

# Zarządzanie danymi gry MMORPG „Big Game Studios”

## 1. Dziedzina problemowa

Projektowany system został zamówiony przez prezesa wschodzącego studia gier MMO „Big Game Studios” i służy do zarządzania danymi związanymi z mającą niedługo premierę grą MMORPG, ale w przyszłości może być użyty jako gotowe rozwiązanie dla innych gier z tego gatunku.

## 2. Cel

Celem projektu jest stworzenie aplikacji, która umożliwi efektywne zarządzanie danymi związanymi z grą MMORPG. Aplikacja ma umożliwiać użytkownikom przechowywanie i aktualizowanie danych postaci, ekwipunku oraz innych danych kont. Studio odpowiedzialne za grę będzie miało możliwość monitorowania i zarządzania kontami graczy, oraz utrzymywać bezpieczeństwo i integralność danych.

## 3. Zakres odpowiedzialności systemu

System będzie odpowiedzialny za:

- Zarządzanie danymi kont graczy, w tym: rejestrację, logowanie oraz zarządzanie postaciami stworzonymi na danym koncie.
- Przechowywanie informacji o postaciach graczy, takich jak imię, klasa, poziom doświadczenia, stan ekwipunku oraz zdobyte osiągnięcia.
- Zarządzanie statusem kont, w tym umożliwienie zakupu rocznej subskrypcji konta Premium
- Zarządzanie stanem ekwipunku postaci graczy, w tym tworzenie i dodawanie przedmiotów.
- Zapewnienie dostępu do danych dla graczy jak i personelu studia, z odpowiednimi uprawnieniami zależnymi od roli (gracz, administrator, deweloper).
- Utrzymywanie bezpieczeństwa danych poprzez implementację mechanizmów uwierzytelniania i autoryzacji oraz regularne tworzenie kopii zapasowych danych.

System nie jest dostępny bezpośrednio z perspektywy gracza, a służy jedynie jako źródło danych dla odpowiednio zaimplementowanego klienta gry. System jest swoistym API dla danego klienta gry MMO, który jest implementowany osobno.

System nie zakłada zarządzania stanem całej gry, a jedynie kontami i danymi bezpośrednio związanymi z graczami i ich kontami, dlatego też:

- System nie zarządza danymi związanymi ze światem gry, jak np. lokacje, misje, NPC etc.

## 4. Użytkownicy systemu

Użytkownikami systemu będą gracze (klient gry), administracja gry, deweloperzy tworzący i rozwijający grę, oraz podsystemy czasu.

## 5. Wymagania użytkownika

1. W systemie należy przechowywać dane dotyczące kont graczy i ich postaci. Konta dzielimy na typy Standard oraz Premium. Konto domyślnie jest Standardowe, i staje się Premium po zakupieniu subskrypcji. Wtedy przechowujemy też koszt zakupionej subskrypcji oraz datę jej zakupu. Subskrypcja trwa rok, po jej upływie konto wraca do stanu Standardowego.
2. Dla każdego konta system musi zapamiętać: e-mail (unikatowy dla każdego konta), hasło, oraz stworzone postaci. Postać nie może istnieć bez przypisanego konta. Należy też zapamiętać i kontrolować maksymalną ilość postaci które mogą zostać stworzone na danym koncie.
3. Dla każdej postaci system zapamiętuje jej nazwę, poziom doświadczenia, zdobyte osiągnięcia, przypisaną klasę oraz ekwipunek. Poziom doświadczenia postaci nie może być mniejszy niż startowy poziom doświadczenia równy 1 oraz większy niż maksymalny poziom doświadczenia równy 100. Wartości te są takie same dla każdej postaci i mogą ulec zmianie.
4. System musi przechowywać nazwę oraz opis każdej klasy.
5. W ramach każdego osiągnięcia system przechowuje nazwę, opis, oraz datę otrzymania go na danej postaci.
6. Każda postać ma dokładnie jeden ekwipunek, w ramach którego system zapamiętuje ilość miejsc na stosy, ilość waluty podstawowej, oraz ilość waluty premium.
7. System ma za zadanie zapamiętywać przedmioty które postaci mogą posiadać w swoim ekwipunku. Ze względu na rodzaj przedmioty dzielimy na:
  - i. Pancerz z dodatkowym atrybutem obrony
  - ii. Broń z dodatkowym atrybutem obrażeń
  - iii. Jednorazowy w ramach którego dodatkowo przechowujemy czas trwania efektu
  - iv. Ubiór, a z nim część ubioru („głowa”, „tułów”, „nogi”, lub „stopy”) oraz kolor.
8. W ramach każdego przedmiotu system przechowuje nazwę, opis oraz maksymalną ilość przechowywalną w ramach jednego stosu.
9. System zapisuje dla każdego stosu przedmiotów do jakiego przedmiotu się odnosi, oraz ich ilość. Ilość ta nie może przekroczyć wcześniej wspomnianej wartości w ramach przedmiotu. Stosy istnieją tylko w kontekście ekwipunku, z tym że należy pilnować, aby nie było ich więcej niż ilość miejsc w ekwipunku danej postaci.

System powinien umożliwić użytkownikom gry:

- Rejestrację oraz zalogowanie się na konto (gracz)
- Stworzenia nowej postaci na koncie (gracz)
- Wyświetlenia wszystkich postaci, wraz z ich osiągnięciami oraz ekwipunkami (gracz)
- Wyświetlenie wszystkich osiągnięć postaci oraz wszystkich możliwych do zdobycia (gracz)
- Wyświetlenie całego ekwipunku postaci (gracz)
- Wyświetlenia wszystkich klas postaci (gracz)
- Dodanie/Usunięcie subskrypcji z konta (administrator)
- Wyświetlenia wszystkich kont, wraz z wyświetleniem ich postaci (administrator)
- Dodania stosu lub osiągnięcia (administrator)
- Dodania klasy postaci (Deweloper)
- Dodania przedmiotu (Deweloper)
- Zwiększenia max poziomu doświadczenia wszystkich postaci (Deweloper)
- Sprowadzenie przedawnionych kont typu Premium na Standard (na koniec każdego miesiąca)

Wymagania niefunkcjonalne systemu:

- Dostępność
- możliwość równoczesnej obsługi wielu użytkowników
- łatwość w obsłudze dla każdego użytkownika
- Niezawodność
- Zapewnienie bezpieczeństwa danych

## 6. Wymagania niefunkcjonalne - miary

| Ograniczenie                                      | Miara  |
|---|--|
| Dostępność  | Czas, w jakim system powinien być dostępny dla użytkowników: 24 godziny przez 7 dni w tygodniu przez 365 dni w roku (dostępność w trybie 24/7/365) |
| Możliwość równoczesnej obsługi wielu użytkowników | Dostęp i poprawna obsługa dla 5000 użytkowników jednocześnie, niezależnie od typu użytkownika (gracz, czy obsługa gry)                             |
| Niezawodność                                      | Średni czas bezawaryjnej pracy systemu nie powinien być krótszy niż 1 rok  |
| Przechowywanie danych w bazie danych              | Serwerem bazy danych powinien być H2 2.1.x lub nowszy  |
| Zapewnienie bezpieczeństwa danych                 | Tworzenie backup-ów zgodnie z harmonogramem, transakcyjne przetwarzanie danych   |

## 7. Opis przyszłej ewolucji systemu

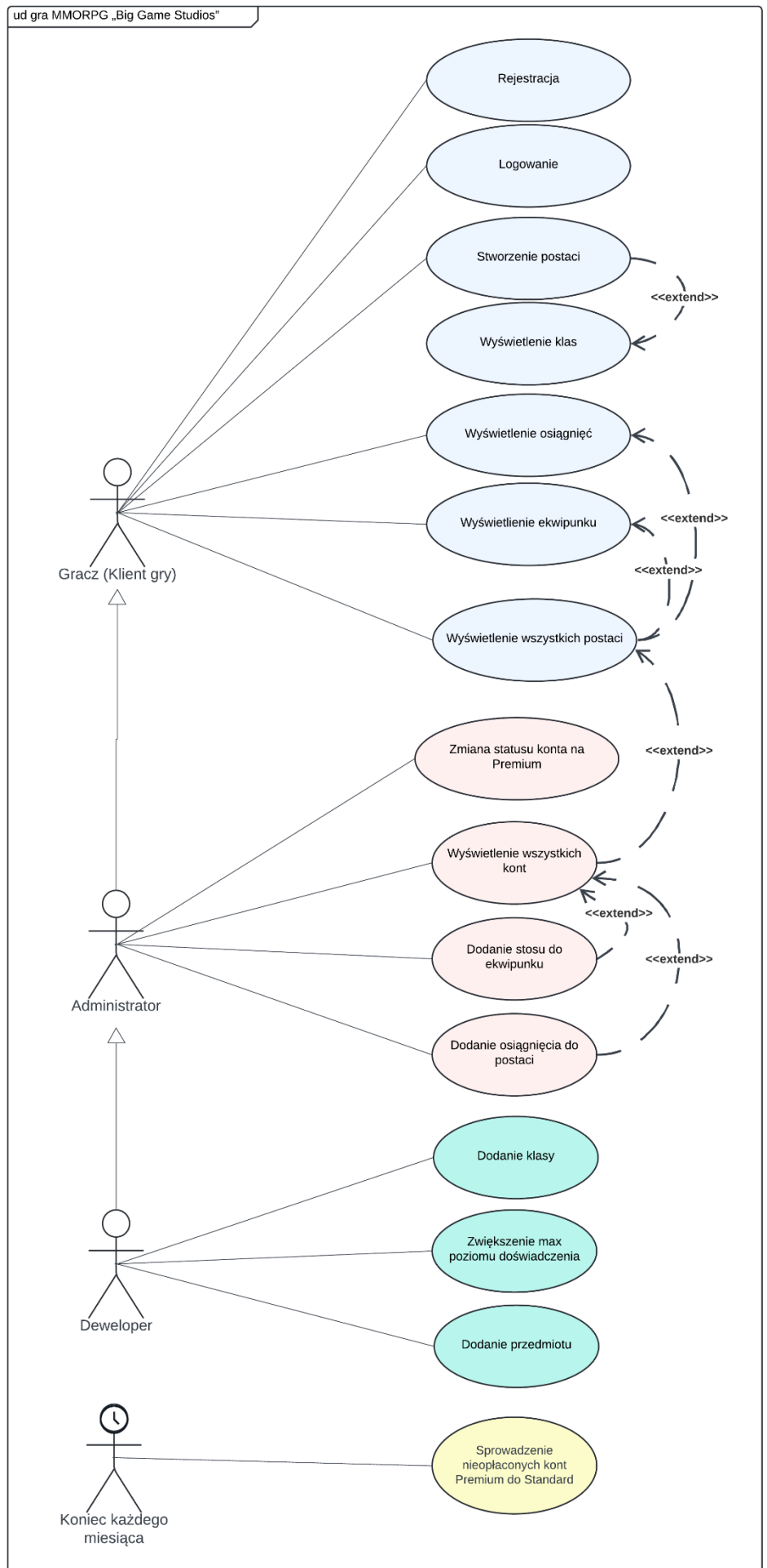
W przyszłości planowana jest rozbudowa systemu o wprowadzenie danych związanych z monetyzacją gry, oraz zapewnienie integralności i spójności danych dotyczących kolejnych elementów gry MMO. Dodatkowym kierunkiem rozwoju aplikacji mogą być dodatkowe narzędzia administracyjne i moderacyjne umożliwiające np. kontrolę i przechowywanie historii komunikacji między graczami w celu utrzymania zdrowej i bezpiecznej społeczności graczy.

## 8. Słownik pojęć z dziedziny problemowej

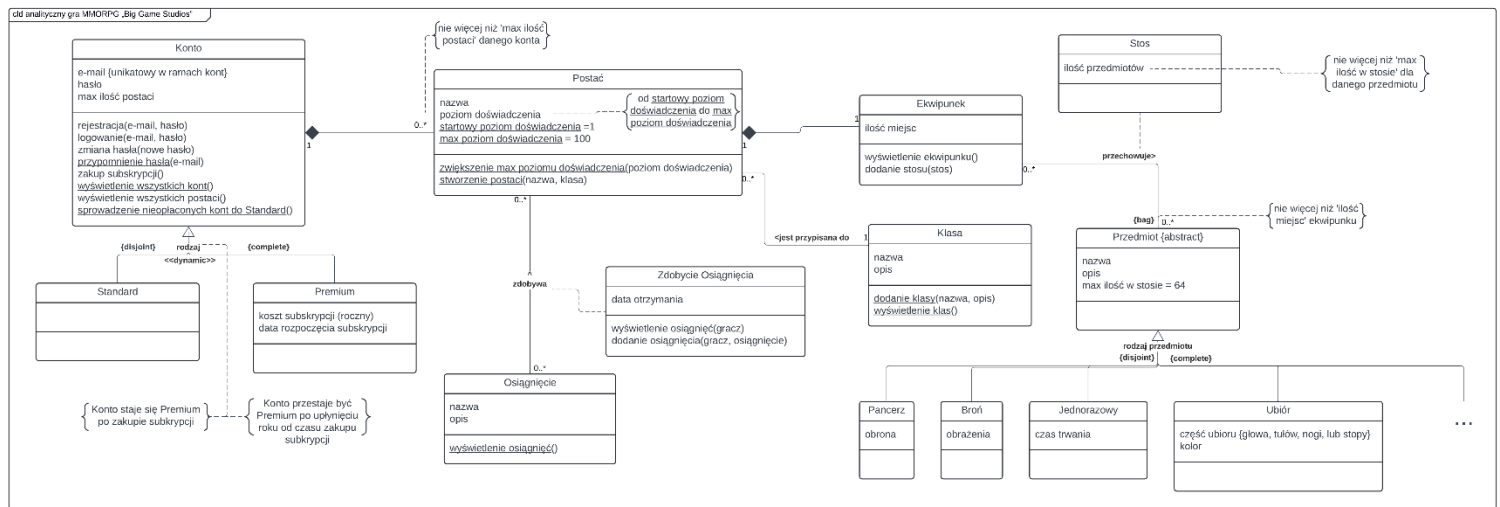
| Pojęcie (termin)      | Objaśnienie   |
|-----------------------|---|
| Konto                 | Konto założone przez gracza, w ramach którego zapisujemy e-mail oraz hasło podane przy rejestracji (e-mail musi być unikatowy w ramach kont), oraz maksymalną ilość postaci która może być stworzona i przechowywana w ramach tego konta.<br><br>Konto jest domyślnie Standardowe, ale może stać się Premium po zakupie subskrypcji.  |
| Postać                | Jest stworzona przez gracza w ramach konkretnego konta w celu przechowywania danych z nią związanych.<br><br>Dla postaci przechowujemy jej nazwę, oraz poziom doświadczenia, który zaczyna się dla każdej postaci od tej samej wartości, i nie może być większy niż maksymalny poziom doświadczenia, jednakowy dla wszystkich postaci |
| Ekwipunek             | Ekwipunek należy do danej postaci, i służy do przechowywania informacji na temat posiadanych stosów przedmiotów.<br><br>Dana postać może posiadać tylko jeden ekwipunek, który jest jej nierozłączną częścią.   |
| Stos                  | Rolą stosu jest zapis informacji jakiego przedmiotu stosem jest, oraz ile sztuk danego przedmiotu w sobie przechowuje.<br><br>Dany stos odnosi się tylko do jednego ekwipunku.  |
| Przedmiot             | Mówi nam o właściwościach przedmiotu posiadanego przez daną postać. Bazowo, w ramach każdego przedmiotu zapisujemy nazwę, opis, oraz maksymalną ilość tego przedmiotu w jednym stosie.  |
| Pancerz               | Pancerz to rodzaj przedmiotu, w ramach którego zapisujemy obronę.   |
| Broń                  | Broń to rodzaj przedmiotu, w ramach którego zapisujemy obrażenia.   |
| Przedmiot Jednorazowy | Przedmiot Jednorazowy to rodzaj przedmiotu, dla którego przechowujemy jego czas trwania.  |
| Ubiór                 | Ubiór to rodzaj przedmiotu, w ramach którego zapisujemy jakiej części ubioru dotyczy (jedno z "głowa", "tułów", "nogi", "stopy"), oraz kolor.   |

|                      |  |
|----------------------|--|
| Osiągnięcie          | W ramach osiągnięcia zapisujemy jego nazwę, oraz opis. System musi pamiętać datę otrzymania każdego osiągnięcia w ramach każdej postaci.                   |
| Zdobycie osiągnięcia | Osiągnięcie może zostać zdobyte przez daną postać. Wówczas zapisujemy datę jego zdobycia przez daną postać. Postać może zdobyć dane osiągnięcie tylko raz. |
| Klasa                | Każda postać jest określonej klasy wybranej przy stworzeniu. Dla każdej klasy zapisujemy jej nazwę, oraz opis.   |

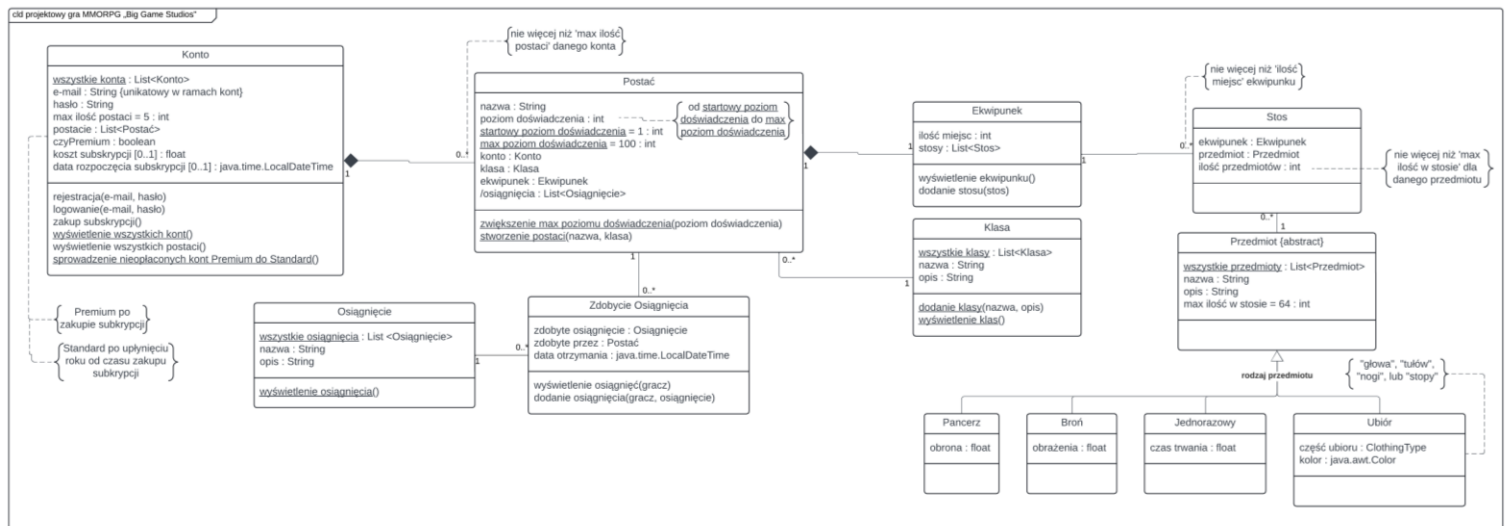
## 9. Diagram przypadków użycia



## 10. Diagram klas – analityczny



## 11. Diagram klas - projektowy



## 12. Tekstowy opis scenariusza przypadku użycia wyświetl osiągnięcia(gracz)

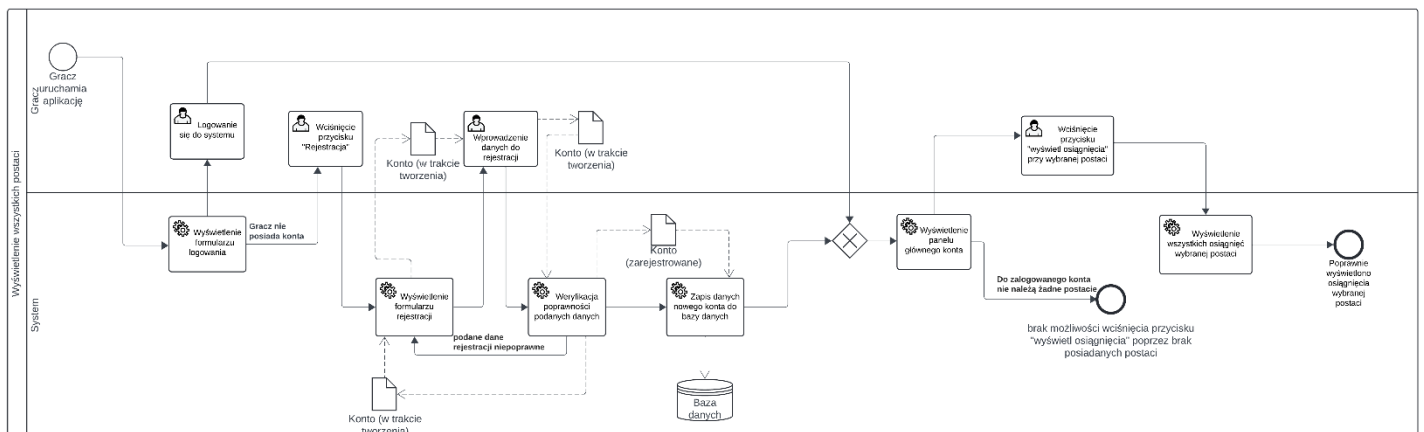
Ta sekcja tekstowo opisuje przypadek użycia wyświetl osiągnięcia(gracz), który zostaje wywołany z przypadku użycia wyświetlenie wszystkich postaci danego konta po zalogowaniu się na nie przez gracza (klienta gry). Gracz wciska przycisk „wyświetl wszystkie postacie”. Jeżeli nie jest jeszcze zalogowany, pojawia się formularz logowania.

Na formularzu jest też dostępny przycisk „Rejestracja”, którego gracz używa jeżeli nie ma jeszcze konta w grze. W przypadku wciśnięcia przycisku „Rejestracja” pojawia się osobny, bardzo podobny formularz rejestracyjny.

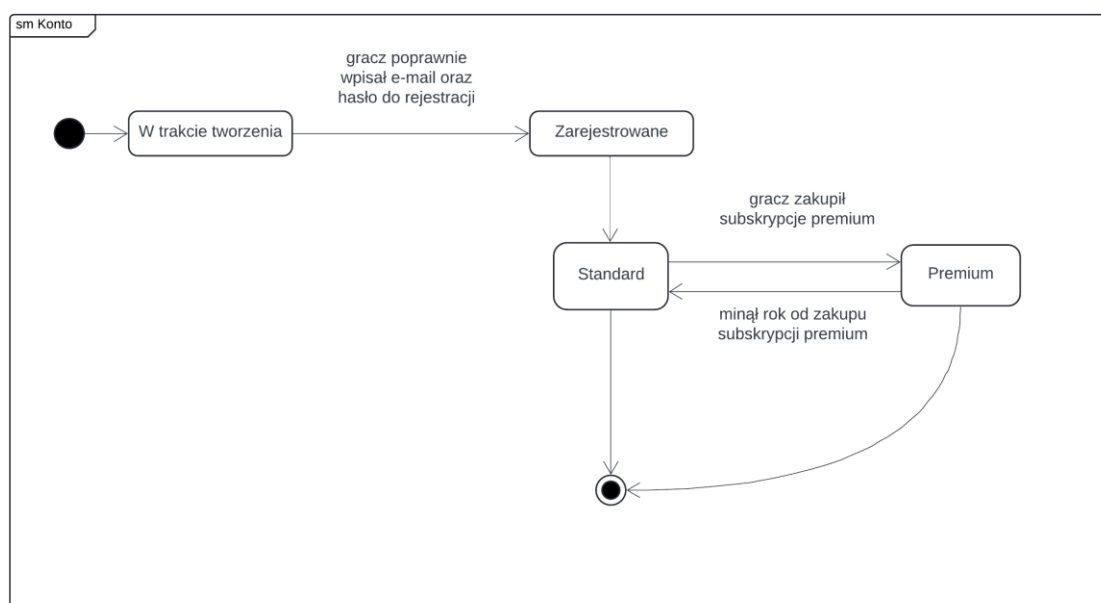
Po pomyślnej rejestracji lub logowaniu, jeżeli na danym koncie nie zostały jeszcze stworzone żadne postacie, system informuje o tym gracza, po czym sugeruje stworzenie swojej pierwszej postaci i proces się w tym momencie kończy.

W przeciwnym wypadku system wyświetla wszystkie postacie posiadane przez konto. Po wybraniu przez Gracza jednej z wyświetlonych postaci wyświetla się lista wszystkich zdobytych przez nią osiągnięć. Każde osiągnięcie może być wybrane w celu wyświetlenia opisu osiągnięcia, oraz daty jego zdobycia. Proces w tym momencie się kończy.

### 13. Diagram aktywności dla przypadku użycia *wyświetl osiągnięcia(gracz)*

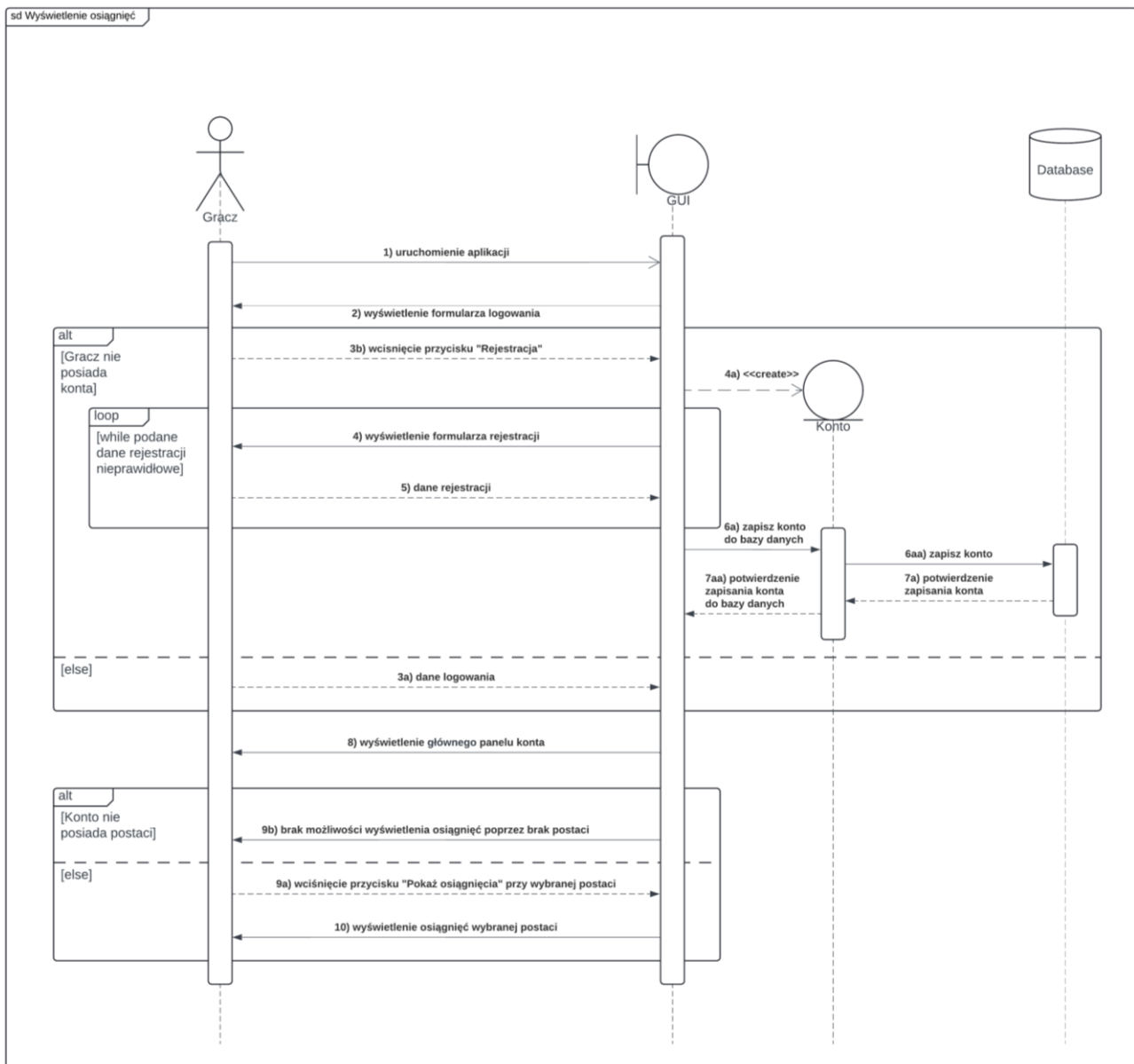


### 16. Diagram stanu klasy Konto



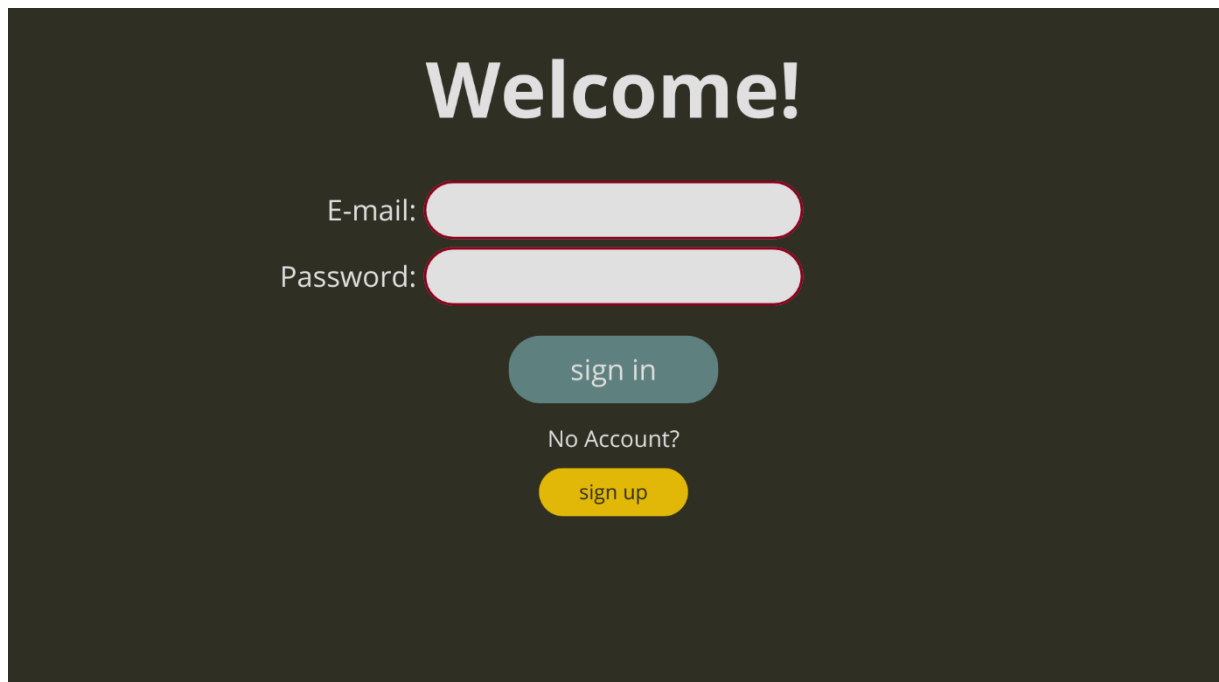


## 17. Diagram interakcji (sekwencji) dla przypadku użycia wyświetl/osiągnięcia(gracz)



## 18. Projekt GUI

### Formularz logowania



A login form UI mockup on a dark gray background. At the top, the word "Welcome!" is displayed in a large, white, sans-serif font. Below it, there are two input fields: "E-mail:" and "Password:". Each label is followed by a light gray rounded rectangle with a thin red border. Below the "Password:" field is a teal rounded button with the text "sign in" in white. Underneath the button is the text "No Account?" in a small, white font. At the bottom is a yellow rounded button with the text "sign up" in black.

Welcome!

E-mail:

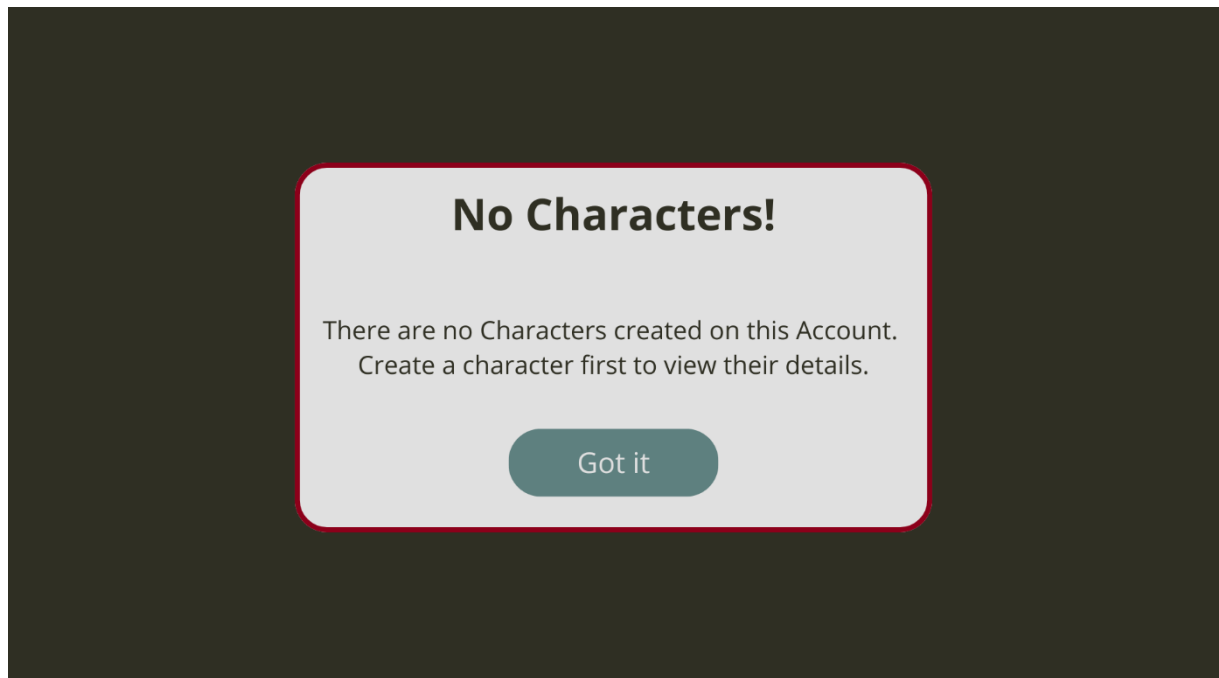
Password:

sign in

No Account?

sign up

### Brak postaci na wybranym koncie



A message box UI mockup on a dark gray background. The message box is a light gray rounded rectangle with a thin red border. Inside, the text "No Characters!" is displayed in a bold, black font. Below it, the text "There are no Characters created on this Account. Create a character first to view their details." is displayed in a smaller, black font. At the bottom is a teal rounded button with the text "Got it" in white.

No Characters!

There are no Characters created on this Account.  
Create a character first to view their details.

Got it

## Wybór konta (administrator)

| <a href="#">Return</a> |         | Accounts   |                 |
|------------------------|---------|------------|-----------------|
| e-mail                 | premium | characters | show characters |
| bob@gmail.com          | yes     | 5          | [SHOW]          |
| alice@gmail.com        | yes     | 2          | [SHOW]          |
| maks@gmail.com         | no      | 0          | [SHOW]          |
| patrycja@gmail.com     | no      | 1          | [SHOW]          |
| ...                    | ...     | ...        | ...             |
|                        |         |            |                 |
|                        |         |            |                 |
|                        |         |            |                 |

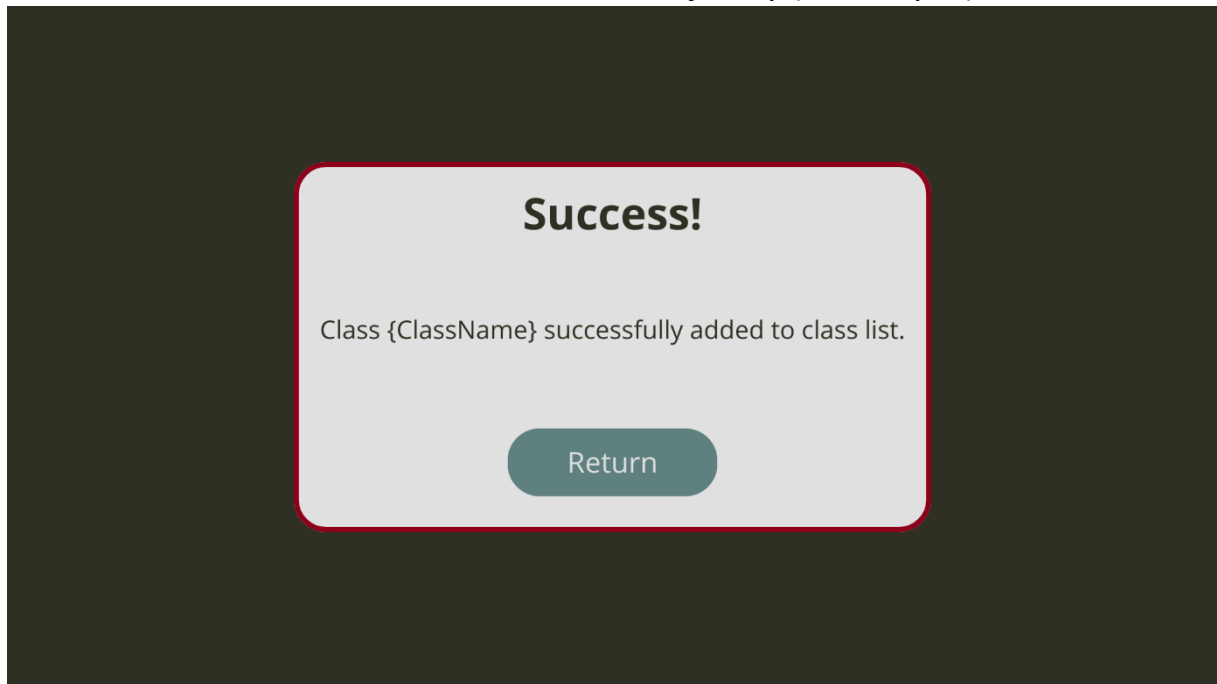
## Wyświetlenie wszystkich klas postaci

[Return](#)

Classes

| name          | description   | choice   |
|---------------|---|----------|
| Shadowblade   | Masters of stealth and subterfuge, these assassins strike from the shadows with deadly precision                    | [CHOOSE] |
| Arcane Knight | Warriors who combine physical prowess with powerful arcane magic, wielding enchanted weapons and armor              | [CHOOSE] |
| Nature Warden | Guardians of the wild, these druids harness the forces of nature to heal allies and control the battlefield         | [CHOOSE] |
| Stormcaller   | Elemental mages who command the power of storms, unleashing devastating lightning and wind attacks on their enemies | [CHOOSE] |
| Blood Warden  | Dark knights who draw power from blood magic, using their own life force to fuel powerful attacks and curses        | [CHOOSE] |

## Potwierdzenie dodania nowej klasy (deweloper)



## 19. Omówienie decyzji projektowych i skutków analizy dynamicznej

Analityczny diagram klas zawiera konstrukcje, które nie występują w języku programowania Java, w którym system zostanie zrealizowany. Na projektowym diagramie klas te konstrukcje zostały przekształcone w następujący sposób:

- Dziedziczenie dynamiczne - dodanie dyskryminatora w postaci zmiennej *czyPremium*; tutaj takie podejście jest proste do zrozumienia oraz implementacji, ponieważ wymaga dodania jednej zmiennej oraz przeniesienia dwóch pól opcjonalnych *koszt subskrypcji* oraz *data rozpoczęcia subskrypcji* do nadklasy.
- Asocjacja z atrybutem - klasa pośrednicząca i dwie asocjacje 1 – 0..\*. Takie rozwiązanie jest intuicyjne i relatywnie proste w implementacji.
- Kompozycja - klasa wewnętrzna. Naturalne rozwiązanie, proste do zrozumienia oraz implementacji.

Zwykłe asocjacje zostały zaimplementowane za pomocą kolekcji, skończone zbiory wartości – za pomocą Enum-ów, uporządkowana ekstensja klasy Osiągnięcie – za pomocą kolekcji typu `List<Osiągnięcie>`.

Opracowane przez:

Maksymilian Spisjak s25303, c17

v.1.0.3