

Serwis łączący graczy „Omni-kom”

Krzysztof Lipski s20901

Wstępny opis projektu

Dziedzina problemowa

Projekt portalu z grami został wymyślony przez grupę znajomych, chcących umożliwić sobie i graczom większej interakcji między sobą w kwestii grania oraz możliwość posiadania biblioteki gier, gdzie będą mogli śledzić postępy.

Cel

Celem systemu jest umożliwienie graczom interakcji między sobą, dokonywanie zakupów wybranych gier oraz śledzenie postępów w grze.

Zakres odpowiedzialności systemu

System powinien umożliwiać zarządzanie informacjami:

- Graczy
- Historii transakcji
- Katalogu gier
- Oceny gier

System powinien umożliwiać

1. Graczom
 - Dokonywanie zakupów gier
 - Kontaktowanie się z innymi graczami
 - Tworzenie czatów oraz zarządzanie nimi
 - Dodawanie znajomych do konta
 - Śledzenie postępów w swoich grach
 - Zdobywanie osiągnięć
 - Dostosowywanie swojego profilu

2. Użytkownikom

- Przeglądanie katalogu sklepu
- Wyszukiwanie pozycji sklepu

3. Gościom

- Przeglądanie katalogu
- Zakładanie konta

Użytkownicy systemu

Potencjalnymi użytkownikami systemu są gracze (członkowie systemu), niezarejestrowani użytkownicy (goście) oraz podsystem czasu.

Wymagania użytkownika

System miałby zastosowanie w ułatwieniu graczom dokonywania zakupów gier komputerowych, konsolowych i nie tylko. Użytkownicy mają dostęp do biblioteki gier, które posiadają. Mogą też zobaczyć szczegóły konkretnej gry, gdzie mają opcje na uruchomienie gry lub zmiana parametrów uruchomienia, np. dodatki do niej.

Gracz dla każdej gry może zdobyć osiągnięcia. Każde z nich posiada takie właściwości jak "rzadkość" - czyli ile procent graczy już posiada to osiągnięcie, status odblokowania, tytuł oraz identyfikator.

System umożliwia również zakup gier dostępnych z katalogu, gdzie każda gra posiada identyfikator, tytuł, opis, cenę, PEGI, producenta i tagi. Gry dzielą się na gatunki, które ułatwiają przeglądanie i filtrowanie katalogu.

Informacje o transakcjach są przechowywane w historii transakcji. Każda jest zapisywana w historii transakcji z następującymi danymi o identyfikatorze, nazwa i zapłacona kwota. Można zarówno obejrzeć historię jak i szczegóły poszczegółnej transakcji.

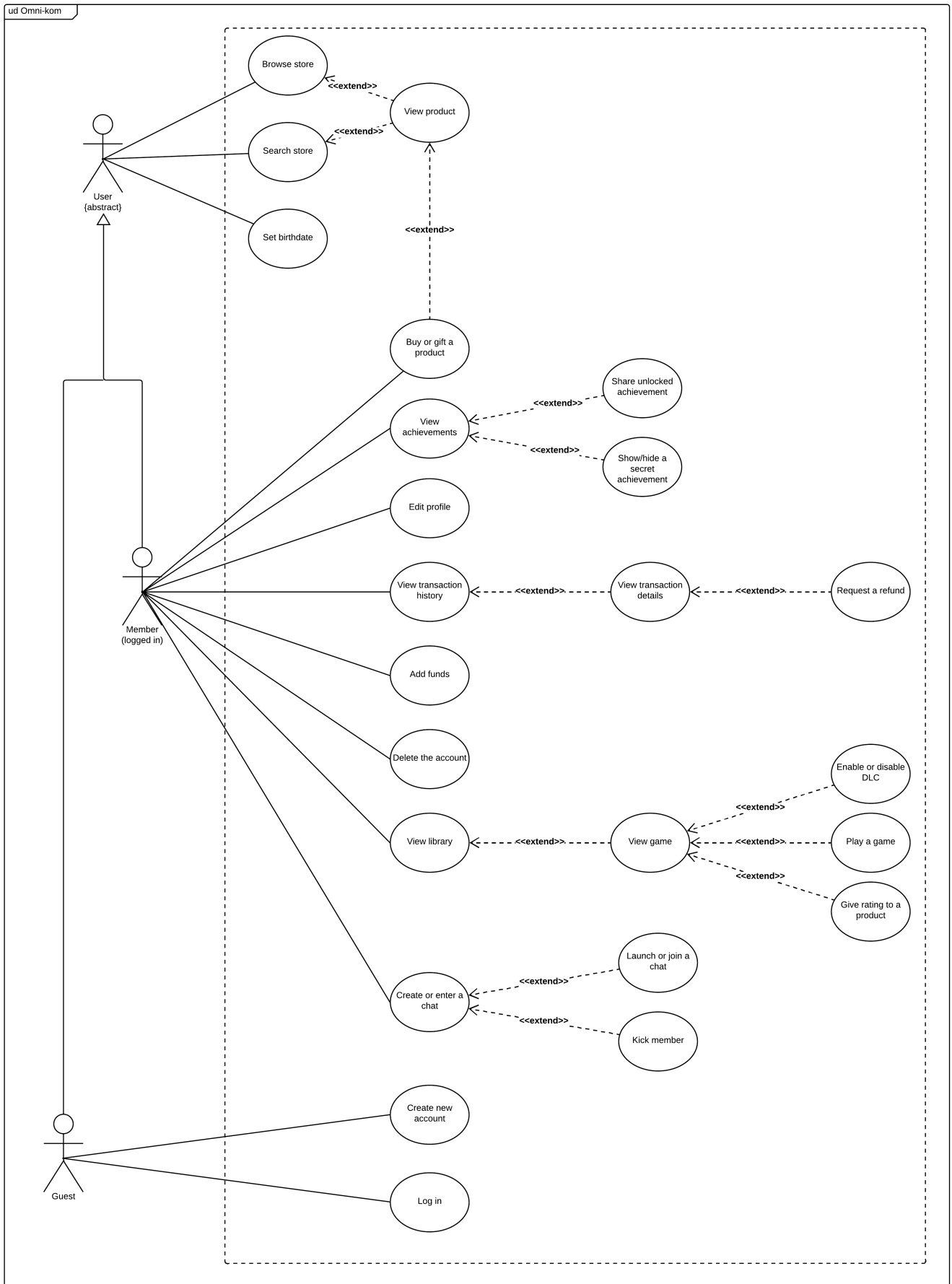
Serwis obsługuje również funkcjonalność posiadania kont zarejestrowanych członków. Każdy z nich ma: identyfikator, imię, nazwisko, adres mailowy oraz stan portfela. Zalogowani członkowie posiadają również możliwość dodania innych użytkowników jako znajomych.

Członek może rozpoczynać rozmowy z dodanymi znajomymi, tworzyć czaty grupowe, o maksymalnej liczebności 99 osób. Czat posiada swój identyfikator, listę członków, właściciela i nazwę.

Czat można zamknąć, dodawać oraz usuwać z niego członków. Tylko właściciel ma uprawnienia do zarządzania utworzonym czatem.

System ma umożliwiać potencjalnym użytkownikom m.in. w realizowanie zadań, których listę przedstawiono poniżej. W nawiasach podano byty uprawnione do ich wywołania:

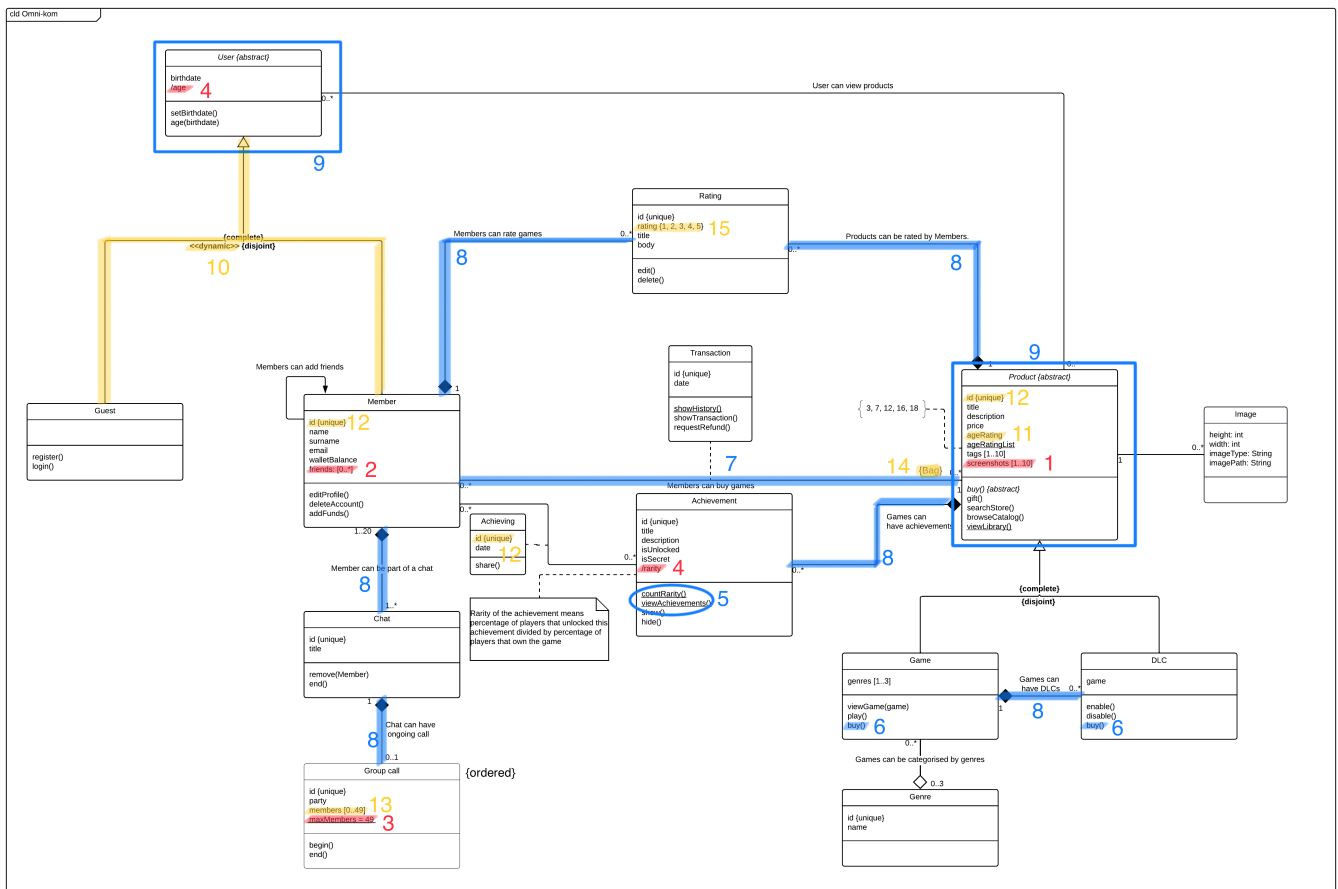
- Przeglądanie sklepu (Użytkownik)
- Przeszukiwanie sklepu (Użytkownik)
 - Wyświetlanie produktu (Użytkownik)
- Podanie daty urodzenia (Użytkownik)
- Kupowanie gry dla siebie lub jako prezent (Członek)
- Wyświetlanie osiągnięć (Członek)
 - Zdobywanie osiągnięć (Członek)
 - Odkrywanie i ukrywanie sekretnych osiągnięć (Członek)
- Edycja profilu (Członek)
- Wyświetlanie historii transakcji (Członek)
 - Wyświetlanie szczegółów transakcji (Członek)
 - Żądanie zwrotów za grę (Członek)
- Dodaj środki (Członek)
- Usuń konto (Członek)
- Pokaż bibliotekę gier (Członek)
 - Pokaż szczegóły gry (Członek)
 - Włącz/wyłącz DLC (Członek)
 - Uruchom grę (Członek)
 - Oceń grę (Członek)
- Utwórz lub dołącz do chatu (Członek)
 - Uruchom lub dołącz do czatu (Członek)
 - Wyrzuć członka (Członek)
- Utwórz nowe konto (Gość)
- Zaloguj się (Gość)



Rysunek 1 - diagram przypadków użycia

Wymagania na system

Wymagania funkcjonalne - analityczny diagram klas



Rysunek 2.1 - analityczny diagram klas

Legenda

1. Atrybut złożony - w implementacji typu `Image`
2. Atrybut opcjonalny - `friends: [0..*]`; `achievements [0..*]`
3. Atrybut klasowy - `maxMembers = 49`
4. Atrybut pochodny - `rarity`; `age`
5. Metoda klasowa - `countRarity()`, `viewAchievements()`; `viewLibrary()`
6. Metoda przesłonięta - `buy`

7. Asocjacja z atrybutem - pomiędzy Member, a *Product*: klasa Transaction

8. Kompozycja

9. Klasa abstrakcyjna - klasy *User* i *Product*

10. Dziedziczenie dynamiczne - po klasie *User*

11. Ograniczenie atrybutu - `ageRating`

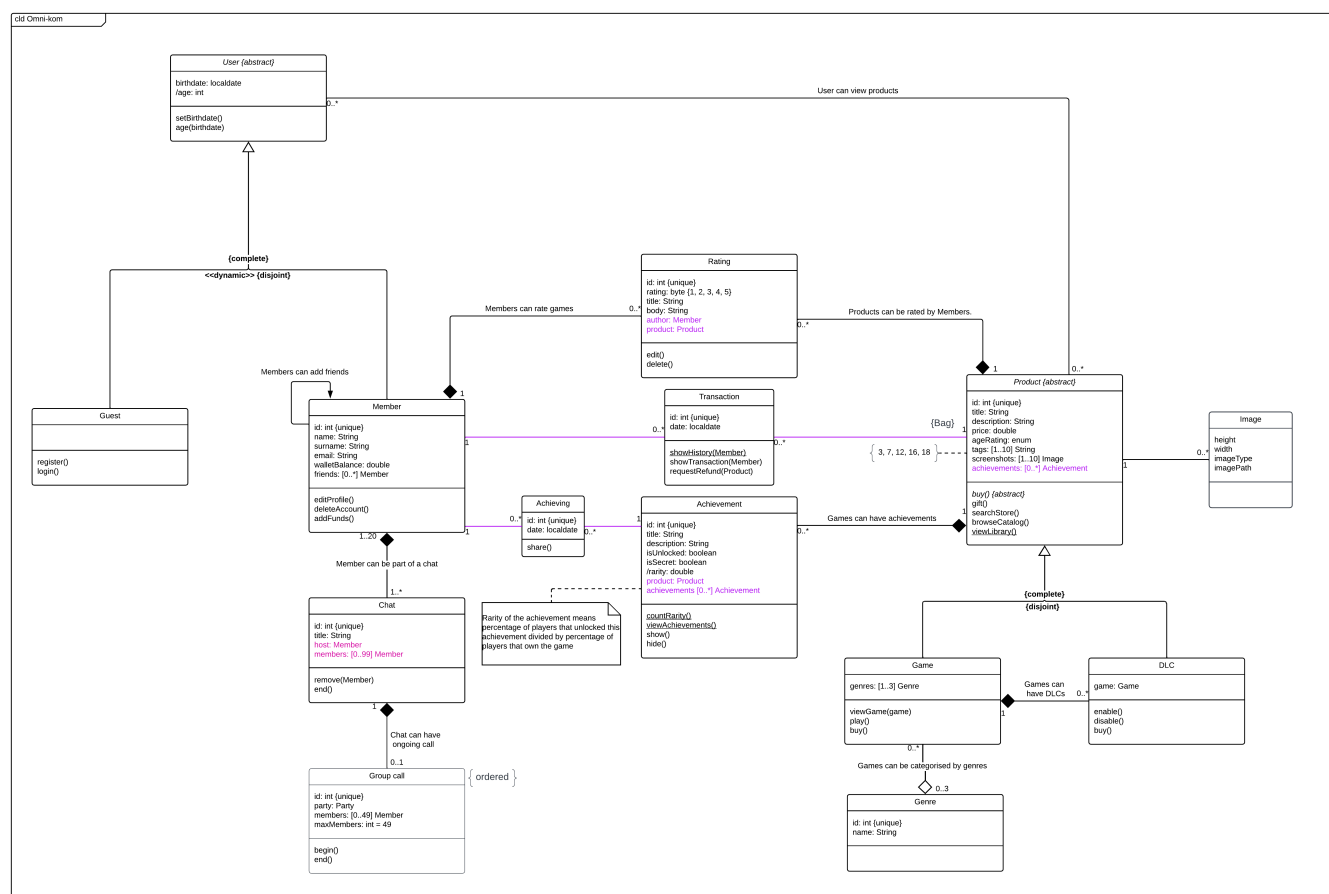
12. Ograniczenie unique - `id`

13. Ograniczenie ordered - `members`

14. Ograniczenie bag - pomiędzy Member, a *Product*: klasa Transaction

15. Ograniczenie własne - `rating {1, 2, 3, 4, 5}`

Opis struktury systemu - projektowy diagram klas



Rysunek 3 - projektowy diagram klas

Zmiany względem diagramu analitycznego oznaczone są kolorem.

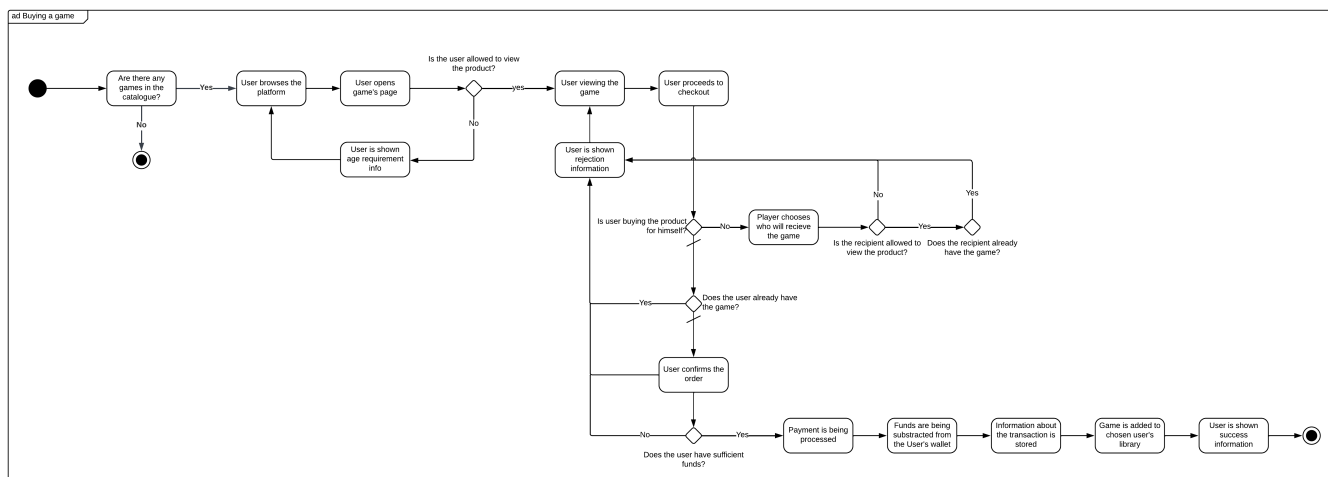
Scenariusz przypadku użycia „Kupowanie gry”

Aktorzy	Gracz, Użytkownik
Warunek początkowy	W katalogu jest przynajmniej 1 gra
Warunek końcowy	Gra jest w ekwipunku gracza

1. Gracz przegląda katalog serwisu
2. Gracz przechodzi do widoku wybranej gry
3. System przeprowadza weryfikację wieku
 1. Wiek gracza nie pozwala na wyświetlenie gry
 1. System wyświetla informację grze dla starszych osób
 2. Powrót do punktu 1.
4. Gracz jest przenoszony do widoku produktu
5. Gracz wybiera opcję zakupu, poprzez wciśnięcie przyciska „Kup”
6. System sprawdza, czy gracz chce kupić grę dla siebie, czy jako prezent
 1. Gracz wybiera opcję zakupu „jako prezent”
 1. Gracz wyznacza znajomego z serwisu, który otrzyma grę
 2. System przeprowadza weryfikację wieku
 1. Docelowy znajomy jest zbyt młody
 1. System wyświetla informację o grze dla starszych osób
 2. Powrót do punktu 4.
 3. Przejście do punktu 7.
 2. Gracz wybiera opcję zakupu „dla siebie”
 1. System przeprowadza weryfikację wieku
 2. W bibliotece gracza znajduje się już ten produkt.
 1. System informuje gracza o posiadaniu już wybranego produktu
 2. Powrót do punktu 4.
 7. Gracz potwierdza zamówienie
 8. System sprawdza, czy saldo przypisanego do konta portfela pozwala na dokonanie zakupu
 1. Gracz ma za małą ilość funduszy
 1. System wyświetla informację o braku środków
 9. System odejmuje określoną kwotę z portfela gracza
 10. System zapisuje informacje o transakcji

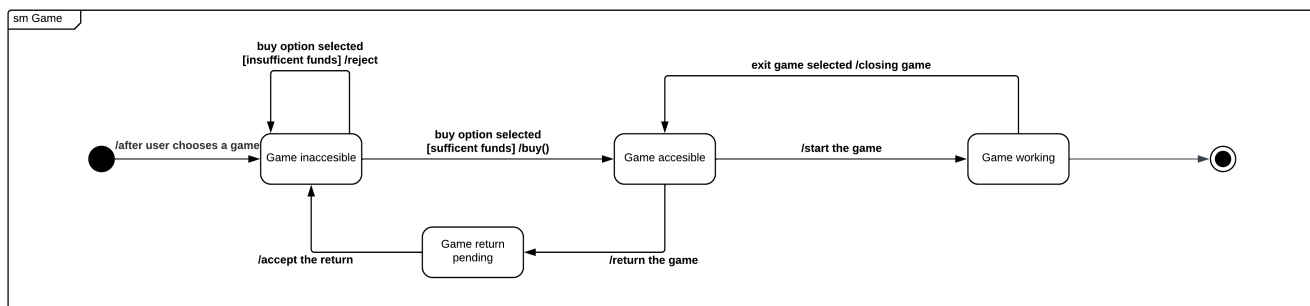
11. System dodaje produkt do biblioteki gracza lub desygnowanego znajomego
12. System pokazuje informację o potwierdzeniu zakupu
13. Przypadek kończy się

Diagram aktywności dla przypadku użycia "Kupowanie gry"



Rysunek 4 - diagram aktywności dla przypadku użycia "Kupowanie gry"

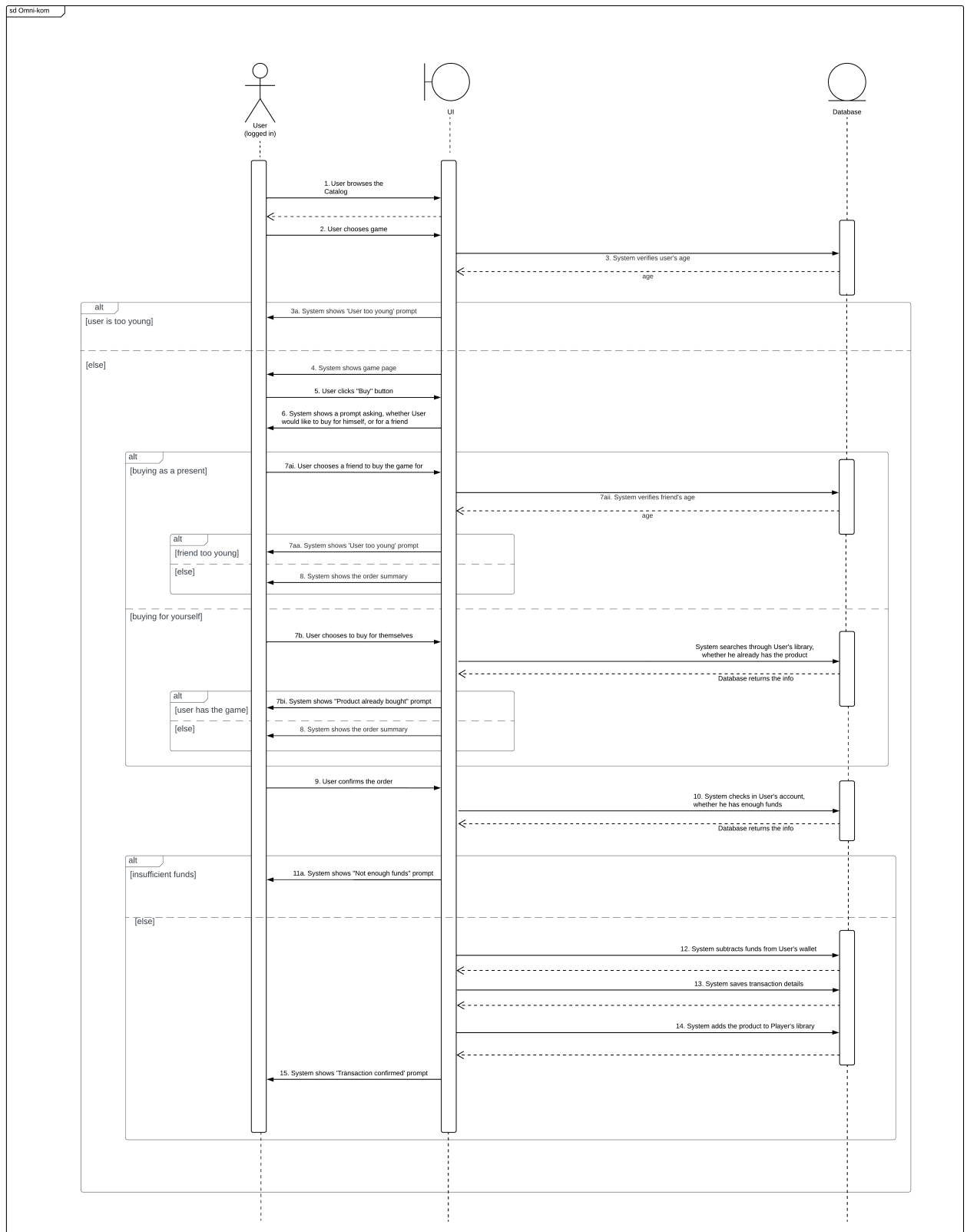
Diagram stanu dla klasy "Gra"



Rysunek 5 - diagram stanu dla przypadku użycia "Kupowanie gry"

Diagramy sekwencji dla przypadku użycia „Kupowanie gry”

Przebieg domyślny - gracz kupuje grę dla siebie



Rysunek 6 - diagram sekwencji dla przypadku użycia "Kupowanie gry"

Wymagania niefunkcjonalne – miary

Ograniczenie	Miara
Wydajność	Ekrany ładują się w czasie mniejszym niż 3 sekundy, przetwarzanie zakupów i płatności trwa mniej niż 5 sekund (nie wliczając czasu pośredników). Obsługa 400 żądań na sekundę bez kolejki dłuższej niż 50.
Bezpieczeństwo	Zapewnienie bezpieczeństwa użytkowników i ich danych poprzez rozlokowanie ich na kilku serwerach (min. 3) z szyfrowaniem ich za pomocą SHA-256 lub mocniejszego. Utrzymywanie zaktualizowanego systemu i elementów w nim zawartych. (Maksymalnie 1 wersja do tyłu)
Łatwość użytkowania	Intuicyjny interfejs zapewniający łatwą dostępność do wszystkich funkcji. Kompatybilność z funkcjami text-to-speech w przeglądarkach. Aktualne FAQ odpowiadające na często zadawane pytania. 7/10 zadowolonych użytkowników.
Dostępność	Dostępność platformy dla użytkowników z całej Europy w 5 językach (polski, angielski, niemiecki, ukraiński, francuski). Dostępność serwisu 24 godziny na dobę, przez 7 dni w tygodniu w ciągu całego roku (Nie licząc przerw serwisowych).
Skalowalność	Zdolność platformy do obsługi rosnącej liczby użytkowników, możliwość przechowywania danych ponad 1 miliona użytkowników. Biblioteka gier będąca w stanie przechowywać ponad 50 000 tytułów gier.

Opis przyszłej ewolucji systemu

- Rozszerzenie biblioteki gier o nowe tytuły i kategorie gier
- Ulepszenie funkcjonalności osiągnięć i dodanie nowych właściwości
- Dodanie funkcjonalności związanych z grami wieloosobowymi, takich jak turnieje i rankingi
- Ulepszenie interfejsu użytkownika i dostosowanie go do różnych urządzeń
- Dodanie funkcjonalności związanych z transakcjami, takich jak płatności mobilne i krypto waluty
- Rozszerzenie funkcjonalności czatu o nowe funkcje, takie jak wideo czat i integracja z innymi platformami społecznościowymi.

Słownik pojęć z dziedziny problemowej

Pojęcie	Termin
Gra	Gra elektroniczna, która wykorzystuje interaktywny ekran wideo do generowania wizualnej informacji w celu interakcji z użytkownikiem.
Platforma gier	Oprogramowanie lub sprzęt, który umożliwia użytkownikom grę w gry wideo
Gracz	Osoba, która zarejestrowała konto w systemie
Gość	Osoba, która nie dokonała rejestracji w systemie
Osiągnięcie	Cel gry stworzony poza światem jej rozgrywki. W przeciwieństwie do zadań lub poziomów, będących celami gier i mającymi bezpośredni wpływ na przebieg ich rozgrywki, system osiągnięć zwykle odbywa się poza środowiskiem i architekturą gry.
Czat	Rodzaj rozmowy między dwoma lub wieloma użytkownikami komputerów za pośrednictwem Internetu lub innej sieci komputerowej, polegającej na naprzemiennym przesyłaniu wiadomości tekstowych

Projekt GUI

Omni-kom

Game's title

Age rating

-Game description-

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ultrices gravida dictum fusce ut placerat orci. Netus et malesuada fames ac turpis egestas integer eget aliquet. Sit amet luctus venenatis lectus magna fringilla. Eleifend donec pretium vulputate sapien nec sagittis aliquam. Fermentum iaculis eu non diam. Amet purus gravida quis blandit turpis. Hac habitasse platea dictumst vestibulum. Tellus id interdum velit laoreet id donec ultrices. Dolor sed viverra ipsum nunc aliquet bibendum enim facilisis gravida. Ultricies mi quis hendrerit dolor magna eget est. Fringilla phasellus faucibus scelerisque eleifend donec pretium. Aliquet enim tortor at auctor urna nunc id cursus. Turpis nunc eget lorem dolor sed viverra ipsum. Feugiat vivamus at augue eget arcu dictum varius duis.

Category

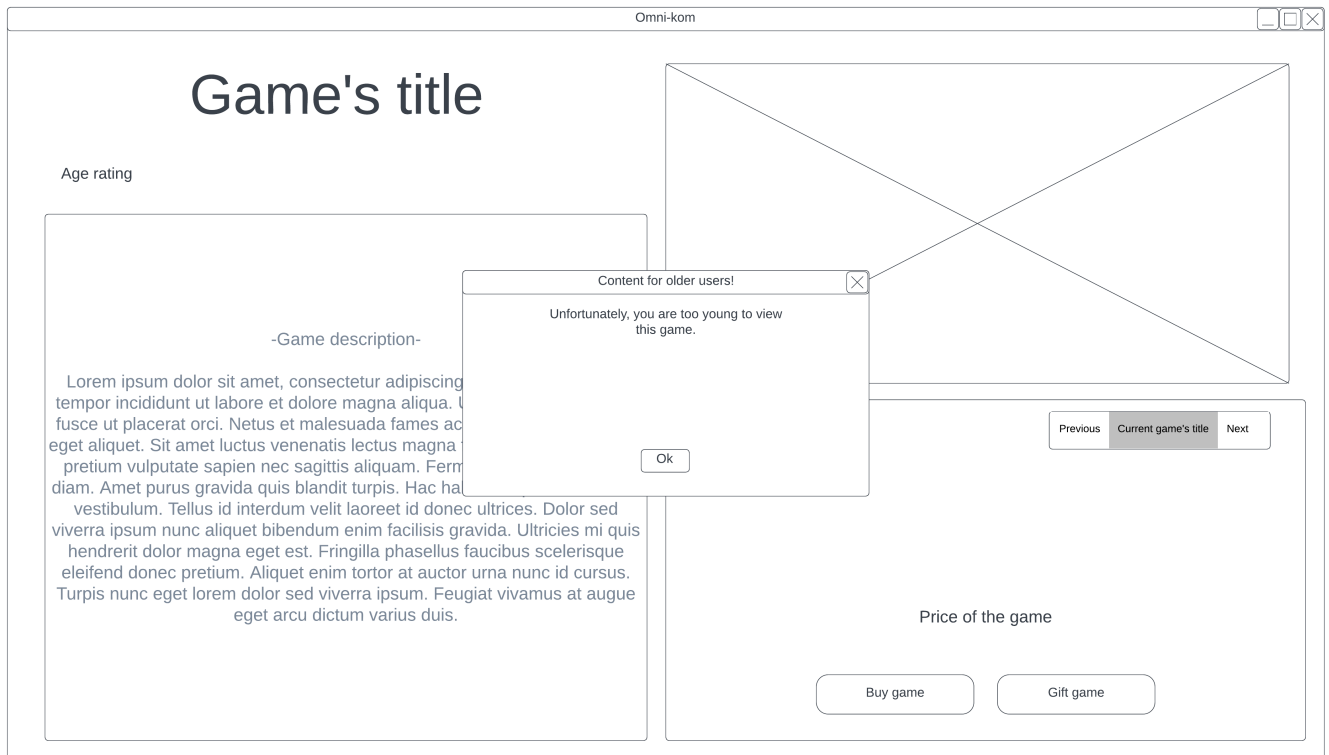
Previous Current game's title Next

Price of the game

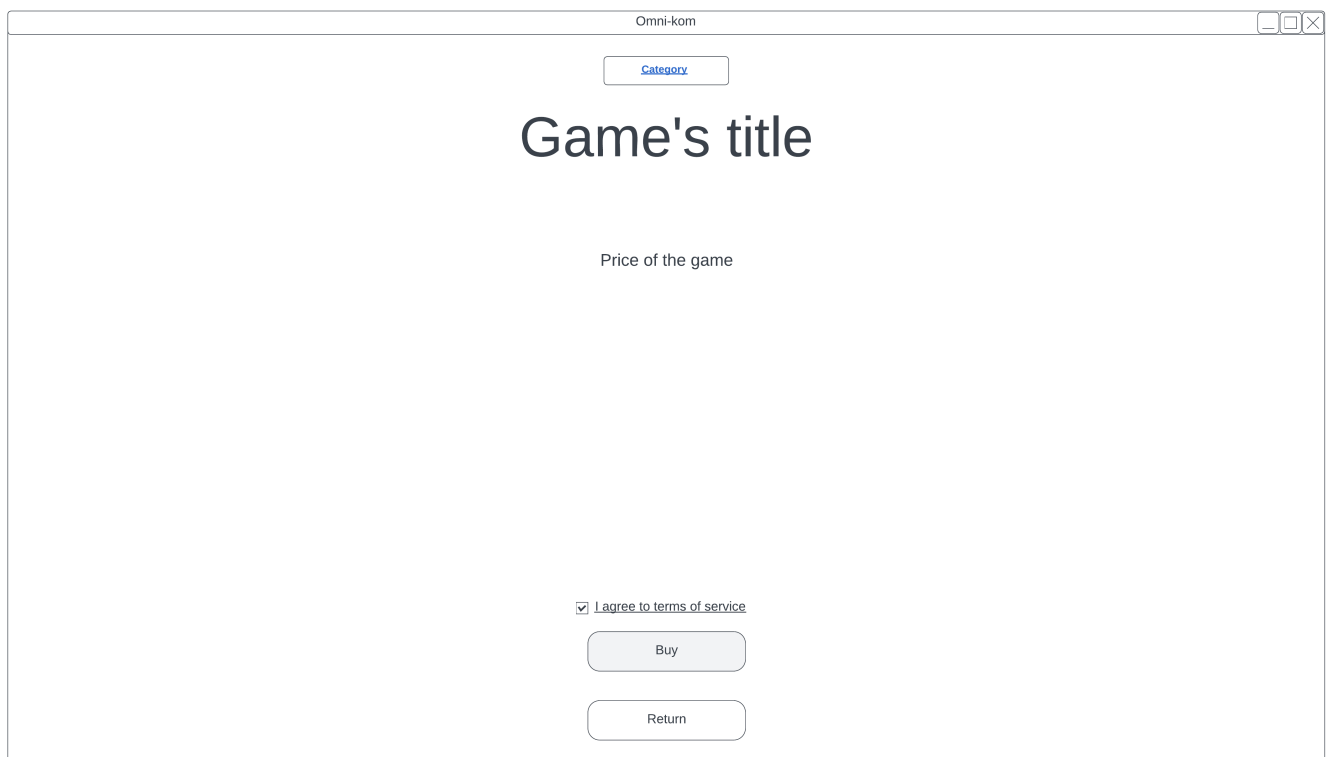
Buy game

Gift game

Rysunek 7 - Karta produktu



Rysunek 7a - Karta produktu z komunikatem



Rysunek 8 - Karta kupowania gry

Omówienie decyzji projektowych

Program został zrealizowany w języku Java. Wszystkie dane, tak jak i ich przechowywanie jest wykonywane poprzez sam język Java. Interfejs graficzny aplikacji jest wyświetlany za pomocą biblioteki graficznej Swing. Wspomagałem się również biblioteką Project Lombok, który pomógł mi w pisaniu czystego i zwięzłego kodu w szybki i praktyczny sposób.

Wybrane technologie

- **Java 17:** Językiem programowania jest Java 17 oferująca wsparcie aż do 2026 roku, z przedłużonym wsparciem do 2029 roku. Jest to istotne, w kwestii zapewnienia trwałości systemu, jego konserwacji i stabilności.
- **Serializacja:** Zapewnienie trwałości jest realizowane przy użyciu serializacji języka Java do plików `.dat`, za pomocą interfejsu `Serializable`.
 - Pliki `.dat` to pliki binarne, zawierające dane zserializowane w formacie binarnym.
 - Wczytywanie danych z plików `.dat` odbywa się przy użyciu strumieni wejściowych.
 - Zapisywanie danych do plików `.dat` odbywa się przy użyciu strumieni wyjściowych.
- **GUI:** Interfejs graficzny (GUI) został zrealizowany za pomocą toolkitu Swing
- **Lombok:** Do obsługi metod dostępu do atrybutów `get`, `set`, `toString` została wykorzystana została biblioteka Lombok
- Atrybuty pochodne są wyliczane w momencie ich wywołania
- Asocjacja z atrybutem została zrealizowana z użyciem klasy pośredniczącej i dwóch asocjacji 1..*
- Asocjacje binarne zaimplementowane za pomocą kolekcji
- Ograniczenie własne zrealizowane za pomocą enumów
- Dziedziczeni dynamiczne zrealizowane poprzez "klonowanie", w klasie `Member` specjalny konstruktor przyjmujący obiekt `Guest`
- Ograniczenie `ordered` kolejność `member` w `Group call` jest ustalana poprzez `Member id`