji turnieju szachowego
 - b. Ułatwienie nadzoru przebiegu turnieju szachowego.
2. Gracz
 - a. Możliwość przygotowania się do partii z konkretnym przeciwnikiem
 - b. Łatwiejszy przepływ informacji na temat harmonogramu rozgrywanych partii.
3. Trener
 - a. Ułatwienie doboru otwarć i taktyk wartych przestudiowania z podopiecznym.

5.1.6 Ograniczenia projektowe i wdrożeniowe

Przepisy prawne	Zgodność z FIDE Zgodność z regulaminem i polityką prywatności sklepów z aplikacjami
Technologia	Dart+Flutter
Baza danych	Firebase
Powiązania z innymi aplikacjami	Konto Google
Docelowa platforma sprzętowa	Urządzenia mobilne z systemami Android i iOS

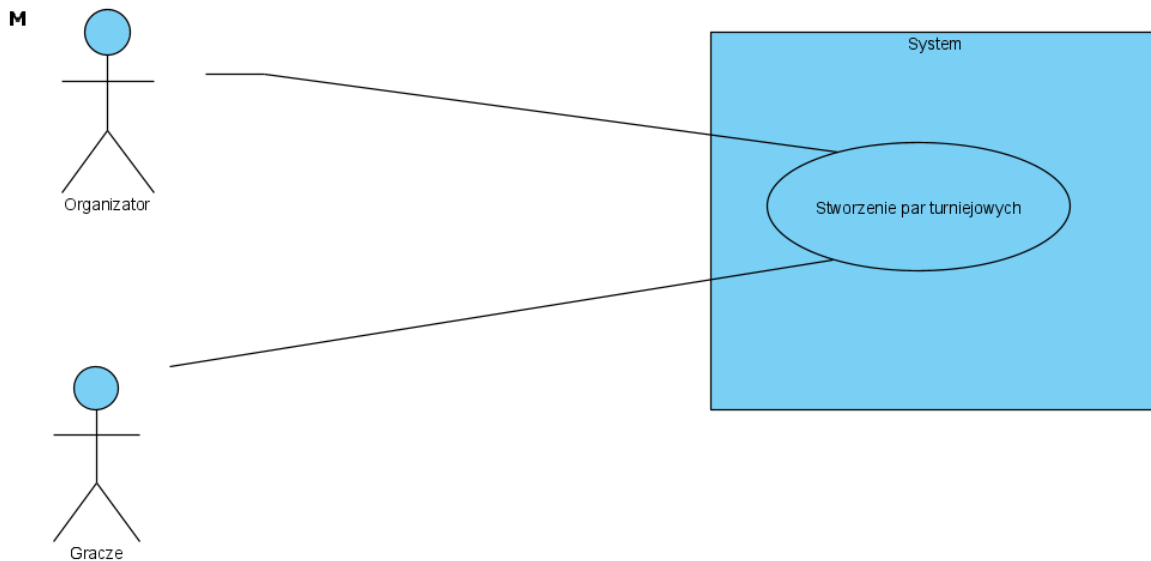
- Zgodność z FIDE – notacja używana do zapisu przebiegu partii w bazie danych powinna być zgodna z oficjalnymi zasadami FIDE. Dodatkowo system powinien śledzić przebieg każdej partii pod kątem nielegalnych posunięć zgodnie z zasadami szachowymi zawartymi w FIDE.
- Zgodność z polityką prywatności sklepów z aplikacjami – wydanie aplikacji poprzez sklep z aplikacjami (np. Google Play) wymaga spełnienia regulaminu i polityki prywatności sklepu.
- Technologia – Zastosowanie języka programowania Dart i pakietu UI Flutter pozwala na łatwe przygotowanie GUI i funkcjonalności aplikacji na systemy Android i iOS bazując na jednej bazie kodu.
- Baza danych – aplikacje Flutter pozwalają na łatwe połączenie z bazami danych Firebase, za pośrednictwem wbudowanego API.
- Powiązania z innymi aplikacjami – baza danych Firebase pozwala na łatwe uwierzytelnianie użytkowników aplikacji za pośrednictwem ich konta Google.
- Docelowa platforma sprzętowa – ponieważ w dzisiejszych czasach każdy nosi przy sobie telefon, aplikacja mobilna pozwala na łatwy dostęp do danych turniejowych nawet w trakcie turnieju.

5.1.7 Lista wymagań

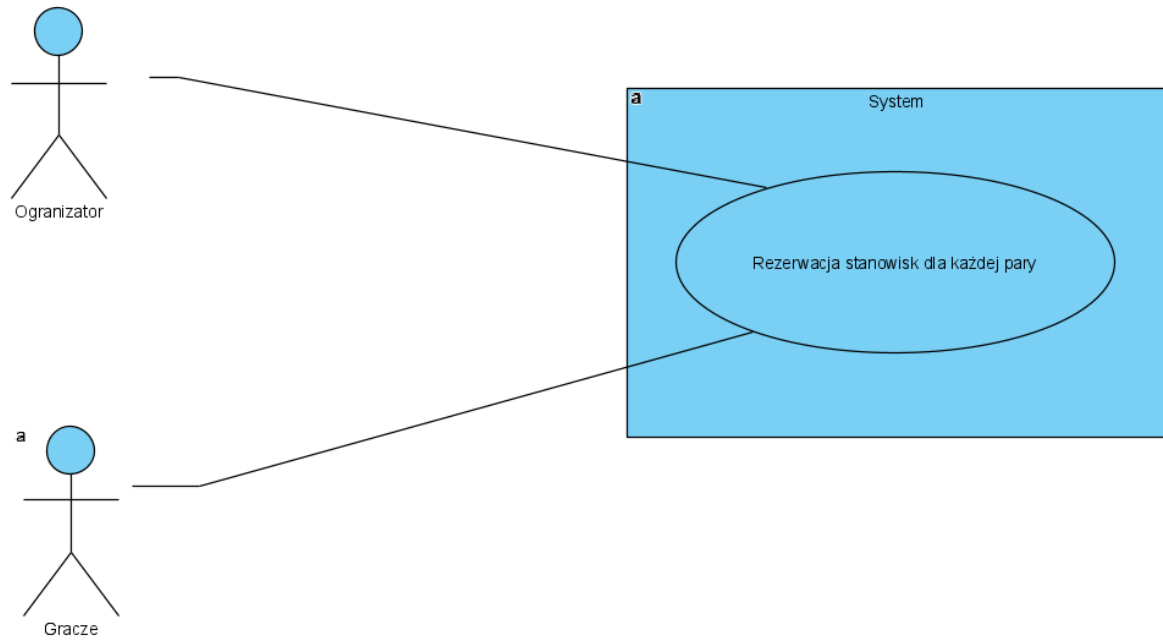
1. Użycie przez organizatora do stworzenia par turniejowych między graczami
2. Użycie przez organizatora do zarezerwowania konkretnych stanowisk dla par
3. Użycie przez gracza do sprawdzenia przeciwnika w kolejnej partii
4. Użycie przez gracza do sprawdzenia przydzielonego stolika przy którym będzie rozgrywana kolejna partia
5. Użycie przez gracza podczas treningu do partii
6. Użycie przez trenera wraz z podopiecznym (Graczem) podczas ostatnich przygotowań do partii

5.1.8 Diagramy przypadków użycia

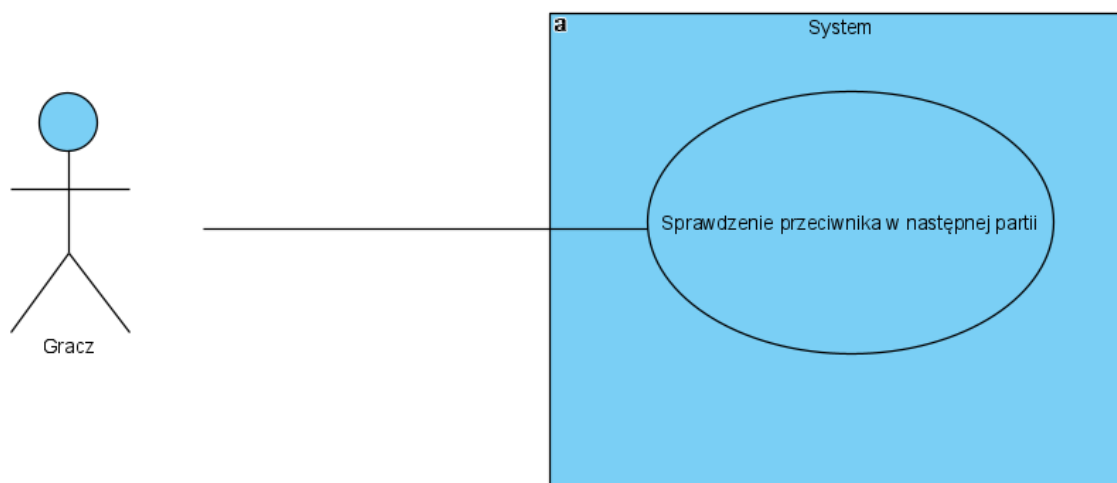
Przypadek 1: Organizator tworzy pary turniejowe



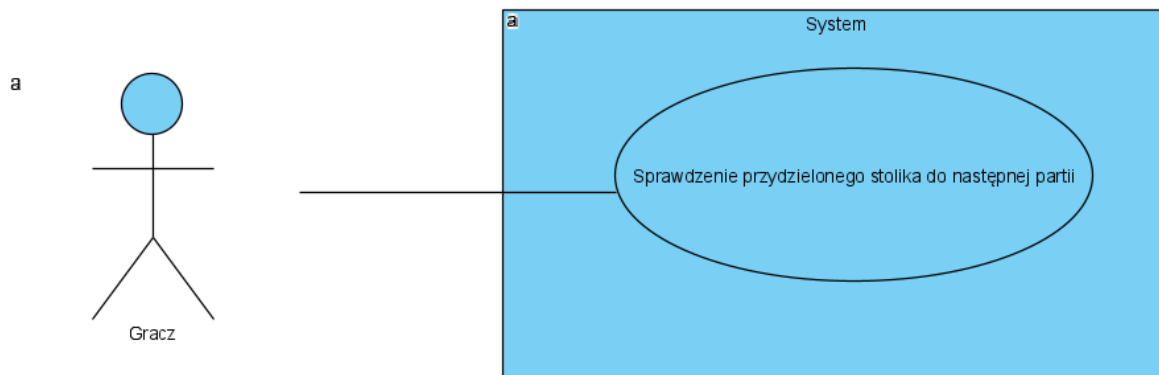
Przypadek 2: Organizator rezerwuje uczestnikom stanowiska



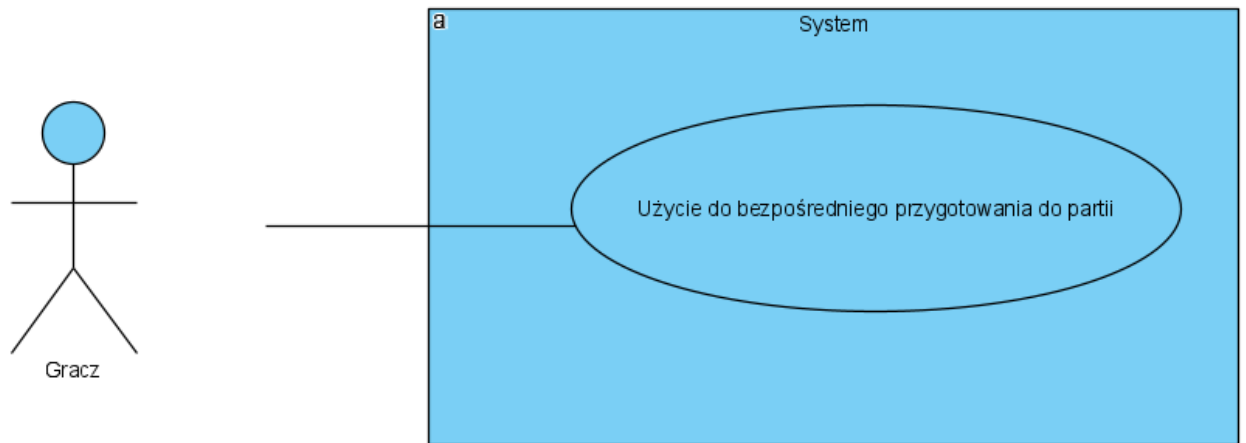
Przypadek 3: Gracz sprawdza dostępne informacje na temat swojego następnego przeciwnika



Przypadek 4: Gracz sprawdza w systemie przydzielone mu stanowisko przy rozgrywaniu następnej partii



Przypadek 5: Gracz trenuje przez partię z danym przeciwnikiem



Przypadek 6: Trener oraz podopieczny (gracz) używają aplikacji bezpośrednio przed partią w celu powtórzenia najważniejszych taktyk, które wypracowali

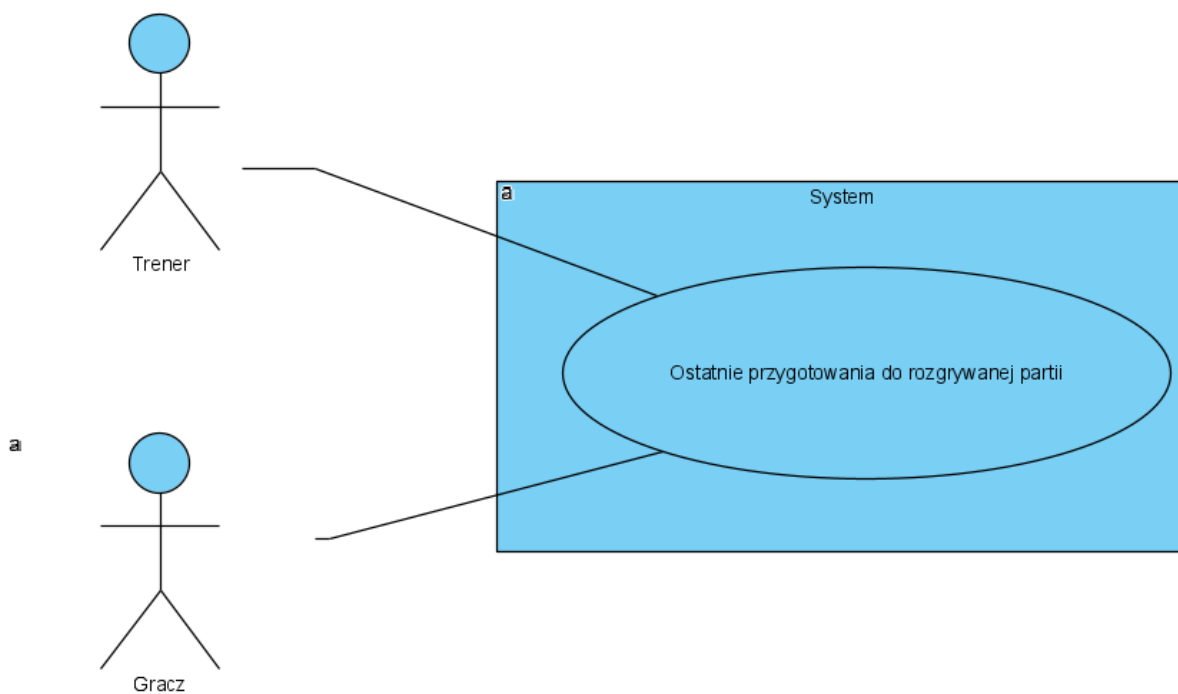
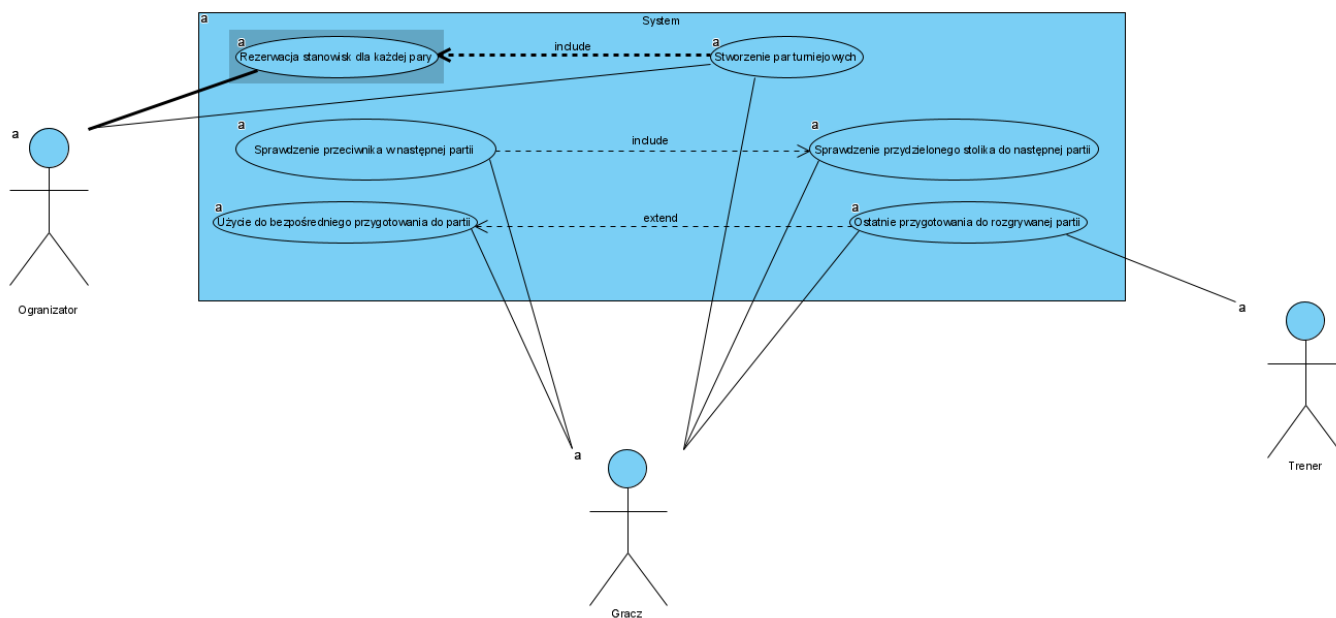


Diagram kontekstowy. (generalizacja systemu względem poszczególnych platform, logowanie do aplikacji)



5.1.9 Szczegółowy opis wymagań

- Numer – 1
- Organizator tworzy pary turniejowe
- ID: 1; Ułatwienie i przyspieszenie procesu organizacji turnieju szachowego, a także nadzoru jego przebiegu
- Organizator, Gracze
- Organizator przypisuje uczestnikom przeciwników:
 - Warunki początkowe: identyfikatory graczy biorących udział w turnieju
 - **Przebieg działań:**
 - .1. Dobranie par w pierwszej turze rozgrywek na podstawie rankingu FIDE lub innego systemu rankingowego
 - .2. Dobranie par w kolejnych turach na podstawie dotychczasowej punktacji
 - Efekty: Przyspieszony proces tworzenia par zmagających się ze sobą zawodników
 - Wymagania нефunkcjonalne: system pozwala na utworzenie par dla turniejów w których udział bierze do 256 uczestników
 - Częstotliwość: 5
 - Istotność: 5

- Numer – 2
- Organizator rezerwuje uczestnikom stanowiska
- ID: 1; Ułatwienie i przyspieszenie procesu organizacji turnieju szachowego, a także nadzoru jego przebiegu
- Organizator, Gracze
- Organizator tworzy listę zajętych stolików turniejowych
 - Warunki początkowe: identyfikatory stolików, spis rozgrywanych partii w danej turze
 - **Przebieg działań:**
 - .1. Przydzielenie stolików do rozgrywanych partii na podstawie wagi partii (do słownika)
 - .2. System przekazuje informacje o dobranych stolikach graczom
 - .3. Aplikacja zwraca listę zajętych stolików oraz przypisanych do nich graczy
 - Efekty: łatwiejsza kontrola nad przebiegiem turnieju, kontrola nad tym czy zawodnicy zajęli przypisane im miejsca przy stolikach
 - Wymagania нефunkcjonalne: system pozwala na przypisanie stolików dla turniejów w których udział bierze do 256 uczestników
 - Częstotliwość: 5
 - Istotność: 5

- Numer – 3
- Gracz sprawdza dostępne informacje na temat swojego następnego przeciwnika
- ID: 2; Możliwość przygotowania się do partii z konkretnym przeciwnikiem oraz łatwiejszy przepływ informacji na temat harmonogramu rozgrywanych partii
- Gracz
- Gracz pozyskuje informacje:
 - Warunki początkowe: Gracz zna swojego następnego przeciwnika, każdy z graczy ma swoją historię gier w systemie
 - **Przebieg działań:**
 - .1. Gracz znajduje swojego przeciwnika w systemie
 - .2. Gracz poszukuje interesujących go informacji na temat rywala
 - Efekty: gracz zna historię gier swojego rywala oraz jego statystyki przez co może przygotować się do rozgrywki
 - Wymagania niefunkcjonalne: uczestnik turnieju może zapoznać się z profilem i statystykami każdego innego zarejestrowanego gracza
 - Częstotliwość: 5
 - Istotność: 4

- Numer – 4
- Gracz sprawdza w systemie przydzielone mu stanowisko przy rozgrywaniu następnej partii
- ID: 2; Możliwość przygotowania się do partii z konkretnym przeciwnikiem oraz łatwiejszy przepływ informacji na temat harmonogramu rozgrywanych partii
- Gracz
- Gracz poszukuje w systemie informacji o swojej następnej partii:
 - Warunki początkowe: gracz, jego przeciwnik oraz turniej są zarejestrowane w systemie jako wydarzenie, które aktualnie się odbywa
 - **Przebieg działań:**
 - .1. Gracz znajduje w systemie swoją partię
 - .2. W szczegółowych informacjach o rozgrywce gracz znajduje informacje o miejscu i numerze zarezerwowanego stolika
 - Efekty: gracz wie w którym miejscu w strefie rozgrywek ma poszukiwać przydzielonego mu stolika
 - Częstotliwość: 5
 - Istotność: 3

- Numer – 5
- Gracz trenuje przez partię z danym przeciwnikiem
- ID: 2; Możliwość przygotowania się do partii z konkretnym przeciwnikiem
- Gracz
- Gracz wykonuje trening:
 - Warunki początkowe: gracz ma zaplanowaną partię z konkretnym przeciwnikiem, który posiada zapisaną w systemie historię partii
 - **Przebieg działań:**
 - .1. Gracz odnajduje swojego rywala w systemie
 - .2. Gracz może przejrzeć interesujące go statystyki przeciwnika
 - Efekty: gracz jest lepiej przygotowany do partii z przeciwnikiem
 - Częstotliwość: 5
 - Istotność: 3

- Numer – 6
- Trener oraz podopieczny (gracz) używają aplikacji bezpośrednio przed partią w celu powtórzenia najważniejszych taktyk, które wypracowali
- ID: 3; Ułatwienie doboru otwarć i taktyk wartych przestudiowania z podopiecznym
- Gracz, Trener
- Gracz utrwała sobie przydatne ruchy:
 - Warunki początkowe:
 - **Przebieg działań:**
 1. Gracz tworzy statystyki najczęstszych ruchów przeciwnika
 2. Gracz ma okazję na wypracowanie odpowiedzi na ruchy przeciwnika
 - Efekty: gracz ma okazję do szczegółowego przygotowania się do partii
 - Wymagania нефункционалне: gracz może zaprosić do analizy partii dowolnego zarejestrowanego użytkownika
 - Częstość: 5
 - Istotność: 3
- Trener przekazuje wskazówki zawodnikowi:
 - Warunki początkowe: gracz i trener wspólnie ustalają przydatne zagrania w starciu w rywalu
 - **Przebieg działań:**
 - .1. Trener ustala z graczem listę przydatnych zagrań
 - .2. Lista ta posłuży do wypracowania zagrań, które mają za zadanie pomóc graczowi wygrać partię
 - .3. Po przeprowadzeniu takiego przygotowania gracz jest lepiej przygotowany do partii
 - Efekty: większe szanse na rozegranie całej partii pod większą kontrolą sytuacji gracza
 - Wymagania нефункционалне: trener może dołączyć do analizy partii dowolnego zarejestrowanego użytkownika
 - Częstość: 3
 - Istotność: 2

5.2 Wymagania niefunkcjonalne

1. Aplikacja wydana na systemy Android oraz iOS jest zoptymalizowana pod kątem wielu urządzeń mobilnych. Sam system nie potrzebuje bardzo wysokich wymagań sprzętowych, jest w stanie działać płynnie na praktycznie każdym telefonie pracującym pod kontrolą podanego wyżej systemu operacyjnego.
2. Zaszyfrowane dane użytkowników są przechowywane na wewnętrznym zabezpieczonym serwerze. Dodatkowo, istnieje kopia zapasowa wszystkich plików na serwerze zewnętrznym w razie awarii bądź uszkodzenia czy ataku wymierzonego w naszą firmę.
3. Pliki użytkowników są zaszyfrowane, co uniemożliwia pozyskanie z nich cennych danych osobowych osobom, które nie są do tego upoważnione. Dodatkowo, każda osoba posiada własne konto, które jest zabezpieczone hasłem. Oprócz tego, można użyć opcji weryfikacji dwuetapowej przy logowaniu: token będzie wysłany jako SMS lub na podany przy rejestracji adres e-mail.
4. Inne cechy jakości:
 - 4.1. Adaptowalność: technologia Flutter+Dart pozwala na utworzenia aplikacji z tzw. adaptowalnych Widgetów, które automatycznie dostosowują swoją wielkość, aby aplikacja działała poprawnie i była przejrzysta niezależnie od parametrów technicznych urządzenia.
 - 4.2. Dostępność: udostępniony w systemowych sklepach z aplikacjami produkt dostępny jest przez całą dobę, dzięki zastosowaniu bazy danych Firebase pozwalającej na zapis do 20 tys. i odczyt do 50 tys. kolekcji danych dziennie w ramach darmowego planu, pozwalać będzie na zapisanie do 700 tys. partii, i odtworzenie ich w ramach systemu 1.5 mln. razy rocznie, baza danych Firebase nie posiada limitu indywidualnych tokenów uwierzytelniania użytkowników, co pozwala na nielimitowaną liczbę indywidualnych kont użytkownika w ramach systemu
 - 4.3. Elastyczność: użycie układu interfejsu i kolorów, które zostały przetestowane pod kątem skupiania uwagi użytkownika i przejrzystości
 - 4.4. Łatwość konserwacji: oparcie projektu na technologii Flutter+Dart pozwala na generowanie aplikacji na systemy Android i iOS z jednego kodu źródłowego, co ułatwia przyszłą konserwację.
 - 4.5. Przenośność: system oparty o jedną bazę danych Firebase zarówno dla użytkowników systemu Android jak i iOS, co pozwala na korzystanie ze swojego konta z różnych urządzeń
 - 4.6. Awaryjność: jedyny awaryjny czynnik zewnętrzny to baza danych Firebase, która jest usługą bardzo stabilną
 - 4.7. Testowalność: emulatory systemu Android i iOS pozwalają na przetestowanie działania aplikacji na wielu urządzeniach o różnych parametrach technicznych z poziomu komputera programisty

6 Zarządzanie projektem

6.1 Zasoby ludzkie

Implementacja projektu w podstawowych założeniach nie jest procesem skomplikowanym, ponieważ sama budowa i działanie systemu nie jest problematyczna. Do realizacji, wdrożenia i monitorowania działania projektu nie wymaga się obecności osób innych niż autorów projektu (Wiszwaty i Osiak). Cały system nie jest projektem, który wymaga dużych nakładów pracy ludzkiej przy jego utrzymaniu i konserwacji. Wdrażanie nowych osób, które miałyby być odpowiedzialne za nadzór nad systemem byłoby bardziej kosztowne i czasochłonne, niż ewentualne wynikające z tego zyski. Ewentualne zatrudnienie specjalistów w razie krytycznych problemów z zarządzaniem i rozwojem systemu.

Czas przewidziany na zrealizowanie projektu wynosi 24 tygodnie. Daje to łącznie 1920 roboczogodzin pracy nad systemem (każda osoba po 960 roboczogodzin). Po uwzględnieniu podziału na poszczególne etapy otrzymujemy:

- 4 tygodnie na testowanie produktu
- 5 tygodni na wdrożenie i szkolenie
- 15 tygodni na implementację

6.2 Harmonogram prac

Zadanie	Data rozpoczęcia	Czas trwania	Data zakończenia	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Projektowanie GUI	10.05.2021	1	17.05.2021																								
Implementacja GUI	17.05.2021	1	24.05.2021																								
Podłączenie bazy danych	24.05.2021	1	31.05.2021																								
Generowanie turniejów i możliwość dołączania do nich	24.05.2021	2	07.06.2021																								
Możliwość zapisu przebiegu partii w bazie danych	07.06.2021	1	14.06.2021																								
Przeglądanie przebiegów partii z poziomu aplikacji	14.06.2021	3	05.07.2021																								
Implementacja harmonogramu partii użytkownika	05.07.2021	2	19.07.2021																								
Testowanie i poprawa błędów	19.07.2021	2	02.08.2021																								
Projekt w fazie 0.1																											
Automatyczne dobieranie stanowisk każdej partii w turnieju	02.08.2021	2	16.08.2021																								
Implementacja funkcji wspólnej analizy partii	16.08.2021	3	06.09.2021																								
Testowanie i poprawa błędów	06.09.2021	2	20.09.2021																								
Projekt w fazie 1.0 - gotowy do wdrożenia																											
Szkolenie z klientem	20.09.2021	1	27.09.2021																								
Serwisowanie platformy	27.09.2021	4	25.10.2021																								

6.3 Etapy/kamienie milowe projektu

1. Założenie celu do realizacji w trakcie trwania projektu.
2. Stworzenie dokumentacji projektowej.
3. Stworzenie działającego prototypu systemu.
4. Rozwój prototypu i wdrażanie kolejnych funkcjonalności.
5. Zakończenie prac nad zakładanymi funkcjonalnościami i wejście w fazę testowania produktu.
6. Wdrożenie aplikacji na rynek.

7 Zarządzanie ryzykiem

7.1 Lista czynników ryzyka

1. Chwilowa niedostępność usługi bazy danych Firebase
2. Przekroczenie założonego przez zleceniodawcę budżetu
3. Nietrzymanie zaplanowanego terminu pracy
4. Część funkcjonalności jest niewykonalna technicznie
5. Brak sprecyzowanych wymagań ze strony zleceniodawcy projektu

7.2 Ocena ryzyka i plan reakcji na ryzyko

Lp.	Plan reakcji	P	W
1	Kontakt z obsługą techniczną usługi Firebase, uruchomienie kopii zapasowych w przypadku długiego czasu niedostępności usługi Firebase	~0%	5
2	Poszukiwanie tańszych rozwiązań lub negocjacje odnośnie zwiększenia budżetu	25%	4
3	Konsultacja ze specjalistą do spraw zarządzania projektem w celu optymalizacji terminowego wykonania prac	40%	3
4	Ograniczenie zaplanowanych funkcjonalności systemu	20%	z
5	Konsultacje ze zleceniodawcą aby wyeliminować ewentualne błędy powstałe przy realizacji projektu	15%	4

Lp. – liczba porządkowa

P – prawdopodobieństwo

W – wpływ

z – zależny od funkcjonalności, której dotyczy ryzyko

8 Zarządzanie jakością

8.1 Scenariusze i przypadki testowe

ID	1
Nazwa scenariusza	Generowanie testowych turniejów
Kategoria	Poprawność działania
Opis	Weryfikacja poprawności generowanych turniejów z podanych danych
Tester	Wisżowaty
Termin	05.06.2021
Narzędzia wspomagające	Emulator systemu mobilnego

Przebieg działań

Lp.	Działania testera	Działania systemu
1	Utworzenie turnieju	Prośba o zaproszenie graczy
2	Dołączenie do turnieju	Oczekiwanie na rozpoczęcie turnieju
3	Rozpoczęcie turnieju	Generowanie par turniejowych

Zestaw danych testowych

Dane użyte przez testera	Dane zwrócone przez system	Warunek zaliczenia testu
Tester nie używa żadnych danych	Wygenerowany turniej i pary	Poprawnie wygenerowany turniej

Czas na testy (w tygodniach)	2
-------------------------------------	---

ID	2
Nazwa scenariusza	Zapis partii do bazy danych
Kategoria	Poprawność działania
Opis	Testowanie poprawnego działania zapisu przebiegu partii do bazy danych
Tester	Osiak
Termin	14.06.2021
Narzędzia wspomagające	Debugger Android, panel administracyjny Firebase

Przebieg działań

Lp.	Działania testera	Działania systemu
1	Wprowadzenie partii z poziomu aplikacji	Zapis partii do bazy danych, wyświetlenie informacji o powodzeniu lub niepowodzeniu
2	Sprawdzenie czy partia została prawidłowo zapisana	Wyświetlenie tabelki

Zestaw danych testowych

Dane użyte przez testera	Dane zwrócone przez system	Warunek zaliczenia testu
Przebieg przykładowej partii	Powodzenie lub niepowodzenie zapisu	Zapis przebiegu partii w bazie danych przebiegł bez zakłóceń

Czas na testy (w tygodniach)	1
-------------------------------------	---

ID	3
Nazwa scenariusza	Próba zapisu nieprawidłowej partii
Kategoria	Poprawność działania
Opis	Testowanie próby zapisu niepoprawnej partii do bazy danych
Tester	Wisżowaty
Termin	14.06.2021
Narzędzia wspomagające	Debugger Android, panel administracyjny Firebase

Przebieg działań

Lp.	Działania testera	Działania systemu
1	Wprowadzenie niepoprawnej partii z poziomu aplikacji	Wyświetlenie błędu zapisu
2	Sprawdzenie czy partia została zapisana	Wyświetlenie tabelki

Zestaw danych testowych

Dane użyte przez testera	Dane zwrócone przez system	Warunek zaliczenia testu
Przebieg partii zawierającej nielegalne posunięcia	Powodzenie lub niepowodzenie zapisu	Partia nie została zapisana w bazie danych

Czas na testy (w tygodniach)	1
-------------------------------------	---

ID	4
Nazwa scenariusza	Przegląd przebiegu partii
Kategoria	Poprawność działania
Opis	Funkcjonalność umożliwiającą przeglądanie partii, które się odbyły, z poziomu aplikacji
Tester	Osiak
Termin	05.07.2021
Narzędzia wspomagające	Emulator Android

Przebieg działań

Lp.	Działania testera	Działania systemu
1	Próba przeglądu przebiegu partii	Wyświetlenie przebiegu partii

Zestaw danych testowych

Dane użyte przez testera	Dane zwrócone przez system	Warunek zaliczenia testu
brak	Wyświetlenie przebiegu partii i ruchów	Poprawne wyświetlenie przebiegu partii

Czas na testy (w tygodniach)	3
-------------------------------------	---

ID	5
Nazwa scenariusza	Testowanie wspólnej analizy partii
Kategoria	Poprawność działania
Opis	Przeprowadzanie testów wspólnej analizy partii
Tester	Wiszowaty
Termin	12.07.2021
Narzędzia wspomagające	Debugger Android

Przebieg działań

Lp.	Działania testera	Działania systemu
1	Rozpoczęcie przeglądania przebiegu partii, włączenie opcji wspólnej analizy	Utworzenie wspólnej sesji, zwrócenie kodu zaproszenia
2	Dołączenie do sesji za pośrednictwem kodu z innego konta	Dołączenie nowego użytkownika do sesji
3	Wykonanie dowolnego ruchu	Synchronizacja ruchu na obu urządzeniach

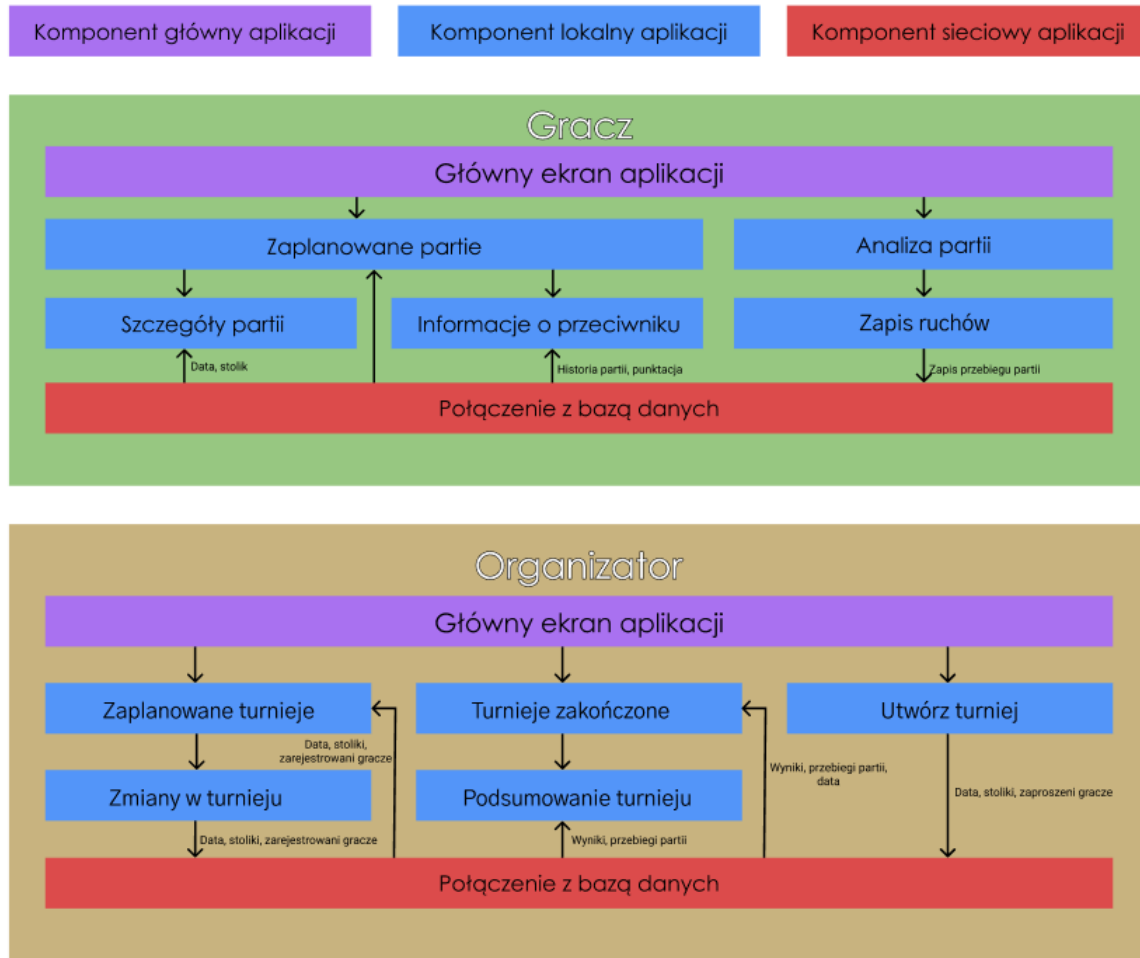
Zestaw danych testowych

Dane użyte przez testera	Dane zwrócone przez system	Warunek zaliczenia testu
brak	Kod zaproszenia, potwierdzenie utworzenia sesji	Poprawnie zsynchronizowana sesja analizy partii

Czas na testy (w tygodniach)	3
-------------------------------------	---

9 Projekt techniczny

9.1 Opis architektury systemu

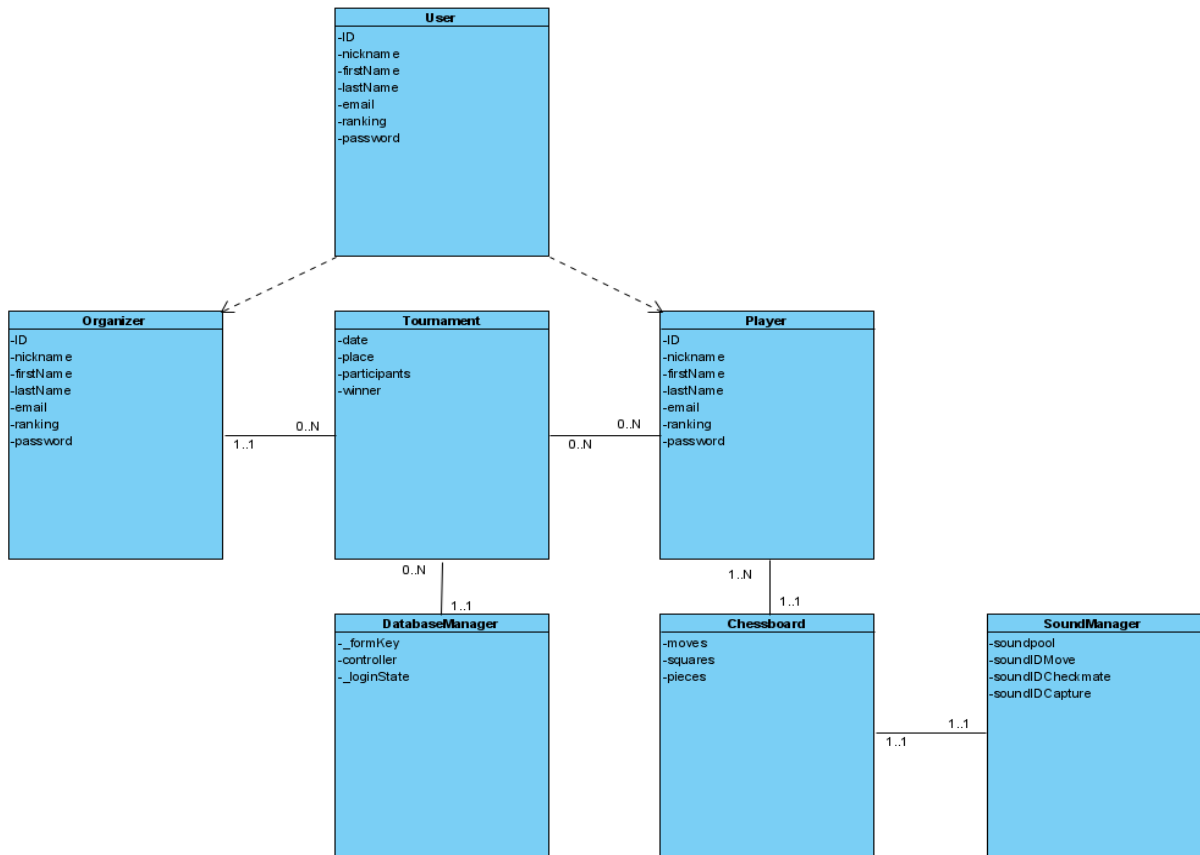


9.2 Technologie implementacji systemu

Technologia	Uzasadnienie
Dart + Flutter	Zdecydowanie ułatwienie pracy w stosunku do innych narzędzi oferujących podobną funkcjonalność
Firebase	Wysoka integralność usługi Firebase z aplikacjami napisanymi w technologii Flutter

9.3 Diagramy UML

9.3.1 Diagram(-y) klas



9.3.2 Diagramy czynności

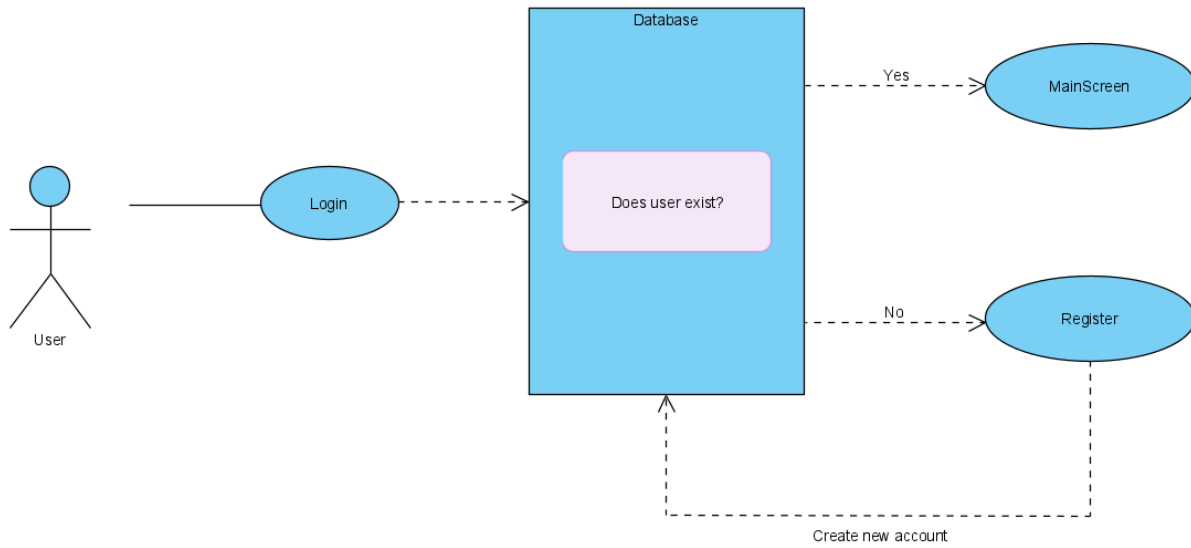


Diagram przedstawiający logowanie użytkownika do systemu.

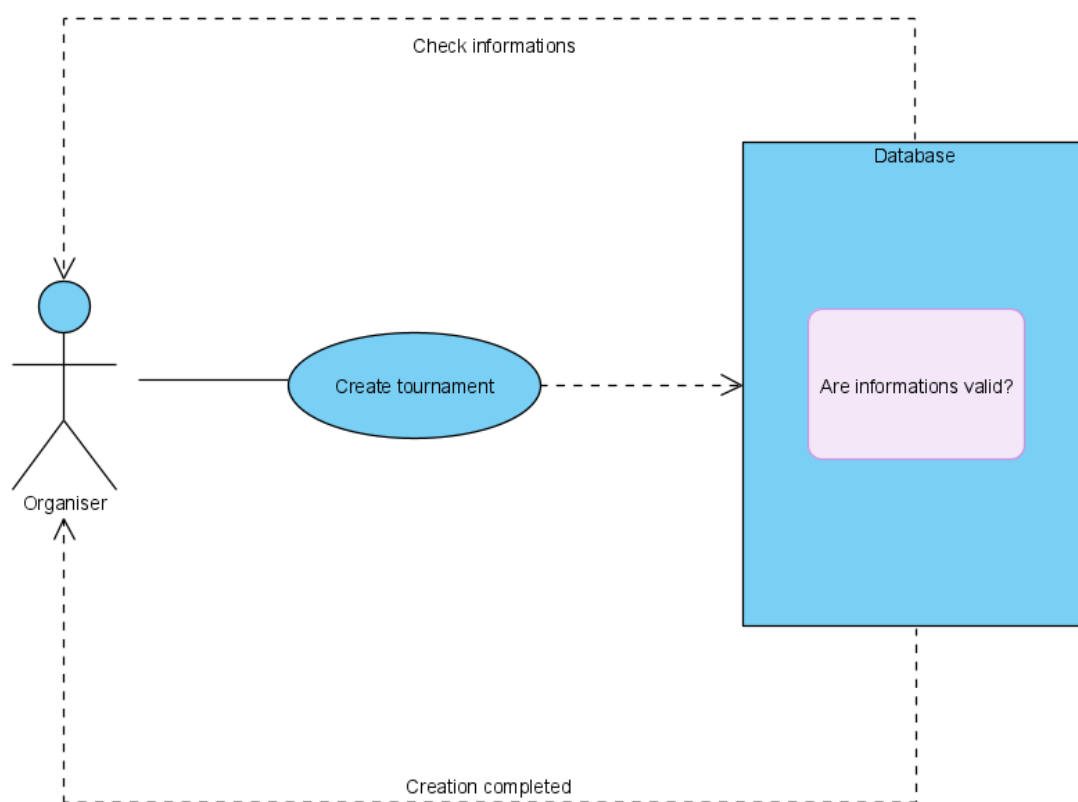


Diagram przedstawiający założenie turnieju przez organizatora.

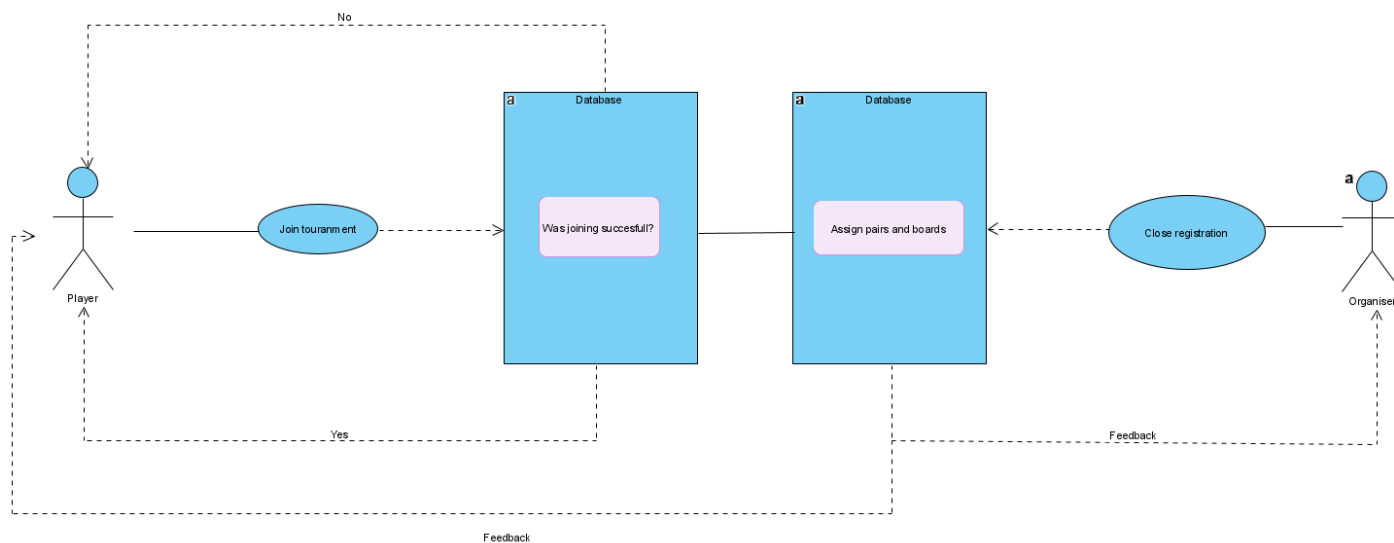
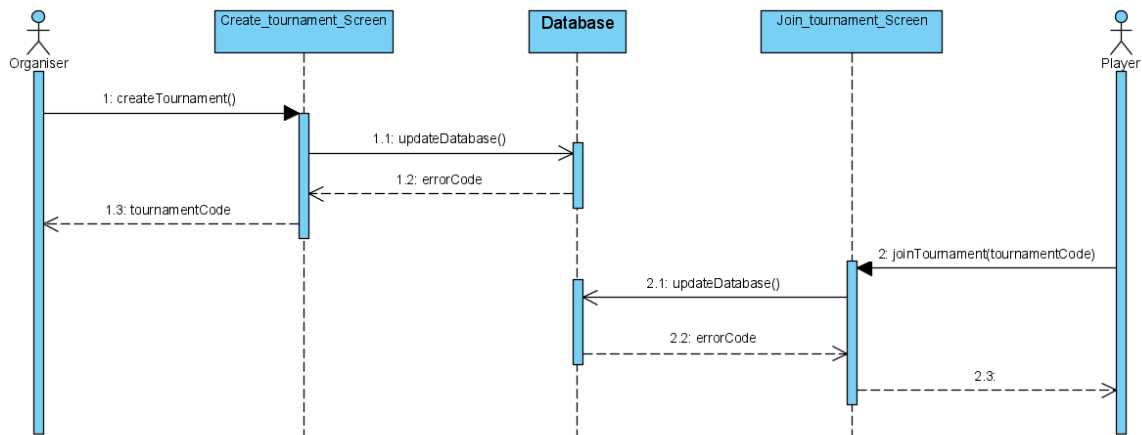
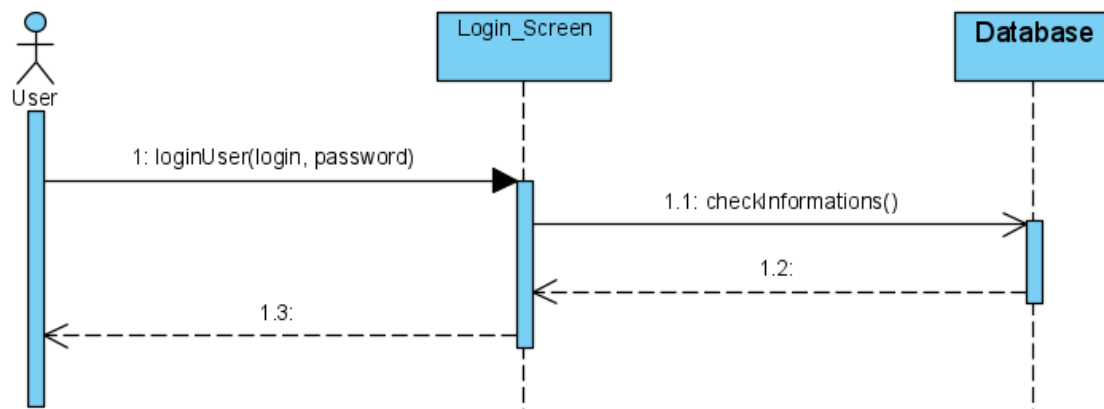


Diagram przedstawiający proces dołączania graczy do turnieju.

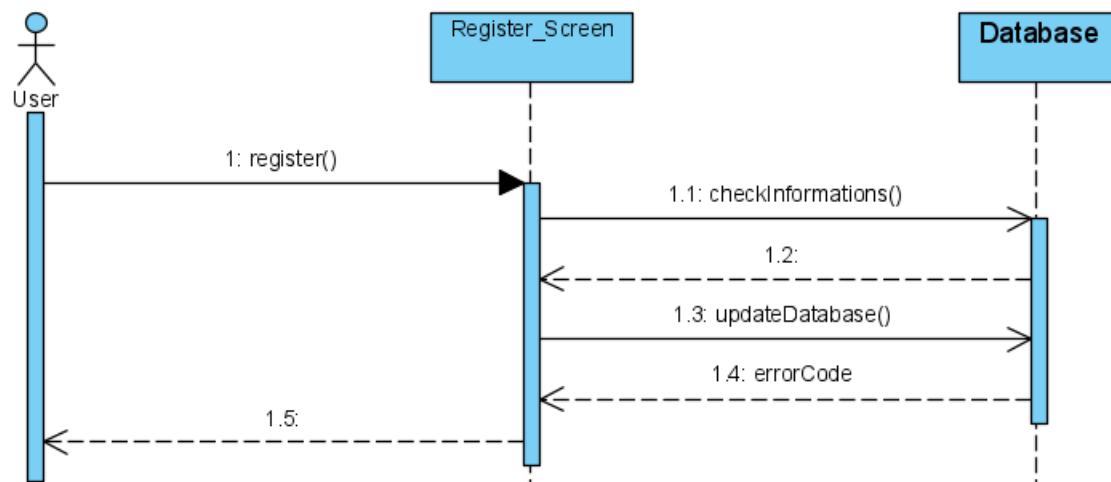
9.3.3 Diagramy sekwencji



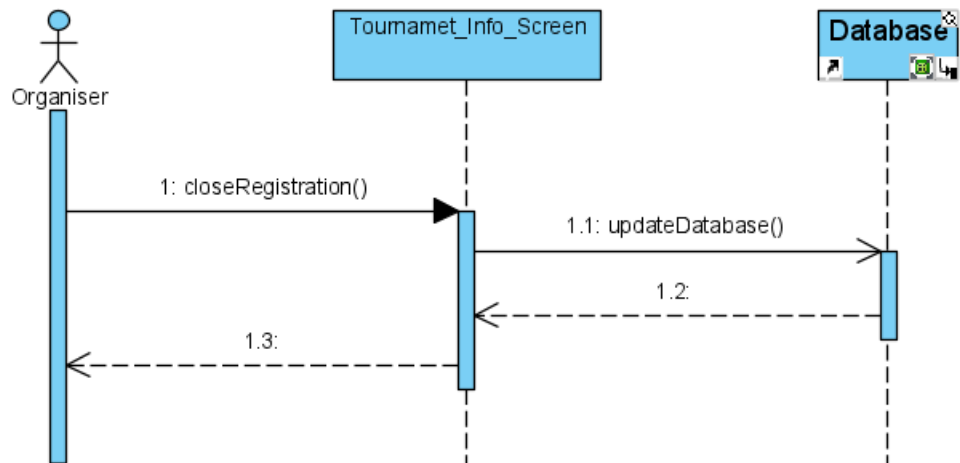
Organizator tworzy turniej otwarty dla wszystkich użytkowników.



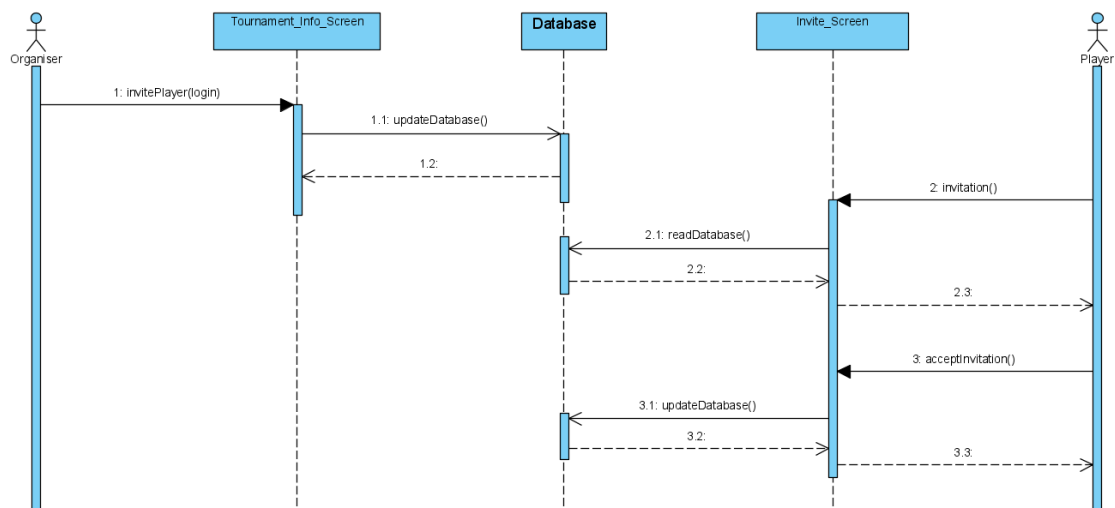
Logowanie użytkownika do systemu.



Zakładanie konta przez użytkownika.



Zamknięcie rejestracji do turnieju przez organizatora.



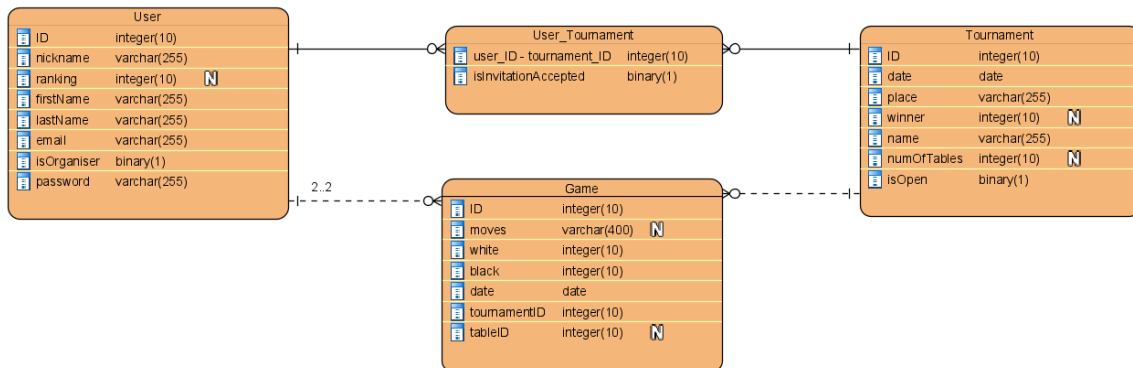
Organizator zaprasza konkretnego gracza do turnieju.

9.3.4 Inne diagramy

co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.

9.4 Projekt bazy danych

9.4.1 Schemat



9.4.2 Projekty szczegółowe tabel

Tabela Users:

- ID – indywidualny identyfikator pozwalający na weryfikację użytkownika, np. 1
- nickname – nazwa użytkownika wyświetlana w aplikacji, np. leOsiak
- password – hasło do konta użytkownika np. haslo
- ranking – wewnętrzny system rankingowy aplikacji, np. 580
- firstName – imię użytkownika, np. Wojciech
- lastName – nazwisko użytkownika, np. Wiszowaty
- email – email użytkownika, np. Jan.kowalski@gmail.com
- isOrganiser – informacja czy konto posiada możliwość organizacji turniejów, np. True/False

Tabela Game:

- ID – identyfikator partii, np. 1085
- moves – lista zapisanych ruchów, np. Kf3, d5, d4
- white – token osoby grającej białymi bierkami, np. 1234567890
- black – token osoby grającej czarnymi bierkami, np. 1234567809
- date – data rozegrania partii, np. 28-02-2021
- tournamentID – identyfikator turnieju w którym została rozegrana partia, np. 5841
- tableID – identyfikator stolika, np. 16

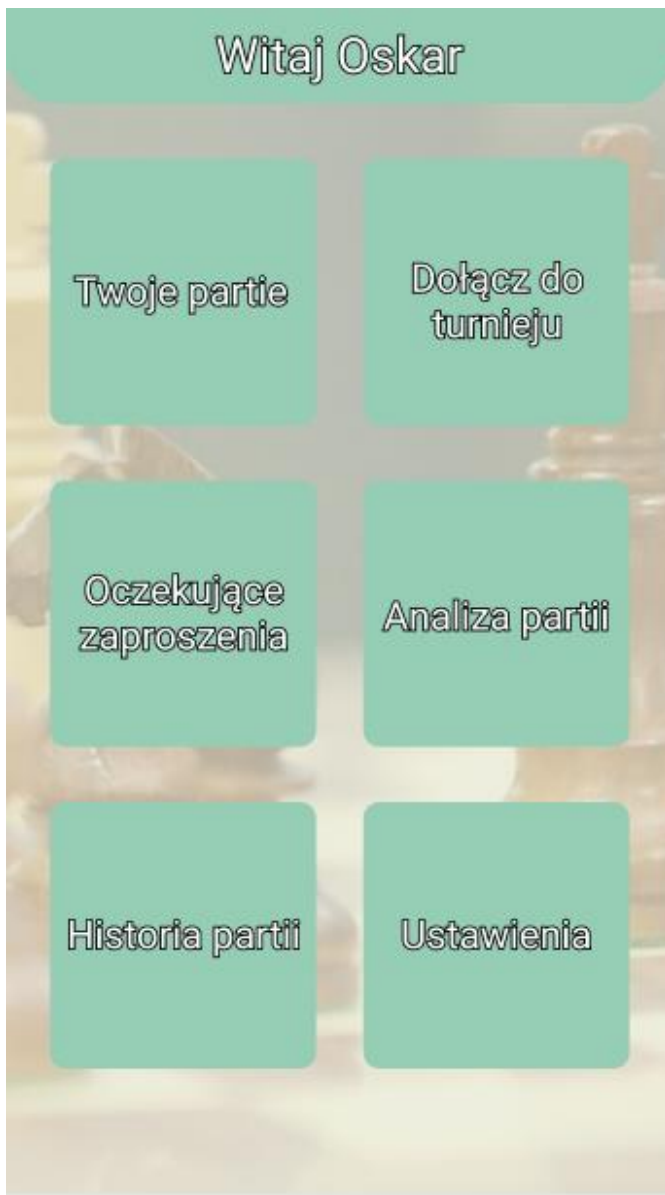
Tabela Tournament:

- ID – identyfikator turnieju, np. 54
- date – data rozpoczęcia turnieju, np. 23-05-2021
- place – miejsce rozegrania turnieju, np. Warszawa, ul. Wojska Polskiego 20
- winner – zwycięzca turnieju, np. 1234567890
- name – nazwa turnieju, np. Turniej Mistrzów
- numOfTables – liczba stolików, np. 15
- isOpen – informacja czy turniej jest dostępny dla wszystkich użytkowników

Relacja User_Tournament:

- user_ID – kolumna z tabeli User
- tournament_ID – kolumna z tabeli Tournament
- isInvitationAccepted – informacja czy gracz już potwierdził udział w turnieju

9.5 Projekt interfejsu użytkownika

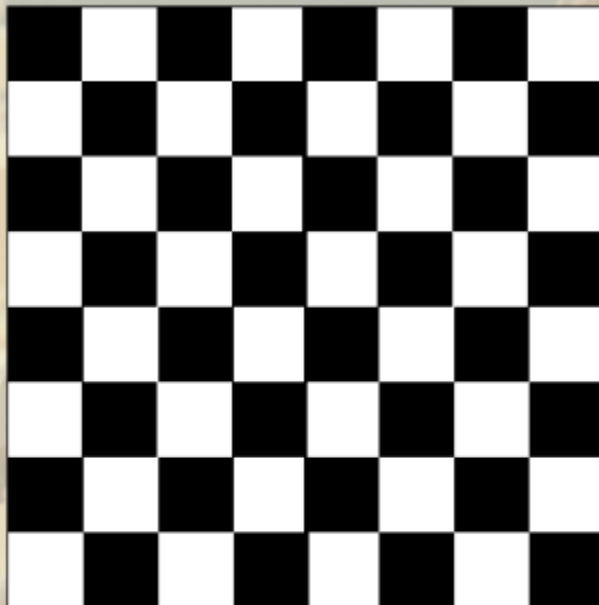


Ekran główny aplikacji



Ekran typu lista

Analiza partii



- | | | |
|----|------|------|
| 1. | e4 | c6 |
| 2. | d4 | d5 |
| 3. | Kc3 | dxe4 |
| 4. | Kxe4 | |

Ekran analizy partii

9.5.1 Lista głównych elementów interfejsu

- Ekran główny aplikacji
- Twoje partie – ekran typu lista
- Dołącz do turnieju
- Oczekujące zaproszenia – ekran typu lista
- Analiza partii
- Historia partii – ekran typu lista
- Ustawienia

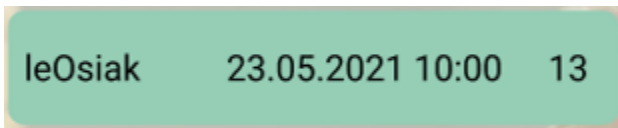
9.5.2 Przejścia między głównymi elementami

Przejścia pomiędzy ekranami aplikacji odbywają się przy pomocy przycisków:

- Przyciski w głównym oknie aplikacji



- Przyciski na ekranie typu lista



9.5.3 Projekty szczegółowe poszczególnych elementów

- 1
- Ekran główny aplikacji
- Rysunek 1
- Ekran główny pozwala na przeniesienie się do kolejnych ekranów przy użyciu odpowiednich przycisków
- Z bazy danych pobierana jest nazwa użytkownika, wyświetlana na górnym pasku aplikacji
- Przyciski powodują wywoływanie odpowiednich ekranów aplikacji

- 2
- Twoje partie
- Rysunek 2
- Lista partii, do których zapisany jest zawodnik
- Z bazy danych pobierana jest lista partii o dacie późniejszej niż aktualna, których uczestnikiem jest zalogowany zawodnik
- Kliknięcie w partię powoduje wyświetlenie się dokładnych informacji o niej (kolor bierek, numer stolika)

- 3
- Dołącz do turnieju
- Pole tekstowe pozwalające na wpisanie kodu turnieju, oraz przycisk pozwalający wysłać zaproszenie
- Kliknięcie w przycisk powoduje wyszukanie w bazie danych turnieju o podanym kodzie, oraz wysłanie do organizatora prośby o dołączenie do turnieju

- 4
- Oczekujące zaproszenia
- Rysunek 2
- Lista oczekujących zaproszeń wysłanych przez gracza do organizatorów, oraz przez organizatorów do gracza
- Lista oczekujących zaproszeń przypisanych do gracza
- Kliknięcie w zaproszenie na liście, powoduje wyświetlenie szczegółowych informacji (Nazwa turnieju, data, ilość graczy, informacje o organizatorze)

- 5
- Analiza partii
- Rysunek 3
- Ekran wizualizujący zapisaną w bazie danych partię
- Z bazy danych pobierany zapis przebiegu partii oraz nazwy graczy którzy ją rozgrywali

- 6
- Historia partii
- Rysunek 2
- Historia rozegranych przez gracza partii
- Z bazy danych pobierana będzie lista partii, do których przypisany jest użytkownik, a które zostały rozegrane wcześniej niż aktualna data
- Kliknięcie w partię wyświetli szczegółowe informacje (przebieg, wynik)

9.6 Procedura wdrożenia

Wdrażanie aplikacji odbywa się etapami. Każdy element systemu przed końcowym zaakceptowaniem jest testowany, aby wyeliminować ewentualne błędy lub konflikty. Poszczególne elementy i nowe funkcjonalności aplikacji będą oddawane do wstępnej oceny funkcjonalnej i graficznej.

W przypadku wystąpienia opóźnień w oddawaniu kolejnych etapów możliwe są konsultacje ze zleceniodawcą w celu zmiany ustalonego terminu.

Proces wdrożenia obejmuje również czterotygodniowe serwisowanie aplikacji pod kątem ewentualnych problemów wynikających z użytkowania przez większą liczbę osób.

10 Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

- pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
- może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

11 Podsumowanie

11.1 Szczegółowe nakłady projektowe członków zespołu

Nad dokumentacją pracowaliśmy razem poprzez udostępnianie ekranu na komunikatorze głosowym. Elementy dokumentacji wykonane przez jedną osobę poza czasem zajęć są uwzględnione w tabeli poniżej:

Wiszowaty	Osiak
Projekt GUI – punkt 9.5	Projekt Bazy Danych – punkt 9.4
Projekt architektury systemu – punkt 9.1	Punkt 6 – diagram i opis słowny
Analiza SWOT – punkt 4.4	Słownik – punkt 3
Logo aplikacji	Lista czynników ryzyka – punkt 7.1

12 Inne informacje

Gotowy system będzie wspierany przez założony przez zleceniodawcę czas od momentu wydania. W tym czasie użytkownicy mogą sugerować pomysły na zmiany, które chcieliby, aby zostały wprowadzone. Ponadto, w tym czasie będą naprawiane wszelkie błędy, które zostaną zauważone bądź zgłoszone przez użytkowników systemu.

W przyszłości aplikacja może zostać przeniesiona jako aplikacja webowa na dowolne urządzenia. Podjęcie tego typu działań zależy od zapotrzebowania rynku, odbioru aplikacji przez samych użytkowników oraz rozwoju technologii Flutter Web.