Dokumentacja Projektu Zespołowego

Nazwa Aplikacji: Magnaci i Czarodzieje

Rodzaj aplikacji: Aplikacja webowa typu - Gra internetowa

Skład Grupy:

-Paweł Szapiel
-Jakub Kozłowski
-Michał Andrejczuk

0. Wstęp

Aplikacja ma służyć do zarządzania danymi i obsługiwać prostą grę Rpg. Umożliwia ona stworzenie i rozwój swojej postaci. Kluczowym w rozwoju postaci jest duży wpływ na rozmaite statystyki, które użytkownik może modyfikować za pomocą przedmiotów i zdobywając doświadczenie. Celem gry jest wykonywanie zadań za które jest nagradzany wirtualnymi nagrodami. Jest to dość autonomiczny system do którego nie potrzeba nadzoru ponieważ wszystko jest zautomatyzowane, ale jeśli pojawiałyby się problemy to całość jest zarządzana przez administratorów sprawujących pieczę nad graczami.

1. Analiza wymagań systemu

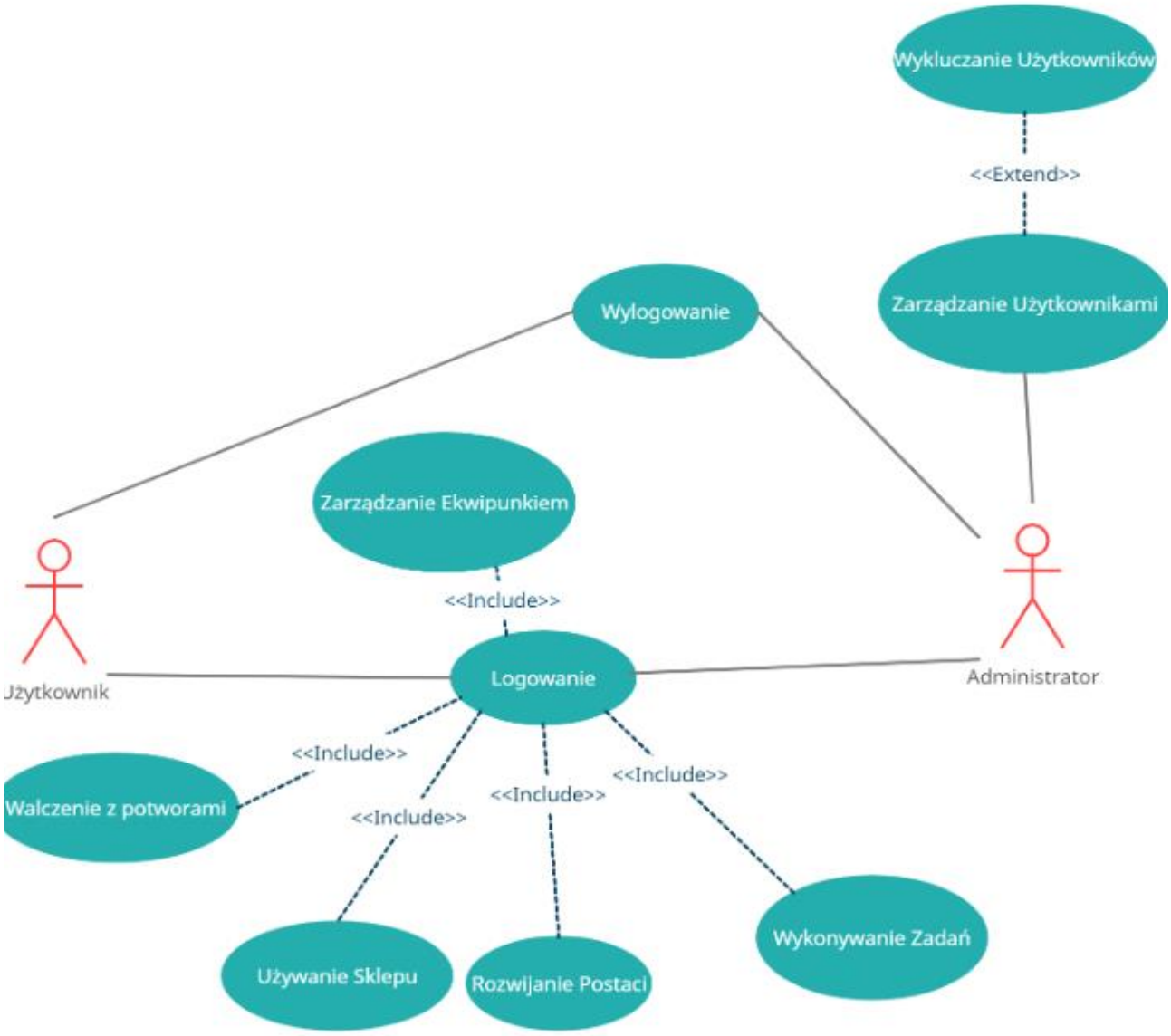
1.1 Wymagania funkcjonalne

-Możliwość zakładania konta.
-Możliwość tworzenia postaci.
-Korzystanie ze sklepu wewnątrz gry.
-Edycja statystyk.
-Wykonywanie zadań.
-Zdobywanie poziomów.

1.2 Wymagania niefunkcjonalne

-Wygodne użytkowanie aplikacji przez internet
-Szybka reakcja na działanie użytkownika
-Przyjemny interfejs

2. Diagram przypadków użycia



3. Wykorzystane technologie

Aplikacja Frontendowa:

ReactJS v17
React Redux v7
React Router-Dom v5

Aplikacja korzysta ze sposobu przechowywania danych zapewnionego przez React-Redux. Umożliwia operacje CRUD na tabelach w sposób przypominający grę rpg. Do wykonywania zapytań REST użyta została biblioteka axios.

Widoki:

React-Bootstrap v2
FontAwesome for React

```
"@fortawesome/fontawesome-svg-core": "^1.3.0-beta2",
"@fortawesome/free-regular-svg-icons": "^6.1.1",
"@fortawesome/free-solid-svg-icons": "^5.15.4",
"@fortawesome/react-fontawesome": "^0.1.16",
"@reduxjs/toolkit": "^1.7.0",
"@testing-library/jest-dom": "^5.11.4",
"@testing-library/react": "^11.1.0",
"@testing-library/user-event": "^12.1.10",
"axios": "^0.24.0",
"bootstrap": "^5.1.3",
"react": "^17.0.2",
"react-bootstrap": "^2.0.3",
"react-dom": "^17.0.2",
"react-hook-form": "^7.22.2",
"react-notifications": "^1.7.3",
"react-redux": "^7.2.6",
"react-router-dom": "^5.3.0",
"react-scripts": "4.0.3",
"web-vitals": "^1.0.1"
```

Lista zależności aplikacji

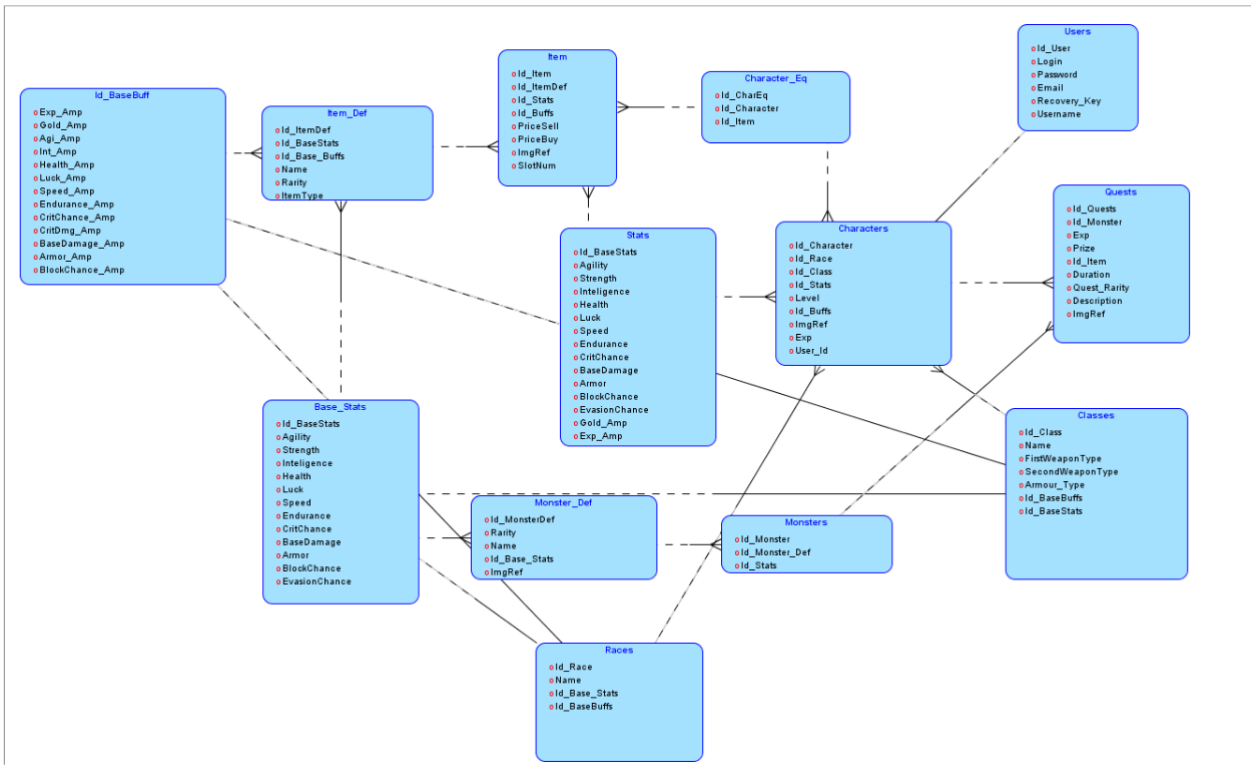
Aplikacja Backendowa:

Spring Boot v2.5.6 (Jdk v11)

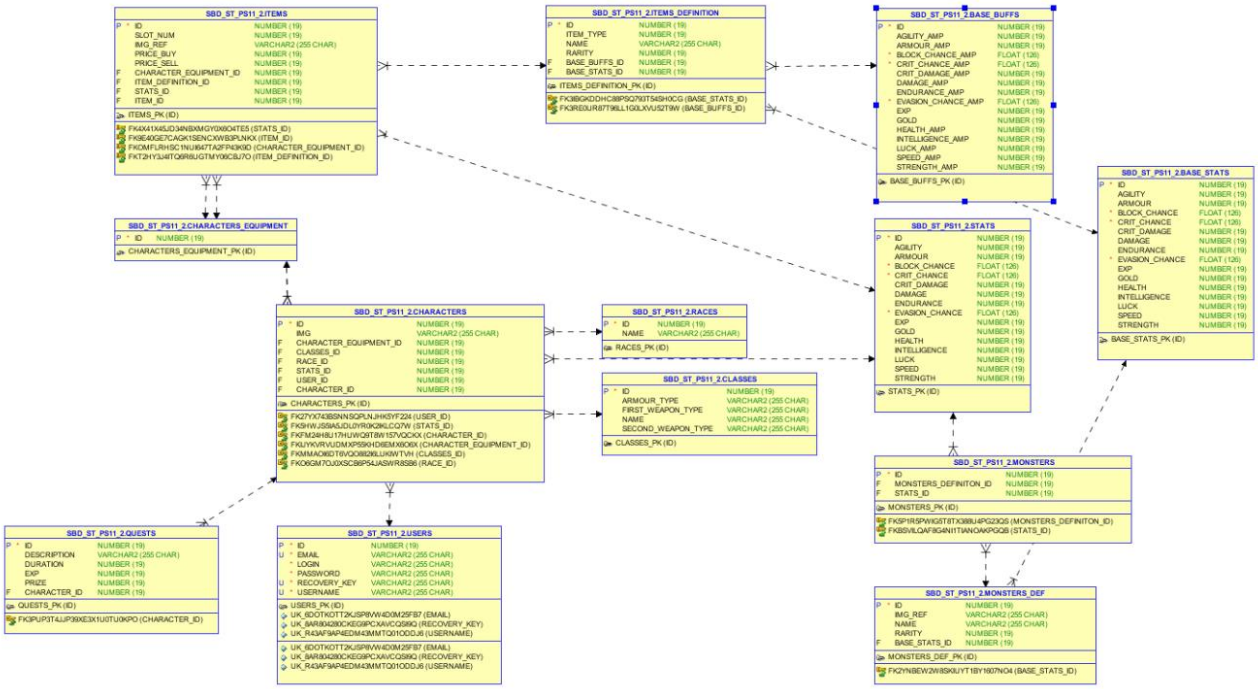
Architektura aplikacji opiera się na wzorcu MVC (Model-View-Controller). Użytkownik ma możliwość
poruszania się na serwisie po różnych widokach stron. Połączone są one z odpowiadającymi im
kontrolerami, które mogą modyfikować dane w bazie danych, korzystając z odpowiednich modeli.
Po uaktualnieniu danych zmieniane konkretne widoki, które następnie są prezentowane użytkownikowi. Aplikacja wykorzystuje zewnętrzny serwer bazy danych Oracle. Dodatkowo,
wykorzystuje ona Hibernate + JPA do wykonania procesu mapowania oraz korzystania z bazy
danych.
Baza Danych:

Postgresql

4. Schemat bazy danych
4.1 Encje



4.2 Szczegółowy schemat relacji



4.3 Dokładny opis tabel bazy danych:

- **User** – tabela przechowuje dane użytkowników systemu(klientów i administratorów)

Każdy rekord z tabeli User przechowuje ID, email, login, password, username, recovery_key

- **Character** – tabela przechowuje dane postaci

Każdy rekord z tabeli Character przechowuje ID, img (zdjęcie wybrane przez użytkownika, ID ekwipunku , ID klasy, ID rasyID statystyk ,ID użytkownika

- **Character_equipment** – tabele przechowuje dane o przedmiotach przypisanych do postaci

Każdy rekord z tabeli CHARACTER_EQUIPMENT zawiera ID

- **Classes** – tabela przechowuje dane o klasach

Każdy rekord z tabeli Classes przechowuje ID, rodzaj zbroi, którą dana klasa może założyć, rodzaj głównej broni, która dana klasa może walczyć(miecz), rodzaj drugiej broni, która dana klasa może się posługiwać(tarcza), nazwa klasy, ID podstawowych statystyk klasy, ID podstawowych wzmocnień(buffów) klas.

- **Races** – tabela przechowuje dane o rasach

Każdy rekord z tabeli Races przechowuje ID, nazwę rasy, ID podstawowych statystyk rasy, ID podstawowych wzmocnień(buffów) rasy

- **Monsters_def** – tabela przechowuje dane o definicjach potworów

Każdy rekord z tabeli Monsters_def przechowuje ID, zdjęcie potwora, nazwę, rzadkość, ID podstawowych statystyk potwora

- **Monsters** – tabela przechowuje dane o potworach

Każdy rekord z tabeli Monsters przechowuje ID, ID definicji potwora, ID zadania, ID statystyk

- **Quests** – tabela przechowuje dane o zadaniach

Każdy rekord z tabeli Quests przechowuje ID, opis zadania, długość trwania zadania, ilość zdobywanego doświadczenia, nagrodę(złoto), rzadkość zadania, ID postaci

- **Items_defintion** – tabela przechowuje dane o definicjach itemu

Każdy rekord z tabeli Items_definition przechowuje ID, rodzaj przedmiotu(łuk,miecz), nazwa przedmiotu, ID bazowych statystyk przedmiotu, ID bazowych wzmocnień przedmiotu

- **Items** – tabela przechowuje dane o itemach

Każdy rekord z tabeli Items przechowuje ID, numer miejsca przyporządkowanemu przedmiotowi, zdjęcie przedmiotu, Cena zakupu, Cena sprzedaży, ID ekwipunku, ID definicji przedmiotu, ID statystyk

- **Stats** –Tabela przechowuje dane dotyczące o statystykach postaci.

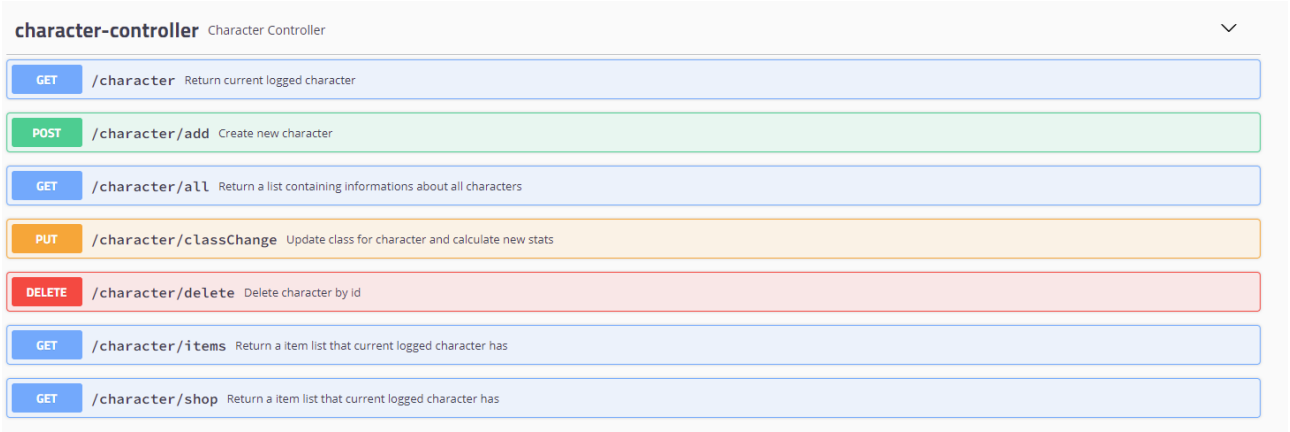
Każdy rekord z tabeli przechowuje dane o sile, zwinności, obronie, szansie bloku, szansie trafienia krytycznego, obrażeń, wytrzymałości, szansie uniku, ilości doświadczenia, punktach zdrowia, inteligencji, szczęścia, szybkości.

- **Base_Stats** – Są to statystyki które są bazowo przypisane do konkretnych postaci przed modyfikacją.

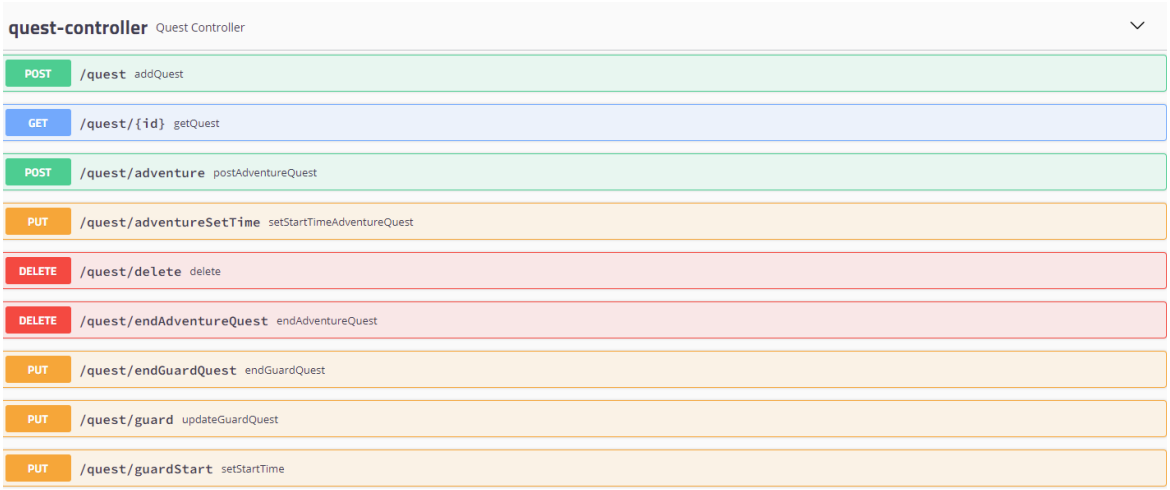
- **Base_Buffs** – Jest to mnożnik do statystyk strength_Amp jest mnożnikiem siły itd.

5. Dokumentacja RestAPI

5.1 Kontroler Character

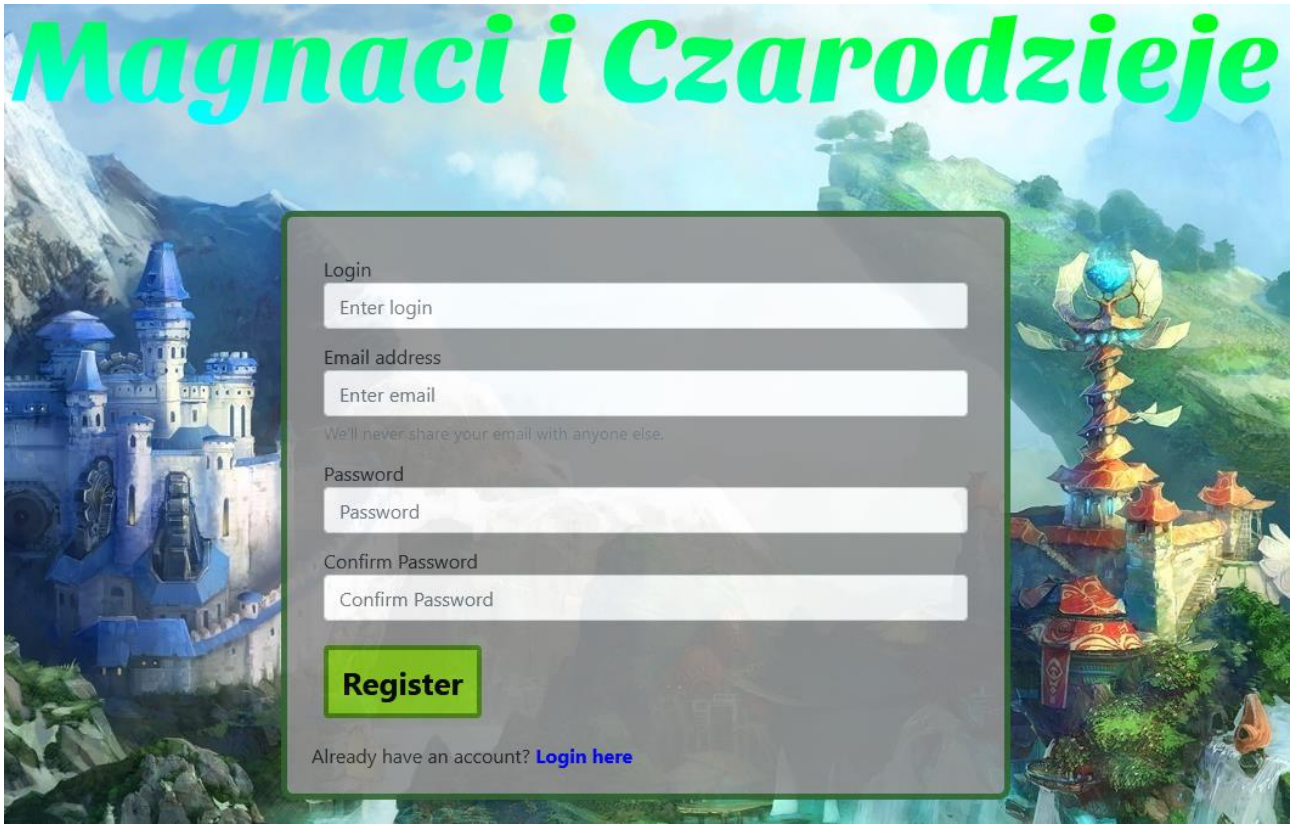


5.2 Kontroler Quest

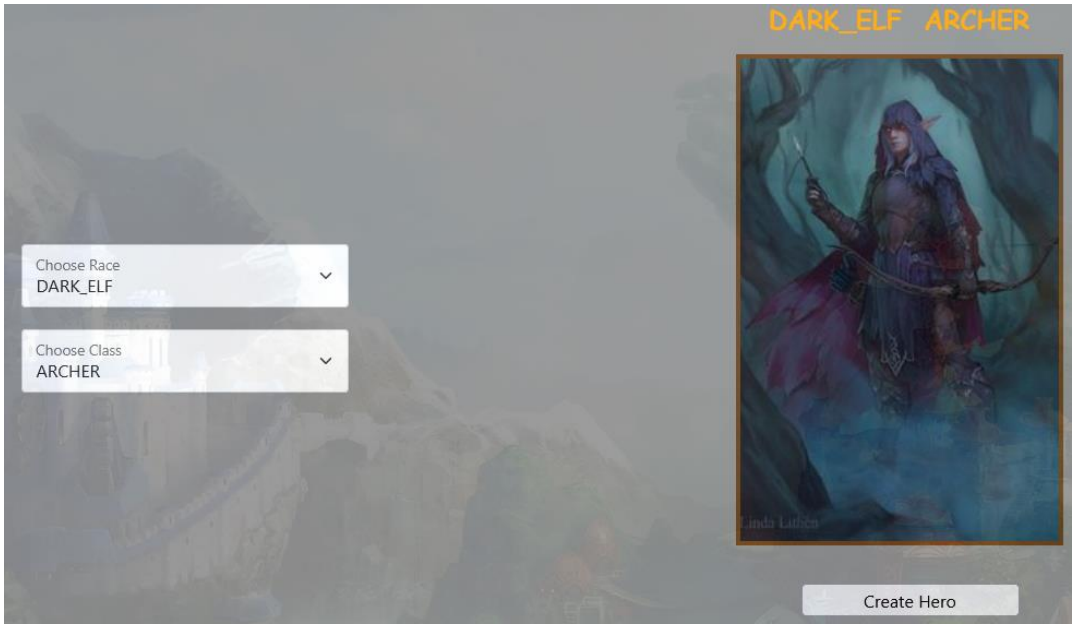


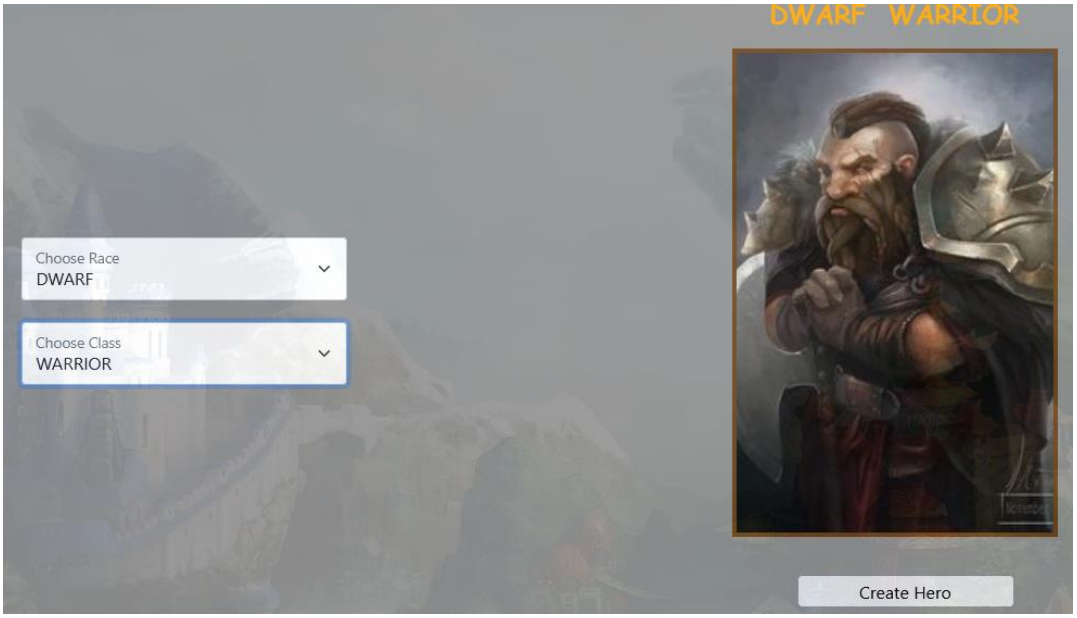
6. Interfejs użytkownika i przedstawienie funkcjonalności

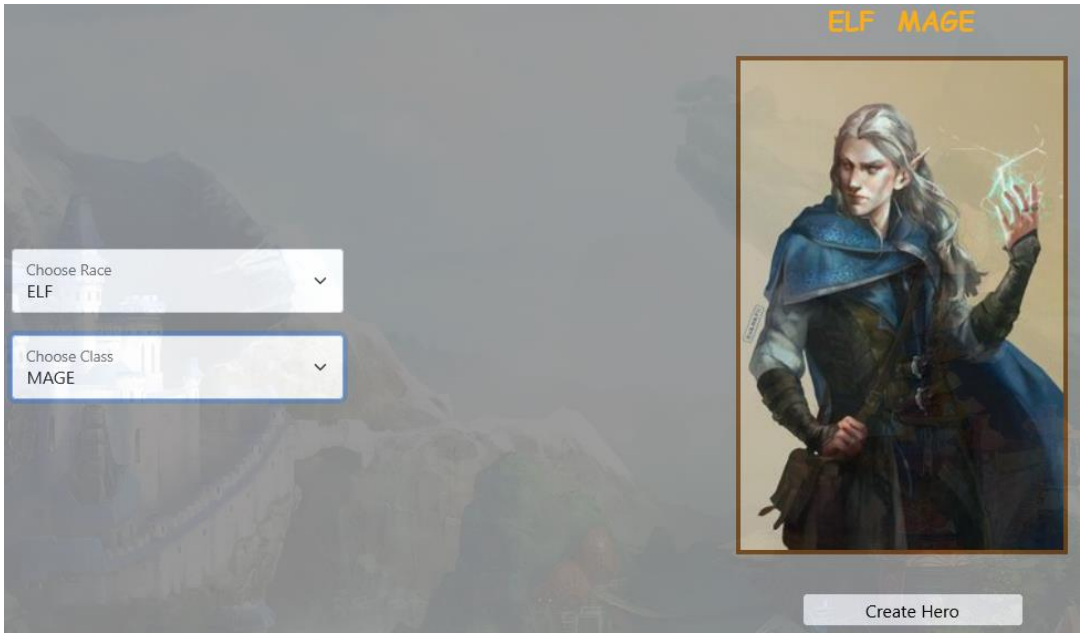
6.1 Aby rozpocząć grę należy założyć konto w serwisie, ważne jest to, że podając swój login ustawiamy automatycznie nazwy naszej postaci.



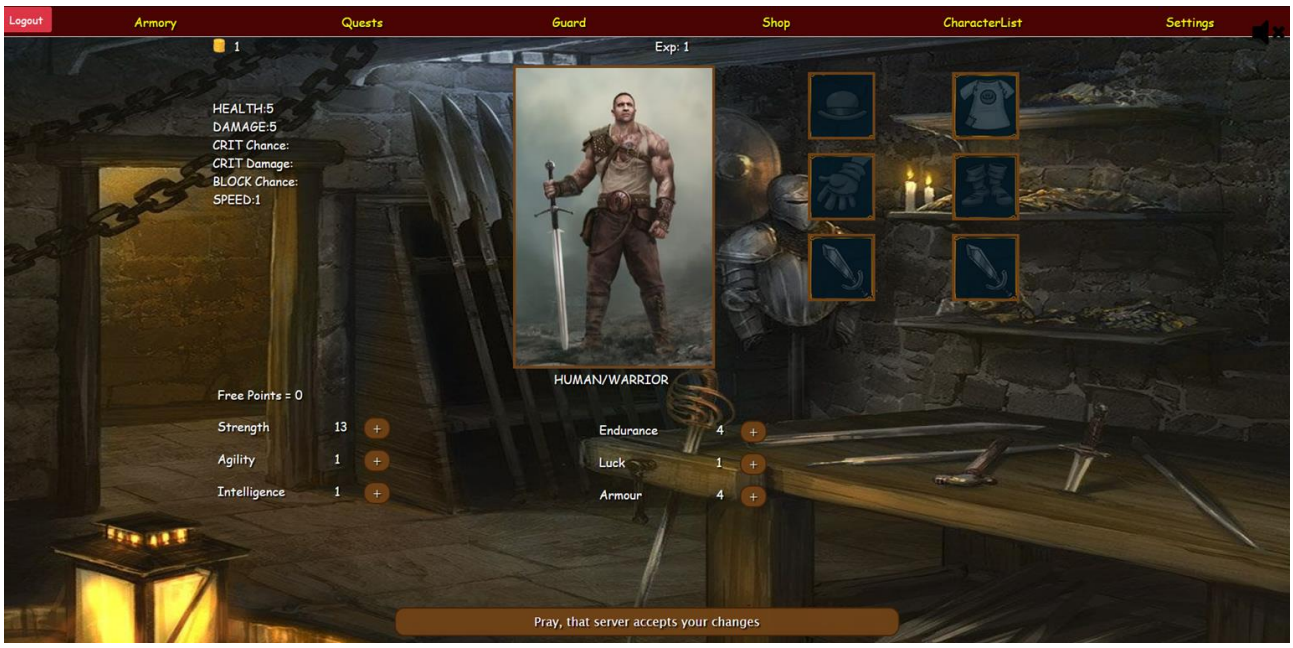
6.2 Następnym krokiem jest wybranie klasy i pochodzenia naszej postaci. Mamy do wyboru 5 ras oraz 3 klasy. Każda konfiguracja jest inna i zapewnia inne bonusy startowe. Poniżej kilka dostępnych postaci:





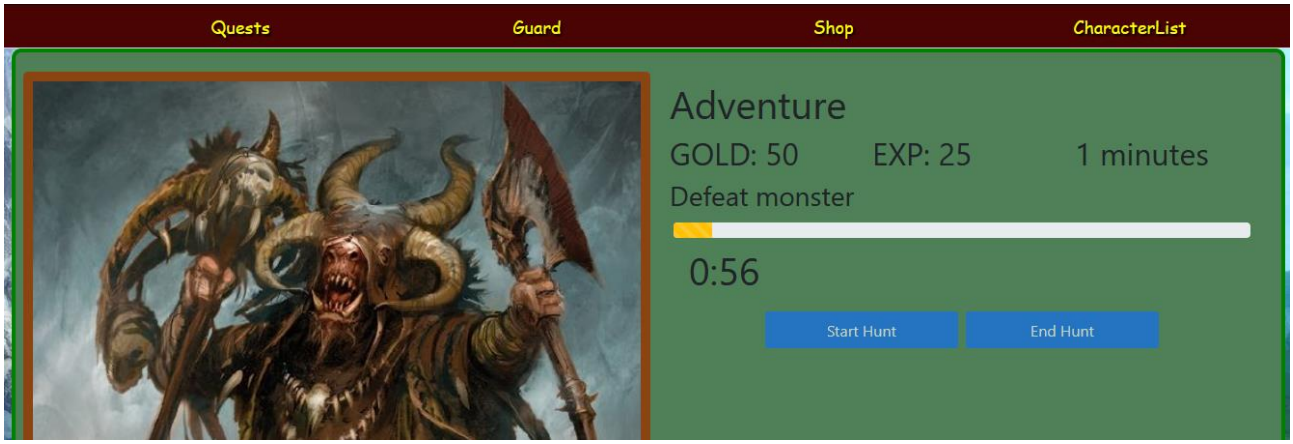


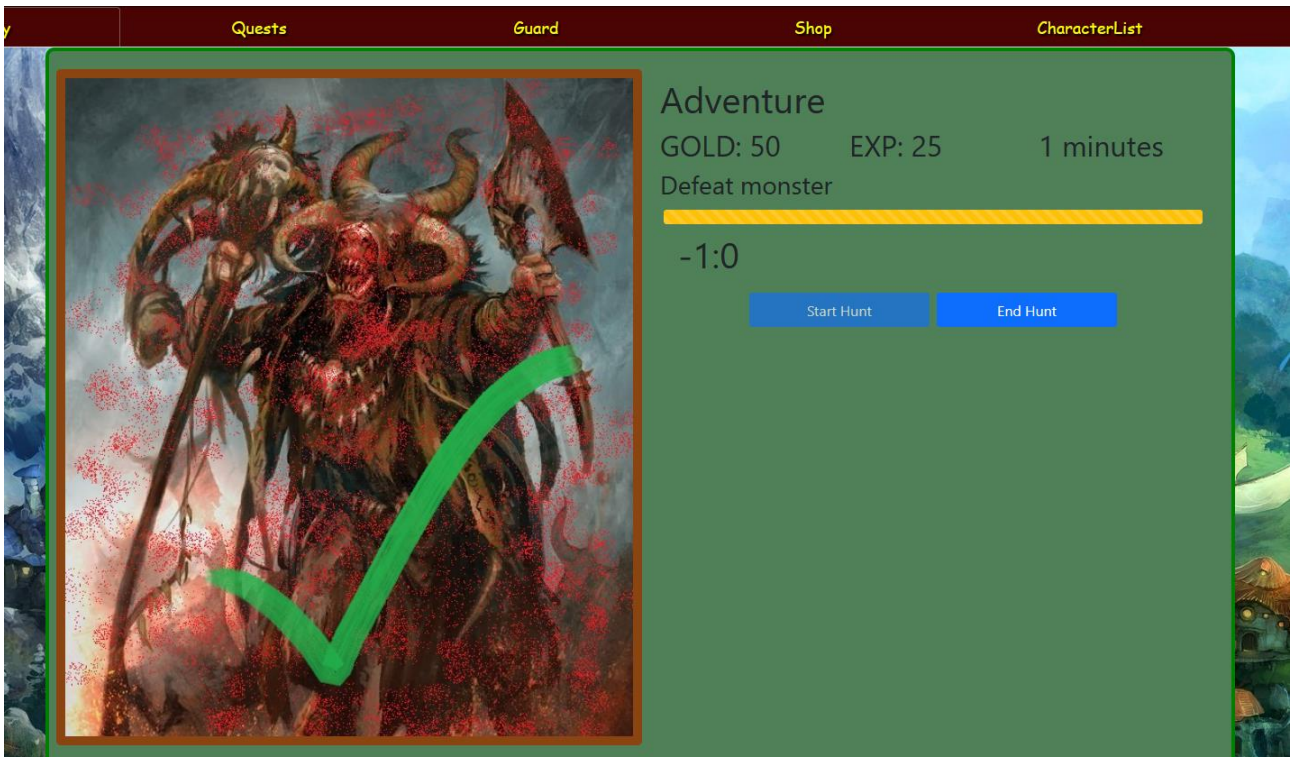
6.3 Po utworzeniu postaci użytkownik zostaje przekierowany do widoku zbrojowni. Z tego miejsca wydaje punkty rozwoju, aby ulepszać statystyki postaci. Widzi również swój obecny ekwipunek oraz statystyki dynamiczne, wyliczane na podstawie statystyk podstawowych.



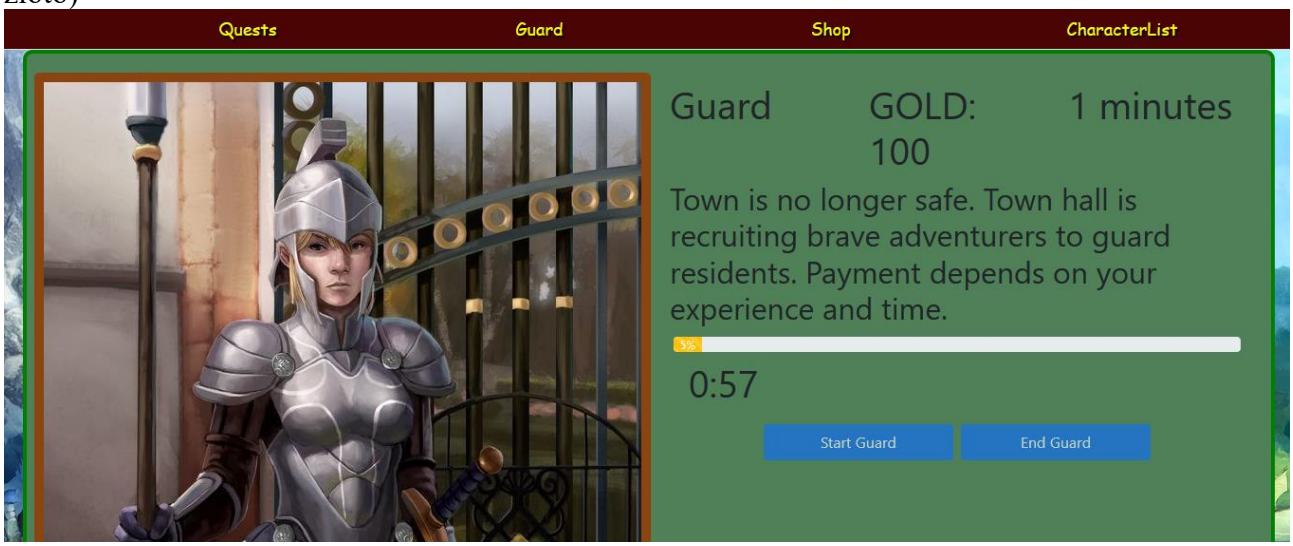
Zwiększenie statystyk następuje po zatwierdzeniu zmian przyciskiem na dole strony.

6.4 Przechodząc do zakładki nr.2 „Quests”, wyświetla nam się panel startu misji. Przed jej rozpoczęciem możemy zobaczyć ilość złota i doświadczenia przydzielanego po zakończeniu misji, a także czas który użytkownik musi poczekać, aby misję zakończyć. W tym czasie użytkownik może przejść do innej zakładki lub wylogować się w celu późniejszego powrotu do rozgrywki.

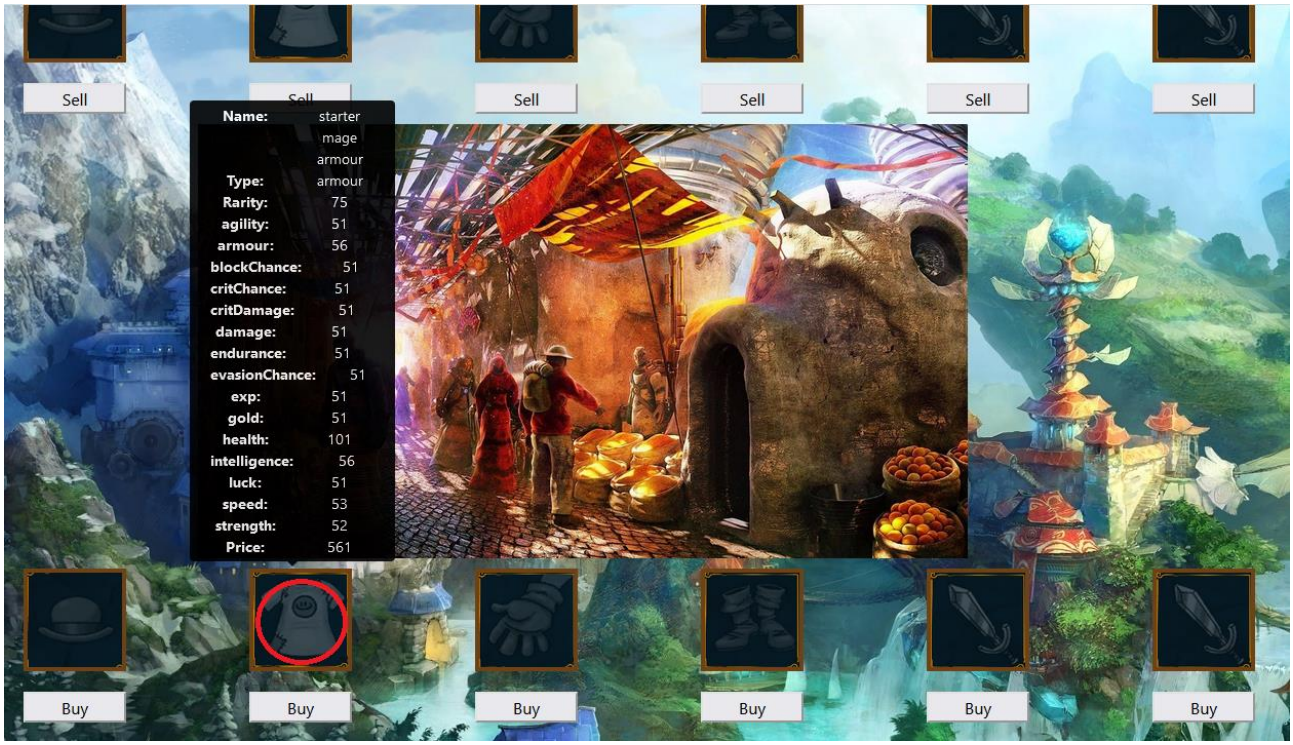




6.5 W podobny sposób gracz może wykonywać mniej wymagające warty (okresy czekania na złoto)



6.6 W zakładce sklep w rzędzie na górze widzimy swoje posiadane itemy, a w rzędzie na dole itemy które możemy zakupić za wcześniej zdobyte złoto. Item jest automatycznie podmieniany po naciśnięciu ‘Buy’ lub usuwany z ekwipunku po naciśnięciu ‘Sell’.



6.7 W zakładce „Ranking” użytkownik widzi listę postaci, które są posortowane według punktów doświadczenia.

Place	Username	Class	Race	Exp
1	HotMagnat69	Mage	elf_race	8798978
2	alfredo321	Mage	dark_elf_race	19887
3	masterPL	Mage	orc_race	1
4	ale	Archer	orc_race	0