Kamil Kłodawski

MAS – SYSTEM ZARZADZANIA FIRMA GIER KOMPUTEROWYCH „gAMEScOMPANYaSSISTANT”

Dokument projektu

Spis treści

[Dziedzina biznesowa i cel 1](#_Toc168913018)

[Zakres odpowiedzialności systemu 2](#_Toc168913019)

[Użytkownicy systemu 2](#_Toc168913020)

[Wymagania użytkownika 2](#_Toc168913021)

[System powinien przechowywać informacje o: 2](#_Toc168913022)

[Z następującymi ograniczeniami: 3](#_Toc168913023)

[Dane który powinny być zapamiętane dla każdej klasy: 3](#_Toc168913024)

[System powinien umożliwiać następujące funkcjonalności: 4](#_Toc168913025)

[Diagram przypadków użycia 6](#_Toc168913026)

[Analityczny diagram klas 7](#_Toc168913027)

[Projektowy diagram klas 8](#_Toc168913028)

[Wymagania niefunkcjonalne 9](#_Toc168913029)

[Przyszła ewolucja systemu 9](#_Toc168913030)

[Scenariusz przypadku użycia „Stwórz drużynę” 10](#_Toc168913031)

[Diagram stanów dla klasy Zadanie 11](#_Toc168913032)

[Diagram aktywności dla przypadku użycia „Stwórz drużynę” 12](#_Toc168913033)

[Projekt GUI dla przypadku użycia „Stwórz drużynę” 13](#_Toc168913034)

[Decyzje projektowe i skutki analizy dynamicznej 18](#_Toc168913035)

# Dziedzina biznesowa i cel

W dzisiejszym dynamicznym świecie branża gier komputerowych stanowi istotny segment rynku rozrywki, przyciągając miliony graczy na całym świecie. Firmy tworzące gry komputerowe znajdują się w środowisku o intensywnej konkurencji, gdzie sukces zależy nie tylko od jakości produktu, ale także od skutecznego zarządzania informacjami oraz szybkiego reagowania na zmiany na rynku. Wobec tego istotne staje się posiadanie systemu, który umożliwi kompleksowe zarządzanie produktami oraz śledzenie wszystkich aspektów związanych z ich tworzeniem, testowaniem i oceną. Obecnie używane narzędzia nie spełniają wymagań firm z branży gier komputerowych. Często charakteryzują się one nieintuicyjnym interfejsem użytkownika, ograniczoną funkcjonalnością oraz niską wydajnością, co prowadzi do utrudnionego zarządzania informacjami i opóźnień w reakcji na zmieniające się warunki rynkowe. Dlatego też powstała potrzeba stworzenia nowego systemu, który będzie odpowiadał specyficznym wymaganiom tej branży.

Celem projektu jest dostarczenie rozwiązania, które pozwoli firmom tworzącym gry komputerowe na skuteczne zarządzanie informacjami o swoich produktach oraz monitorowanie wszystkich działań z nimi związanych. Nowy system ma za zadanie zastąpić obecnie używane narzędzia poprzez dostarczenie intuicyjnego, wydajnego i kompleksowego rozwiązania.

# Zakres odpowiedzialności systemu

System ma umożliwić użytkownikowi (czyli osobie odpowiedzialnej za zarządzanie informacjami w firmie tj. Menadżerowi) sprawne dodawanie informacji o produktach firmy, testach, przeglądach wydajności, oraz wszystkich informacji niezbędnych do stworzenia i zarządzania takimi produktami. Jak i również zarządzanie pracownikami jak i osobami które przyczyniły się do rozwoju produktu takich jak testerzy, aktorzy czy krytycy.

Dodatkowo powinien umożliwić testerom dodawanie informacji o testach oraz krytykom o przeglądach przez nich wykonanych.

# Użytkownicy systemu

System powinien być w stanie przełączać się pomiędzy widokami menadżera który zarządza informacjami, testera, który może zapisywać wykonane testy oraz, krytyka który może zapisywać swoje przeglądy wydajności.

# Wymagania użytkownika

## System powinien przechowywać informacje o:

* Osobach, które pracują przy grze jak i o personelu który monitoruje pracę pracowników. Wyszczególniając
  + Pracownika, którzy mogą należeć do jakiejś drużyny
  + Aktora, który może odgrywać jakieś role w grze
  + Krytyka, który ocenia pracowników.
  + Testera, który dodaje testy do gier.
  + Menedżera, który zarządza informacjami związanymi z grą, drużynami, zadaniami i personelem.
  + PracownikaAktora
* Przeglądach wydajności, które przechowują informacje o rezultatach pracowników
* Testach, które przechowują informacje o tym jaki test został wykonany na grze oraz z jakim rezultatem
* Grach, które zostały/są tworzone przez firmę oraz posiadają
  + Postacie które odgrywane są przez\
    - Aktorów
* Drużynach, które odpowiedzialne są za wykonywanie:
  + Zadań

## Z następującymi ograniczeniami:

* Osoba może być pracownikiem, aktorem lub pracownikiemaktorem
* Osoba może być tylko jednym na raz: Pracownikiem, Aktorem, Krytykiem, Testerem. Lecz może ona zmieniać swoją przynależność
* Pracownik może być przypisany tylko do jednej drużyny.
* Zadanie może zmieniać swój status na „w trakcie”, „zakończone” lub „przerwane”, a termin takiego zadania może ulec zmianie.

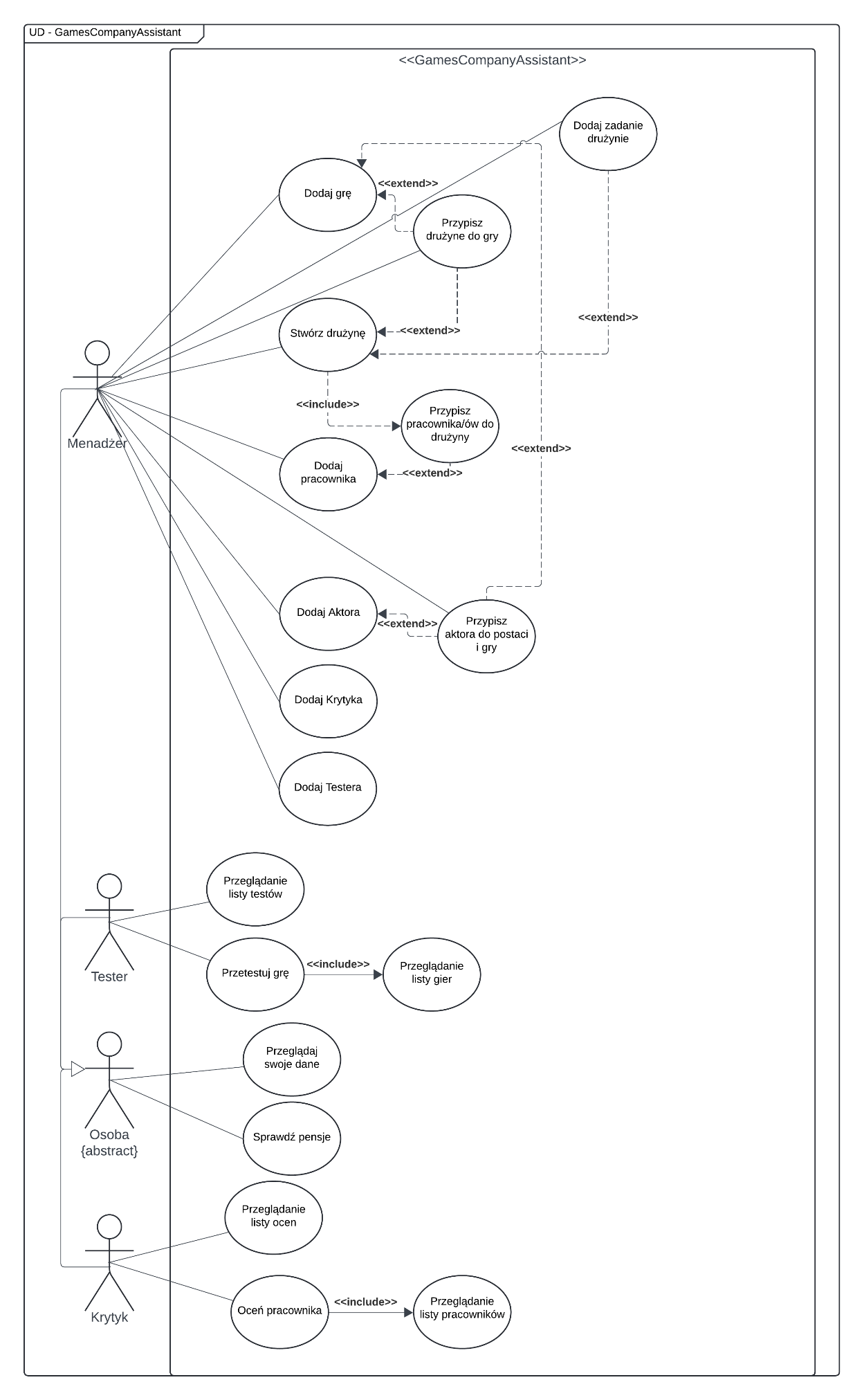
## Dane który powinny być zapamiętane dla każdej klasy:

* Osoba
  + Imię
  + Nazwisko
  + Data urodzenia
  + Email
  + Telefon
  + Pesel (unikalny)
  + Pensja
* Krytyk
  + Wynagrodzenie za ocenę
  + Specjalizacje
* Pracownik
  + Data zatrudnienia
  + Adres
    - Miasto
    - Ulica
    - Nr Domu
    - Kod pocztowy
  + Czas pracy (ilość godzin przepracowywanych w ciągu miesiąca)
* Aktor
  + Zapłatę za linię
  + Ilość linii
* Tester
  + Wynagrodzenie za test
  + Doświadczenie
* Przegląd wydajności
  + Data przeglądu
  + Komentarz
  + Ocena (w skali 1 do 10)
* Drużynę
  + Nazwa drużyny
  + Rola drużyny
* Zadanie
  + Nazwa zadania
  + Opis zadania
  + Termin wykonania zadania
  + Status zadania (może być „w trakcie”, „zakończone” lub „przerwane”)
* Gra
  + Tytuł gry
  + Lista gatunków gry
  + Data wydania (opcjonalna)
* Test
  + Data wykonania testu
  + Komentarz do testu
  + Wynik testu (w postaci „zaliczony” lub „niezaliczony”)

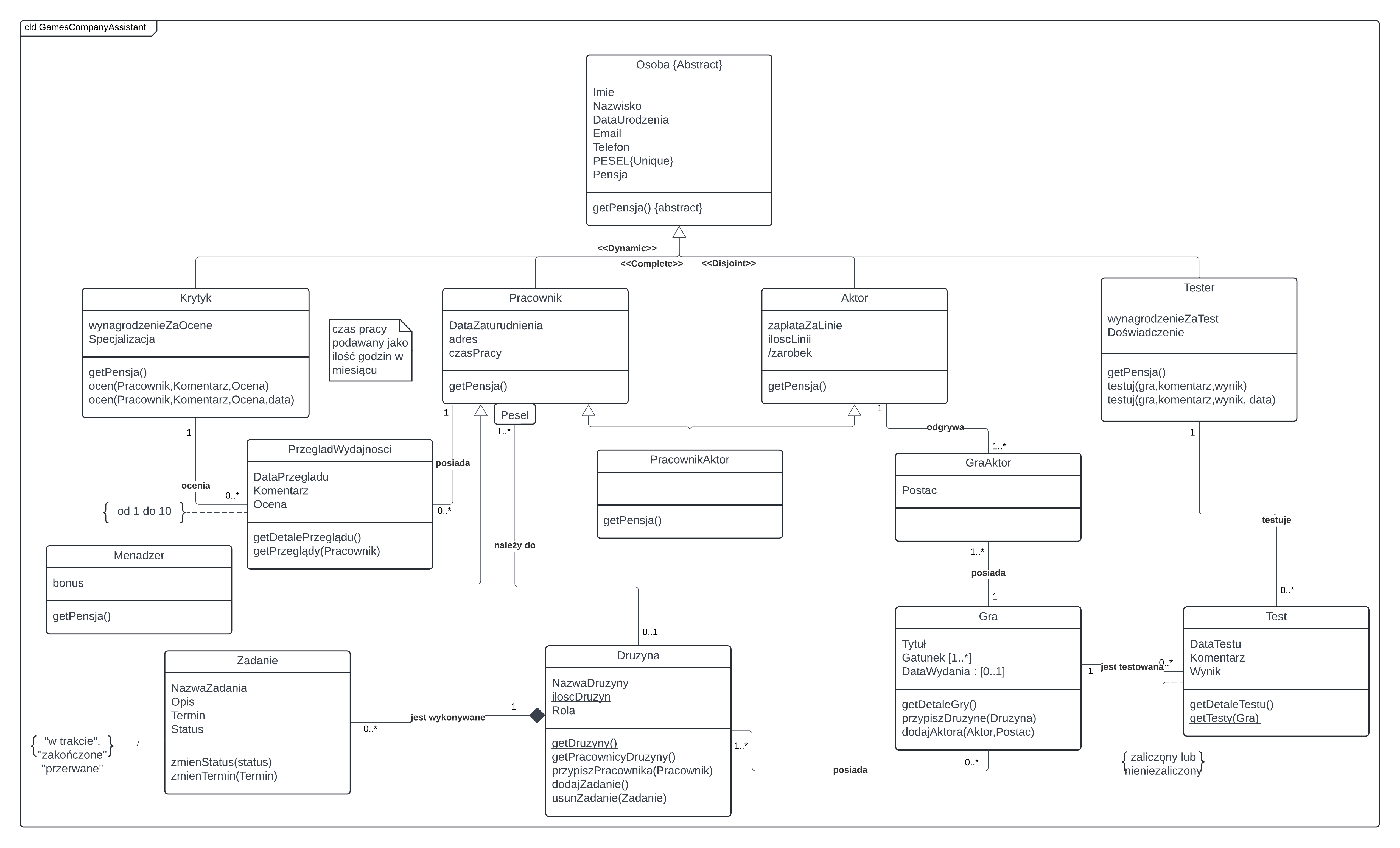
## System powinien umożliwiać następujące funkcjonalności:

* Przeglądanie danych osobowych
* Sprawdzenie pensji osoby
* Przeglądanie listy ocen
* Ocenę pracownika
* Przeglądanie listy pracowników
* Przeglądanie listy testów
* Testowanie gry
* Przeglądanie listy gier
* Dodanie Testera
* Dodanie Krytyka
* Dodanie Aktora
* Dodanie Pracownika
* Stworzenie drużyny
* Dodanie gry
* Przypisanie aktora do postaci i gry
* Przypisanie pracownika do drużyny
* Przypisanie drużyny do gry
* Dodanie zadanie i przypisanie je drużynie

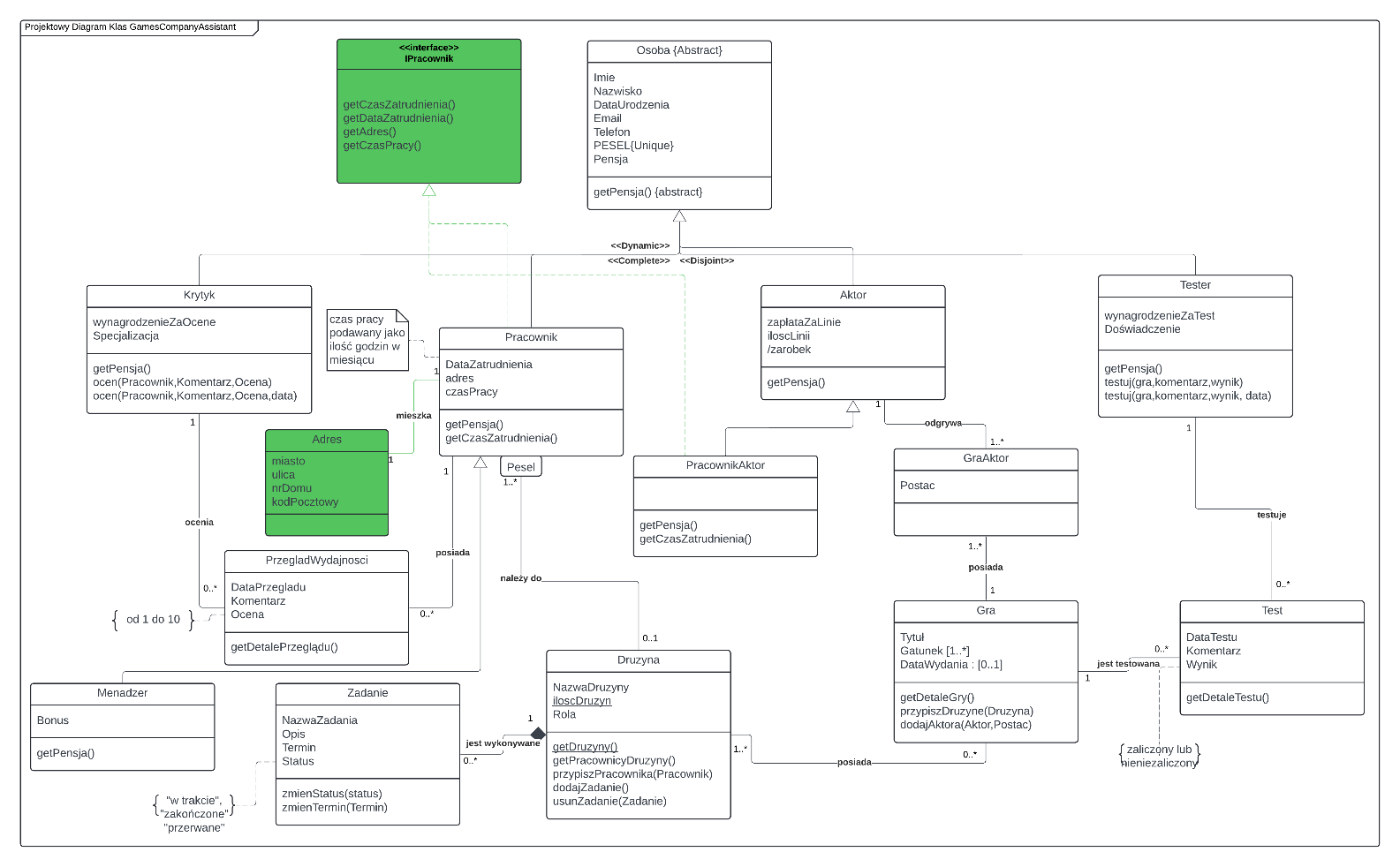
# Diagram przypadków użycia



Analityczny diagram klas

****

Projektowy diagram klas

****

# Wymagania niefunkcjonalne

1. Łatwość użytkowania

System powinien być prosty i intuicyjny do zrozumienia dla użytkowników, którzy nie są zaznajomieni z pracą na komputerze. Wszystkie funkcjonalności powinny być łatwo dostępne i zrozumiałe bez potrzeby długiego szkolenia.

1. Wydajność

System powinien działać płynnie i szybko na aktualnych sprzętach, bez opóźnień i zawieszeń.

1. Bezpieczeństwo

System powinien przechowywać wszystkie dane w zabezpieczonej bazie danych, która jest chroniona przed nieautoryzowanym dostępem i wyciekami.

1. Łatwość rozwoju

Kod systemu powinien umożliwiać łatwe modyfikacje i rozwój aplikacji.

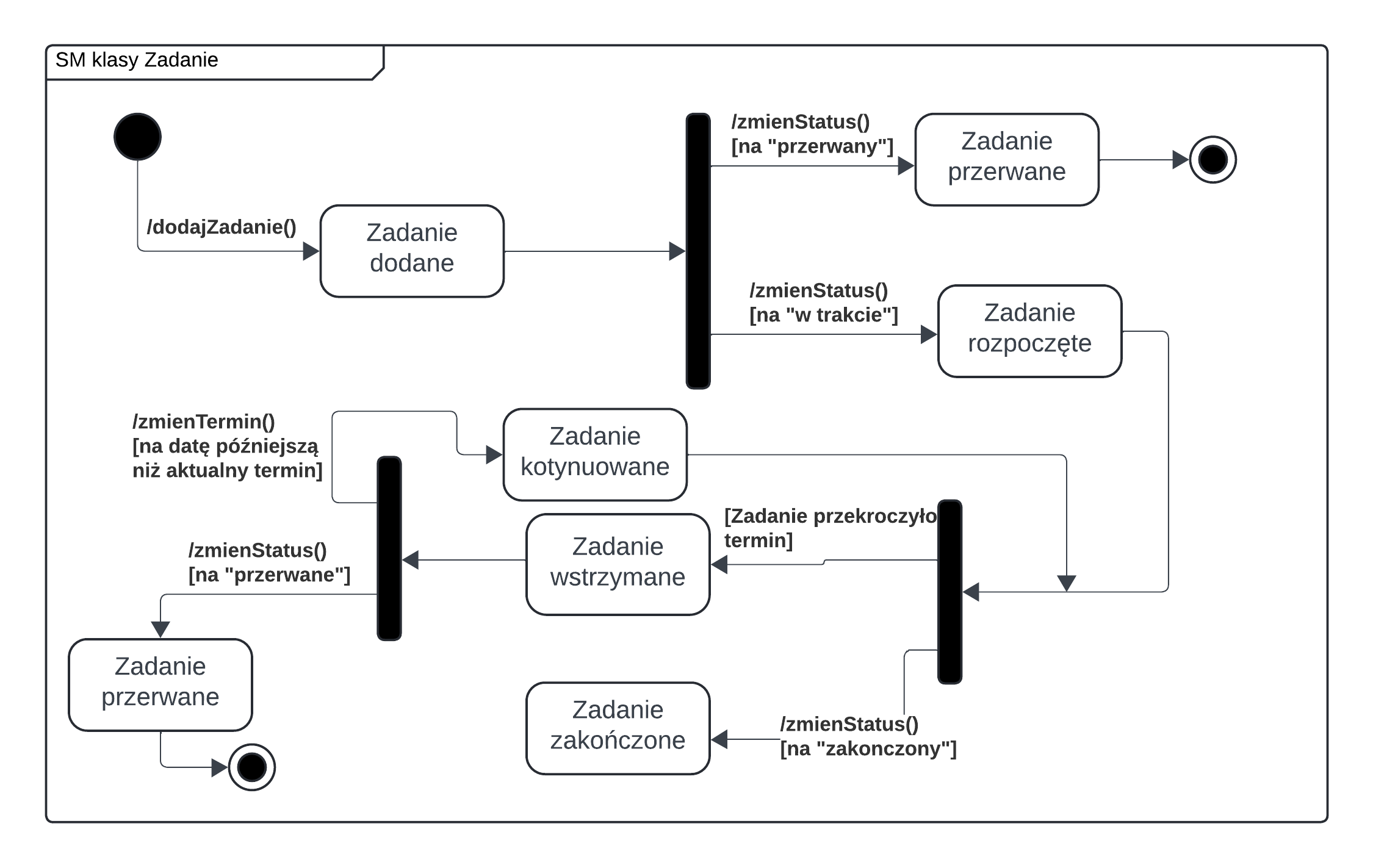
# Przyszła ewolucja systemu

W przyszłości planowane jest dodanie systemu funkcjonalności które, pozwolą na tworzenie większej ilości i bardziej rozbudowanych raportów wraz z możliwością eksportowania ich do najpopularniejszych formatów. Również planowane jest stworzenie aplikacji mobilnej systemu która umożliwi dostęp do systemu z dowolnego miejsca i dowolnym czasie oraz dodanie samouczków oraz systemu podpowiedzi, które pozwolą użytkownikom w pełni wykorzystać możliwości systemu.

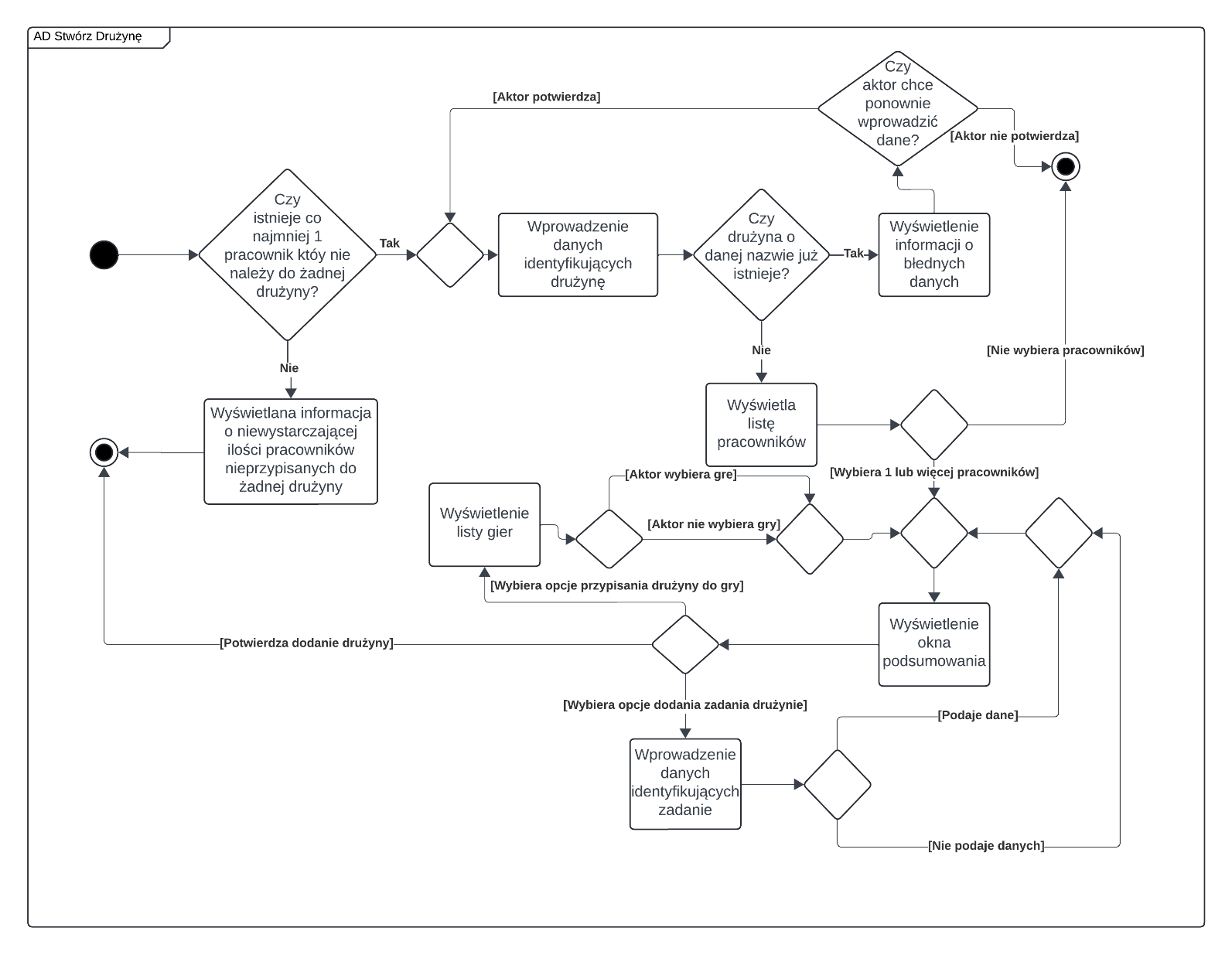
# Scenariusz przypadku użycia „Stwórz drużynę”

|  |  |
| --- | --- |
| **Sekcja** | **Treść** |
| Aktorzy | Menedżer |
| Warunek początkowe | Minimum 1 pracownik w firmie który nie należy do żadnej drużyny |
| Główny przebieg zdarzeń | 1) Aktor rozpoczyna przypadek użycia  2) System prosi o podanie danych identyfikujących drużynę.  3) Aktor podaje dane  4) System sprawdza czy drużyna o podanej nazwie już istnieje w systemie, wyświetla informacje o utworzeniu drużyny oraz wyświetla listę pracowników którzy nie są przypisani do żadnej drużyny  5)Aktor zaznacza 1 pracownika lub więcej  6) System zatwierdza przypisanie pracowników oraz przechodzi do okna podsumowania  7) Aktor potwierdza dodanie drużyny  8) System zapisuje drużynę oraz kończy przypadek użycia |
| Alternatywny przebieg zdarzeń | 4a) System informuje o istnieniu drużyny o danej nazwie, oraz pyta czy aktor chce wprowadzić dane powownie  4aa) Aktor potwierdza. Powrót do punktu 2  4ab) Aktor nie potwierdza. Koniec przypadku użycia.  5a) Aktor nie wybiera pracowników. Koniec przypadku użycia  7a) Aktor wybiera opcje przypisania drużyny do gry  7aa) System wyświetla listę gier oraz prosi o wybranie gry do której aktor chce przypisać drużynę  7aaa) Aktor wybiera grę. Powrót do punktu 6  7aab) Aktor nie wybiera gry. Powrót do punktu 6  7ab) System wyświetla informacje o braku gier. Powrót do punktu 6  7b) Aktor wybiera opcje dodania zadania drużynie.  7ba) System prosi o podanie danych identyfikujących zadanie.  7baa) Aktor podaje dane. Powrót do punktu 6  7bab) Aktor nie podaje danych. Powrót do punktu 6 |
| Warunek końcowy | Utworzona zostanie drużyna oraz przypisany do niej co najmniej 1 pracownik ewentualnie zostanie przypisana do gry i/lub dodane zostanie do niej zadanie |

# Diagram stanów dla klasy Zadanie



# Diagram aktywności dla przypadku użycia „Stwórz drużynę”

****

# 

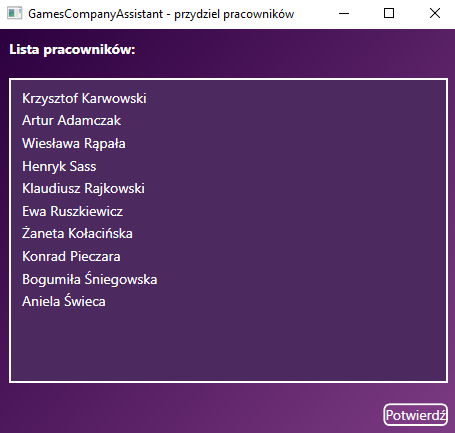
# Projekt GUI dla przypadku użycia „Stwórz drużynę”

- Ekran początkowy przypadku Stwórz drużynę



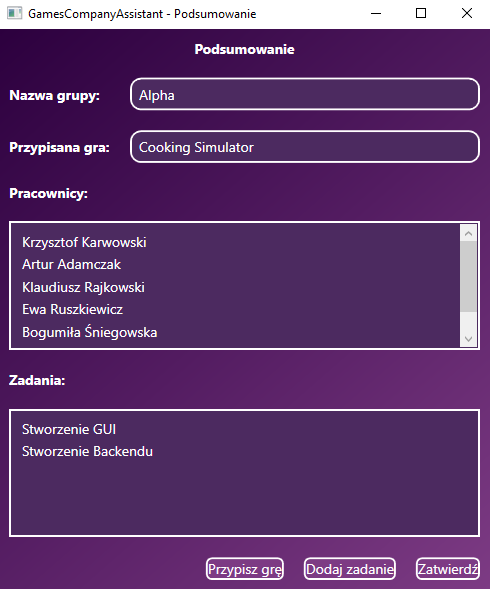
Jest to ekran, który jest uruchamiany podczas rozpoczęcia przypadku „Stwórz drużynę”. Ma on na celu wprowadzenie danych identyfikujących drużynę tj. Nazwę drużyny oraz rolę drużyny.

- Ekran przydzielenia pracowników



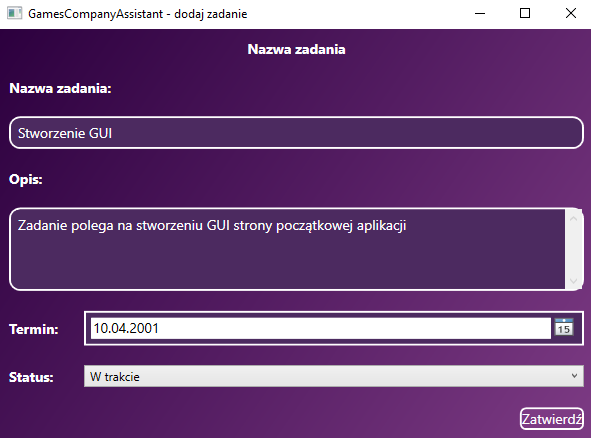
Jest to ekran uruchamiany po zatwierdzeniu danych identyfikujących drużynę, i ma on na celu wielokrotny wybór pracowników w celu przypisania ich do grupy. Jest to lista wielokrotnego wyboru a dane w niej wyświetlane są to pracownicy nieprzypisani jeszcze do żadnej grupy oraz są oni identyfikowani poprzez ich imię oraz nazwisko.

- Ekran podsumowania

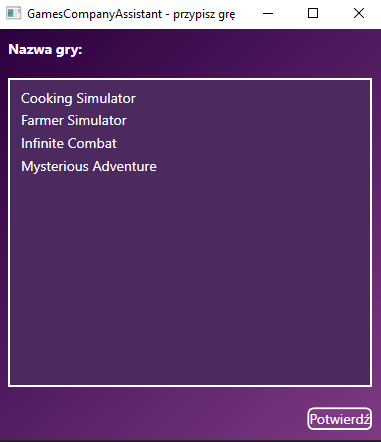


Jest to ekran uruchamiany po zatwierdzeniu listy pracowników którzy mają należeć do nowej grupy. Ekran służy podsumowaniu wszystkich dodanych dotychczasowo danych. Pokazuje on Nazwę którą grupa będzie posiadała, Grę do której grupa może zostać przypisana, Listę pracowników wybranych pracowników oraz zadań które zostały dla niej stworzone. Posiada ona również możliwość kliknięcia w przycisk „Przypisz grę” który otworzy nowe okno służące do przypisania istniejącej gry do grupy, przycisk „Dodaj zadanie” który otworzy nowe okno służące do stworzenia nowego zadania oraz przypisaniu go tej grupie, jak i również przycisk „Zatwierdź” który zapisze nowo dodaną grupę w systemie.

- Ekran dodania nowego zdania



Jest to ekran uruchamiany poprzez kliknięcie w przycisk „Dodaj zadanie” w ekranie podsumowania, służy on stworzeniu nowego zadania i przypisaniu go grupie. Ma on na celu wprowadzeniu danych identyfikujących dane zadania tj. Nazwę zadania, Opis zadania, Termin wykonania zadania wybierany poprzez DatePicker, oraz wybór statusu poprzez rozwijalną listę.

- Ekran przypisania gry  


Jest to ekran uruchamiany poprzez kliknięcie w przycisk „Przypisz grę” w ekranie podsumowania, służy on przypisaniu grupie istniejącej już w systemie gry. Ma on na celu jednokrotny wybór z listy gier, które są identyfikowane poprzez Tytuł danej gry.

# Decyzje projektowe i skutki analizy dynamicznej

* Niektóre klasy posiadają dodatkowe metody, które są niezbędne do prawidłowego działa aplikacji takie jak metody do łączenia z bazą danych.
* Dziedziczenie wieloaspektowe zostało zaimplementowane za pomocą interfejsu
* Atrybut pochodny został zaimplementowany za pomocą klasy wewnętrznej
* Dziedziczenie dynamic zostało zaimplementowane za pomocą dodatkowych konstruktorów w każdej z podklas
* Trwałość danych w systemie uzyskiwana jest poprzez wykorzystanie bazy danych MySQL
* W każdej klasie zostaną zaimplementowane niezbędne dla niej gettery oraz setery
* GUI zostało stworzone poprzez elementy WPF w języku programowania C#, który jest wyborem do implementacji całego projektu.