

System do zarządzania Gildią w grze MMORPG.
„COMPANION”

Opracował:
mikoszo8

Spis treści

System do zarządzania Gildią w grze MMORPG	1
Spis treści	2
1.Dziedzina Problemu	3
2.Cel.....	4
3.Zakres Odpowiedzialności Systemu	5
4.Użytkownicy Systemu	6
5.Wymagania Użytkownika	7
6.Wymagania Funkcjonalne	9
7.Opis Struktury Systemu	10
8. Analiza dynamiczna dla przypadków użycia.....	11
9. Analiza dynamiczna dla wybranej klasy obiektów	18
10.Struktura systemu z uwzględnieniem wyników analizy dynamicznej.....	19
11. Decyzje projektowe	20
12.Schemat logiczny systemu.....	21
13. Projekt interfejsu użytkownika w oparciu o przypadki użycia	22
14. Wymagania Niefunkcjonalne	24
15.Opis Ewolucji Systemu	25
16.Słownik.....	26

1.Dziedzina Problemu

System znajdzie swoje zastosowanie w grach komputerowych typu **massively multiplayer online role-playing game**, czyli w skrócie grach **MMORPG**.

W rozgrywce tego typu wielką rolę odgrywa współpraca pomiędzy graczami.

Dzięki systemowi zarządzania Gildią, grupa potencjalnych poszukiwaczy przygód będzie miała możliwość połączenia się w jedną drużynę, co ułatwi wykonywanie aktywności wewnątrz gry.

2.Cel

Celem oprogramowania będzie udostępnienie graczom przejrzystego i intuicyjnego interfejsu umożliwiającego tworzenie, dołączanie i usuwanie Gildii dla graczy.

Wbudowany system wiadomości i sentencji dnia ułatwi graczom proklamację ważnych informacji tj.: uzgadnianie miejsca i czasu zbiórki do wspólnej zabawy podczas eventu, terminów rajdów lub turniejów PVP czy reklamę swoich przedmiotów na aukcjach(wiadomości dnia).

Założyciel jak i oficerowie, dzięki licznikowi ofiarowanej reputacji przez członków grupy, będą mieli wgląd w poziom zaangażowania i postęp gildii, a zebrane punkty, głowa gildii, będzie mogła wymienić w Sklepie Gildyjnym, na różne bonusy usprawniające rozgrywkę.

3.Zakres Odpowiedzialności Systemu

System powinien umożliwić sprawny proces rekrutacji nowych członków gildii. Będzie on przetrzymywał najważniejsze informacje o tym na jakim **poziomie** postaci aktualnie znajduje się gracz, **gdzie jest** i ile **reputacji** zdobył w imieniu swojego klanu.

Udostępniony zostanie również dział **eventów** – zarządca będzie miał możliwość tworzenia różnego typu wydarzeń dla wszystkich członków gildii.

Założyciel dzięki **punktom reputacji** zdobytym dzięki wkładowi członków, będzie miał możliwość zakupu dodatkowych **bonusów** oddziałujących w korzystny sposób na wszystkich członków.

Dodatkowo system umożliwi funkcjonalności takie jak **wiadomość dnia**, **sentencje dnia** u założyciela, oraz **ranking** osób przynależących.

4.Użytkownicy Systemu

- Założyciel
- Oficer
- Aplikant
- Członek Gildii

5. Wymagania Użytkownika

System powinien przechowywać informacje o:

- A. Gildii
- B. Członkach gildii.
- C. Oficerach
- D. Założycielu
- E. Aplikantach
- F. Sklepie Gildyjnym

Gildia

Dane powinny zawierać:

- Nazwa Gildii
- Data założenia
- Nazwę założyciela
- Frakcję Gildii
- Listę Osiągnięć
- Listę Eventów
- Listę Logów
- Listę Aplikantów
- Listę Członków
- Listę Formularzy Aplikacyjnych

Sklep Gildyjny

Dane powinny zawierać:

- Listę Boosterów

Członek gildii

Dane powinny zawierać:

- Nazwę postaci
- Poziom
- Klasę
- Wiadomość dnia
- Gdzie gracz się znajduje
- Posiadane punkty reputacji
- Wkład punktów reputacji
- Datę przystąpienia
- Listę Ekwipunku
- Listę Eventów
- Do jakiej Gildii należy

Oficer

Oficer traktowany jest jak Członek gildii.

Założyciel

Założyciel traktowany jest jak Oficer.

Dane powinny zawierać:

- Sentencje dnia (edytowalna, krótka, wiadomość do wszystkich członków)

Aplikant

Dane powinny zawierać:

- Listę Formularzy Aplikacyjnych
-

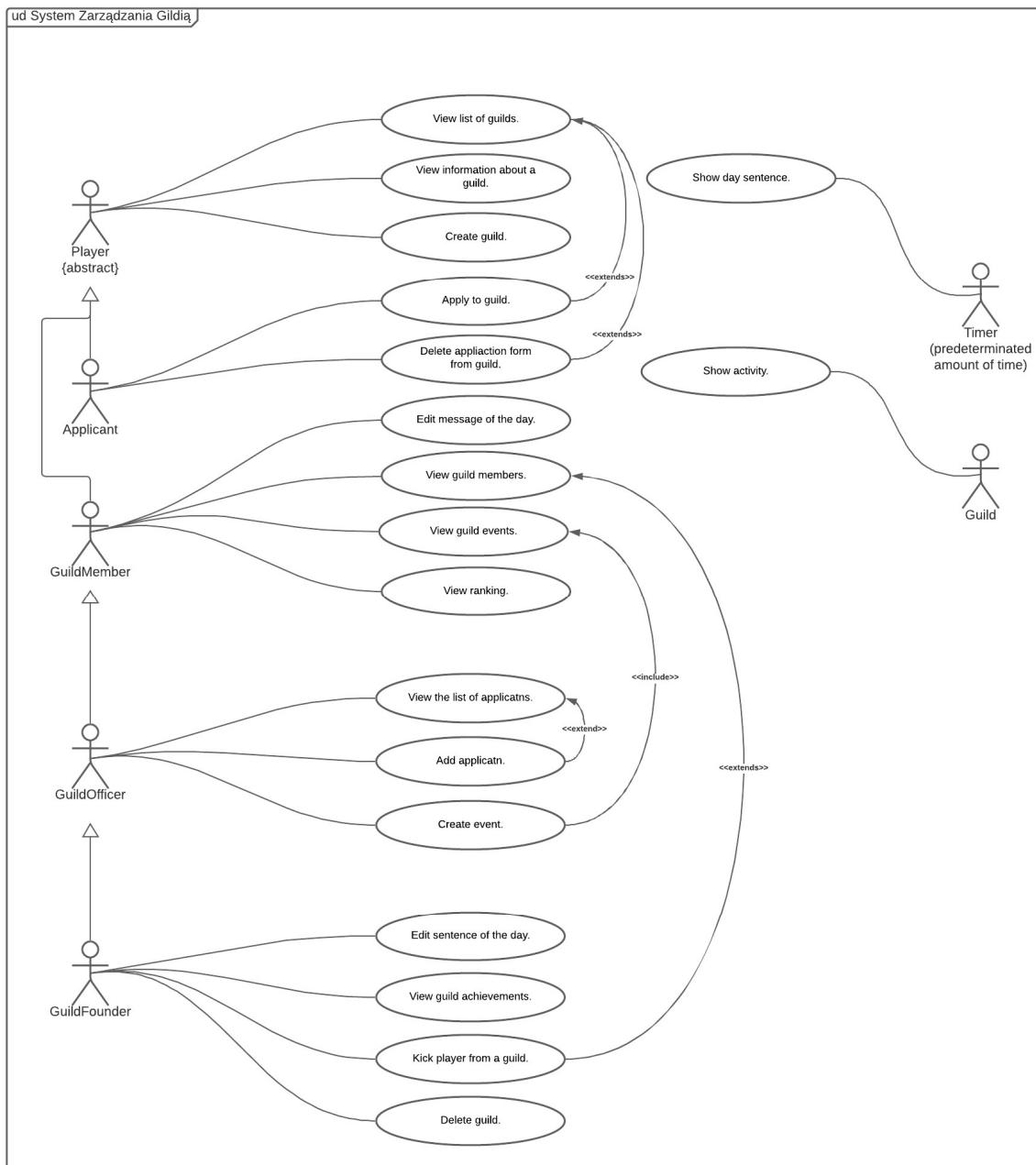
Oczekuje się, że system będzie wspomagał użytkowników w następujący sposób:

- Członkom powinien umożliwić funkcję dołączenia do nowej gildii, podgląd informacji wewnętrz samej gildii (listy członków, eventów itp.), możliwość zapisania się na wydarzenie, edycję swojej wiadomości dnia oraz podgląd rankingu członków gildii.
 - Oficer posiada wszystkie funkcjonalności członka gildii, dodatkowo ma dostęp do listy aplikantów, dzięki czemu będzie mógł dodawać nowych członków (nie usuwać), dodatkowo może tworzyć nowe Eventy dla graczy.
 - Założycielowi powinien umożliwić dostęp do wszystkich opcji oficera, dodatkowo edycję sentencji dnia, możliwość usunięcia członka gildii jak i samej gildii.
 - Aplikant ma możliwość wysłania prośby o dodanie do grupy, może dodać opcjonalną wiadomość polecającą, dodatkowo może usunąć wysłany wcześniej formularz.
 - Gildia powinna mieć możliwość wyświetlania sentencji dnia w wybranym przez założyciela czasie jak i logować najważniejsze zdarzenia wykonane przez m.in. członków gildii.
-

Aplikacja zarządzania gildią powinna spełniać następujące wymagania:

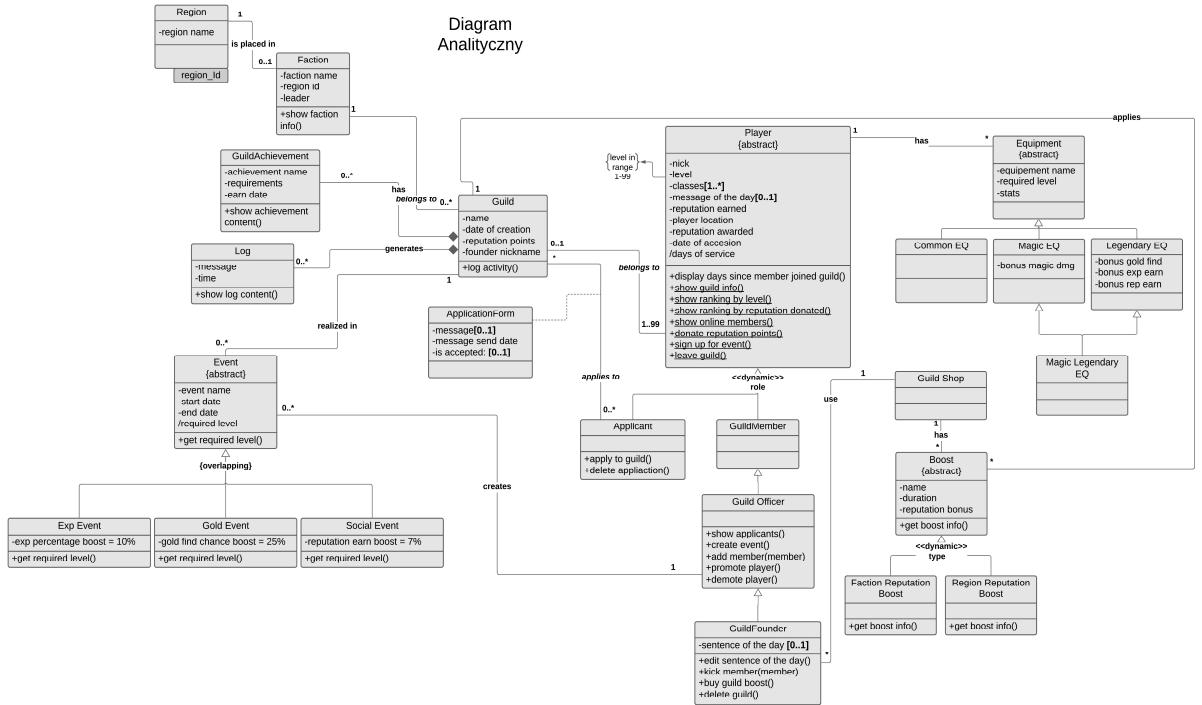
- Powinna działać w środowisku Windows 10 oraz Windows 7
- Będzie zaimplementowana w środowisku Inteli J Idea w języku Java 16.
- W przypadku awarii powinna zostać naprawiona w przeciągu 24h.

6. Wymagania Funkcjonalne



Rysunek 1 Nowy Diagram przypadków użycia

7. Opis Struktury Systemu

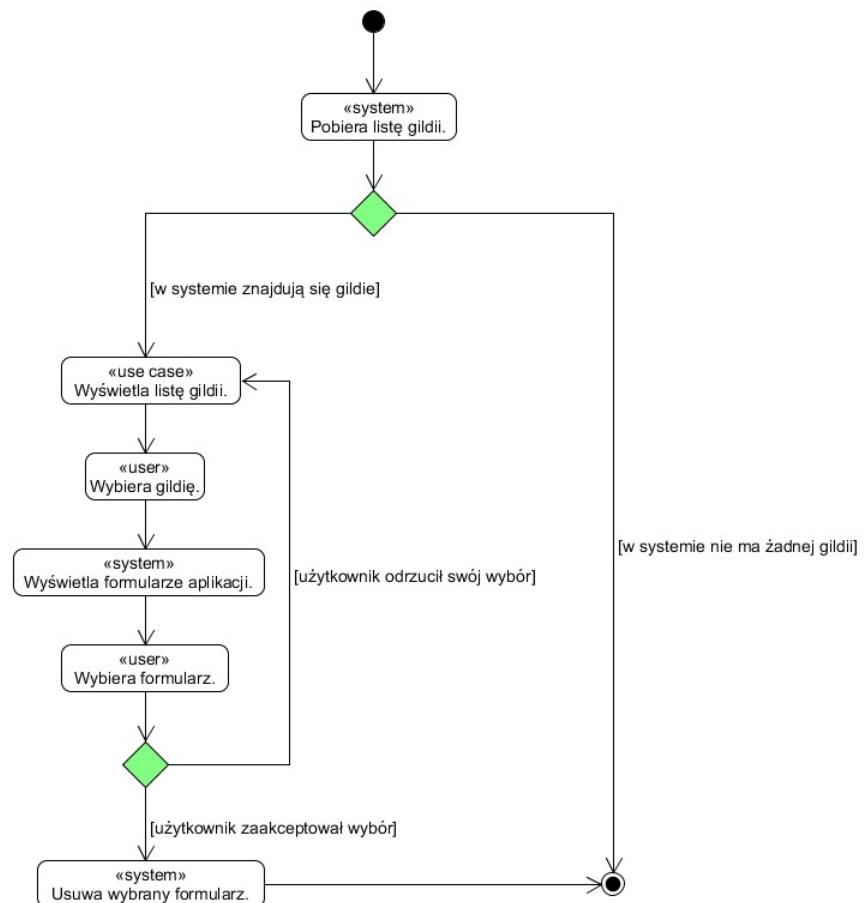


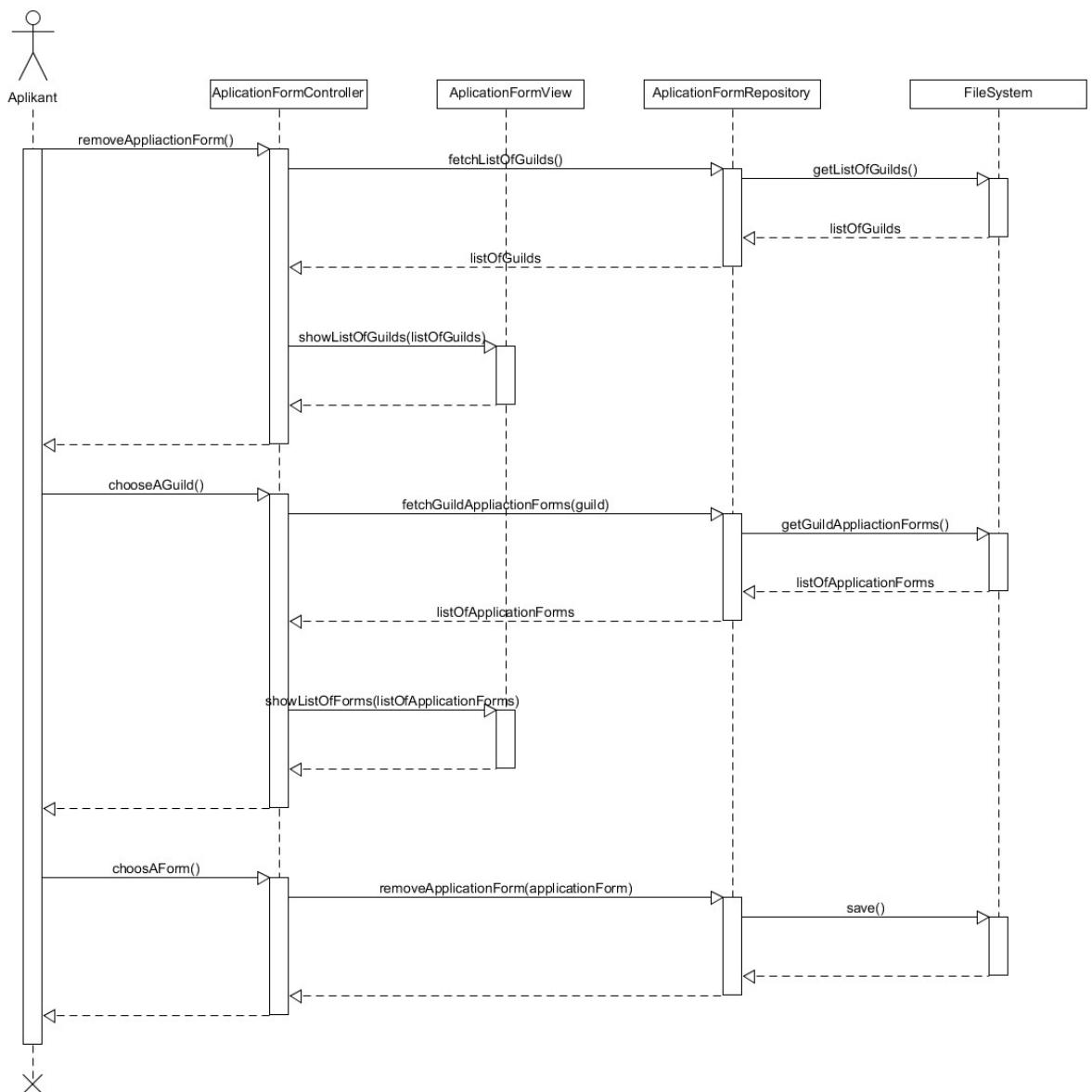
Rysunek 2 Diagram Analityczny

8. Analiza dynamiczna dla przypadków użycia

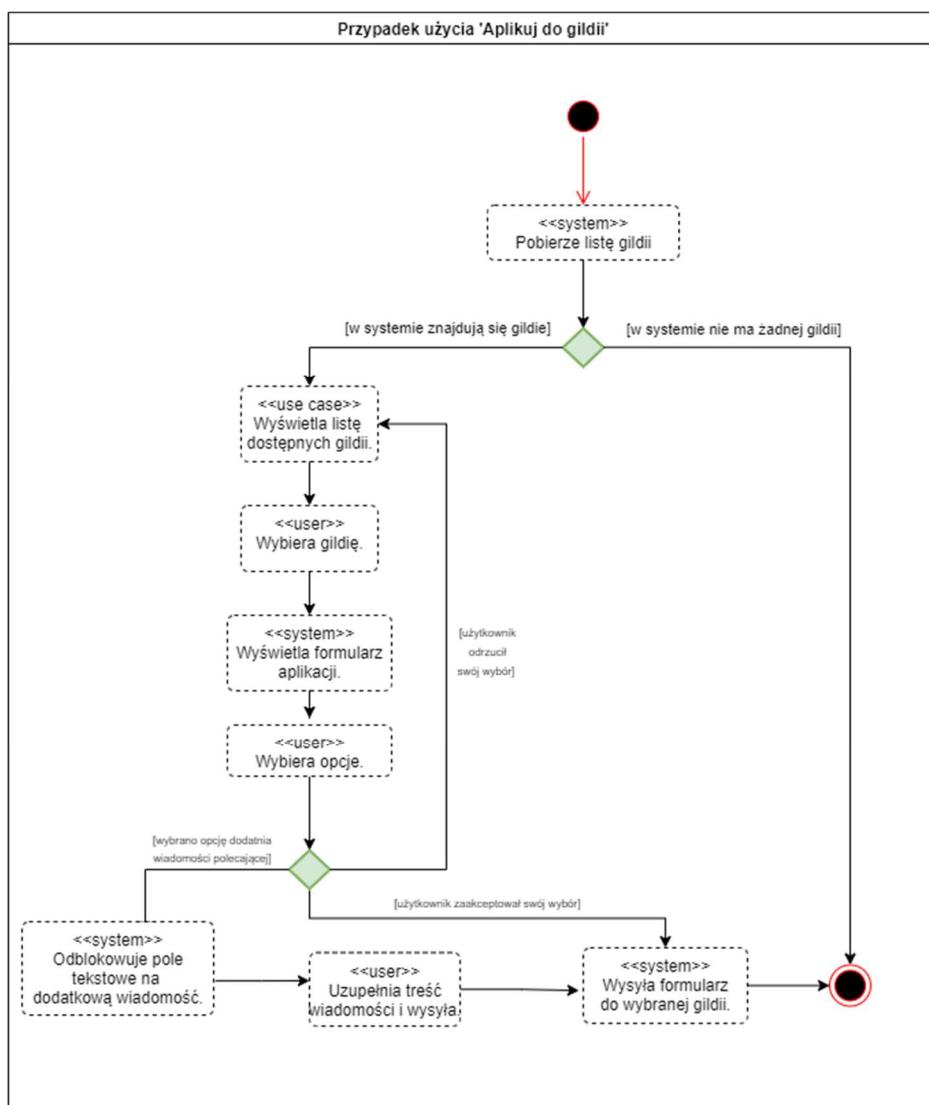
	Usuń aplikację
Warunek początkowy	Istnienie Gildii w systemie. Istnienie Formularza Aplikacji w gildii.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor Aplikant uruchamia przypadek użycia. 2. System wyświetla listę gildii. 3. Aktor wybiera gildię, z której chce usunąć aplikację. 4. System wyświetla listę formularzy Aplikanta. 5. Aktor wybiera, który formularz ma zostać usunięty i akceptuje swój wybór.. 6. System usuwa wybrany formularz. 7. System kończy przypadek użycia.
Alternatywny przepływ zdarzeń	
Zakończenie	W dowolnym momencie

(Przykład nawigacji po asocjacji do punktu 4.2.4.)

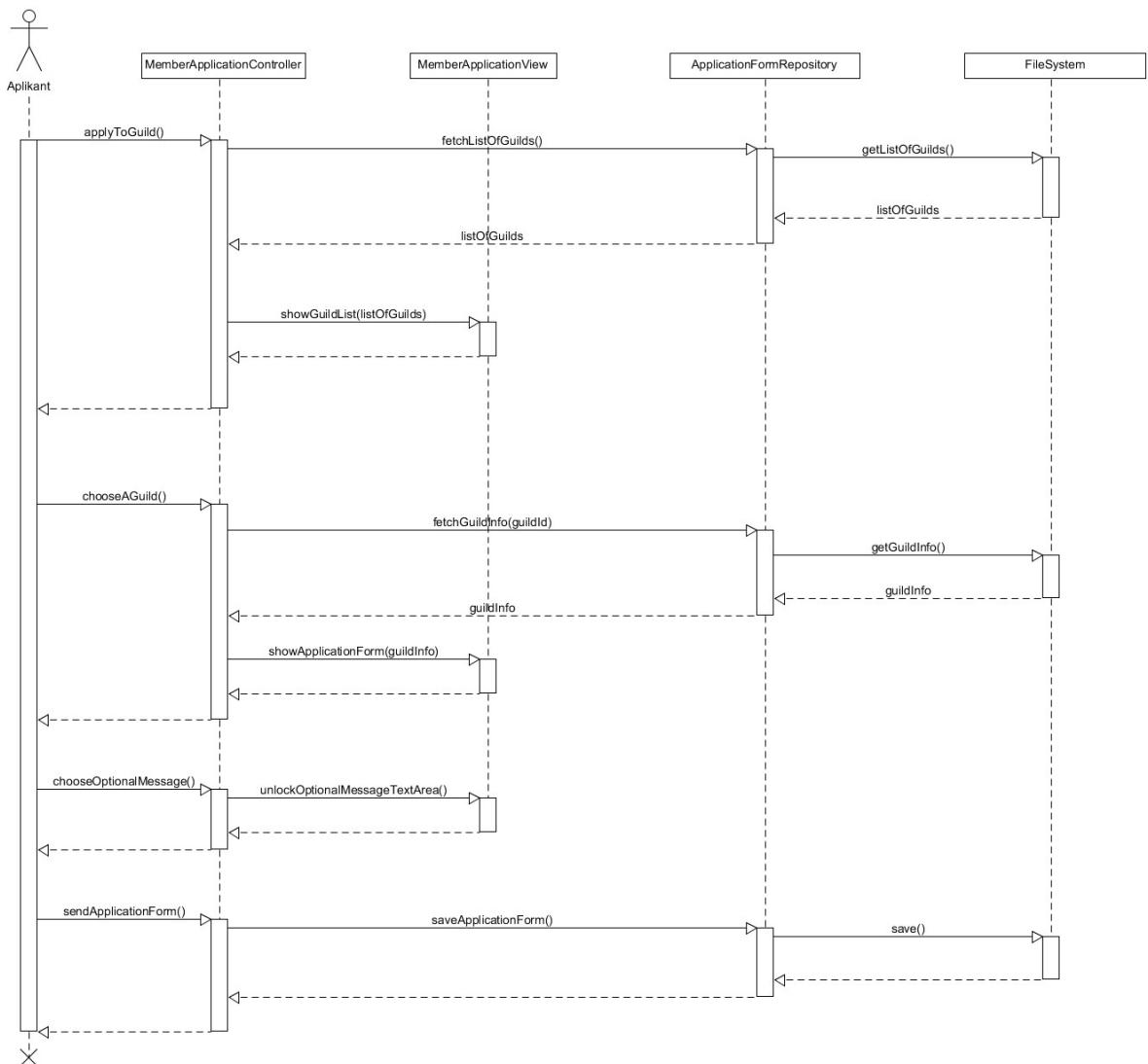




Aplikuj do gildii	
Warunek początkowy	Istnienie Gildii w systemie.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor Aplikant uruchamia przypadek użycia. 2. System wyświetla listę gildii. 3. Aktor wybiera odpowiadającą mu gildię. 4. System wyświetla formularz aplikacji. 5. Aktor akceptuje swój wybór i wysyła prośbę do danej gildii. 6. System kończy przypadek użycia.
Alternatywny przepływ zdarzeń	<ol style="list-style-type: none"> 5a. Aktor wybiera możliwość dodania wiadomości polecającej. 6. System odblokowuje odpowiednie pole tekstowe na wprowadzenie tekstu. 7. Aktor uzupełnia treść wiadomości i akceptuje swój wybór. 8. System kończy przypadek użycia.
Zakończenie	W dowolnym momencie



Rysunek 3 Diagram aktywności Aplikuj do Gildii

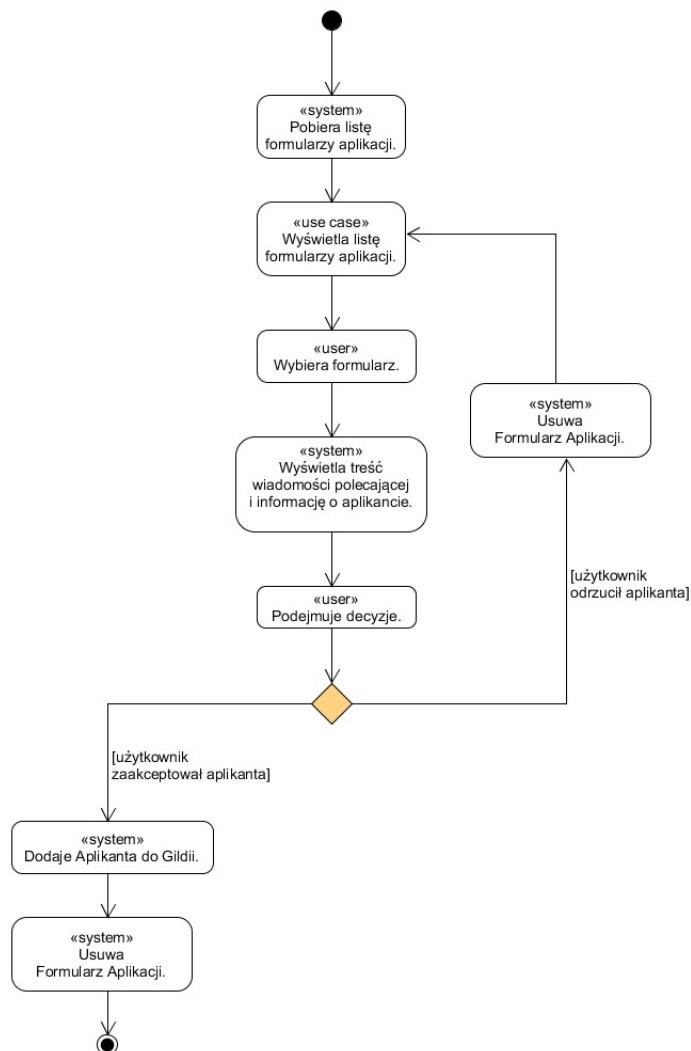


Rysunek 4 Diagram Sekwencji Aplikuj do Gildii; ścieżka z opcjonalną wiadomością

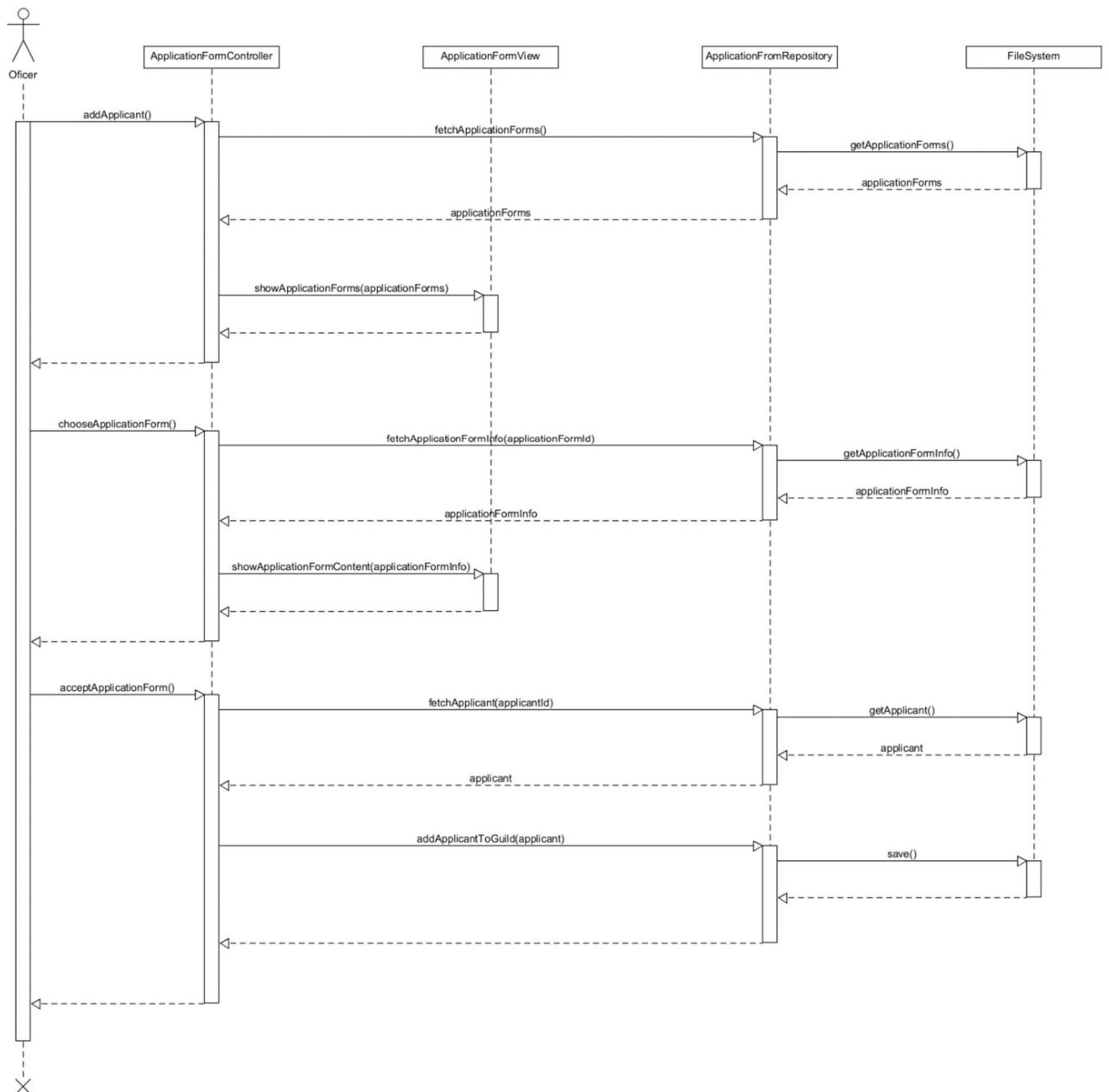
W tym przypadku użycia metodami niezbędnymi będą te pobierające dane z plików. Potrzebnymi informacjami będą Gildie i informacje o nich, oprócz tego ważnym procesem będzie jeszcze zapisanie tych danych do pliku, umożliwią nam to metody, umieszczone w klasie odpowiedzialnej za komunikację z lokalnym repozytorium.

Dodatkowo nowymi metodami niezbędnymi do zrealizowania tego przypadku użycia będą te odpowiedzialne za wybór Gildii z listy, odblokowanie pola na opcjonalną wiadomość polecającą jak i samo wysłanie aplikacji do zapisu.

	Dodaj aplikanta
Warunek początkowy	Istnienie Gildii w systemie. Istnienie Formularza Aplikacji w systemie. Istnienie Aplikanta w systemie
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Aktor Oficer uruchamia przypadek użycia. 2. System wyświetla listę Formularzy Aplikacji 3. Aktor wybiera Formularz Aplikacji 4. System wyświetla informację o Aplikancie i treść opcjonalnej wiadomości polecającej. 5. Aktor akceptuje decyzje o przyjęciu nowego członka i zatwierdza wybór. 6. System dodaje Aplikanta do Gildii 7. System usuwa Formularz Aplikacji 8. System kończy przypadek użycia
Alternatywny przepływ zdarzeń	<ol style="list-style-type: none"> 5a. Aktor odrzuca aplikacje 6. System usuwa Formularz Aplikacji 7. Powrót do punktu 2
Zakończenie	W dowolnym momencie

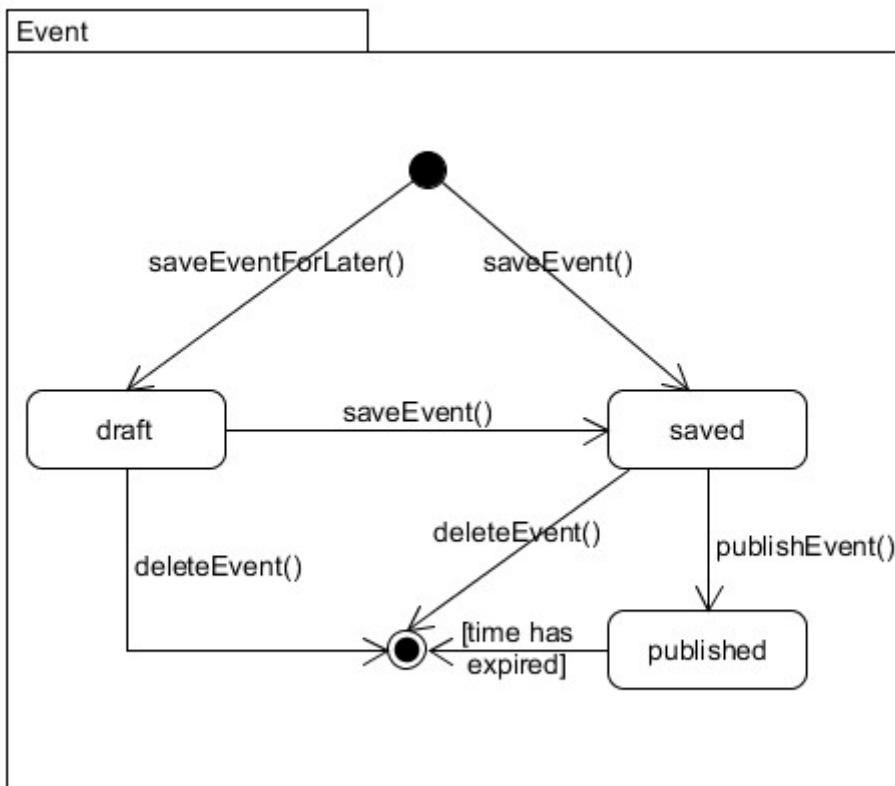


Rysunek 5 Diagram aktywności Dodaj Aplikanta



Rysunek 6 Diagram sekwencji Dodaj Aplikanta

9. Analiza dynamiczna dla wybranej klasy obiektów

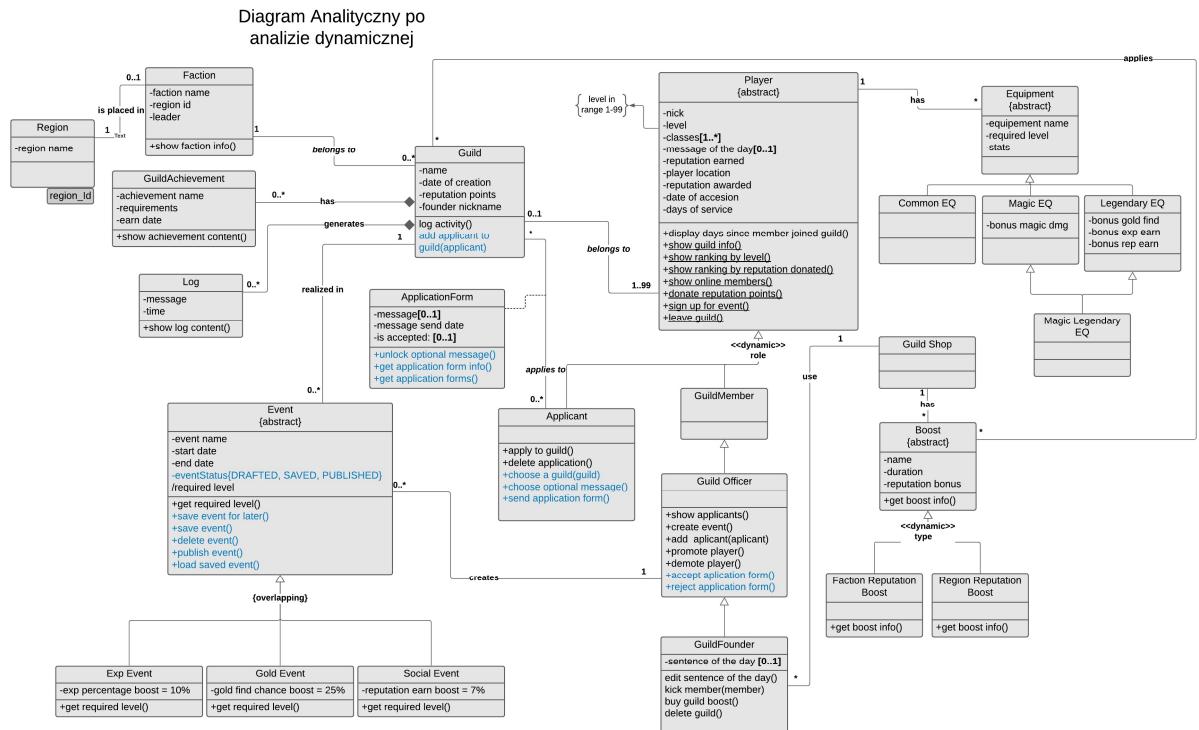


Rysunek 7 Diagram Sekwencji dla Eventu

Skutkiem przeprowadzonej analizy było wykrycie nowych funkcjonalności systemu.

Oficer, który będzie tworzył dany Event będzie musiał mieć dostęp do metod umożliwiających pracę nad nowymi wydarzeniami tj. zapis jako **kopia robocza**, zapis jako **wersja gotowa** do publikacji, **usunięcie** gdy posiada stan **draft** lub **saved**, **publikację** dla wszystkich członków danej Gildii oraz **wczytanie** zapisanej wcześniej wersji.

10. Struktura systemu z uwzględnieniem wyników analizy dynamicznej



Rysunek 8 Diagram Analityczny po analizie dynamicznej

11. Decyzje projektowe

Implementacja konstrukcji

Atrybuty pochodne – wartość takiego atrybutu będzie wyliczana w metodach pomocniczych na podstawie innych atrybutów.

Asocjacje – związki między klasami będą określone poprzez zapamiętywanie w nich, w listach lub pojedynczych obiektach, informacji, że są częścią jakiegoś innego elementu programu.

Związki wiele-do-wielu – asocjacja z atrybutem zostanie zamieniona na 3-klasową asocjacje [1..*] i [*..1] (np.: Gildia [1..*] Formularz_Aplikacji [*..1] Aplikant).

Ograniczenia – wszelkie modyfikacje atrybutów ograniczonych jakąś zasadą będą wcześniej sprawdzane i przetwarzane poprzez metody set i get, czyli zastosowana zostanie hermetyzacja.

Dziedziczenie overlapping – w przypadku Eventu użyto metody nakładających się na siebie klas, powodem takiej implementacji jest fakt, że w systemie będzie istnieć wiele wersji danego wydarzenia np.: Event wzmacniający przyrost złota i reputacji jednocześnie.

Dziedziczenie dynamiczne – w przypadku Graczy i Boostów w sklepie gildijnym użyto sposobu dynamicznej zmiany jednego obiektu w drugi, bez potrzeby wcześniejszego usuwania i tworzenia instancji na nowo. To znaczy, że każdy gracz w każdej chwili może na przykład opuścić gildię i stać się Aplikantem, lub zostanie awansowany na stanowisko Oficera.

Wielo-dziedziczenie – w przypadku Ekwipunku gracza możemy natknąć się na przypadek przedmiotu, który byłby Magiczny i zarazem Legendarny, przez co odziedziczyłyby atrybuty obu tych typów.

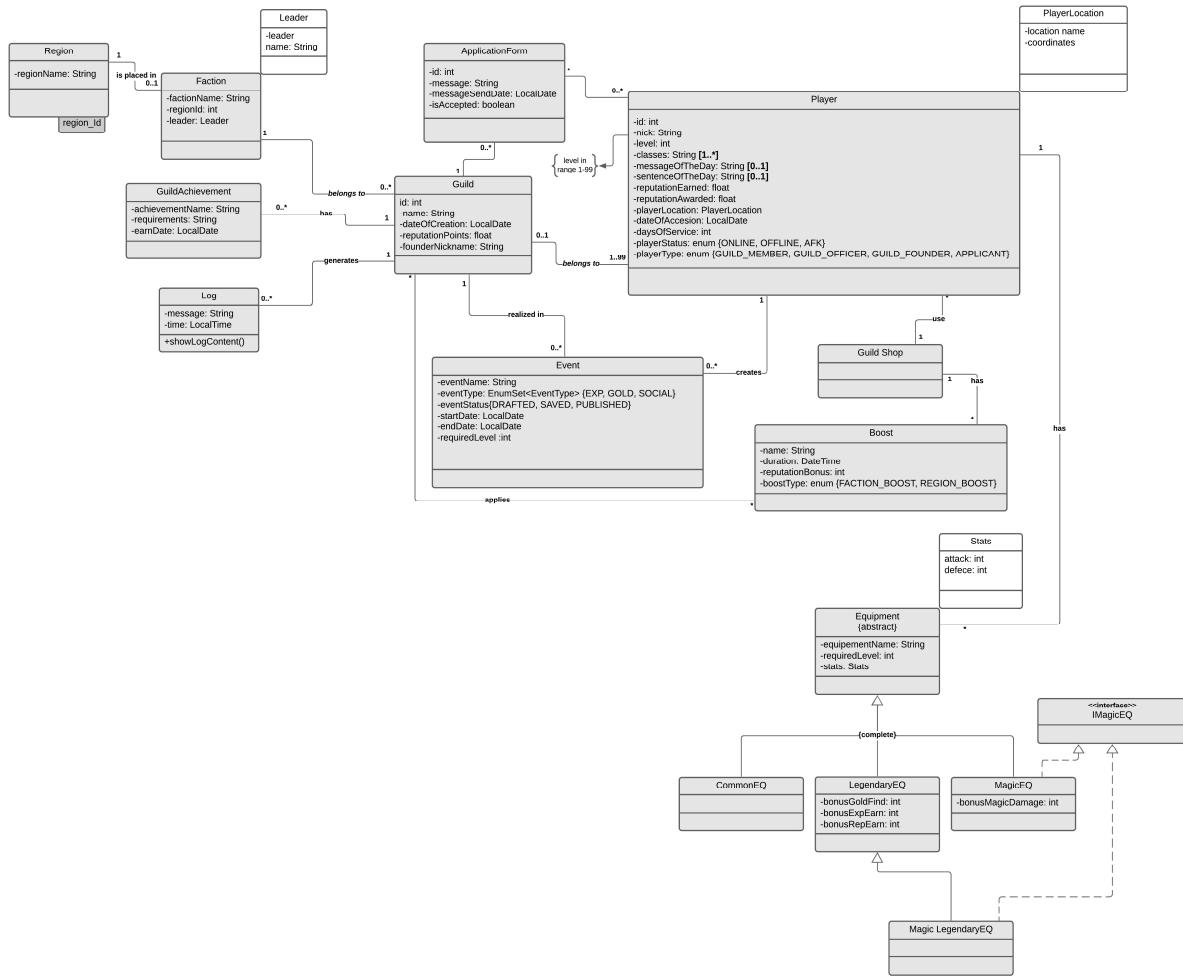
Przechowywanie danych

Informację o obiektach i ich asocjacjach będą przechowywane przy pomocy plików systemowych i serializacji.

Zrezygnowano z użycia dziedziczenia po klasie pomocniczej ‘ObjectPlus’, w zamian ekstensje wszystkich klas będą zapisywane przy pomocy mechanizmu, który będzie zapisywał, bądź wczytywał je w jednym momencie z i do pliku.

12. Schemat logiczny systemu

Diagram
Projektowy



Rysunek 9 Diagram Projektowy

13. Projekt interfejsu użytkownika w oparciu o przypadki użycia



Rysunek 10 Projekt GUI

Przypadek użycia: „Usuń aplikację”.

(Przykład nawigacji po asocjacji do punktu 4.2.4.)

Krok I

Aplikant uruchamia przypadek użycia przyciskiem „Remove Appliaction”.

Krok II

System wyświetla listę Gildii.

Aplikant wybiera Gildię.



Rysunek 11 okno z listą gildii

Krok III

System wyświetla połączone, z wybraną Gildią, Formularze Aplikacji wysłane przez Aplikanta. Aplikant wybiera Formularz Aplikacji do usunięcia.



Rysunek 12 okno z listą aplikacji

Krok IV

System usuwa wybrany formularz i kończy przypadek użycia.

Przypadek użycia: „Dodaj Aplikanta”

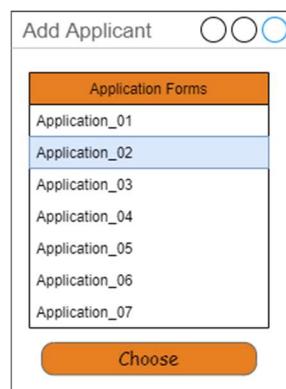
Krok I

Założyciel lub Oficer poprzez przycisk „Add Applicant” uruchamiają przypadek użycia.

Krok II

System wyświetla przed użytkownikiem nowe okno z listą zapisanych formularzy aplikacji.

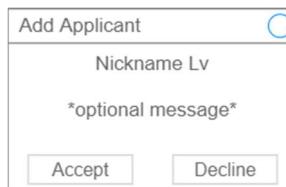
Oficer w tym momencie może zaznaczyć i wybrać odpowiadającego mu kandydata.



Rysunek 13 okno z listą formularzy aplikacji danej gildii

Krok III

System wyświetla okno dialogowe z informacjami wydobytymi z formularza aplikacji, tj. nazwa gracza, poziom, opcjonalna wiadomość polecająca.



Rysunek 14 okno decyzji dodania członka do gildii

Krok IV

Użytkownik podejmuje decyzję.

System dodaje aplikanta w przypadku akceptacji lub usuwa formularz zgłoszeniowy w drugim przypadku i kończy przypadek użycia.

14. Wymagania Niefunkcjonalne

Reputacja – liczby zmiennoprzecinkowe, nieujemne.

Np.: ‘gracz przetransferował 1135.15 punktów reputacji na rzecz gildii.’

Poziom – liczby całkowite nieujemne od 0 do 99.

Lokalizacja – nazwa dokładnej lokalizacji gracza z koordynatami.

Np.: ‘Wichrowy Gród (x:15, y:34)’

Liczba członków – maksymalnie 99 członków w jednej gildii.

Boost do reputacji – liczba niezmienna przecinkowa, nieujemna.

Np.: ‘+15% do zdobywanego złota.

15.Opis Ewolucji Systemu

System w przyszłości można rozwinąć o dodatkową funkcjonalność jaką byłby panel aukcyjny. Gracze mogliby tworzyć oferty kupna/sprzedaży swojego ekwipunku.

16. Słownik

MMORPG - rodzaj komputerowych gier fabularnych, w których duża liczba graczy może grać ze sobą w wirtualnym świecie.

Gildia – grupa osób grających w konkretną grę komputerową wieloosobową przez Internet.

Nick – przewisko, pseudonim.

Poziom/Level Postaci – symbolizowany najczęściej liczbą w zakresie 0-99. Symbolizuje stopień doświadczenia zdobytego przez danego gracza.

Event – wydarzenie trwające przez określoną ilość czasu, zwykle oferujące dodatkowe nagrody.

Boost – czasowe ulepszenie, podnoszące statystyki.

EQ – inaczej ekwipunek gracza.

Achievement – osiągnięcie zdobywana po spełnieniu określonych warunków.