

AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział Nauk Inżynierijnych
Informatyka

DOKUMENTACJA PROJEKTOWA INŻYNIERIA OPROGRAMOWANIA

Platforma dystrybucji cyfrowej "GameZone"

Autorzy:
Adamczyk Mateusz
Bazielich Tomasz
Cetera Maciej

Prowadzący:
mgr inż. Daniel Drozd

Nowy Sącz 2022

Spis treści

1. Opis projektu	3
2. Podobne aplikacje	4
3. Definicja wymagań funkcjonalnych i niefunkcjonalne	5
4. Identyfikacja problemów oraz proponowane rozwiązania	6
5. Dobór technologii.	7
6. Diagram przypadków użycia	8
7. Scenariusze przypadków użycia	9
8. Diagram ERD	17
9. Diagram Klas	18
10. Diagram Sekwencji	19
11. Diagram Aktywności	20
12. Wzorce projektowe	21
13. Testowanie	24
14. Bibliografia	38
Spis rysunków	40
Spis tabel	41

1. Opis projektu

Celem projektu jest stworzenie aplikacji internetowej - sklepu z grami komputerowymi. Do realizacji projektu wykorzystujemy framework do aplikacji internetowych Laravel w wersji 9.35.1 wraz z serwerem Apache oraz język programowania PHP. Platforma będzie przeznaczona przede wszystkim dla graczy. Każdy gracz będzie mógł kupować oraz kolekcjonować swoje ulubione tytuły gier na stronie profilu osobistego konta oraz wyszukiwać je w sklepie.

Administrator będzie miał dostęp do funkcji zarządzających aplikacją które umożliwiają mu m.in na dodawanie nowych produktów do sklepu oraz ich usuwanie, modyfikacja cen oraz kont użytkowników.

2. Podobne aplikacje

- Steam
- Epic games store
- GOG galaxy
- Origin
- Ubisoft Store
- Microsoft Store
- Battle.net
- G2A

Nasza strona połączy najistotniejsze funkcje wyżej wymienionych. Nasza aplikacja będzie posiadała przejrzysty i intuicyjny interfejs przyjazny dla użytkownika. Profil każdego użytkownika będzie posiadał swój poziom, który może wzrosnąć dzięki punktom, które można zdobyć przy zakupach gier. Coraz wyższe poziomy pozwolą użytkownikowi na odblokowanie kolejnych nagród do swojego profilu.

3. Definicja wymagań funkcjonalnych i niefunkcjonalne

Wymagania funkcjonalne:

- rejestracja
- logowanie
- kupowanie gry
- wyszukiwanie gier w sklepie
- dodanie środków do konta użytkownika
- poziom profilu
- lista życzeń informująca użytkownika o promocjach
- dodanie, usunięcie, modyfikowanie gry do sklepu (Admin)
- moderowanie kontami użytkowników (nałożenie blokady) (Admin)

Wymagania niefunkcjonalne:

- przejrzysty i intuicyjny interfejs
- szybkość działania aplikacji

4. Identyfikacja problemów oraz proponowane rozwiązania

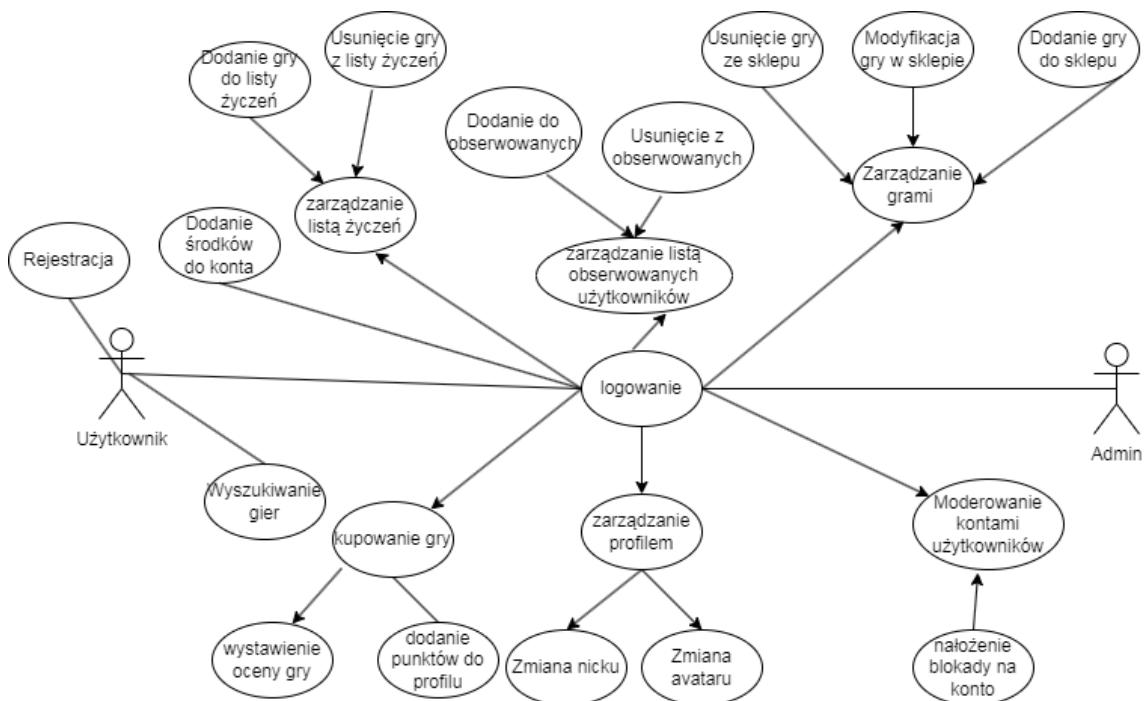
W naszym projekcie przewidujemy problemy podczas implementacji niektórych z naszych założeń. Najbardziej skomplikowanymi do wdrożenia funkcjonalnościami wydają się być lista obserwowanych profili oraz poziomy profilu.

Lista obserwowanych umożliwi na śledzenie kont innych użytkowników, ich poziomu, który będzie zależał od ilości posiadanych punktów.

5. Dobór technologii.

- Język PHP - framework Laravel
- Wygląd - Biblioteka Bootstrap z HTML oraz CSS
- Kompilator - Visual Studio Code
- Powłoka systemowa - Shell

6. Diagram przypadków użycia



Rys. 6.1. Diagram przypadków użycia

7. Scenariusze przypadków użycia

Tab. 7.1. Scenariusz 01.

Nazwa: Scenariusz 01 - Rejestracja
Aktor: Użytkownik
Warunki początkowe: -
Opis: Użytkownik wybiera opcje "Rejestracja"
Ścieżki główne: 1. Użytkownik kliką opcje "Rejestracja" 2. Wypełnia formularz rejestracyjny danymi 3. Kliką "Zarejestruj" 4. System sprawdza poprawność danych
Ścieżka alternatywna: 4a. Wyświetlenie komunikatu o niepoprawnych danych
Warunki końcowe: Użytkownik został pomyślnie zarejestrowany

Tab. 7.2. Scenariusz 02.

Nazwa: Scenariusz 02 - Logowanie
Aktor: Użytkownik, Admin
Warunki początkowe: Użytkownik musi być zarejestrowany
Opis: Użytkownik wybiera opcje "Logowanie"
Ścieżki główne: 1. Użytkownik wybiera opcje "Logowanie" 2. Wypełnia formularz logowania danymi 3. Kliką "Zaloguj" 4. System sprawdza poprawność danych
Ścieżka alternatywna: 4a. Wyświetlenie komunikatu o niepoprawnych danych
Warunki końcowe: Użytkownik został pomyślnie zalogowany

Tab. 7.3. Scenariusz 03.

Nazwa: Scenariusz 03 - Wyszukiwanie gier
Aktor: Użytkownik
Warunki początkowe: -
Opis: Użytkownik wybiera opcje "Wyszukaj"
Ścieżki główne:
1. Użytkownik wpisuje nazwę gry
2. System sprawdza czy wyszukiwana fraza znajduje się w bazie danych
Ścieżka alternatywna:
3. Wyświetlenie komunikatu o nie znalezieniu wyszukiwanej frazy
Warunki końcowe: System wyświetla wyniki wyszukiwania

Tab. 7.4. Scenariusz 04.

Nazwa: Scenariusz 04 - Dodanie środków do konta
Aktor: Użytkownik
Warunki początkowe: Użytkownik musi być zalogowany
Opis: Użytkownik wybiera opcję "Dodaj środki"
Ścieżki główne:
1. Użytkownik kliką opcję "Dodaj środki"
2. Określa kwotę którą chce doładować
3. Wybiera sposób płatności
4. Użytkownik wybiera "Zatwierdź transakcję"
5. System sprawdza wykonanie transakcji
Ścieżka alternatywna:
5a. System odrzuca wykonanie transakcji
Warunki końcowe: Środki zostały przelane na konto użytkownika

Tab. 7.5. Scenariusz 05.

Nazwa: Scenariusz 05 - Dodanie gry do listy życzeń
Aktor: Użytkownik
Warunki początkowe: Użytkownik musi być zalogowany oraz wyszukał grę
Opis: Użytkownik dodaje grę do listy życzeń
Ścieżki główne:
1. System sprawdza, czy dana gra jest już w liście życzeń
Ścieżka alternatywna:
2. Jeśli dana gra jest już w liście życzeń gra nie zostanie dodana ponownie
Warunki końcowe: Gra została dodana do listy życzeń

Tab. 7.6. Scenariusz 06.

Nazwa: Scenariusz 06 - Kupowanie gry
Aktor: Użytkownik
Warunki początkowe: Użytkownik musi być zalogowany, wybrana gra znajduje się w koszyku, użytkownik posiada środki na koncie
Opis: Użytkownik wybiera opcje "Dokonaj zakupu"
Ścieżki główne:
1. Użytkownik dokonuje zakupu wybranej gry.
2. System sprawdza dostępne środki na koncie.
3. System pobiera odpowiednią sumę z konta.
4. System przypisuje zakupioną grę do konta użytkownika.
5. System przyznaje odpowiednią ilość punktów za zakup gry.
Ścieżka alternatywna:
1a. Użytkownik nie posiada gry w koszyku.
2a. Użytkownik nie posiada odpowiednich środków na koncie.
Warunki końcowe:
Gra została pomyślnie zakupiona
Punkty za zakup gry zostały dodane do konta

Tab. 7.7. Scenariusz 07.

Nazwa: Scenariusz 07 - Usunięcie gry z listy życzeń
Aktor: Użytkownik
Warunki początkowe: Użytkownik jest zalogowany oraz gra jest na liście życzeń
Opis: Użytkownik usuwa grę z listy życzeń
Ścieżki główne:
1. Użytkownik przy wybranej grze z listy kliką "usuń grę z listy życzeń"
2. System usuwa grę z listy życzeń
Ścieżka alternatywna: -
Warunki końcowe: Gra została usunięta z listy życzeń

Tab. 7.8. Scenariusz 08.

Nazwa: Scenariusz 08 - Wystawienie oceny gry
Aktor: Użytkownik
Warunki początkowe: Użytkownik posiada ocenianą grę
Opis: Użytkownik wybiera opcję "Oceń grę"
Ścieżki główne:
1. Użytkownik wybiera stopień oceny od 1 do 5
2. System przypisuje ocenę użytkownika grze
Ścieżka alternatywna: -
Warunki końcowe: Gra została oceniona przez użytkownika

Tab. 7.9. Scenariusz 09.

Nazwa: Scenariusz 09 - Zmiana nicku
Aktor: Użytkownik
Warunki początkowe: Użytkownik jest zalogowany
Opis: Użytkownik wybiera opcję "Zmień nazwę profilu"
Ścieżki główne: 1. Użytkownik wybiera opcję "Zarządzaj profilem" 2. W panelu zarządzania użytkownik wypisuje nową nazwę profilu 3. System zmienia nazwę na podaną
Ścieżka alternatywna: - 2a. Użytkownik podał tą samą nazwę lub nieobsługiwane znaki
Warunki końcowe: Nazwa profilu została pomyślnie zmieniona na nową

Tab. 7.10. Scenariusz 10.

Nazwa: Scenariusz 10 - Zmiana avataru
Aktor: Użytkownik
Warunki początkowe: Użytkownik jest zalogowany
Opis: Użytkownik wybiera opcję "Zmień ikonę profilu"
Ścieżki główne: 1. Użytkownik wybiera opcję "Zarządzaj profilem" 2. W panelu zarządzania użytkownik wybiera nową ikonę profilu 3. System zmienia ikonę na wybraną
Ścieżka alternatywna: -
Warunki końcowe: Ikona profilu została pomyślnie zmieniona.

Tab. 7.11. Scenariusz 11.

Nazwa: Scenariusz 11 - Dodanie gry do sklepu
Aktor: Admin
Warunki początkowe: Admin jest zalogowany
Opis: Admin dodaje grę do sklepu
Ścieżki główne: 1. Admin wybiera opcję "Zarządzaj grami" 2. Admin wybiera opcję "Dodaj grę do sklepu" 3. Admin wypełnia formularz danymi 4. Admin wybiera "Zatwierdź" 5. System sprawdza poprawność danych
Ścieżka alternatywna: 6. Jeśli dane są niepoprawne lub gra jest już w sklepie, gra nie zostanie dodana i wyświetli się stosowny komunikat
Warunki końcowe: Gra została dodana do sklepu

Tab. 7.12. Scenariusz 12.

Nazwa: Scenariusz 12 - Nałożenie blokady na konto
Aktor: Admin
Warunki początkowe: Admin jest zalogowany
Opis: Admin nakłada nakłada blokade na danego użytkownika
Ścieżki główne: 1. Admin wyszukuje danego użytkownika 2. Admin wybiera opcję "Zablokuj użytkownika" 3. Admin określa długość trwania blokady i zatwierdza wybór 4. System nakłada blokadę na konto użytkownika
Ścieżka alternatywna: 2a. Użytkownik zablokowany permanentnie nie może być ponownie zablokowany
Warunki końcowe: Na użytkownika zostaje nałożona blokada

Tab. 7.13. Scenariusz 13.

Nazwa: Scenariusz 13 - Zdjęcie blokady z konta
Aktor: Admin
Warunki początkowe: Admin jest zalogowany, dany użytkownik jest zablokowany
Opis: Admin usuwa blokadę z danego użytkownika
Ścieżki główne: 1. Admin wyszukuje danego użytkownika 2. Admin wybiera opcję "Odblokuj użytkownika" 3. System zdejmuje blokadę z konta użytkownika
Ścieżka alternatywna: -
Warunki końcowe: Z Użytkownika zostaje zdjęta blokada

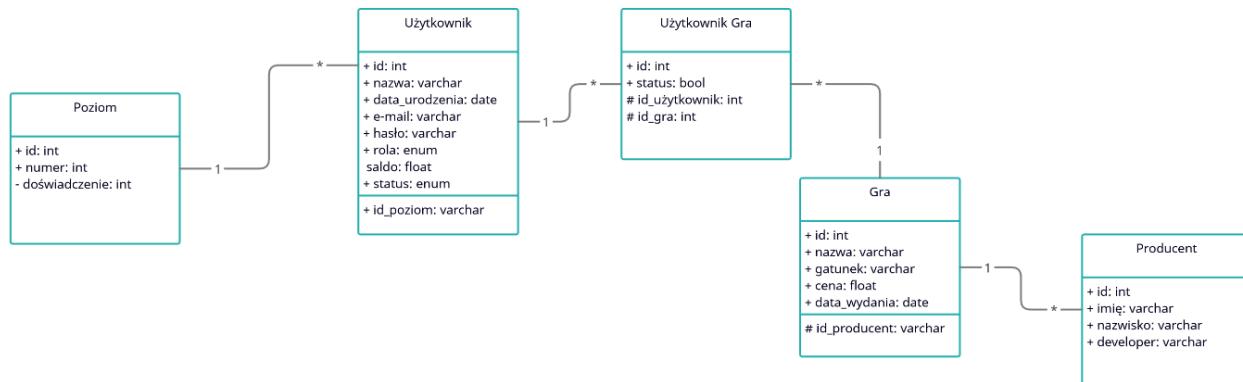
Tab. 7.14. Scenariusz 14.

Nazwa: Scenariusz 17 - Usunięcie gry ze sklepu
Aktor: Admin
Warunki początkowe: Admin jest zalogowany oraz gra jest w sklepie
Opis: Admin usuwa grę ze sklepu
Ścieżki główne: 1. Admin wybiera opcję "Zarządzaj grami" 2. Admin wybiera opcję "Usuń grę do sklepu" 3. System usuwa grę ze sklepu
Ścieżka alternatywna: -
Warunki końcowe: Gra została usunięta ze sklepu

Tab. 7.15. Scenariusz 15.

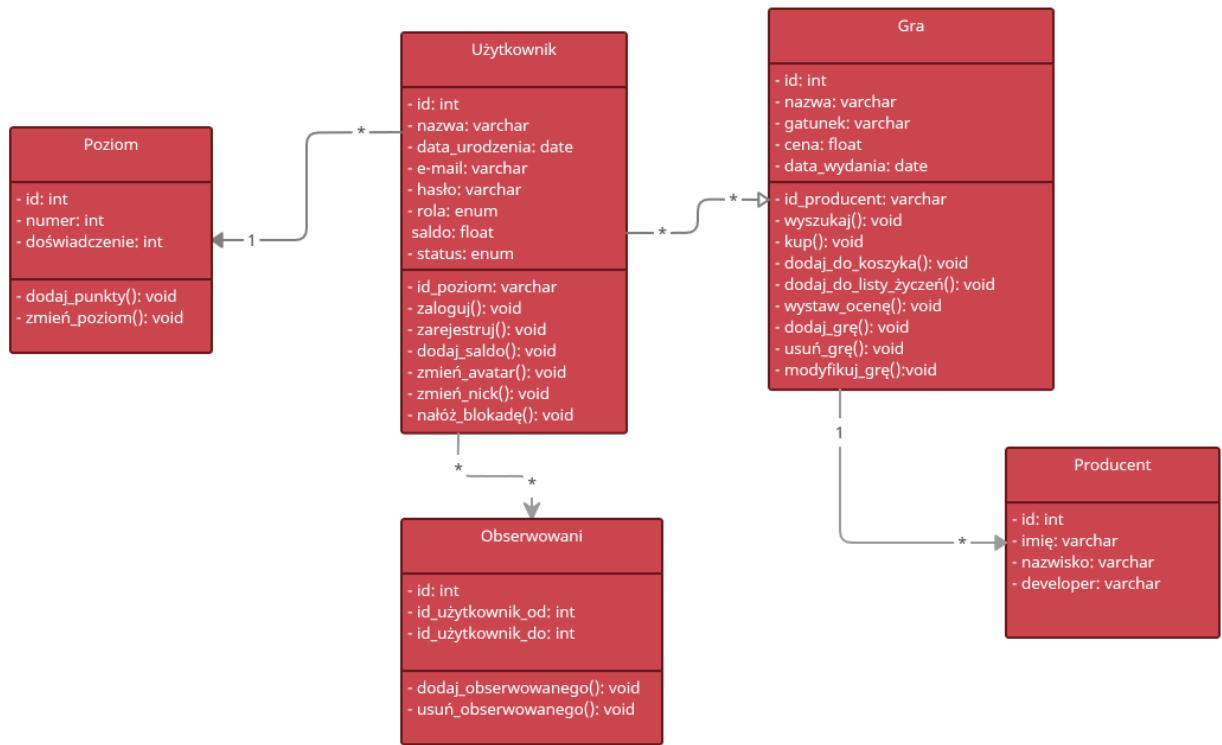
Nazwa: Scenariusz 18 - Modyfikacja gry ze sklepu
Aktor: Admin
Warunki początkowe: Admin jest zalogowany oraz gra jest w sklepie
Opis: Admin modyfikuje grę ze sklepu
Ścieżki główne: 1. Admin wybiera opcję "Zarządzaj grami" 2. Admin wybiera opcję "Modyfikuj grę" 3. Admin wypełnia formularz danymi 4. Admin wybiera "Zatwierdź" 5. System sprawdza poprawność danych
Ścieżka alternatywna: 6. Jeśli dane są niepoprawne lub gra jest już w sklepie, gra nie zostanie dodana i wyświetli się stosowny komunikat
Warunki końcowe: Gra została zmodyfikowana w sklepie

8. Diagram ERD



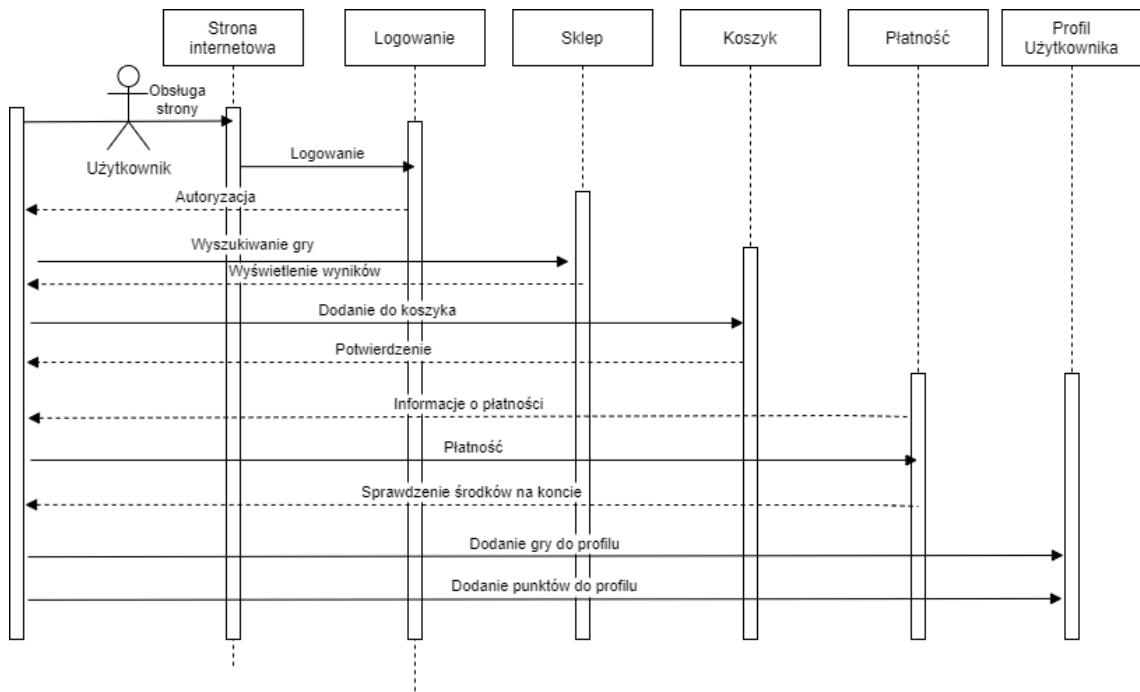
Rys. 8.1. Diagram ERD

9. Diagram Klas



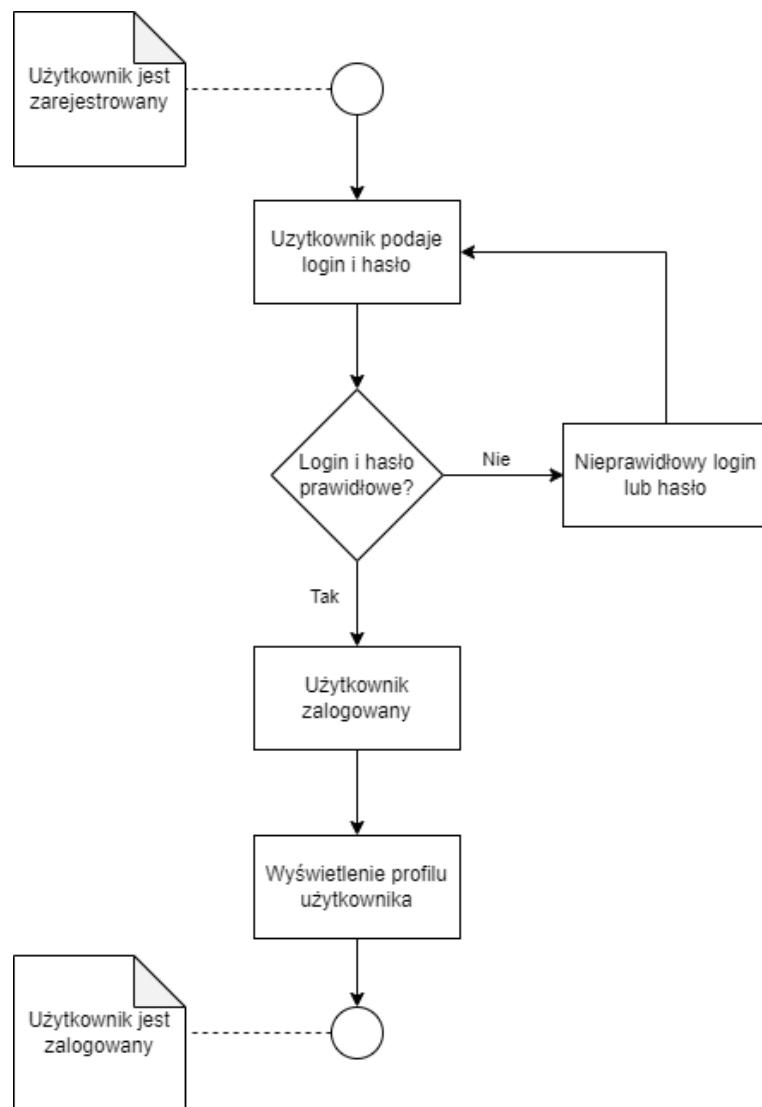
Rys. 9.1. Diagram klas

10. Diagram Sekwencji



Rys. 10.1. Diagram sekwencji

11. Diagram Aktywności



Rys. 11.1. Diagram aktywnosci

12. Wzorce projektowe

1. Metoda szablonowa

Jej zadaniem jest zdefiniowanie metody będącej szkieletem algorytmu. Jest to wzorzec projektowy który pozwala na nadpisanie pewnych etapów algorytmu bez konieczności zmiany jego struktury. Metoda szablonowa jest stosowana w przypadkach gdy chcemy pozwolić klientom na rozszerzanie niektórych etapów algorytmu bez dokonywania zmian jego całości. Wzorzec jest również przydatny w sytuacjach gdy kilka klas posiada prawie identyczne algorytmy

2. Budowniczy

Budowniczy jest to kreacyjny wzorzec projektowy, który umożliwia na budowanie zaawansowanych obiektów wg etapów. Tworzone Obiekty mogą być różnego typu oraz nie mają dostępu do innych obiektów przed ich ukończeniem. Główną zaletą jest możliwość ponownego użycia stworzonej struktury do tworzenia innych przedmiotów, wiąże się to jednak ze znacznym stopniem skomplikowania kodu.

3. Adapter

Adapter umożliwia współpracę ze sobą elementów obiektów, aby mogły ze sobą współpracować, lecz pojedynczo są one niekompatybilne. Pozwala on konwertować interfejsy w taki sposób że obiekty się rozumią. Adapter nie tylko potrafi konwertować formaty ale pozwala na współpracę obiektów. Możliwe jest nawet stworzenie adaptera dwukierunkowego, który konwertuje dane w obu kierunkach. Analogią to prawdziwego życia jest przejściówka między wtyczką europejską a brytyjską.

4. Singleton

Singleton to wzorzec kreacyjny, ale często określa się go jako antywzorzec ponieważ poprzez ograniczenie możliwości tworzenia obiektów tylko do pojedynczej instancji, czyli do tworzenia wielu obiektów w jednym wykonaniu kodu, łamie się zasady obiektowego projektowania. Zaletą takiego rozwiązania

jest pewność, że dany obiekt zostanie stworzony po raz pierwszy wtedy, kiedy rzeczywiście jest on potrzebny. Wadą są wspomniane złamane zasady projektowania obiektowego.

5. Odwiedzający

Odwiedzający jest behawioralnym wzorcem projektowym. Umożliwia on na dokonanie rozdzielenia algorytmów od obiektów, które je wykorzystują w swoim działaniu. Pozwala to na odpowiednie dopasowanie działań do typu obiektu poprzez “odwiedzającego”, który zbiera informacje o danym obiekcie. Minusem takiego rozwiązania jest brak dostępu do niektórych obiektów, np. obiektów prywatnych.

6. Dekorator

Dekorator jest to strukturalny wzorzec projektowy pozwalający dodawać nowe zadania obiektom przy pomocy innych obiektów które zawierają odpowiednie zachowania. Pozwala on na przypisanie dodatkowych obowiązków obiektom w trakcie działania programu bez niszczenia kodu z którego korzystają obiekty. Jest to wzorzec projektowy znany również pod nazwą “nakładka” oznacza to że może być połączony z innym docelowym obiektem. Nakładka zawiera ten sam zestaw metod jak element docelowy i przydziela mu odpowiednio otrzymywane żądania.

7. Fasada

Fasada umożliwia uproszczenie interfejsu ze złożonych klas, bibliotek oraz frameworku. Fasada to klasa stanowiąca prosty interfejs dla złożonego podsystemu, zawierającego mnóstwo ruchomych części. Fasada ogranicza funkcjonalność, ale za to eksponuje tylko te elementy, których klient naprawdę potrzebuje. Stworzenie fasady jest wygodnym sposobem integracji swojej aplikacji ze skomplikowaną biblioteką posiadającą wiele funkcji, gdy potrzebujemy tylko wąskiego zakresu jej funkcji.

Struktura fasady

Fasada zbudowana jest na zasadzie klasy, może wystąpić również dodatkowa, pomocnicza klasa odciążającą główną część fasady. Fasada jest głównie wykorzystywana przez klienta, który korzystając z fasady nie wywołuje bezpośrednio obiektów podsystemu.

Wewnątrz fasady znajduje się złożony podsystem składający się z wielu różnych obiektów, fasada ustala hierarchię ich wykonywania tak, aby ich inicjalizacja była wykonywana w odpowiedniej kolejności oraz dba o zachowanie danych w odpowiednim formacie.

W naszym projekcie autoryzacja podczas logowania i rejestracji korzysta ze wzorca fasady.

8. Kompozyt

Jest to strukturalny wzorzec projektowy który pozwala na dzielenie obiektów w struktury drzewiaste i traktowanie ich jako osobne obiekty. Jest stosowany tylko w przypadku gdy główny model aplikacji może być przedstawiony w formie drzewa. Kompozyt określa dwa typy elementów które współdzielą ten sam interfejs: proste liście i złożone kontenery. Kontener może składać się z liści oraz z innych kontenerów co pozwala na skonstruowanie zagnieżdżonej, rekurencyjnej struktury obiektów przypominającej drzewo.

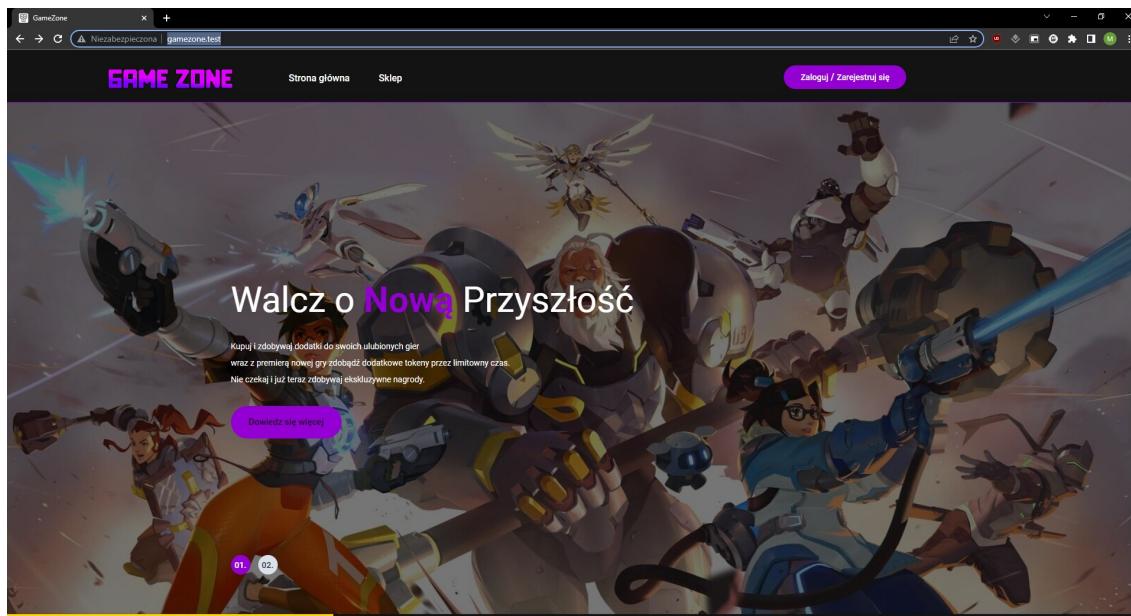
9. Obserwator

Obserwator jest behawioralnym wzorcem projektowym, który umożliwia na pośrednictwie innych obiektów o wydarzeniach mających miejsce w innym, obserwowanym obiekcie. Po każdej zmianie w monitorowanym obiekcie do innych obiektów zostają wysyłane powiadomienia zawierające informacje o zmianach w obiekcie. Wadą jest powiadamianie "subskrybentów" w nieokreślonej, przypadkowej kolejności.

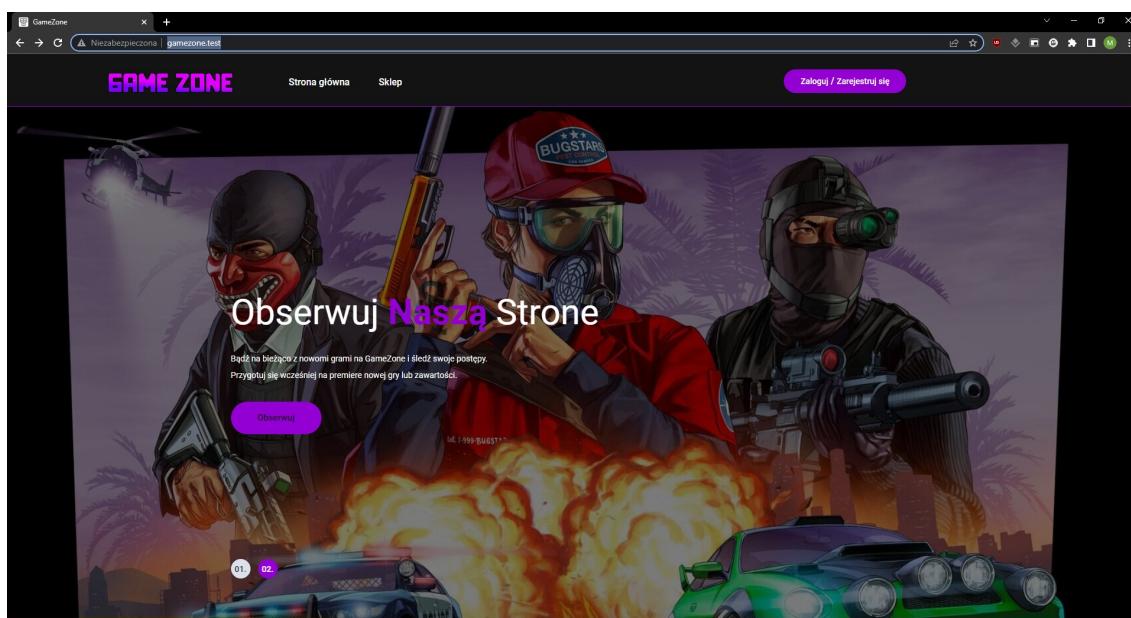
13. Testowanie

1. Strona główna

Aby wejść na stronę główną projektu użytkownik musi wpisać w pasku adresu przeglądarki: "gamezone.test".



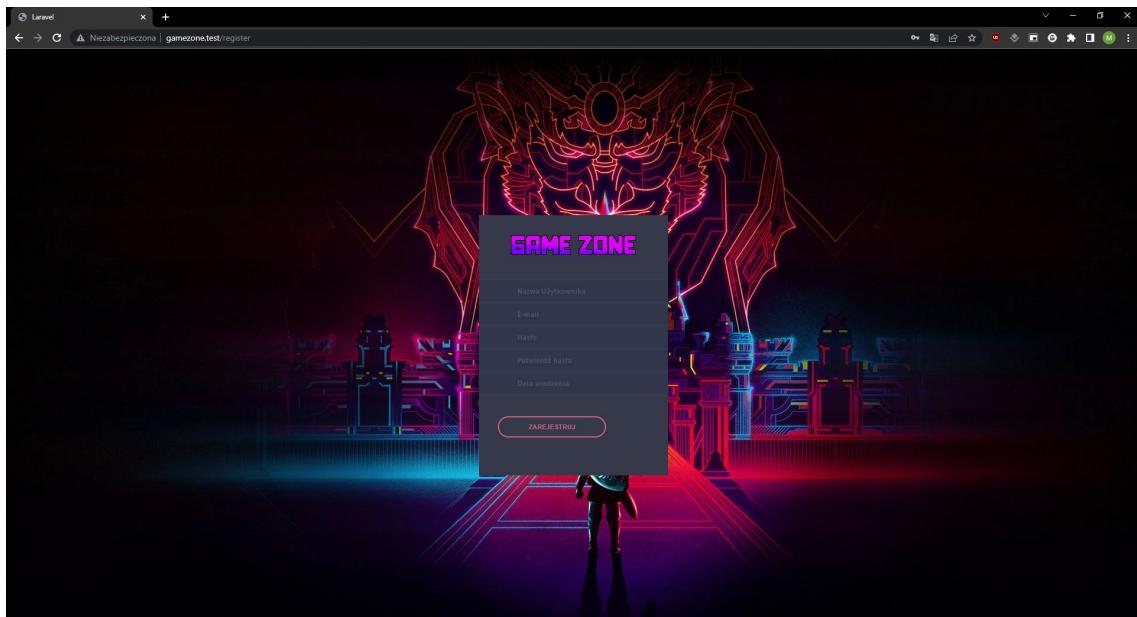
Rys. 13.1. Strona główna



Rys. 13.2. Strona główna

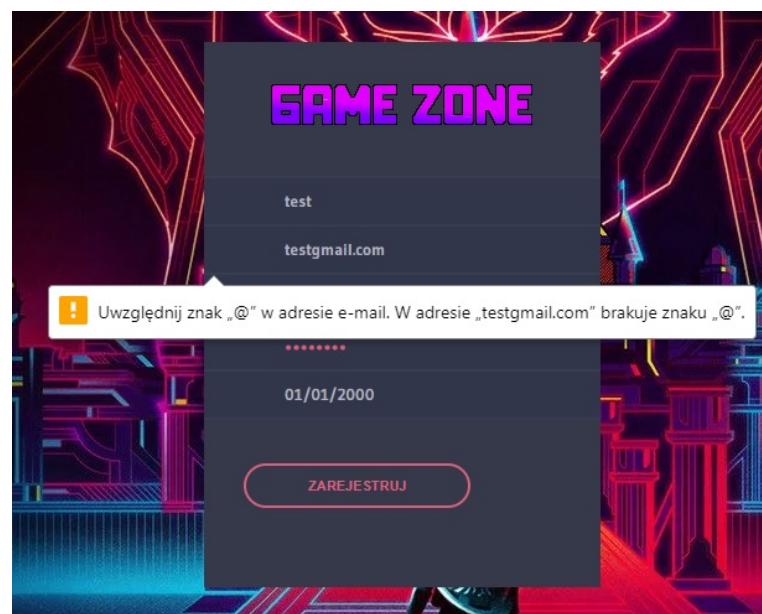
2. Rejestracja

Aby dokonać rejestracji na stronie użytkownik musi kliknąć w przycisk "Zarejestruj się", następnie wypełnić formularz rejestracji.



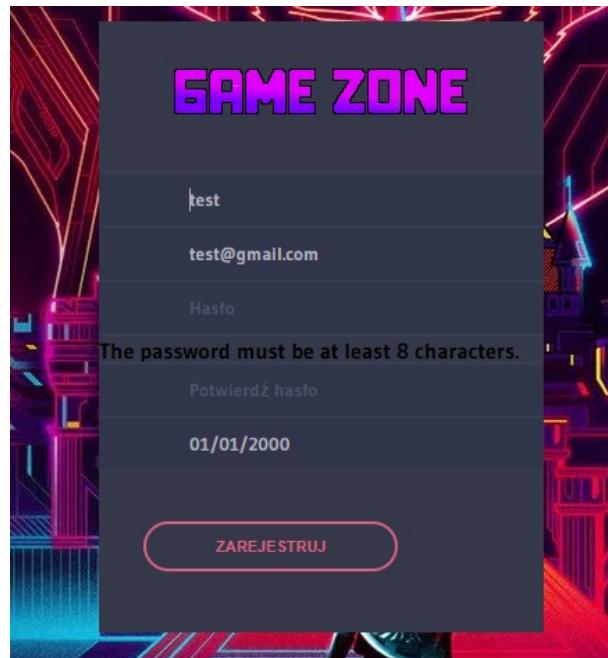
Rys. 13.3. Rejestracja

Pole "E-mail" musi zawierać znak "@".



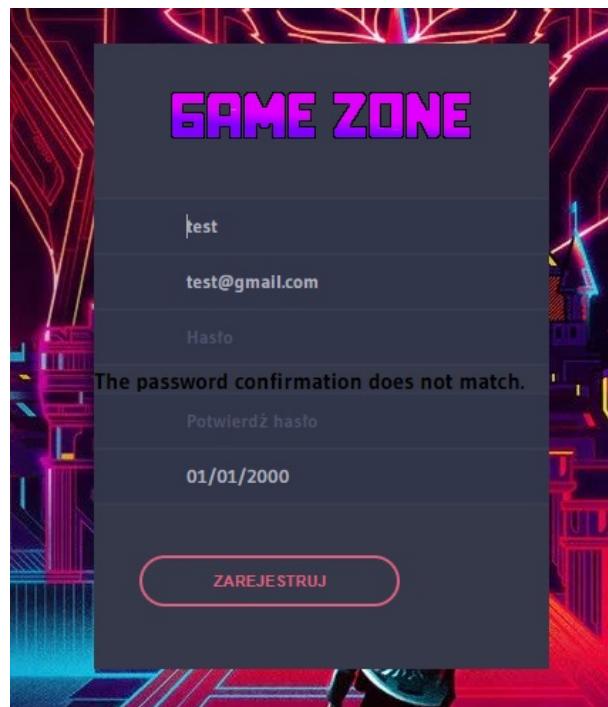
Rys. 13.4. Rejestracja validacja e-mail

Pole ”Hasło” musi zawierać przynajmniej 8 znaków.



Rys. 13.5. Rejestracja walidacja hasło

Pole ”Hasło” oraz ”Potwierdź hasło” muszą zawierać ten sam tekst.



Rys. 13.6. Rejestracja walidacja potwierdź hasło

Po naciśnięciu w pole ”Data urodzenia” wyskakuje okienko z wyborem daty urodzenia.



Rys. 13.7. Rejestracja data urodzenia

Data urodzenia jest zamieniana w modelu na odpowiedni format.

```
49     public function setBirthDateAttribute($value)
50     {
51         $this->attributes['birth_date'] = Carbon::createFromFormat('m/d/Y', $value)->format('Y-m-d');
52     }
53
54     public function getBirthDateAttribute()
55     {
56         return Carbon::createFromFormat('Y-m-d', $this->attributes['birth_date'])->format('m/d/Y');
57     }
58
59     public function games()
60     {
61         return $this->hasMany(Game::class);
62     }
63 }
```

Rys. 13.8. Rejestracja data urodzenia - format

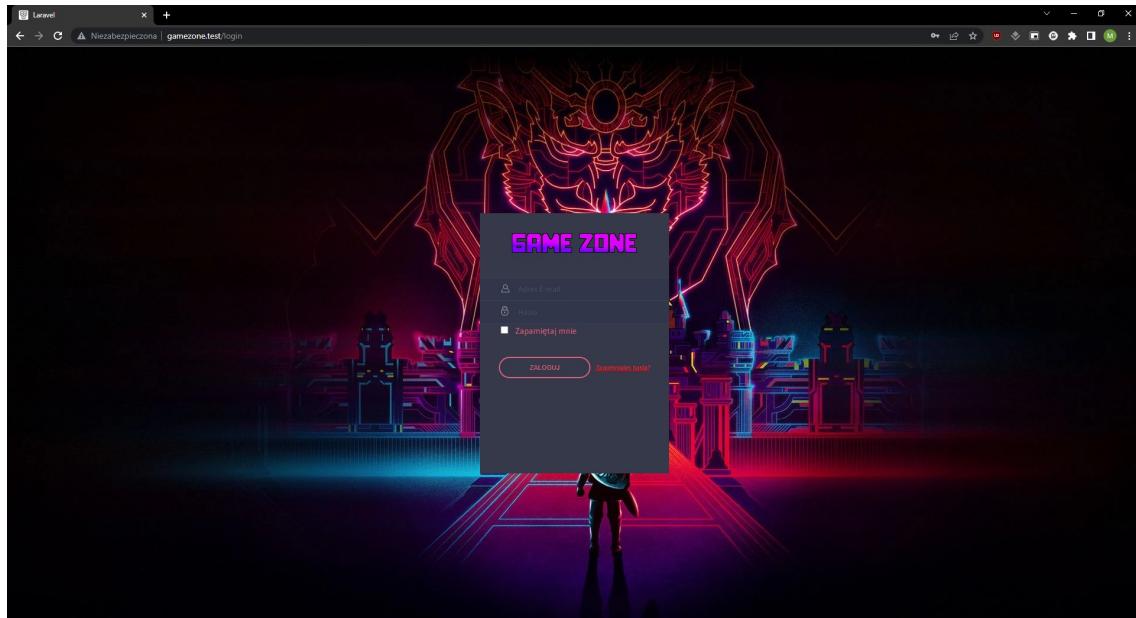
Po udanej walidacji danych dane użytkownika zostają dodane do bazy danych

<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	1 admin	admin@gmail.com	2000-01-01	0
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	2 user	user@gmail.com	2001-02-11	0
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	3 user2	user2@gmail.com	1999-07-13	100
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	4 user3	user3@gmail.com	1996-12-27	500
<input checked="" type="checkbox"/>	Edytuj	Kopiuj	Usuń	5 test	test@gmail.com	2000-01-20	0

Rys. 13.9. Rejestracja - wpis w bazie danych

3. Logowanie

Aby zalogować się na stronę użytkownik musi kliknąć w przycisk "Zaloguj", następnie wypełnić formularz logowania.



Rys. 13.10. Logowanie

Aby walidacja danych przebiegła pomyślnie pola z loginem oraz z hasłem muszą zgadzać się z wartościami znajdującymi się w bazie danych. Hasło w bazie danych jest przechowywane w zahaszowanej postaci.



Rys. 13.11. Logowanie - hasło

Za walidację danych odpowiadają kontrolery z grupy Auth - tutaj kontroler walidacji logowania i przekierowania do odpowiedniego panelu w zależności od typu użytkownika.

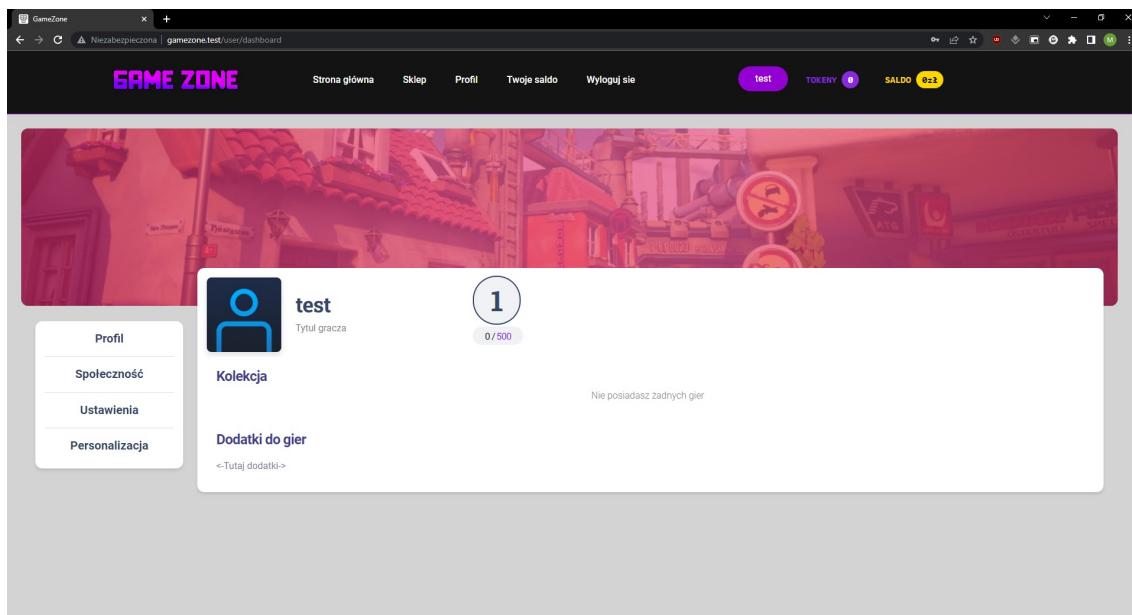


The screenshot shows a code editor with a Laravel application structure. The current file is `LoginController.php` located at `app/Http/Controllers/Auth`. The code implements a login controller with methods for handling login requests and redirecting users based on their role.

```
file Edit Selection View Go Run Terminal Help
... gameone
EXPLORER GAMEZONE ...
app Http Controllers Auth > LoginController.php
22 use AuthenticatedUsers;
23
24 /**
25 * Where to redirect users after login,
26 *
27 * @var string
28 */
29 protected $redirectTo = RouteServiceProvider::HOME;
30
31 protected function redirectTo(){
32     if ($this->auth()->user()->role == 1){
33         return route('admin.dashboard');
34     }
35     elseif ($this->auth()->user()->role == 2){
36         return route('user.dashboard');
37     }
38 }
39
40 /**
41 * Create a new controller instance.
42 *
43 * @return void
44 */
45 public function __construct()
46 {
47     $this->middleware('guest')->except('logout');
48 }
49
50 public function login(Request $request){
51     $input = $request->all();
52     $this->validate($input, [
53         'email'=>required|email,
54         'password'=>required
55     ]);
56
57     if($auth()=>attempt(['email'=>$input['email'], 'password'=>$input['password']])) {
58         if($auth()->user()->role == 1) {
59             return redirect()->route('admin.dashboard');
60         }
61         elseif ($auth()->user()->role == 2) {
62             return redirect()->route('user.dashboard');
63         }
64     }
65     else{
66         return redirect()->route('login')->with('error', 'Email and password are wrong');
67     }
68 }
69 }
```

Rys. 13.12. Logowanie - walidacja

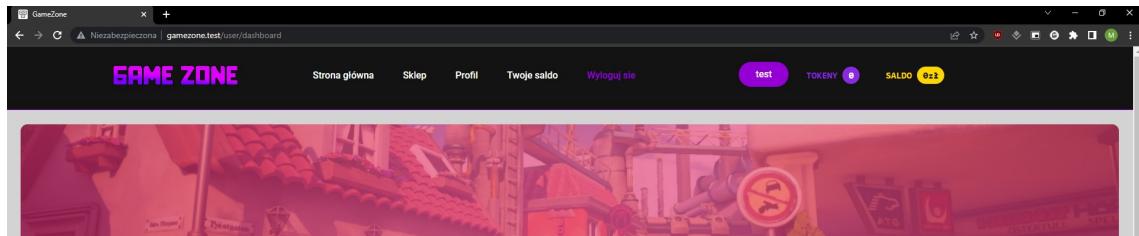
Po udanym logowaniu użytkownik zostaje przekierowany na swój panel użytkownika



Rys. 13.13. Logowanie - panel użytkownika

4. Wylogowanie

Aby wylogować się użytkownik musi kliknąć w przycisk "Wyloguj się" na górnym pasku aplikacji. Przycisk jest dostępny jedynie wtedy, jeśli użytkownik jest zalogowany.



Rys. 13.14. Wylogowanie

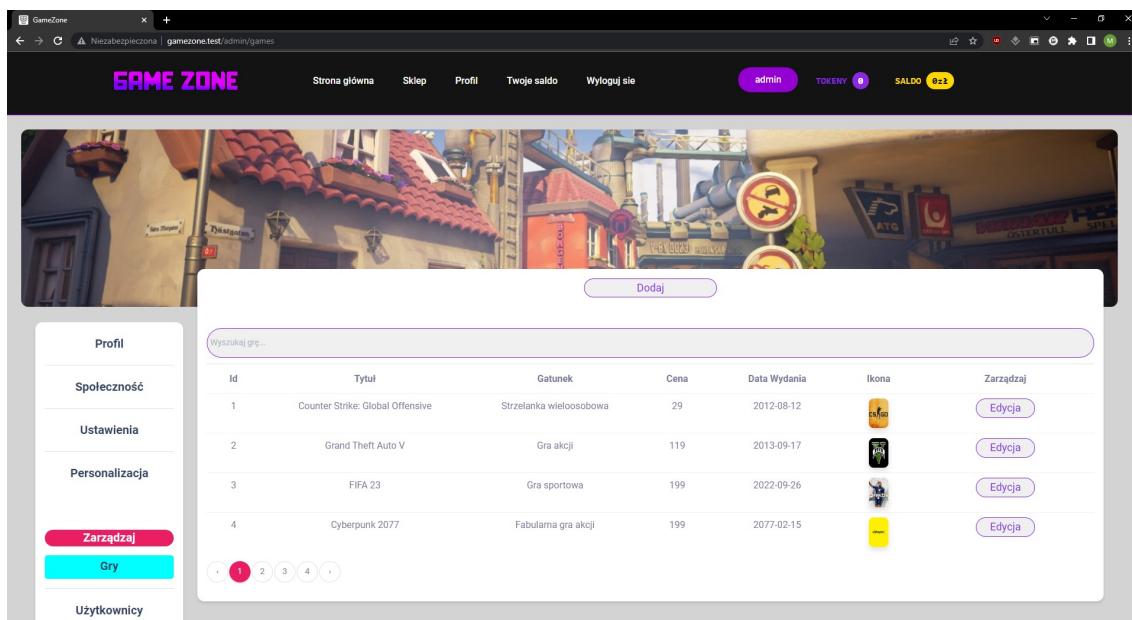
Po wylogowaniu się użytkownik zostaje przeniesiony na stronę główną. Jeśli w tym momencie kliknie w przycisk powrotu na pasku przeglądarki nie zostanie przeniesiony na swój panel, lecz aplikacja poprosi o ponowne zalogowanie z panelu logowania. Za to zabezpieczenie odpowiada odpowiedni kontroler.

```
PreventBackHistory.php
app > Http > Middleware > PreventBackHistory.php
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7
8 class PreventBackHistory
9 {
10     /**
11      * Handle an incoming request.
12      *
13      * @param \Illuminate\Http\Request $request
14      * @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next
15      * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
16     */
17     public function handle(Request $request, Closure $next)
18     {
19         $response = $next($request);
20         return $response->header('Cache-Control','nocache,no-store,max-age=0,must-revalidate')
21             ->header('Pragma','no-cache')
22             ->header('Expires','Sun, 02 Jan 1990 00:00:00 GMT');
23     }
24 }
25
```

Rys. 13.15. Wylogowanie - zabezpieczenie

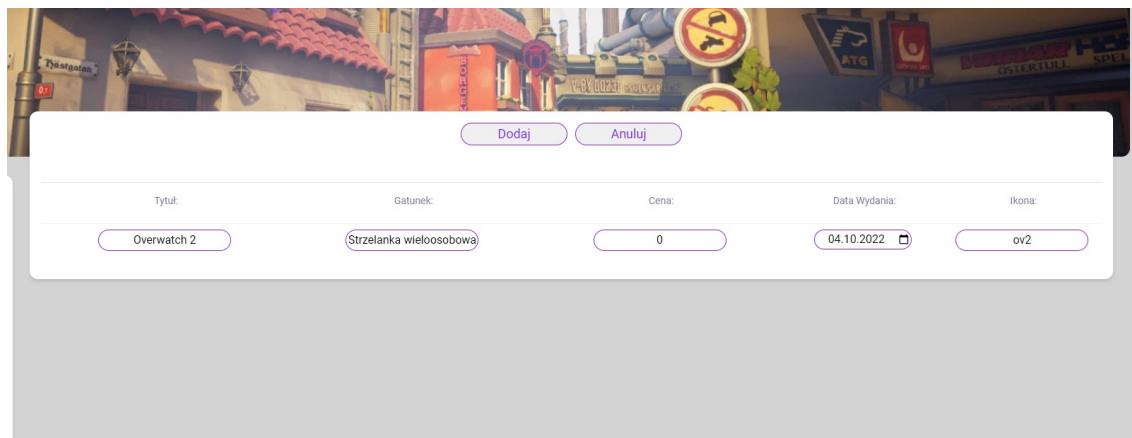
5. Dodanie gry do sklepu

Grę do sklepu może dodać tylko admin. Robi on to ze zbudowanego w tym celu panelu.



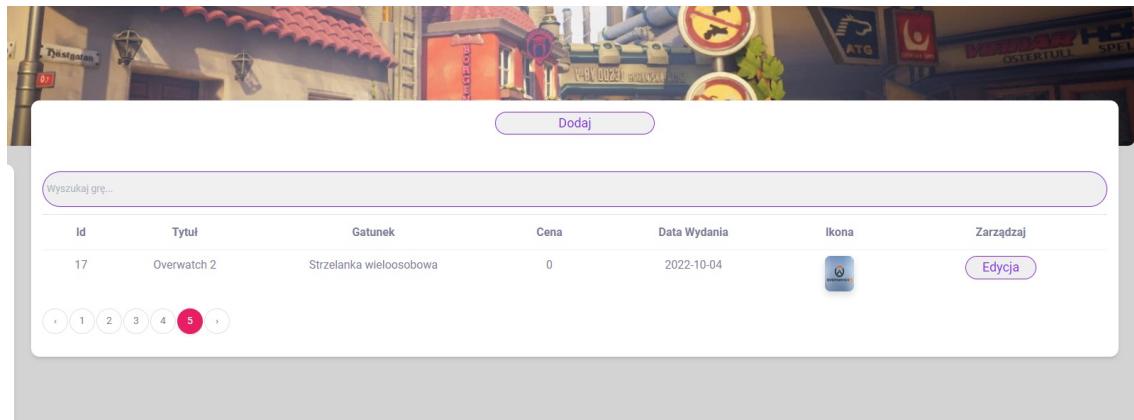
Rys. 13.16. Dodanie gry

Aby dodać grę admin musi kliknąć w przycisk "Dodaj". Zostanie on wtedy przeniesiony do panelu z widokiem formularza dodania gry.



Rys. 13.17. Dodanie gry - formularz

Po pomyślnej walidacji danych gra zostaje dodana do sklepu.



Rys. 13.18. Dodanie gry - po dodaniu

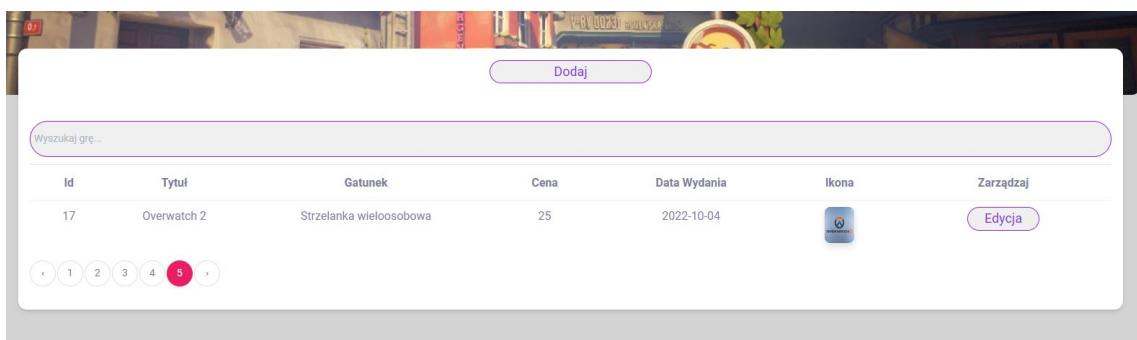
6. Edycja gry ze sklepu

Aby edytować grę admin musi kliknąć w przycisk "Edycja" przy odpowiedniej grze, którą chce zedytować. Zostanie on wtedy przeniesiony do panelu z widokiem formularza edytowania gry.



Rys. 13.19. Edytowanie gry

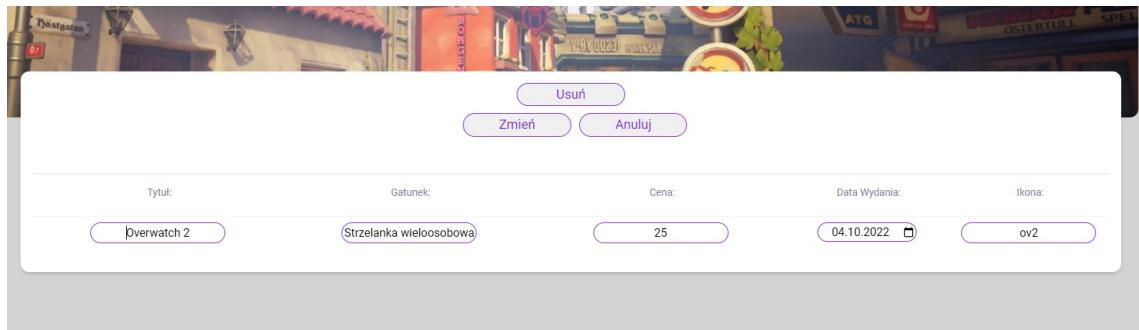
Po pomyślnej walidacji danych widzimy, że cena zmieniła się z "0" na wprowadzoną przez nas wartość "25".



Rys. 13.20. Edytowanie gry - efekt

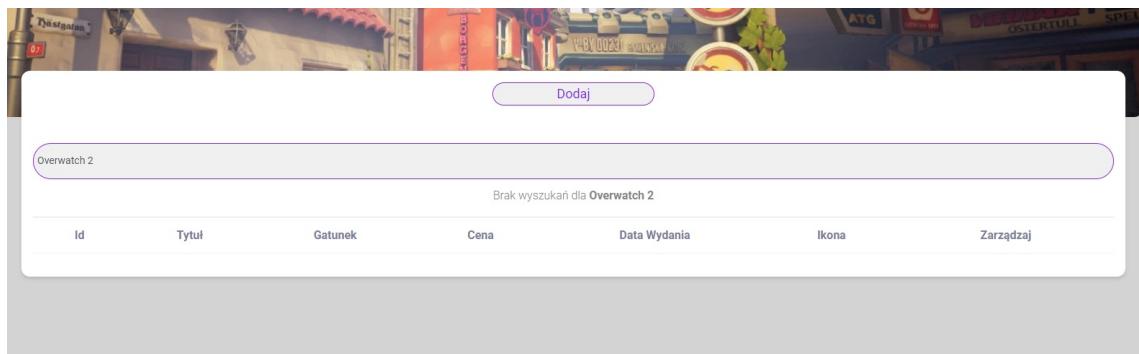
7. Usuwanie gry ze sklepu

Aby usunąć grę admin musi kliknąć w przycisk "Edycja" przy odpowiedniej grze, którą chce usunąć. Zostanie on wtedy przeniesiony do panelu z widokiem formularza edytowania gry gdzie może usunąć grę przyciskiem "Usuń".



Rys. 13.21. Usuwanie gry

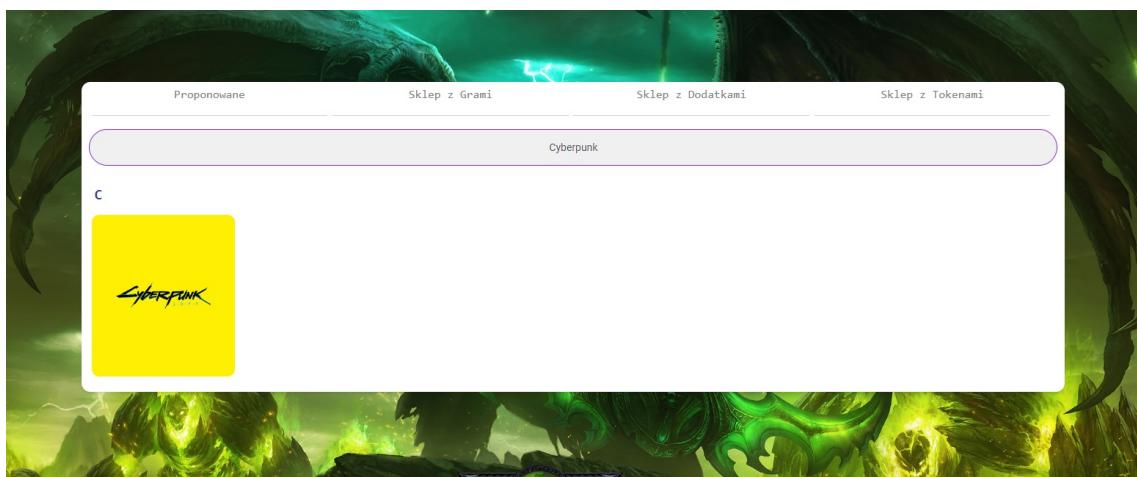
Po pomyślnej walidacji danych widzimy, że gra została usunięta.



Rys. 13.22. Usuwanie gry - efekt

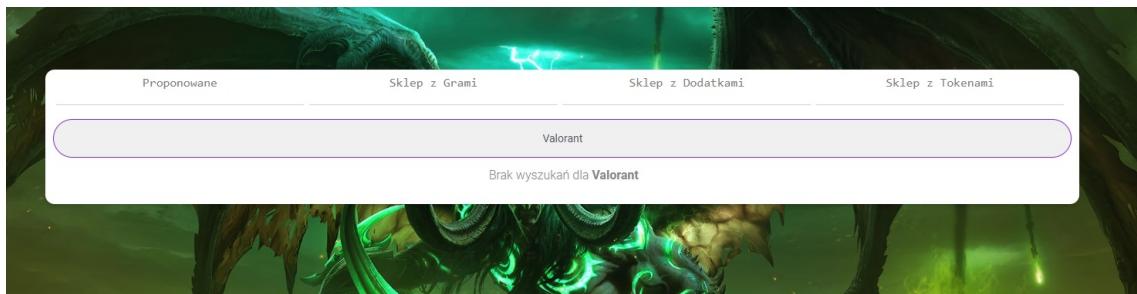
8. Wyszukiwanie gry w sklepie

Aby wyszukać grę użytkownik musi wejść w sklep, następnie w "Sklep z grami" oraz wyszukać grę wpisując jej tytuł.



Rys. 13.23. Wyszukiwanie gry

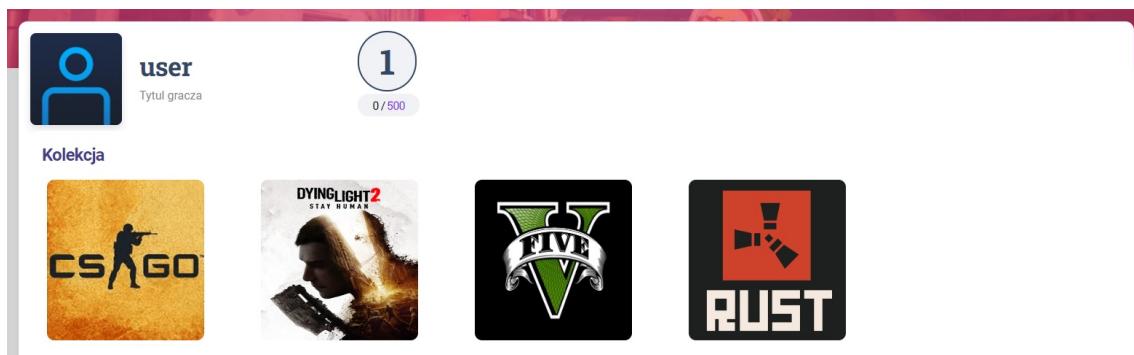
Jeśli podana fraza nie zostanie wyszukana wyświetli się stosowny komunikat o braku znalezienia gry.



Rys. 13.24. Wyszukiwanie gry - brak znalezienia

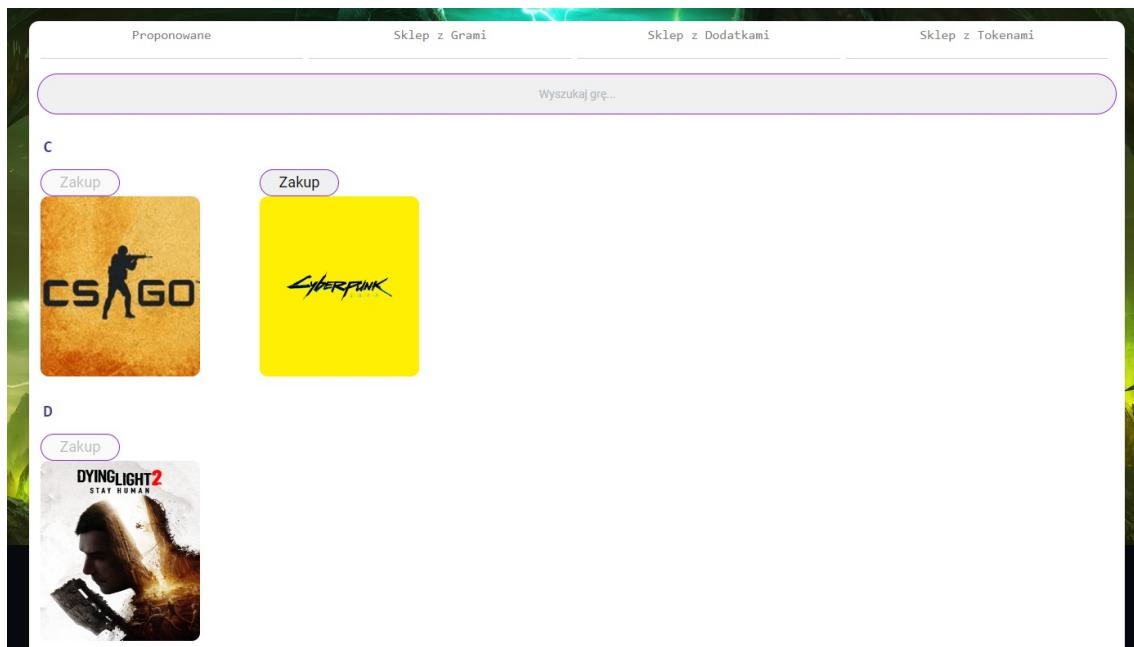
9. Kupno gry

Obecnie posiadane gry użytkownika.



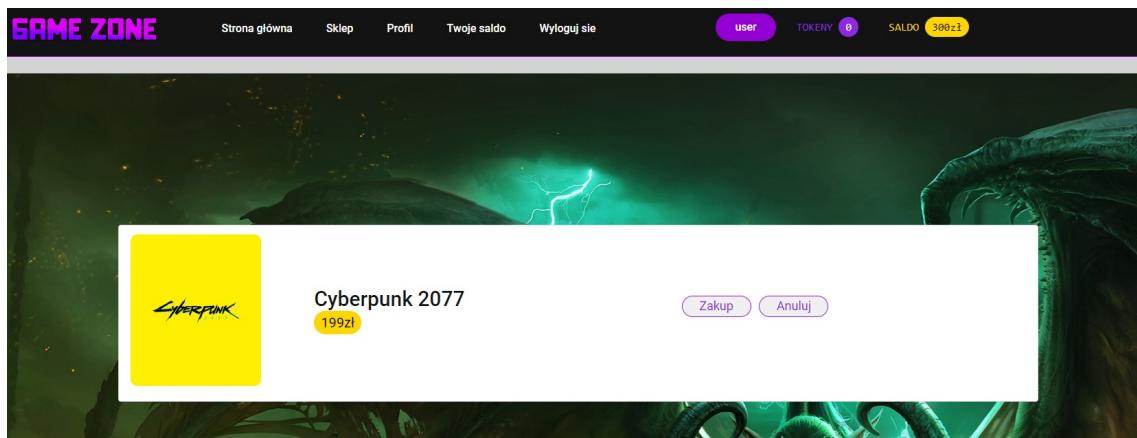
Rys. 13.25. Kupno gry - przed

Aby kupić grę użytkownik musi wejść w sklep, następnie w "Sklep z grami". Jeśli gra jest już w jego posiadaniu to przycisk "Zakup" będzie wyłączony.



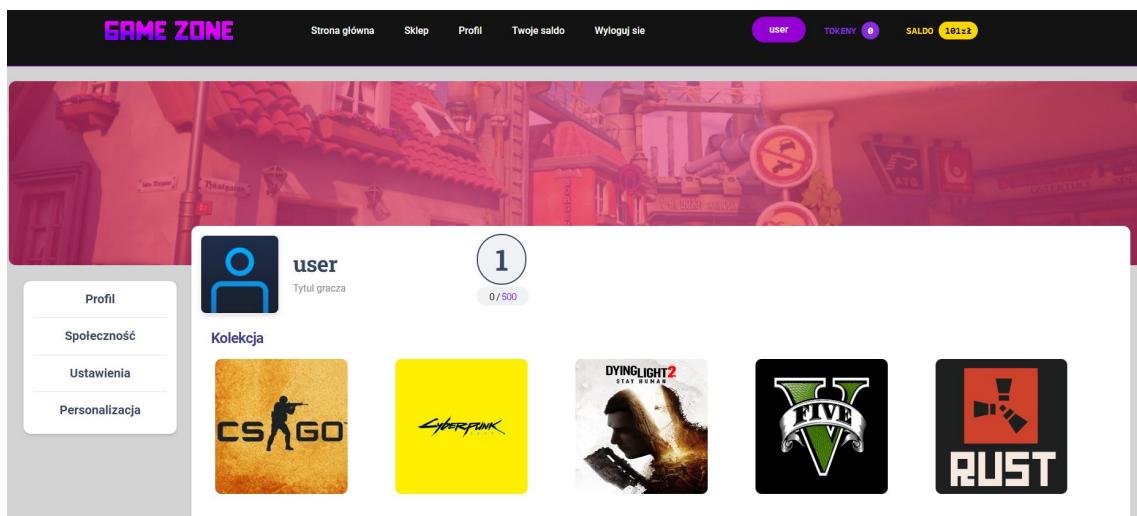
Rys. 13.26. Kupno gry - sklep

Saldo użytkownika jest większe niż cena gry, zakup zatem jest możliwy.



Rys. 13.27. Kupno gry - kupno

Widzimy, że użytkownik pomyślnie nabył nową grę. Saldo użytkownika zostało również pomniejszone o cenę zakupionej gry.



Rys. 13.28. Kupno gry - po

14. Bibliografia

- [1] Dokumentacja Laravel <https://laravel.com/docs/9.x>
- [2] Szablon strony <https://colorlib.com/>
- [3] Grafika https://www.creocommunity.pl/wp-content/uploads/2022/10/1664947809_4_Witamy-
- [4] Grafika <https://brief.pl/wp-content/uploads/2022/03/Overwatch-2-750x430.jpg>
- [5] Grafika https://i.blogs.es/15d106/gta-online/1366_2000.webp

Spis rysunków

6.1.	Diagram przypadków użycia	8
8.1.	Diagram ERD	17
9.1.	Diagram klas	18
10.1.	Diagram sekwencji	19
11.1.	Diagram aktywnosci	20
13.1.	Strona główna	24
13.2.	Strona główna	24
13.3.	Rejestracja	25
13.4.	Rejestracja validacja e-mail	25
13.5.	Rejestracja validacja hasło	26
13.6.	Rejestracja validacja potwierdź hasło	26
13.7.	Rejestracja data urodzenia	27
13.8.	Rejestracja data urodzenia - format	27
13.9.	Rejestracja - wpis w bazie danych	27
13.10	Logowanie	28
13.11	Logowanie - hasło	28
13.12	Logowanie - walidacja	29
13.13	Logowanie - panel użytkownika	29
13.14	Wylogowanie	30
13.15	Wylogowanie - zabezpieczenie	30
13.16	Dodanie gry	31
13.17	Dodanie gry - formularz	31
13.18	Dodanie gry - po dodaniu	32
13.19	Edytowanie gry	33
13.20	Edytowanie gry - efekt	33
13.21	Usuwanie gry	34
13.22	Usuwanie gry - efekt	34
13.23	Wyszukiwanie gry	35
13.24	Wyszukiwanie gry - brak znalezienia	35
13.25	Kupno gry - przed	36
13.26	Kupno gry - sklep	36

13.27Kupno gry - kupno	37
13.28Kupno gry - po	37

Spis tabel

7.1. Scenariusz 01	9
7.2. Scenariusz 02	9
7.3. Scenariusz 03	10
7.4. Scenariusz 04	10
7.5. Scenariusz 05	11
7.6. Scenariusz 06	11
7.7. Scenariusz 07	12
7.8. Scenariusz 08	12
7.9. Scenariusz 09	13
7.10. Scenariusz 10	13
7.11. Scenariusz 11	14
7.12. Scenariusz 12	14
7.13. Scenariusz 13	15
7.14. Scenariusz 14	15
7.15. Scenariusz 15	16