AKADEMIA
NAUK STOSOWANYCH
W ŁOMŻY

Wydział Nauk Informatyczno-Technologicznych

Kierunek studiów: **Informatyka I stopnia**

Projekt Zespołowy

Temat projektu: Przeglądarkowa gra karciana.

Marcin Kiczyński

10286

Mateusz Bloch

10000

Adam Kania

10012

26 października 2022

Streszczenie

Dokumentacja projektu zawiera szczegółowy opis przebiegu całego projektu. Wstęp zawiera cel projektu, a także krótkie wprowadzenie do tematu. Cel biznesowy określa, jakie korzyści będzie przynosić projekt dla firmy lub organizacji. Wymagania funkcjonalne i niefunkcjonalne opisują, jakie funkcje musi posiadać system, aby spełnić potrzeby użytkowników. Metodyka pracy opisuje, jakie metody i narzędzia zostaną wykorzystane do realizacji projektu. Harmonogram zawiera planowany przebieg projektu, z uwzględnieniem poszczególnych etapów i terminów ich realizacji. Diagramy UML przedstawiają wizualizację struktury i działania systemu. Podsumowanie zawiera opis celów zrealizowanych oraz niezrealizowanych, Opis problemów z którymi zetknął się zespół podczas realizacji projektu i jak sobie z nimi poradzono. Kierunki rozbudowy systemu opisują możliwe rozwinięcia i ulepszenia systemu po jego wdrożeniu.

Historia zmian

Data	Autor	Opis zmiany	Wersja
26.10.22r.	Mateusz Bloch	Utworzenie dokumentu	1.0
27.10.22r	Marcin Kiczyński	Opis wymagań funkcjonalnych	1.0
27.10.22r.	Adam Kania	Opis wymagań niefunkcjonalnych,	1.0
10.11.22r.	Adam Kania	Diagram przypadków użycia, diagram komponentów	1.0
10.11.22r.	Mateusz Bloch, Marcin Kiczyński	Modyfikacja dokumentu	1.1
23.11.22r.	Adam Kania	Dodanie diagramów stanów	1.1
20.01.23r.	Adam Kania, Marcin Kiczyński	Podsumowanie	1.1
25.01.23r.	Adam Kania	Streszczenie dokumentacji, Diagram wdrożenia	1.2

Spis treści

1. Wstęp

1.1 Ogólna charakterystyka projektu

Celem projektu jest stworzenie aplikacji webowej pozwalającej na rozegranie między dwoma graczami pojedynku w formacie 1 kontra 1. Gracze mierzą się ze sobą w turowej grze karcianej, której celem jest pozbawienie oponenta jego punktów życia. Format rozgrywki pozwala na potencjalne utworzenie sceny e-sportowej wokół tego tytułu co pomogłoby w promocji.

1.2 Przegląd istniejących rozwiązań

1. Hearthstone - gra karciana fantasy opracowana przez firmę Blizzard Entertainment, gracze kolekcjonują karty i używają ich do pojedynku z innymi graczami w trybie online.

2. Magic: The Gathering Arena - cyfrowa wersja kultowej gry karcianej opracowanej przez firmę Wizards of the Coast, gracze konstruują talie z kart i walczą z innymi graczami w trybie online.

3. Gwent: The Witcher Card Game - gra karciana stworzona przez studio CD Projekt RED, oparta na uniwersum Wiedźmina. Gracze konstruują talie z kart i używają ich do pojedynków z innymi graczami w trybie online.

2. Cel biznesowy

Gra bazować ma na modelu bezpłatnym z mikro transakcjami w postaci paczek z nowymi kartami, nowym wyglądzie areny oraz na zakupie ulepszonego wyglądu najpopularniejszych kart (np. grafiki stworzone w innym stylu, kolorowe obwódki kart itd.). Karty bazowały będą na znanych poetach i autorach książek.

3. Wymagania funkcjonalne

WF.01	Użytkownik ma możliwość zarejestrowania się. Jeśli posiada konto może dokonać zalogowania się.
WF.02	Zalogowany użytkownik może zapisać się do kolejki w celu znalezienia rozgrywki.
WF.03	Zalogowany użytkownik może rozegrać pojedynek z innym graczem

4. Wymaganie niefunkcjonalne

Użyteczność	WPF.01	System powinien zapewniać dostęp użytkownikom popularniejszych przeglądarek.
Niezawodność	WPF.02	System powinien działać cały czas z wyjątkiem konserwacji występujących co 2 tygodnie.
Bezpieczeństwo	WPF.03	System powinien bezpiecznie przechowywać dane użytkowników.
Wydajność	WPF.04	System powinien zapewniać rozegranie pojedynku dla 2 graczy.
Utrzymanie	WPF.05	Aplikacja powinna być stworzona w czytelny i zrozumiały sposób.
Skalowalność	WPF.06	System powinien zapewniać możliwość tworzenia nowych kart oraz aktywności.

5. Metodyka pracy

Do zarządzania projektu użyliśmy oprogramowania JIRA, która pozwala na dzielenie pracy na etapy i przypisywania odpowiednim osobom, śledzenie zgłoszeń, backlogi projektu.

Komunikacje w zespolu realizowaliśmy za pomocą komunikatora Discord

W projkecie używane także były Google Docs do tworzenia podstawowej dokumentacji.

Środki implementacyjne zostawane przez nas to następujące technologie:

Node.js - wieloplatformowe środowisko uruchomieniowe o otwartym kodzie do tworzenia aplikacji typu server-side napisanych w języku JavaScript.

Socket.io - biblioteka do języka javascript, która pozwala na komunikację z serwerem w czasie rzeczywistym.

MongoDB - otwarty, nierelacyjny system zarządzania bazą danych.

Git - rozproszony system kontroli wersji.

Amazon Web Services (AWS) - publiczna platforma chmurowa oraz hostingowy serwis internetowy.

Render – Serwis hostingowy.

UML – Metoda modelowania

6. Harmonogram

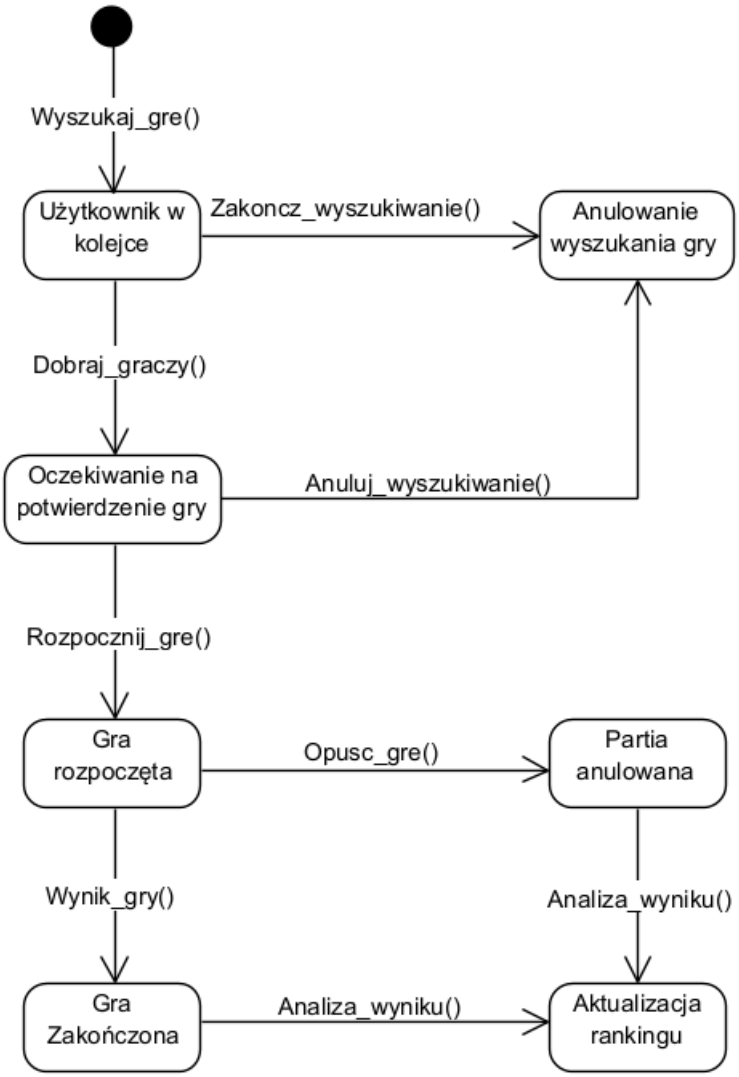
Zadanie	Termin	Wykonawca
Przygotowanie założeń projektu	Październik 2022r.	Adam Kania, Mateusz Bloch, Marcin Kiczyński
Podstawowy zarys projektu	Październik 2022r.	Adam Kania, Mateusz Bloch, Marcin Kiczyński
Diagramy	I połowa grudzień 2022r.	Adam Kania, Marcin Kiczyński
Działający program	10 styczeń 2023r.	Marcin Kiczyński, Mateusz Bloch, Adam Kania
Rozwiązanie ewentualnych problemów	25 styczeń 2023r.	Adam Kania
Oddanie projektu	26 styczeń 2023r.	Adam Kania, Mateusz Bloch, Marcin Kiczyński

7. Diagramy

Rysunek 1. Przypadki użycia

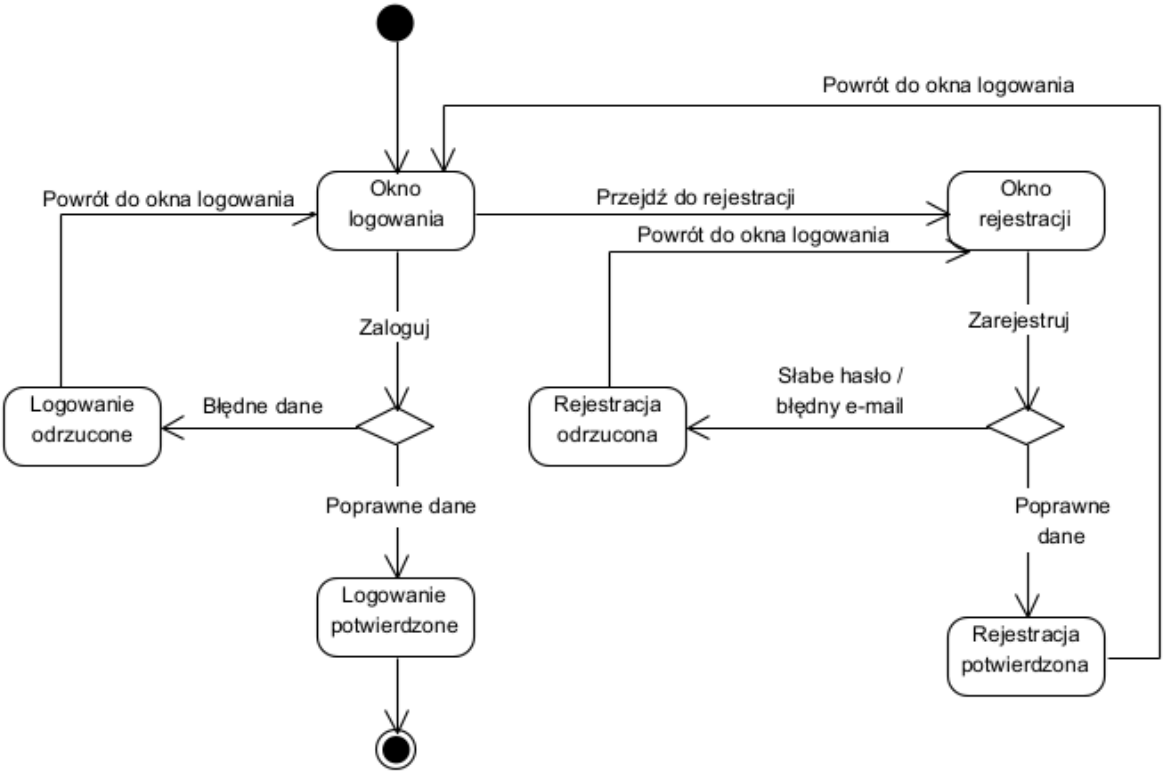
Powyższy diagram przypadków przedstawia sposób korzystanie systemu przez użytkownika.
Użytkownik ma możliwość rejestracji oraz logowania. Zalogowany użytkownik posiada możliwość edycji danych, rozpoczynania oraz dołączania do rozgrywki, a także zarządzania własną kolekcją kart poprzez dodawania ich czy usuwania.

System automatycznie zarządza meczami, rankingiem a także użytkownikami.



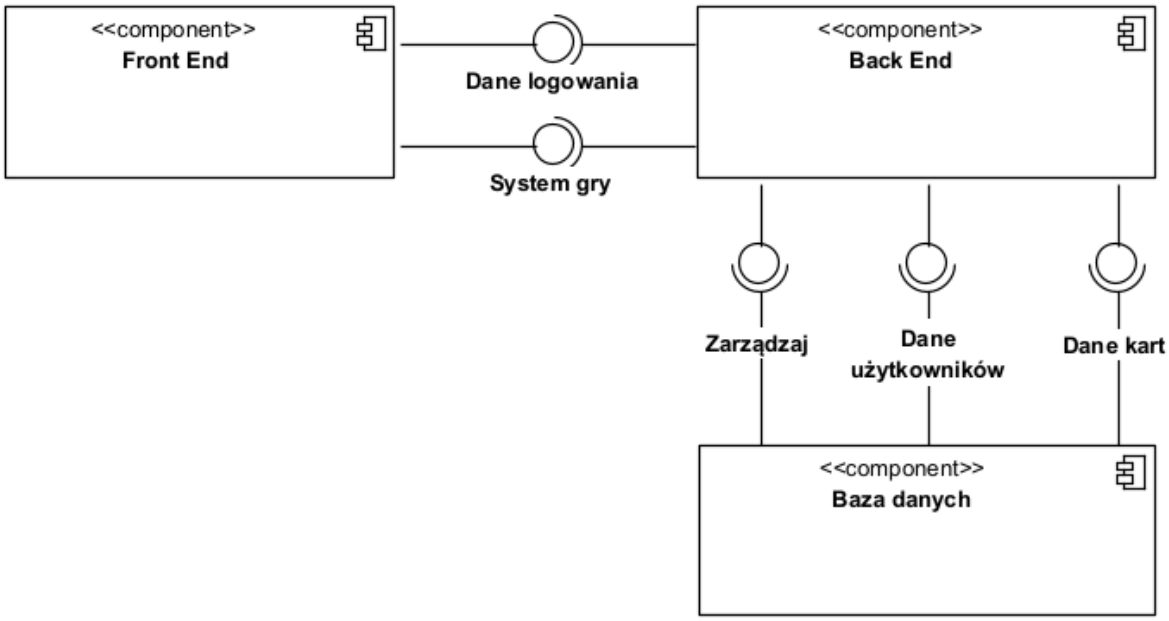
Rysunek 2. Protokołowa maszyna stanowa – przebieg rozgrywki

Powyższy diagram stanów przedstawia przebieg rozgrywki. Po wejściu do kolejki przez użytkownika system rozpoczyna dobieranie drugiego gracza pod względem posiadających punktów rankingowych. W każdym momencie wyszukiwania użytkownik może anulować wyszukiwanie gry. Po znalezieniu gry przez system obaj gracze muszą zaakceptować grę w celu potwierdzenia obecności, w innym przypadku wyszukiwanie gry jest zakończone. Po rozpoczęciu gry uczestnicy mają możliwość opuszczenia rozgrywki co spowoduje aktualizację rankingu na korzyść gracza, który pozostał w grze. Po zakończeniu gry w domyślny sposób system aktualizuje ranking na podstawie punktów rankingowych posiadanych przez graczy oraz wyniku meczu.



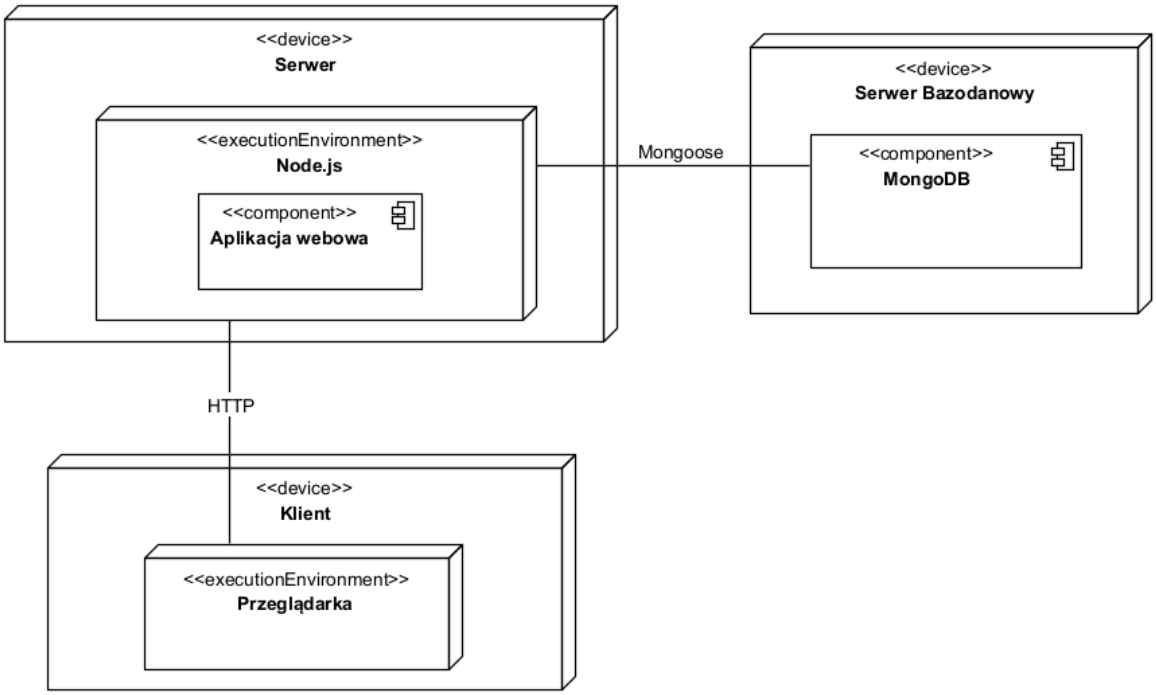
Rysunek 3. Diagram stanów – przebieg logowania

Powyższy diagram stanów przedstawia przebieg logowania. W celu zalogowania użytkownik musi podać poprawne dane, w innym przypadku powraca do okna logowania. W przypadku gdy użytkownik nie ma konta przechodzi do rejestracji. W celu zarejestrowania użytkownik musi podać poprawny e-mail oraz silne hasło po czym powraca do okna logowania.



Rysunek 4.Diagram komponentów

Back-end zarządza bazą danych oraz komunikuje się z nią poprzez odpowiedni interfejs. Back-end udostępnia rozwiązania front-endowi na podstawie informacji zapisanych w bazie danych.



Rysunek 5. Diagram wdrożenia

Diagram wdrożenia przedstawia rozmieszczenie elementów systemu oraz komunikacje między nimi. Urządzenie Serwer zawiera aplikację webową uruchomioną za pomocą Node.js. Serwer został połączony z bazą danych z pomocą narzędzia Mongoose natomiast w celu połączenia się z klientem został wykorzystany protokół komunikacyjny HTTP.

8. Użytkowanie

8.1 Instrukcja wdrożeniowa

Aby uruchomić projekt należy:

1. Pobrać i zainstalować Node.js (wersja 18.13.0)
2. Pobrać projekt z GitHuba
3. Otworzyć pobrany folder w visual studio code

4. Utworzyć w folderze plik o nazwie .env i wkleić w nim następujący kod:

```
DATABASE_URI=mongodb+srv://bloch:uF3yiuiuR04n1TaP@cluster0.szimj8c.mongodb.net/baza?retryWrites=true&w=majority
```

5. Otworzyć terminal, użyć komendy „npm install” a następnie „npm start”
6. W przeglądarce wpisać http://localhost:3000

8.2 Instrukcja użytkowania

Po otworzeniu aplikacji w przeglądarce pojawi się okno logowania widoczne na Rysunek 6

Rysunek 6. Okno logowania

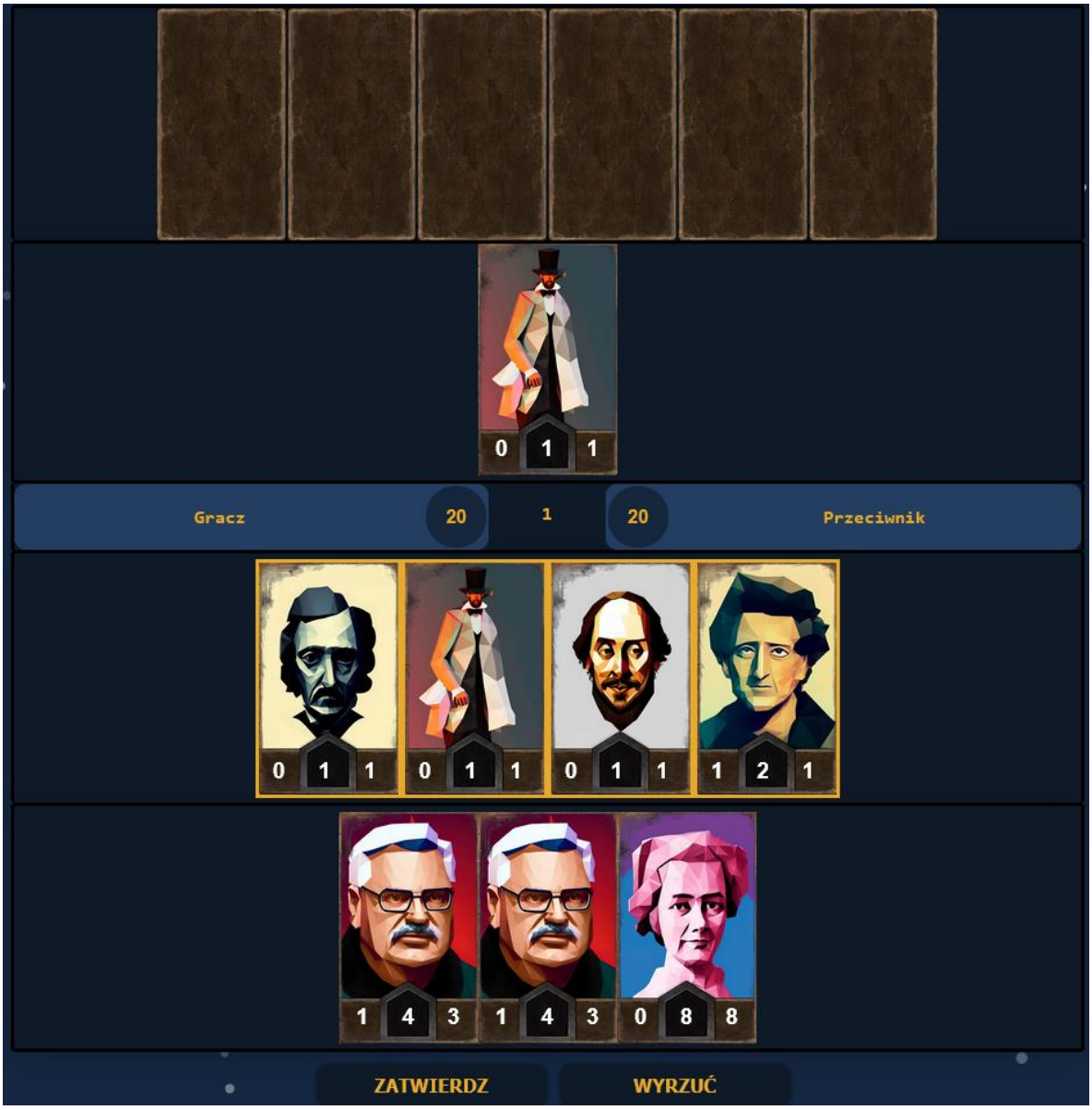
W celu zalogowania się najpierw trzeba zarejestrować się klikając przycisk „Rejestracja”. Okno rejestracji przedstawia Rysunek 7.

Rysunek 7 Okno rejestracji

W celu zarejestrowania się należy wprowadzić login i hasło które trzeba dodatkowo potwierdzić. Po prawidłowym zarejestrowaniu będzie możliwe zalogowanie się. Zalogowany użytkownik będzie miał możliwość wyszukania oponenta lub wylogowania się tak jak jest to przedstawione na Rysunek 8.

Rysunek 8 Okno po zalogowaniu się

Po naciśnięciu przycisku „ZNAJDŹ OPONENTA” rozpoczniemy wyszukiwanie innego zalogowanego użytkownika który aktualnie szuka gry. Po wyszukaniu oponenta będziemy mogli rozpocząć pojedynek z innym graczem, który prezentuje się na Rysunek 9



Rysunek 9 Okno gry

9. Podsumowanie

9.1 Opis celów zrealizowanych i niezrealizowanych

1	Aplikacja mobilna – pozwalająca użytkownikom androida na korzystanie z aplikacji	Niezrealizowany
2	Sklep – pozwalający na zakup kart	Niezrealizowany
3	System rankingowy – Określający miejsce w rankingu graczy oraz pozwalający na dobieranie oponentów na zbliżonym poziomie umiejętności.	Niezrealizowany
4	Aplikacja webowa – Pozwalająca na dostęp do aplikacji z poziomu przeglądarki internetowej.	Zrealizowany
5	Rozegranie pojedynku 1 na 1	Zrealizowany
6	System dodawania kart umożliwiający na łatwe dodawanie nowych kart.	Zrealizowany

9.2 Opis problemów

1	Zmiana funkcjonowania algorytmu strony spowodowała pogorszenie jakości generowanych portetów poetów.	Karty wymagały większego dopracowania w procesie edycji grafiki.
2	Dodanie funkcjonalności usuwania kart – po dodaniu tej funkcjonalności wiele innych funkcjonalności zaprzestało porawnie funkcjonować.	Naprawa niedziałających funkcjonalności.
3	Rozmiar planszy różnił się w zależności do rozdzielczości programu (ucinanie części planszy)	Zmiana wartości skalowania kart.

9.3 Kierunki rozbudowy systemu

System rankingowy – Pozwoli na dobieranie graczy biorąc pod uwagę ich umiejętności zarządzaniem talią kart.

Umiejętności kart – Pozwolą na urozmaicenie rozgrywki, oraz sprawi że strategiczne podejście do gry będzie miało większe znaczenie.

Porty na inne platformy – Pozwolenie użytkownikom korzystających z innych platform grać z użytkownikami przeglądarek.

Ścieżka dźwiękowa oraz animacje kart – Pozwalające milej spędzać czas podczas gry.

Tworzenie talii – Pozwolenie użytkownikom tworzyć własne talie kart.

9.4 Wnioski

W trakcie tworzenia programu wynikło, że część zakładanych funkcjonalności jest niepotrzebna. Istnieje również kilka funkcji które zostały pominięte z powodu trudności implementacji lub z powodu znalezienia bardziej optymalnego rozwiązania.

Cały projekt pozwolił nam na podniesienie naszych umiejętności pracy w grupach oraz na lepsze zapoznanie się z narzędziami które musieliśmy wykorzystać w projekcie.

10. Literatura

[1] "Introdution to Node.js" https://nodejs.dev/en/learn/

[2] Socket.IO Documentation https://socket.io/docs/v4/

[3]”Introduction to MongoDB" https://docs.mongodb.com/manual/introduction/