Marcin Nasiłowski 16.02.2018

WYMAGANIA WSTĘPNE

DZIEDZINA PROBLEMOWA:

Edukacja, popularyzacja nauki, konkursy.

CEL

Celem systemu jest obsługa złożonego konkursu internetowego

ZAKRES ODPOWIEDZIALNOŚCI SYSTEMU:

Projektowany system ma być platformą do obsługi konkursu online. Głównym jego zadaniem będzie rejestrowanie zespołów konkursowych oraz przyjmowanie zrealizowanych przez nie prac. System powinien również umożliwiać ocenę złożonych prac i ich przeglądanie. Przy pomocy tego narzędzia powinno być możliwe wyświetlenie informacji o konkursie. Dodatkową funkcjonalnością systemu będzie możliwość kontaktu z organizatorami oraz mentorami.

UŻYTKOWNICY SYSTEMU:

Uczestnicy konkursu ich opiekunowie, osoby oceniające prace, mentorzy, goście, organizatorzy.

WYMAGANIA UŻYTKOWNIKA:

1. wszystkich osobach zarejestrowanych w systemie mają być przechowywane następujące informacje: tytuł(opcjonalne), imię, nazwisko, data urodzenia, płeć(opcjonalne), adres email, numer telefonu (opcjonalne), afiliacja(opcjonalne), typ rejestracji(uczestnik, opiekun, osoba oceniająca prace, mentor), kraj zamieszkania, krótkie CV (obowiązkowe w przypadku mentorów oraz osób oceniających prace).
2. Uczestnicy i opiekunowie mogą rejestrować zespoły w jednej z trzech kategorii
- od 7 do 13 lat. Zespoły 1 osobowe.
- od 14 do 18 lat. Zespoły 2 lub 3 osobowe.
- Od 19 do 22 lat. Zespoły 1 lub 2 osobowe.
3. Jeden uczestnik może należeć tylko do jednego zespołu, natomiast opiekun może zarejestrować kilka zespołów.
4. System powinien przechowywać następujące dane o zespole: kategorię wiekową, nazwę zespołu, imię i nazwisko poszczególnych członków zespołu oraz ich wiek. W przypadku kategorii wiekowych a i b system powinien również przechowywać dane o opiekunie zespołu.
5. System powinien sprawdzać czy w innych zespołach nie ma tych samych osób.
6. Opiekunowie zespołów muszą być osobami pełnoletnimi.
7. Na podstawie daty urodzenia system powinien sprawdzać czy dana osoba może uczestniczyć w konkursie w danej kategorii wiekowej.
8. System powinien wysyłać uczestnikom przypomnienie o zbliżającym się końcu przyjmowania prac konkursowych.
9. Po zakończonym ocenianiu prac w pierwszym etapie każdy z uczestników powinien otrzymać informację o swoim wyniku wraz z komentarzem od osób oceniających prace oraz informację o tym czy jego zespół zakwalifikował się do kolejnego etapu.
10. System powinien wspomagać użytkowników w realizowaniu funkcjonalności takich jak np.: Organizatorzy powinni móc:
    - akceptować rejestracje osób oceniających prace oraz mentorów
    - przypisywać poszczególne prace konkretnym osobom oceniającym
    - przeglądać w dowolnym momencie dane zarejestrowanych osób oraz złożone przez nie prace.

- Wysyłanie wiadomości do wszystkich osób zarejestrowanych, z możliwością zawężenia listy odbiorców do uczestników danej kategorii wiekowej z danego kraju.

11. Uczestnicy oraz opiekunowie powinni móc:
    - rejestrować zespoły konkursowe. Przy czym uczestnik może zarejestrować tylko zespół którego jest członkiem natomiast opiekun może rejestrować wiele zespołów.
    - Edytować dane członków zarejestrowanego zespołu w terminie określonym przez regulamin.
    - Dodawać i usuwać pliki projektu konkursowego do daty określonej w regulaminie. System powinien sprawdzać czy format przesłanych plików jest zgodny z dozwolonym przez regulamin dla danej kategorii wiekowej.
    - Wyszukiwać Mentorów ze względu na dziedzinę którą się zajmują
12. Osoby oceniające powinny móc:
    - Przeglądać pliki przypisanych prac.
    - Wystawić każdej z przypisanych prac ocenę w kategoriach wyszczególnionych przez regulamin.
    - Dodać komentarz do ocenianej pracy, który zostanie wysłany przez system autorom.
    - Sortować prace ze względu na wystawioną ocenę (ogólną oraz w poszczególnych kategoriach)
    - Umożliwiać kontakt uczestników konkursu z mentorami oraz organizatorami.
    - Po zakończonym procesie składania prac system powinien umożliwić ich przeglądanie wszystkim użytkownikom systemu.
    - Uczestnikom zakwalifikowanym do dalszego etapu konkursu system powinien umożliwiać przesyłanie zmodyfikowanych wersji pracy do terminu półfinałów i finałów.

OPIS PRZYSZŁEJ EWOLUCJI SYSTEMU

Po zakończonym konkursie system mógłby być wykorzystany przy organizacji innych konkursów, ewentualnie można było by go rozszerzyć o możliwość organizacji konkursów cyklicznych.

SŁOWNIK POJĘĆ

- Ewaluator – osoba oceniająca projekty.
- Mentor – osoba udzielająca wsparcia merytorycznego uczestnikom konkursu.
- Organizator – osoba odpowiedzialna za organizację konkursu na terenie danych krajów.

ANALIZA FUNKCJONALNA I STRUKTURALNA

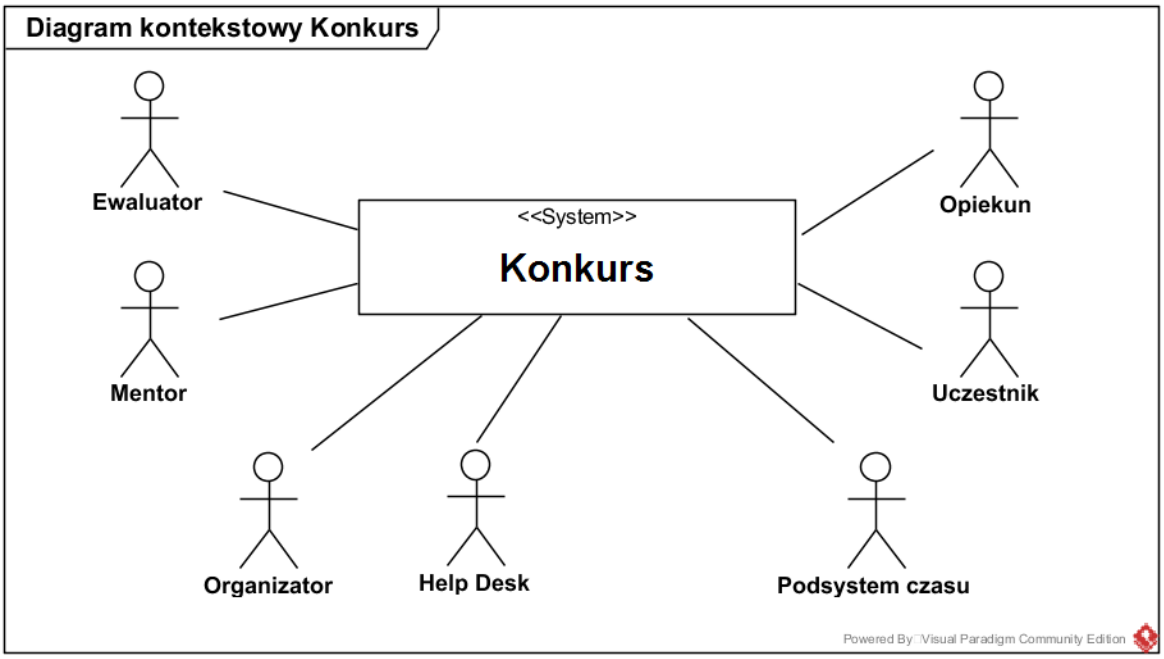


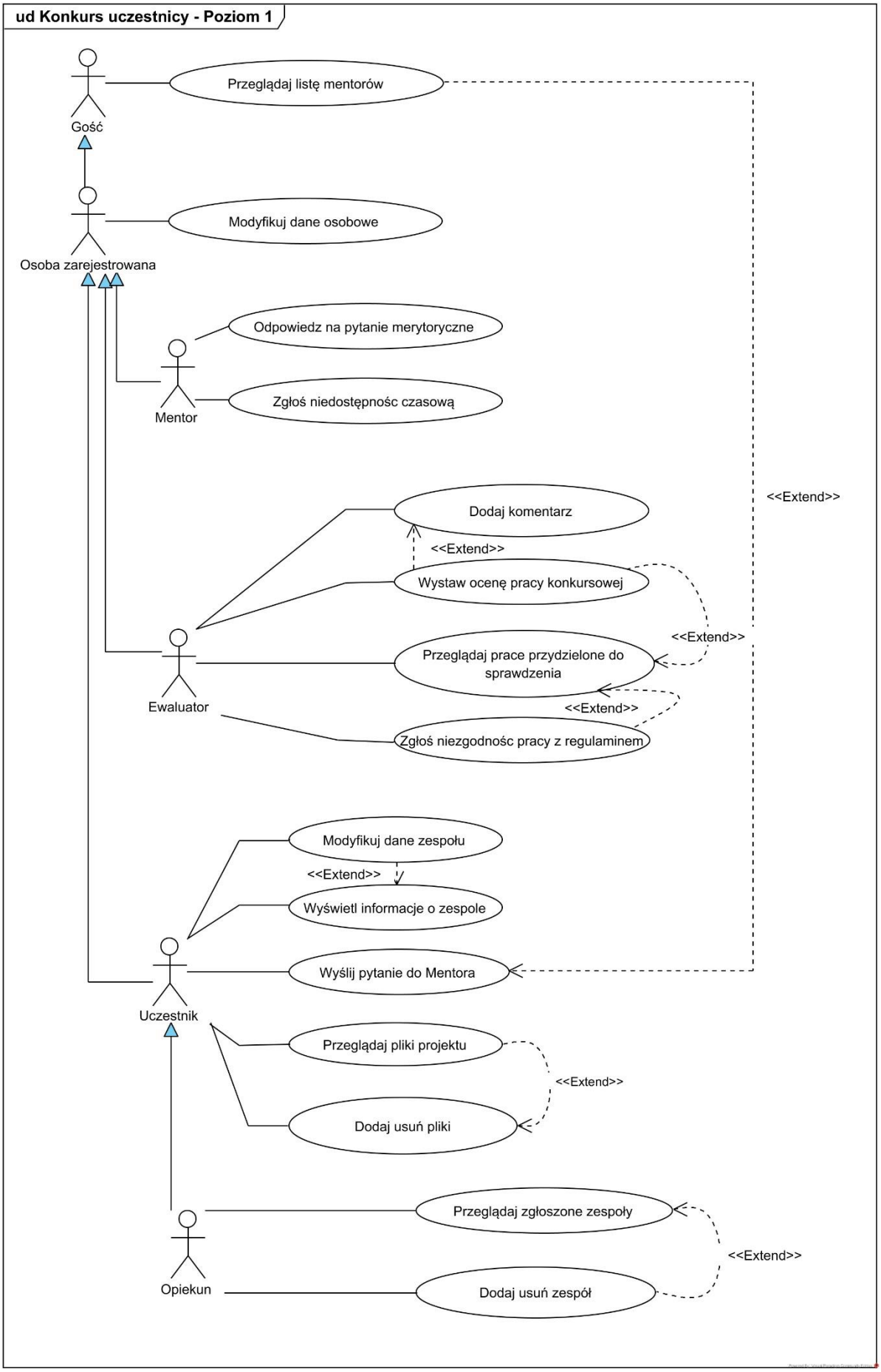
DIAGRAM KONTEKSTOWY

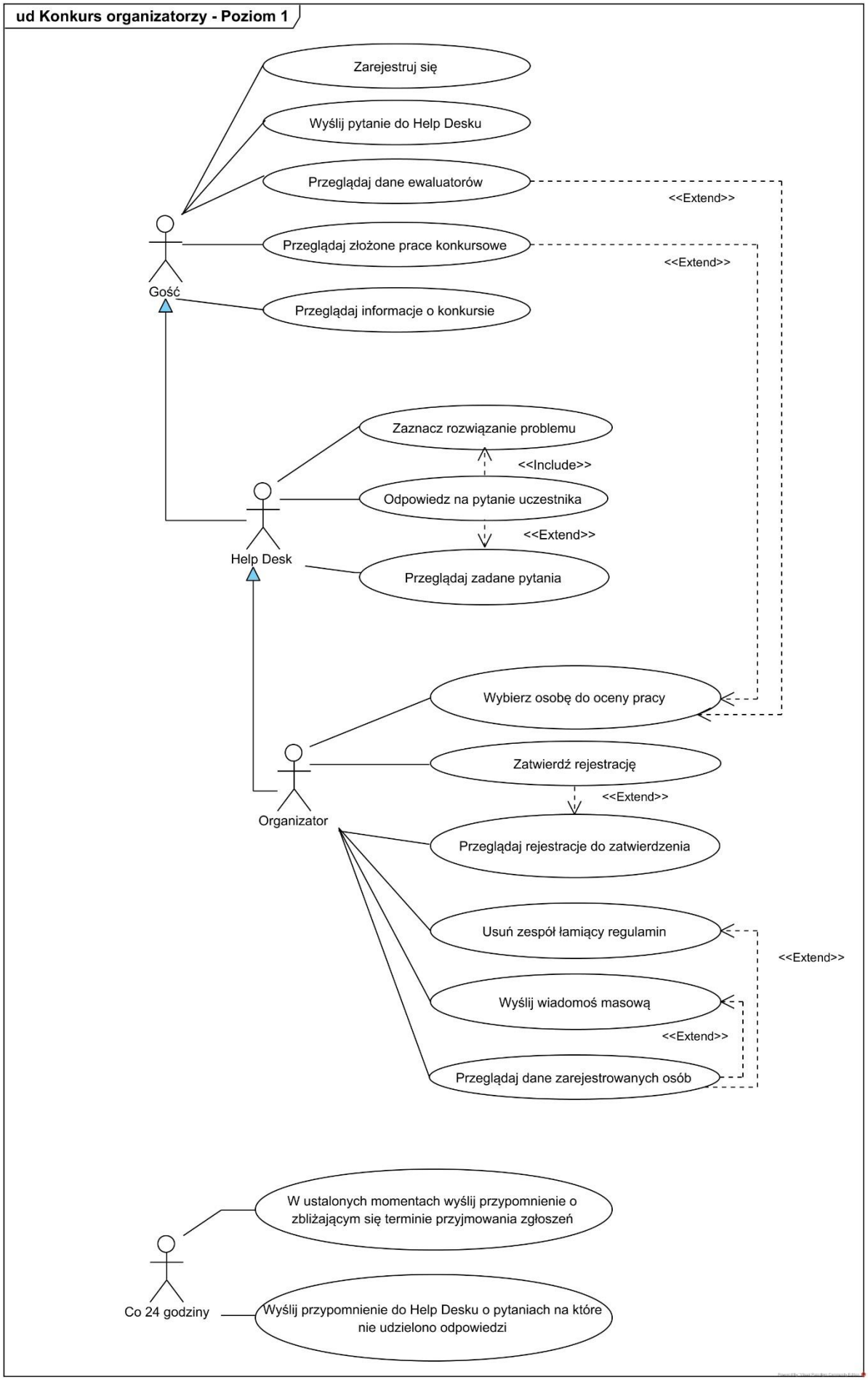
Tabela 1 Lista przypadków użycia. Priorytet przypadku został oceniony w skali od 1 do 4 gdzie 1 oznacza przypadki które powinny zostać

ID	Nazwa przypadku użycia	Rodzaj	Priorytet	Złożoność
1	Zarejestruj się	Złożony	1	3
2	Wyślij Pytanie do Help Desku	Prosty	2	1
3	Przeglądaj dane Ewaluatorów	Złożony	4	2
4	Przeglądaj informacje o Mentorach	Złożony	3	2
5	Przeglądaj złożone prace konkursowe	Złożony	3	2
6	Przeglądaj informacje o konkursie	Złożony	1	2
7	Zaznacz rozwiązanie problemu	Prosty	4	1
8	Odpowiedz na pytanie uczestnika	Prosty	2	1
9	Przeglądaj zadane pytania	Złożony	2	2
10	Przeglądaj dane zarejestrowanych osób	Złożony	2	2
11	Wybierz osobę do oceny pracy	Złożony	1	3
12	Zatwierdź rejestrację	Prosty	2	2
13	Przeglądaj rejestrację do zatwierdzenia	Złożony	2	2
14	Usuń zespół łamiący regulamin	Prosty	3	3
15	Wyślij wiadomość masową	Złożony	4	2
16	W ustalonych momentach wyślij przypomnienie o zbliżającym się terminie przyjmowania zgłoszeń	Prosty	4	2
17	Wyślij przypomnienie do Help Desku o pytaniach na które nie udzielono odpowiedzi	Prosty	4	2
18	Modyfikuj dane osobowe	Prosty	2	2
19	Odpowiedz na pytanie merytoryczne	Prosty	3	2
20	Zgłoś niedostępność czasową	Prosty	3	1
21	Dodaj komentarz	Prosty	3	1
22	Wystaw oceną pracy konkursowej	Złożony	1	3
23	Przeglądaj prace przydzielone do sprawdzenia	Złożony	1	2
24	Zgłoś niezgodność pracy z regulaminem	Prosty	1	1
25	Modyfikuj dane zespołu	Prosty	2	2
26	Wyświetl informacje o zespole	Prosty	2	1
27	Wyślij pytanie do Mentora	Prosty	3	1
28	Przeglądaj pliki projektu	Złożony	1	2
29	Dodaj usuń plik	Prosty	1	3
30	Przeglądaj zgłoszone zespoły	Złożony	1	2
31	Dodaj usuń zespół	Prosty	1	3

zaimplementowane w pierwszej kolejności, 4 oznacza przypadki możliwe do implementacji na końcu. Złożoność oznacza złożoność implementacji przypadku, 1 oznacza przypadek prosty w implementacji 3 skomplikowany.

DIAGRAM PRZYPADKÓW UŻYCIA

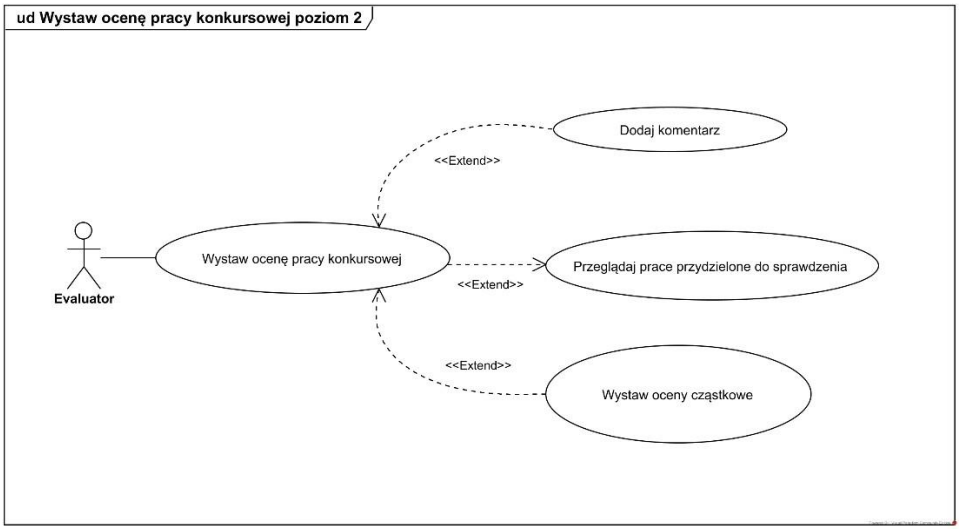


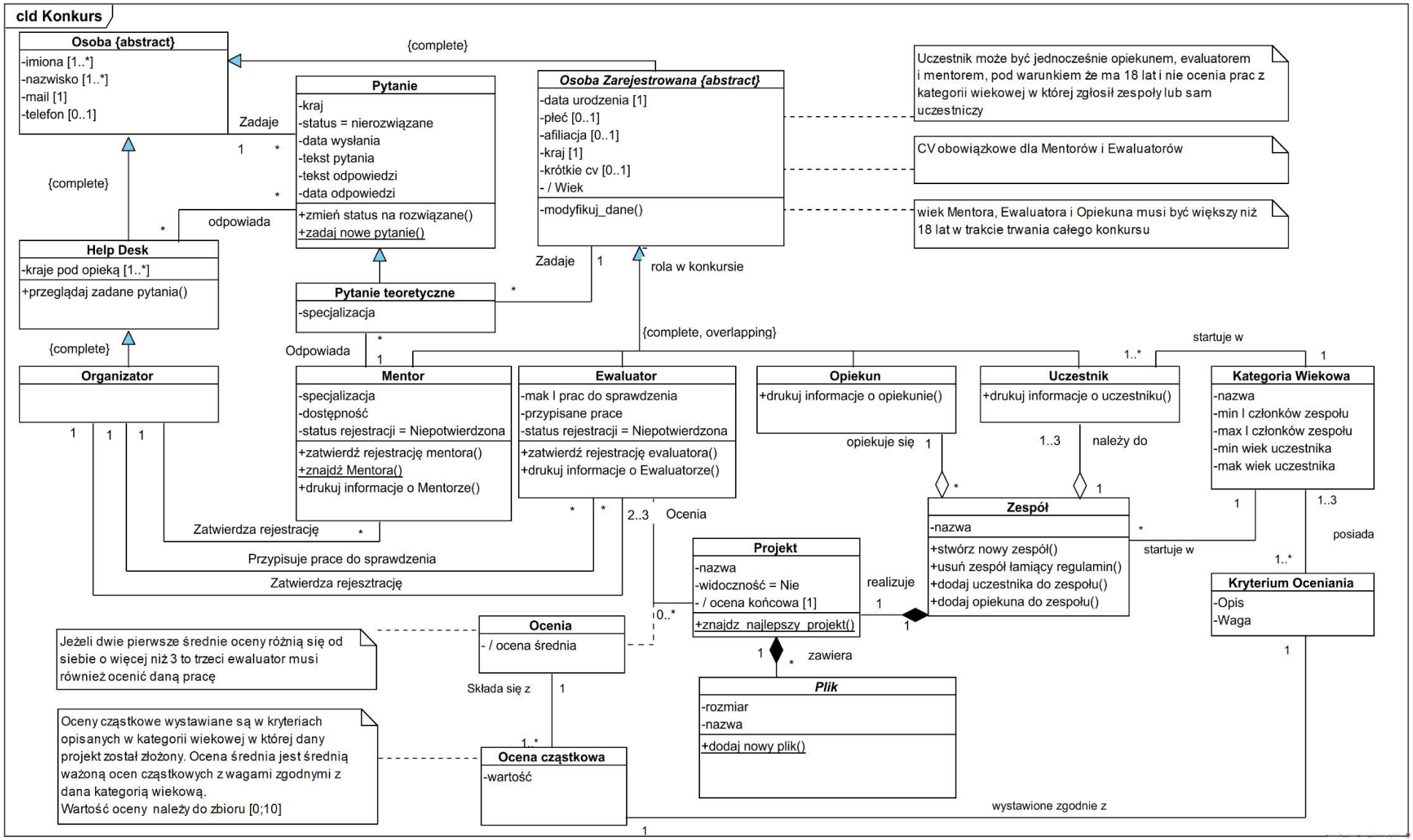

ud Konkurs organizatorzy - Poziom 1
Zarejestruj się
Wyślij pytanie do Help Desku
Przeglądaj dane ewaluatorów
<<Extend>>
Przeglądaj złożone prace konkursowe
<<Extend>>
Gość
Przeglądaj informacje o konkursie
Zaznacz rozwiązanie problemu
<<Include>>
Odpowiedz na pytanie uczestnika
<<Extend>>
Help Desk
Przeglądaj zadane pytania
Wybierz osobę do oceny pracy
Zatwierdź rejestrację
<<Extend>>
Organizator
Przeglądaj rejestracje do zatwierdzenia
Usuń zespół łamiący regulamin
<<Extend>>
Wyślij wiadomoś masową
<<Extend>>
Przeglądaj dane zarejestrowanych osób
W ustalonych momentach wyślij przypomnienie o zbliżającym się terminie przyjmowania zgłoszeń
Co 24 godziny
Wyślij przypomnienie do Help Desku o pytaniach na które nie udzielono odpowiedzi

Nazwa przypadku użycia	**Zarejestruj się**
Warunek początkowy	Brak
Główny przepływ zdarzeń	1. Aktor *Gość* uruchamia przypadek użycia 2. System wyświetla formularz rejestracyjny zawierający. następujące pola: login, kraj pochodzenia, data urodzenia, adres email, typ rejestracji. A 3. System informuje o pozytywnym zakończeniu rejestracji.
Alternatywny przepływ zdarzeń	2a. Aktor wprowadził błędne dane, system wyświetla odpowiednią informację i wraca do punktu 2. 2b. Aktor wprowadził adres email, który już został wykorzystany przy rejestracji. System pyta czy zakończyć. 2ba. Aktor zdecydował się kontynuować przypadek użycia. System powraca do punktu 2. 2bc. Aktor zdecydował się zakończyć przypadek użycia. 2c. Wiek Aktora nie pozwala na zarejestrowanie się w wybranej kategorii. System wyświetla informację o sugerowanym typie rejestracji i pyta czy zmienić typ rejestracji. 2ca. Aktor zmienił typ rejestracji i system przechodzi do punktu 3 2cb. Aktor postanowił powrócić do punktu 2 2cb. Aktor postanowił zakończyć przypadek użycia.
Zakończenie	W dowolnym momencie

Nazwa przypadku użycia	**Wybierz osobę do oceny pracy**
Warunek początkowy	Możliwe dopiero po zakończeniu przyjmowania prac konkursowych. Złożona przynajmniej jedna praca w regionie, którym się opiekuje dany Organizator.
Główny przepływ zdarzeń	1. Aktor Organizator uruchamia przypadek użycia Wybierz osoby do oceny pracy. 2. System wyświetla formularz zawierający listę złożonych prac. Organizator wybiera jedną z nich. 3. System wyświetla formularz zawierający podstawowe informacje o pracy, to znaczy: Abstrakt, kategorię tematyczną i kategorię wiekową i trzy pola do wyboru Ewaluatorów. Organizator przydziela odpowiednie pola, i potwierdza wprowadzone zmiany. 4. System informuje o pozytywnym zapisaniu zmian i pyta czy przydzielić Ewaluatorów do kolejnej pracy. Aktor postanawia zkończyć przypadek użycia.
Alternatywny przepływ zdarzeń	2.a Aktor postanawia przejrzeć wszystkie złożone prace uruchamia przypadek użycia przeglądaj złożone prace konkursowe, po czym wraca do punktu 2. 3.a Aktor postanawia przejrzeć dane Ewaluatorów i uruchamia przypadek użycia Przejrzyj dane Ewaluatorów, po czym wraca do punktu 2. 3.b Organizator wybrał osobę, której przypisano więcej prac niż się na to zgodziła. System pyta czy Aktor chce pozostać przy dokonanym wyborze 3ba. Aktor postanawia wybrać inną osobę do oceny pracy i wraca do punktu 3. 3bb. Aktor nie zmienia swojego wyboru i wraca do punktu trzeciego 3. 3.c Aktor nie wybrał trzech Ewaluatorów, system wyświetla odpowiednią informację i pyta czy na pewno zakończyć. 3ca. Aktor postanowił wybrać brakujące osoby oceniające i system wraca do punktu 3. 3cb. Aktor postanawia zostawić pole bez wyboru i system przechodzi do punktu 4. 4a. Aktor postanawia przypisać osoby oceniające do pozostałych prac i system przechodzi do punktu 2.
Zakończenie	W dowolnym momencie.
Warunek końcowy	Przypisanie osoby oceniającej dla wybranej pracy.

Nazwa przypadku użycia	**Wystaw ocenę pracy konkursowej.**
Warunek początkowy	Przynajmniej jedna praca została przypisana do sprawdzenia.
Główny przepływ zdarzeń	1. Aktor *Ewaluator* uruchamia przypadek użycia. 2. System wyświetla formularz zawierający listę przydzielonych prac do sprawdzenia. Aktor wybiera pracę, którą chce ocenić. 3. System wyświetla formularz, w którym Aktor wpisuje oceny cząstkowe pracy w poszczególnych kategoriach i zapisuje zmiany. 4. System wyświetla informację o ocenie średniej i zapisaniu zmian i pyta czy dodać komentarz do pracy. Aktor postanawia zakończyć przypadek użycia.
Alternatywny przepływ zdarzeń	1.a Aktorowi nie przypisano żadnej pracy do sprawdzenia. System wyświetla odpowiedni komunikat. 2.a Aktor w polu z oceną wstawił niedozwoloną wartość. System wyświetla stosowną informacje i wraca do punktu 3. 3.a Aktor postanowił dodać komentarz do pracy. System uruchamia przypadek użycia *Wystaw komentarz pracy.*
Zakończenie	W dowolnym momencie.
Warunek końcowy	Zostaje zarejestrowana ocena wystawiona przez danego Aktora.




cld Konkurs
Osoba {abstract}
-imiona [1..*]
-nazwisko [1..*]
-mail [1]
-telefon [0..1]
{complete}
Pytanie
-kraj
-status = nierozwiązane
-data wysłania
-tekst pytania
-tekst odpowiedzi
-data odpowiedzi
+zmień status na rozwiązane()
+zadaj nowe pytanie()
Zadaje
odpowiada
Osoba Zarejestrowana {abstract}
-data urodzenia [1]
-płeć [0..1]
-afiliacja [0..1]
-kraj [1]
-krótkie cv [0..1]
- / Wiek
-modyfikuj_dane()
Uczestnik może być jednocześnie opiekunem, evaluatorem i mentorem, pod warunkiem że ma 18 lat i nie ocenia prac z kategorii wiekowej w której zgłosił zespoły lub sam uczestniczy
CV obowiązkowe dla Mentorów i Ewaluatorów
wiek Mentora, Ewaluatora i Opiekuna musi być większy niż 18 lat w trakcie trwania całego konkursu
Help Desk
-kraje pod opieką [1..*]
+przeglądaj zadane pytania()
Pytanie teoretyczne
-specjalizacja
Zadaje
rola w konkursie
{complete, overlapping}
Odpowiada
Organizator
Mentor
-specjalizacja
-dostępność
-status rejestracji = Niepotwierdzona
+zatwierdź rejestrację mentora()
+znajdź Mentora()
+drukuj informacje o Mentorze()
Ewaluator
-mak l prac do sprawdzenia
-przypisane prace
-status rejestracji = Niepotwierdzona
+zatwierdź rejestrację evaluatora()
+drukuj informacje o Ewaluatorze()
Opiekun
+drukuj informacje o opiekunie()
Uczestnik
+drukuj informacje o uczestniku()
Kategoria Wiekowa
-nazwa
-min l członków zespołu
-max l członków zespołu
-min wiek uczestnika
-mak wiek uczestnika
startuje w
opiekuje się
należy do
Zespół
-nazwa
+stwórz nowy zespół()
+usuń zespół łamiący regulamin()
+dodaj uczestnika do zespołu()
+dodaj opiekuna do zespołu()
Zatwierdza rejestrację
Przypisuje prace do sprawdzenia
Zatwierdza rejesztrację
Ocenia
Projekt
-nazwa
-widoczność = Nie
- / ocena końcowa [1]
+znajdz najlepszy projekt()
realizuje
posiada
Kryterium Oceniania
-Opis
-Waga
Ocenia
- / ocena średnia
Jeżeli dwie pierwsze średnie oceny różnią się od siebie o więcej niż 3 to trzeci ewaluator musi również ocenić daną pracę
Składa się z
Ocena cząstkowa
-wartość
Oceny cząstkowe wystawiane są w kryteriach opisanych w kategorii wiekowej w której dany projekt został złożony. Ocena średnia jest średnią ważoną ocen cząstkowych z wagami zgodnymi z dana kategorią wiekową.
Wartość oceny należy do zbioru [0;10]
Plik
-rozmiar
-nazwa
+dodaj nowy plik()
zawiera
wystawione zgodnie z

ANALIZA WARTOŚCI POCHODNYCH, POCZĄTKOWYCH I GRANICZNYCH

ANALIZA WARTOŚCI POCHODNYCH

Na diagramie klas można znaleźć następujące wartości pochodne:

wiek: Jest wyliczany na podstawie atrybutu *data urodzenia* oraz daty końca przyjmowania zgłoszeń.

ocena średnia: jest średnią ważoną ocen cząstkowych oraz odpowiadających im wag z odpowiedniego kryterium oceniania.

ocena końcowa: Jest to średnia arytmetyczna dwóch najbliższych ocen przyznanych przez ewaluatorów.

ANALIZA WARTOŚCI POCZĄTKOWYCH

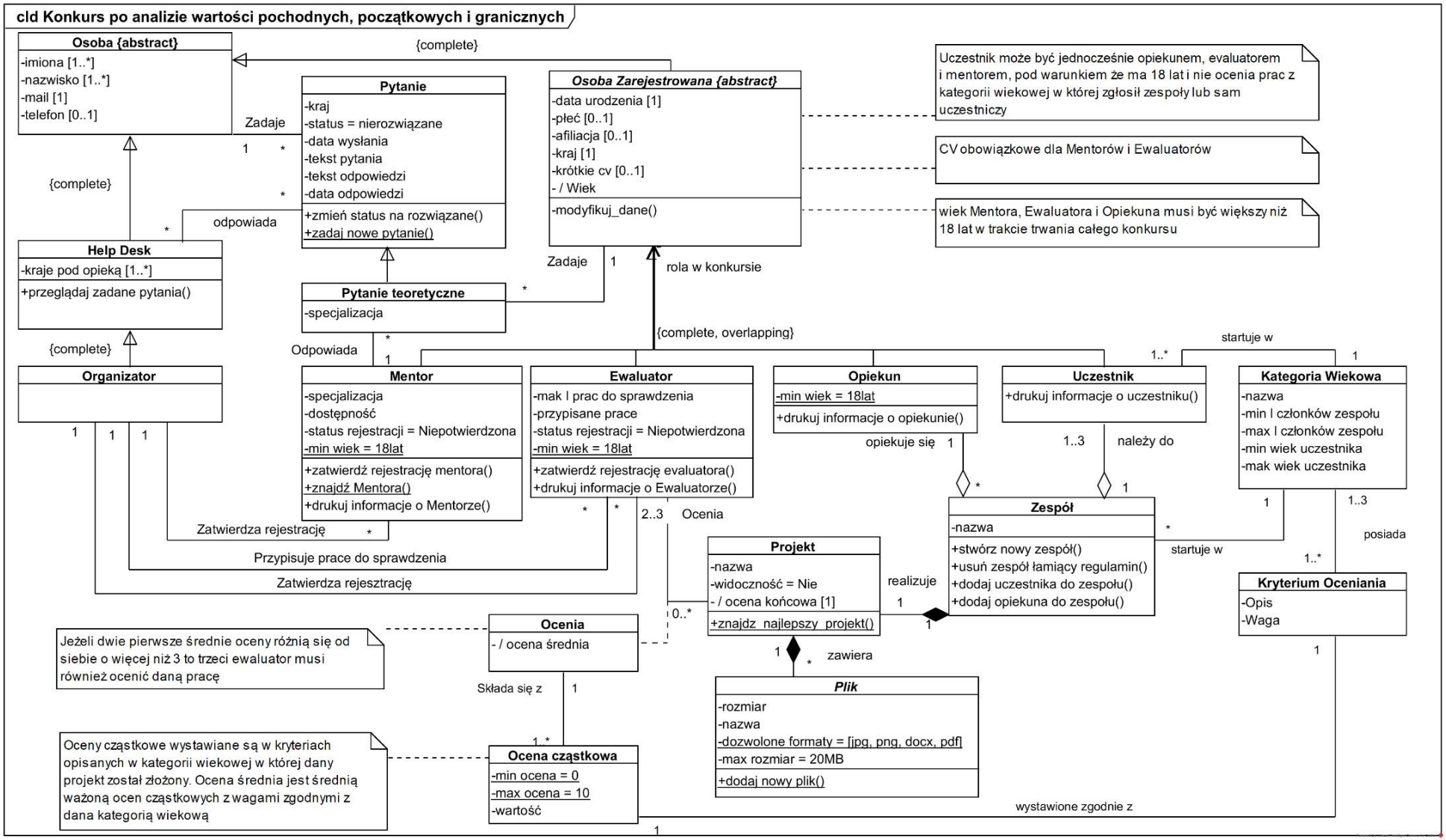
Następujące atrybuty posiadają wartości początkowe

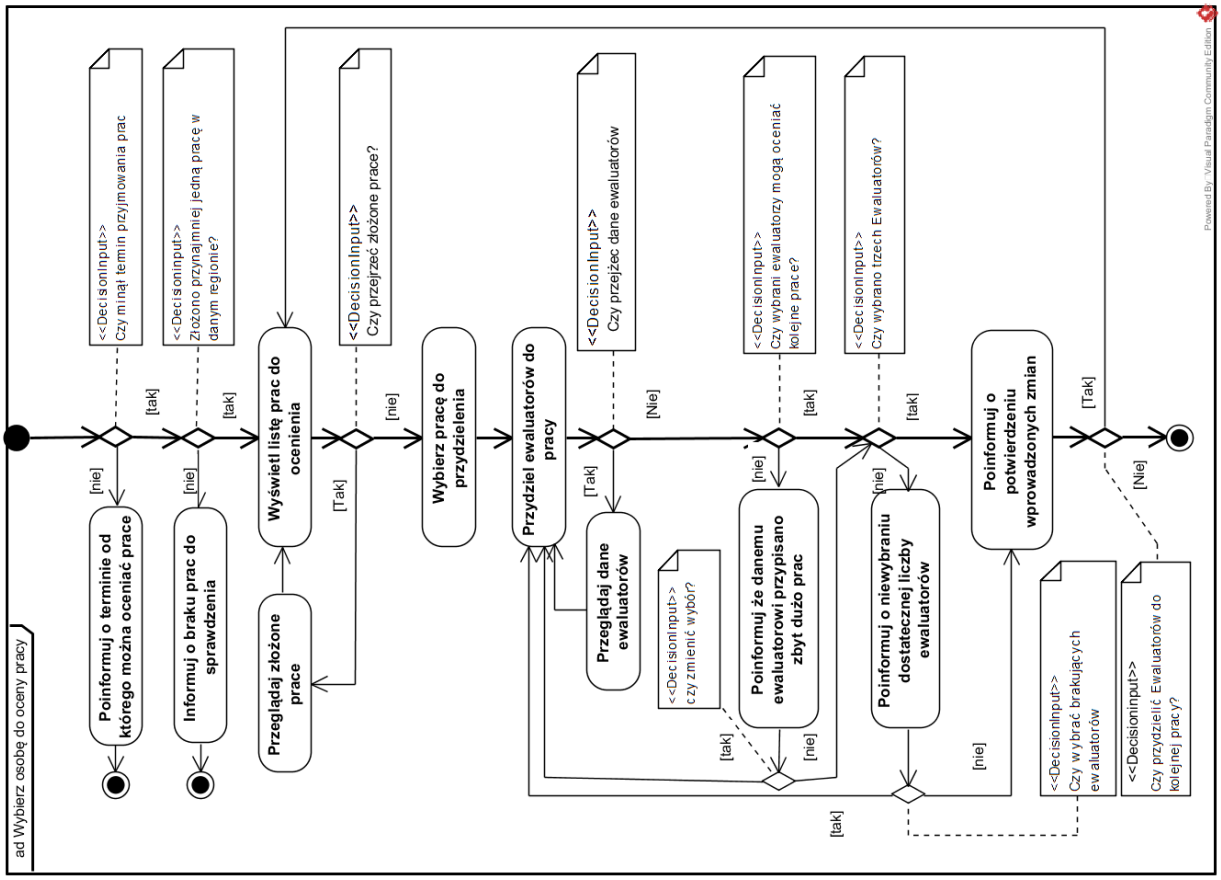
Atrybut **status** klasy Pytanie, posiada wartość początkową wynoszącą „nierozwiązane”

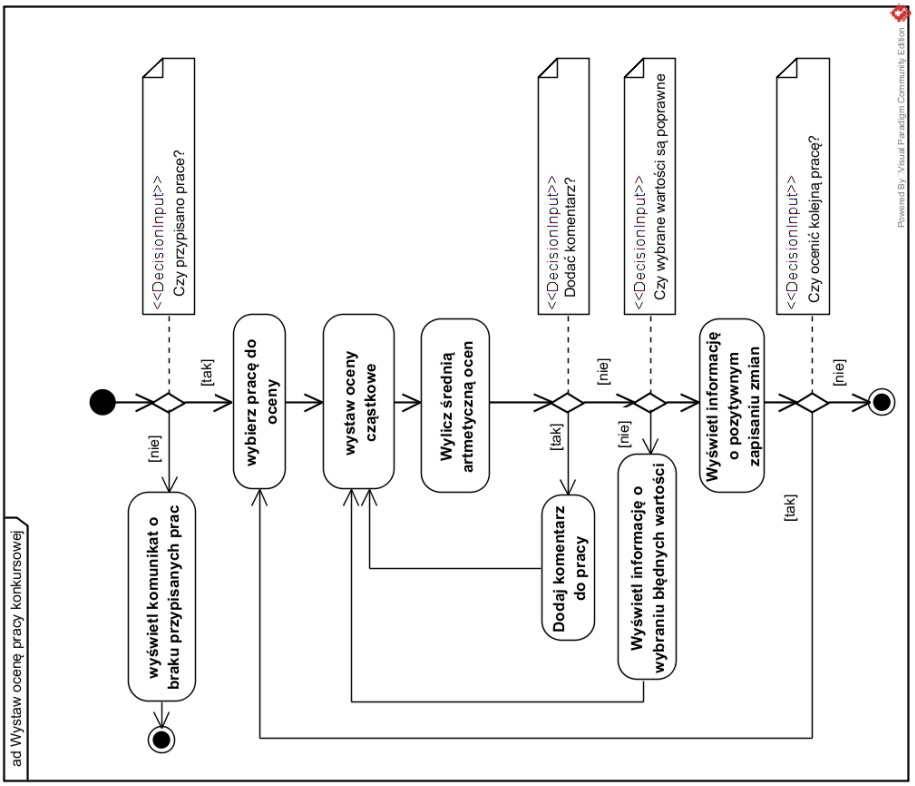
Atrybut **status rejestracji** w klasach Mentor i Ewaluator ma wartość początkową „niepotwierdzone”

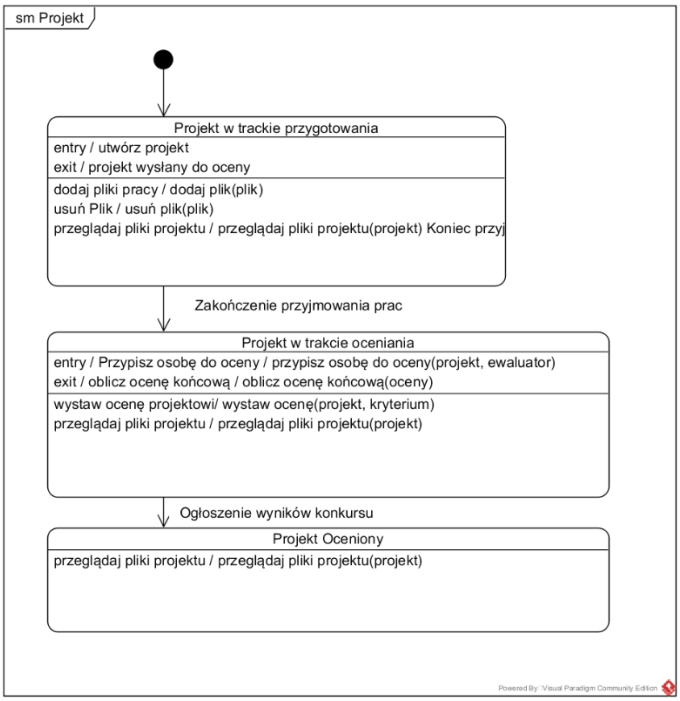
ANALIZA WARTOŚCI GRANICZNYCH

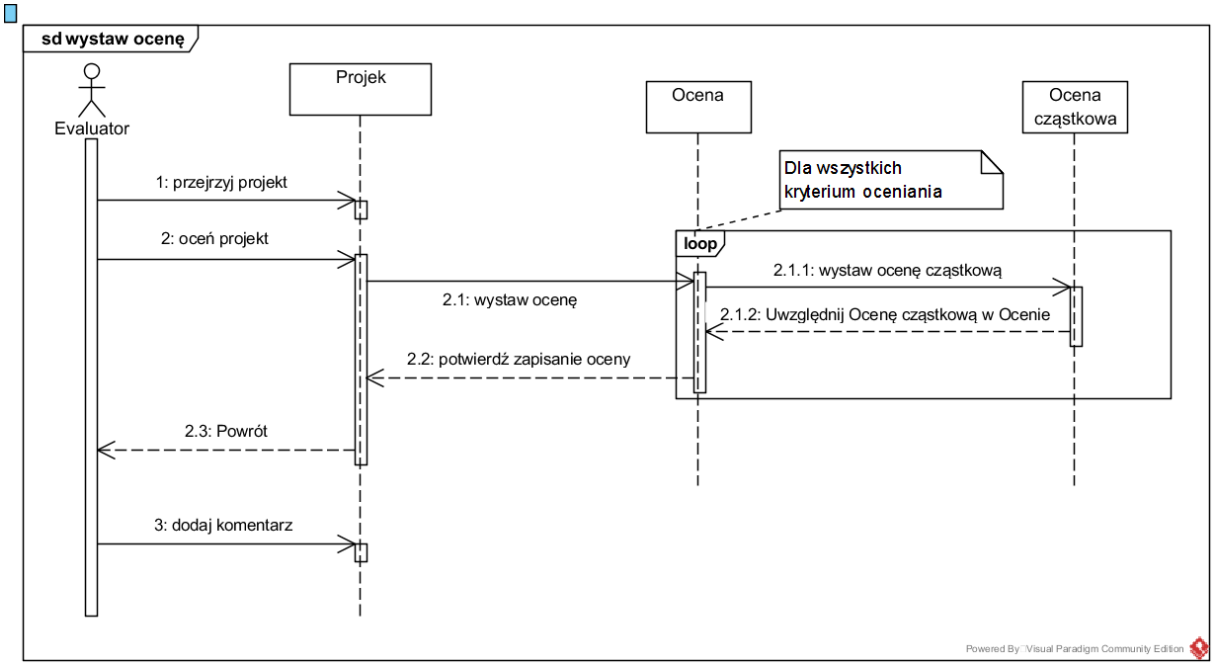
Wartość graniczna	**Ograniczenie**	**Zmiany w schemacie pojęciowym**
20 MB	{nie więcej niż 20 MB}	Dodakowy atrybut o zasięgu klasowy max rozmiar = 20MB w klasie Plik
18 lat	{przynajmniej 18 lat}	Dodatkowy atrybut o zasięgu klasowym, min wiek = 18lat w klasach Ewaluator, Mentor, Opiekun.
0 - 10	{ocena w przedziale od 0 do 10}	Dodatkowe atrybut klasowe min ocena = 0 oraz max ocena = 10 w klasie Ocena cząstkowa.
{.jpg. .png; .docx; .pdf}	{dozwolone formaty plików pracy to: .jpg. .png; .docx; .pdf o rozmiarze nie większym niż 20 MB}	Dodanie atrybutu klasowego dozwolone formaty = [.jpg; .png; .docx; .pdf] Dodanie atrybutu klasowego max rozmiar = 20MB


cld Konkurs po analizie wartości pochodnych, początkowych i granicznych
Osoba {abstract}
-imiona [1..*]
-nazwisko [1..*]
-mail [1]
-telefon [0..1]
{complete}
Zadaje
1
*
Pytanie
-kraj
-status = nierozwiązane
-data wysłania
-tekst pytania
-tekst odpowiedzi
-data odpowiedzi
+zmień status na rozwiązane()
+zadaj nowe pytanie()
Osoba Zarejestrowana {abstract}
-data urodzenia [1]
-płeć [0..1]
-afiliacja [0..1]
-kraj [1]
-krótkie cv [0..1]
- / Wiek
-modyfikuj_dane()
Uczestnik może być jednocześnie opiekunem, evaluatorem i mentorem, pod warunkiem że ma 18 lat i nie ocenia prac z kategorii wiekowej w której zgłosił zespoły lub sam uczestniczy
CV obowiązkowe dla Mentorów i Ewaluatorów
wiek Mentora, Ewaluatora i Opiekuna musi być większy niż 18 lat w trakcie trwania całego konkursu
odpowiada
Help Desk
-kraje pod opieką [1..*]
+przeglądaj zadane pytania()
Pytanie teoretyczne
-specjalizacja
Zadaje
rola w konkursie
{complete, overlapping}
Odpowiada
Organizator
Mentor
-specjalizacja
-dostępność
-status rejestracji = Niepotwierdzona
-min wiek = 18lat
+zatwierdź rejestrację mentora()
+znajdź Mentora()
+drukuj informacje o Mentorze()
Ewaluator
-mak l prac do sprawdzenia
-przypisane prace
-status rejestracji = Niepotwierdzona
-min wiek = 18lat
+zatwierdź rejestrację evaluatora()
+drukuj informacje o Ewaluatorze()
Opiekun
-min wiek = 18lat
+drukuj informacje o opiekunie()
opiekuje się
Uczestnik
+drukuj informacje o uczestniku()
1..3
należy do
startuje w
Kategoria Wiekowa
-nazwa
-min l członków zespołu
-max l członków zespołu
-min wiek uczestnika
-mak wiek uczestnika
Zespół
-nazwa
+stwórz nowy zespół()
+usuń zespół łamiący regulamin()
+dodaj uczestnika do zespołu()
+dodaj opiekuna do zespołu()
startuje w
posiada
Kryterium Oceniania
-Opis
-Waga
Zatwierdza rejestrację
Przypisuje prace do sprawdzenia
Zatwierdza rejesztrację
2..3
Ocenia
Projekt
-nazwa
-widoczność = Nie
- / ocena końcowa [1]
+znajdz_najlepszy_projekt()
realizuje
0..*
zawiera
Plik
-rozmiar
-nazwa
-dozwolone formaty = [jpg, png, docx, pdf]
-max rozmiar = 20MB
+dodaj nowy plik()
Ocenia
- / ocena średnia
Składa się z
Jeżeli dwie pierwsze średnie oceny różnią się od siebie o więcej niż 3 to trzeci ewaluator musi również ocenić daną pracę
Oceny cząstkowe wystawiane są w kryteriach opisanych w kategorii wiekowej w której dany projekt został złożony. Ocena średnia jest średnią ważoną ocen cząstkowych z wagami zgodnymi z dana kategorią wiekową
Ocena cząstkowa
-min ocena = 0
-max ocena = 10
-wartość
wystawione zgodnie z


ad Wybierz osobę do oceny pracy
<<DecisionInput>>
Czy minął termin przyjmowania prac
<<DecisionInput>>
Złożono przynajmniej jedną pracę w danym regionie?
<<DecisionInput>>
Czy przejrzeć złożone prace?
<<DecisionInput>>
Czy przejżec dane ewaluatorów
<<DecisionInput>>
Czy wybrani ewaluatorzy mogą oceniać kolejne prace?
<<DecisionInput>>
Czy wybrano trzech Ewaluatorów?
[nie]
[tak]
Poinformuj o terminie od którego można oceniać prace
Informuj o braku prac do sprawdzenia
Wyświetl listę prac do oceniania
Przeglądaj złożone prace
Wybierz pracę do przydzielenia
Przydziel ewaluatorów do pracy
Przeglądaj dane ewaluatorów
<<DecisionInput>>
czy zmienić wybór?
Poinformuj że danemu ewaluatorowi przypisano zbyt dużo prac
Poinformuj o niewybraniu dostatecznej liczby ewaluatorów
Poinformuj o potwierdzeniu wprowadzonych zmian
<<DecisionInput>>
Czy wybrać brakujących ewaluatorów
<<DecisionInput>>
Czy przydzielić Ewaluatorów do kolejnej pracy?
Powered By Visual Paradigm Community Edition


ad Wystaw ocenę pracy konkursowej
<<DecisionInput>>
Czy przypisano prace?
<<DecisionInput>>
Dodać komentarz?
<<DecisionInput>>
Czy wybrane wartości są poprawne
<<DecisionInput>>
Czy ocenić kolejną pracę?
wyświetl komunikat o braku przypisanych prac
wybierz pracę do oceny
wystaw oceny cząstkowe
Wylicz średnią artmetyczną ocen
Dodaj komentarz do pracy
Wyświetl informację o wybraniu błędnych wartości
Wyświetl informację o pozytywnym zapisaniu zmian
[nie]
[tak]
Powered By Visual Paradigm Community Edition





W wyniku analizy dynamicznej do diagramu kontekstowego postanowiono dodać następujące elementy:

W klasie Projekt postanowiono dodać metody:

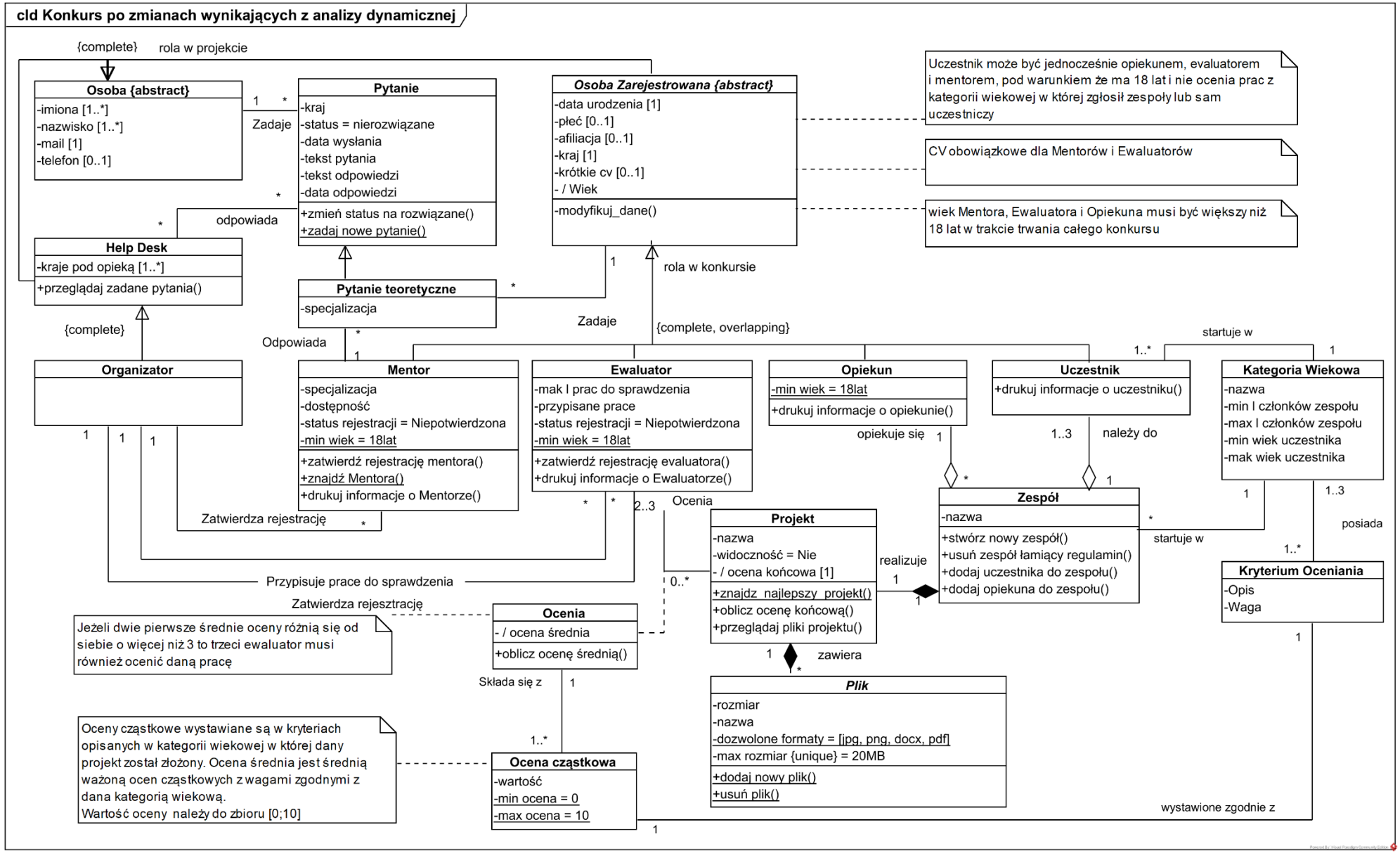
- przeglądaj plik projektu(plik)
- wystaw ocenę projektowi()
- oblicz ocenę końcową()

W klasie Ocena dodano

- Oblicz ocenę średnią()

W klasie Plik postanowiono dodać operację:

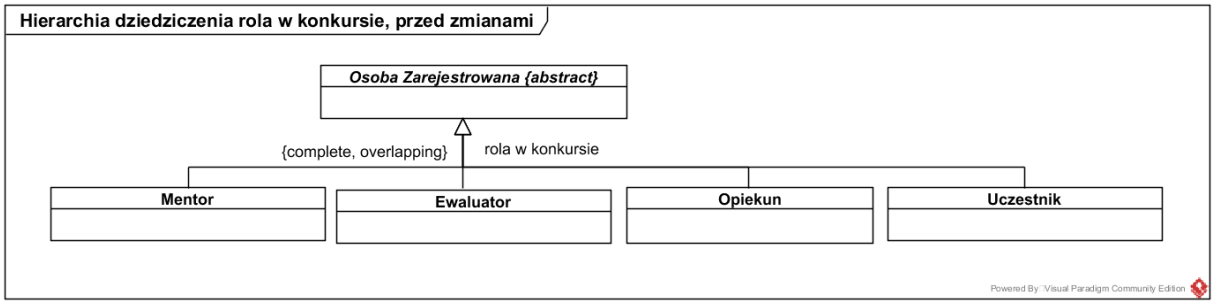
- Usuń plik(plik)

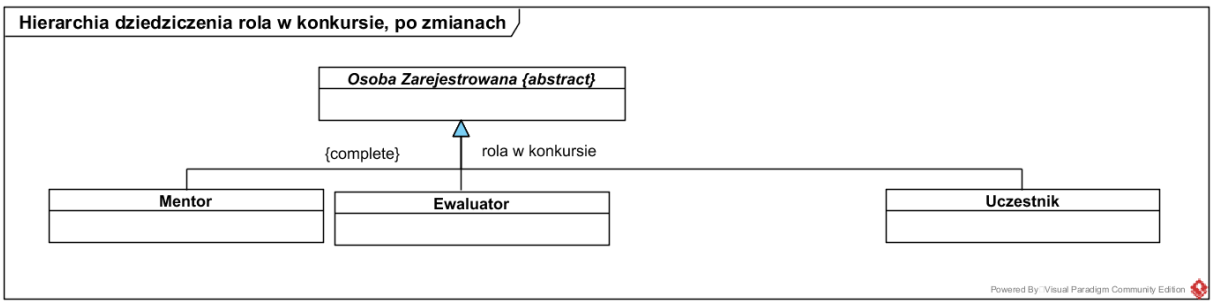

cld Konkurs po zmianach wynikających z analizy dynamicznej
{complete}
rola w projekcie
Osoba {abstract}
-imiona [1..*]
-nazwisko [1..*]
-mail [1]
-telefon [0..1]
1
*
Zadaje
Pytanie
-kraj
-status = nierozwiązane
-data wysłania
-tekst pytania
-tekst odpowiedzi
-data odpowiedzi
+zmień status na rozwiązane()
+zadaj nowe pytanie()
Osoba Zarejestrowana {abstract}
-data urodzenia [1]
-płeć [0..1]
-afiliacja [0..1]
-kraj [1]
-krótkie cv [0..1]
- / Wiek
-modyfikuj_dane()
Uczestnik może być jednocześnie opiekunem, evaluatorem i mentorem, pod warunkiem że ma 18 lat i nie ocenia prac z kategorii wiekowej w której zgłosił zespoły lub sam uczestniczy
CV obowiązkowe dla Mentorów i Ewaluatorów
wiek Mentora, Ewaluatora i Opiekuna musi być większy niż 18 lat w trakcie trwania całego konkursu
odpowiada
Help Desk
-kraje pod opieką [1..*]
+przeglądaj zadane pytania()
Pytanie teoretyczne
-specjalizacja
rola w konkursie
{complete}
Odpowiada
Zadaje
{complete, overlapping}
startuje w
Organizator
Mentor
-specjalizacja
-dostępność
-status rejestracji = Niepotwierdzona
-min wiek = 18lat
+zatwierdź rejestrację mentora()
+znajdź Mentora()
+drukuj informacje o Mentorze()
Ewaluator
-mak l prac do sprawdzenia
-przypisane prace
-status rejestracji = Niepotwierdzona
-min wiek = 18lat
+zatwierdź rejestrację evaluatora()
+drukuj informacje o Ewaluatorze()
Opiekun
-min wiek = 18lat
+drukuj informacje o opiekunie()
Uczestnik
+drukuj informacje o uczestniku()
Kategoria Wiekowa
-nazwa
-min l członków zespołu
-max l członków zespołu
-min wiek uczestnika
-mak wiek uczestnika
opiekuje się
należy do
Zatwierdza rejestrację
Ocenia
Zespół
-nazwa
+stwórz nowy zespół()
+usuń zespół łamiący regulamin()
+dodaj uczestnika do zespołu()
+dodaj opiekuna do zespołu()
startuje w
posiada
Projekt
-nazwa
-widoczność = Nie
- / ocena końcowa [1]
+znajdz_najlepszy_projekt()
+oblicz ocenę końcową()
+przeglądaj pliki projektu()
realizuje
Kryterium Oceniania
-Opis
-Waga
Przypisuje prace do sprawdzenia
Zatwierdza rejesztrację
Jeżeli dwie pierwsze średnie oceny różnią się od siebie o więcej niż 3 to trzeci ewaluator musi również ocenić daną pracę
Ocenia
- / ocena średnia
+oblicz ocenę średnią()
Składa się z
zawiera
Plik
-rozmiar
-nazwa
-dozwolone formaty = [jpg, png, docx, pdf]
-max rozmiar {unique} = 20MB
+dodaj nowy plik()
+usuń plik()
Oceny cząstkowe wystawiane są w kryteriach opisanych w kategorii wiekowej w której dany projekt został złożony. Ocena średnia jest średnią ważoną ocen cząstkowych z wagami zgodnymi z dana kategorią wiekową.
Wartość oceny należy do zbioru [0;10]
Ocena cząstkowa
-wartość
-min ocena = 0
-max ocena = 10
wystawione zgodnie z

DECYZJE PROJEKTOWE

DZIEDZICZENIE NIETRYWIALNE:

Dziedziczenie compleeate overlaping zostanie zrealizowane przez spłaszczenie hierarchii. Uczestnik i Opiekun zostaną zastąpieni jedną klasą Uczestnik. Pozostałe klasy mają na tyle różną funkcjonalność, że zrezygnowano z możliwość wykonywania wszystkich tych trzech zadań z jednego konta.





Uproszczony fragment kodu odpowiadający temu dziedziczeniu będzie wyglądał w następujący sposób:

```
public class osobaZarejestrowana extends Osoba{
//…
}

public class Mentor extends osobaZarejestrowana {
//…
}

public class Ewaluator extends osobaZarejestrowana {
//…
}

public class Uczestnik extends osobaZarejestrowana {
//…
}
```

EKSTENCJEE KLASY:

Ekstencje klasy będą realizowane poprzez stworzenie klasy ObjectPlus po której dziedziczyć będą wszystkie klasy projektu. Dzięki kontenerowi mapującemu zapisywanemu do pliku zapewniać ona będzie trwałość ekstynkcji oraz umożliwi zarządzanie nią. Ze względu na czytelność diagramu dziedziczenie tej klasy nie zostanie umieszczone na diagramie klas.

```
public class ObjectPlus implements Serializable {
    private static Hashtable ekstensje = new Hashtable();
//ekstencje są kontenerem mapującym przechowuje on klucze oraz wartości
zawierające kolekcje z referencjami do instancji danej klasy.
//Dzięki implementacji Serializable mamy trwałość ekstynkcji.
    public ObjectPlus() {
        Vector ekstensja = null;
        Class klasa = this.getClass();
        if(ekstensje.containsKey(klasa)) {
            // Ekstensja tej klasy istnieje w kolekcji ekstensji
            ekstensja = (Vector) ekstensje.get(klasa);
        }
        else {
            // Ekstensji tej klasy jeszcze nie ma -> dodaj ją
            ekstensja = new Vector();
            ekstensje.put(klasa, ekstensja);
        }
        ekstensja.add(this);
    }
//Zapisywanie wszystkich ekstynkcji do pliku:
    public static void zapiszEkstensje(ObjectOutputStream stream) throws
IOException {
        stream.writeObject(ekstensje);
    }
 //Odczytywanie wszystkich ekstynkcji z pliku
    public static void odczytajEkstensje(ObjectInputStream stream) throws
IOException {
        ekstensje = (Hashtable) stream.readObject();
    }
}
```

ATRYBUTY,

ATRYBUTY WIELOKROTNE

Atrybuty wielokrotne Imię, nazwisko w klasie Osoba zostały zamienione na pojedyncze typu string, W trakcie korzystania z systemu nie jest istotne czy imię jest pojedyncze czy składa się z kilku członów.

```
public abstract class Osoba {
    public String imiona;
    public String nazwisko;
//…
}
```

Wielokrotność argumentu kraje pod opieką została wprowadzona poprzez dodanie nowej klasy Kraj oraz połączenie jej asocjacją wiele do wielu z klasą Help Desk. Rozwiązanie to umożliwi w przyszłości ograniczanie Konkursu do wybranych krajów oraz uniemożliwi opiekę nad nieistniejącymi krajami.

```
public class HelpDesk extends Osoba {
	public ArrayList<Kraj> krajePodOpieka; //Argument wielokrotny
//…
}
```

ATRYBUTY OPCJONALNE

Atrybuty opcjonalne, mają przypisaną wartość null w przypadku gdy nie występują. Np. argument telefon w klasie Osoba:

```
public String telefon = null;
```

ATRYBUTY O ZASIĘGU KLASOWYM

Atrybuty o zasięgu klasowe zostały zaimplementowane poprzez deklaracje zmiennych typu static. Na przykład zmienna max rozmiar w klasie Plik została zaimplementowana w następujący sposób:

```
final static float maxRozmiar;
```

ATRYBUTY POCHODNE

Atrybut pochodny wiek w klasie Osoba zarejestrowana została zaimplementowana przez metodę getWiek

```
    public static int getWiek(LocalDate dataUrodzenia, LocalDate teraz) {
            return Period.between(dataUrodzenia, teraz).getYears();
    }
```

Atrybut ocena średnia w klasie Ocenia został zaimplementowany przez funkcję zwracającą sumę wartości ocen cząstkowych pomnożonych przez ich wagę. Mnożenie przez wagę odbywa się w klasie Ocena Cząstkowa.

```
import java.util.ArrayList;
public class Ocenia {
public ArrayList<OcenaCzastkowa> ocenyCzastkowe;
	//…
	public static int getOcenaSrednia(ArrayList<OcenaCzastkowa> ocenyCzastkowe)
	{
	int suma = 0;
	for (int i = 0; i < ocenyCzastkowe.size(); i++) {
		suma += ocenyCzastkowe.get(i).wartosc;
	}
	return suma;
	}
	//…
}
```

W analogiczny sposób została zaimplementowany atrybut pochodny ocena końcowa.

ATRYBUTY UNIKALNE

Ograniczenie na unikalność atrybutu zostanie zaimplementowana poprzez sprawdzanie w ekstynkcji klasy, czy nie istnieje obiekt posiadający już ten sam atrybut. To znaczy przy tworzeniu nowego obiektu

METODY

METODY ZASIĘGU KLASOWYM,

Przykładem metody o zasięgu klasowym w projekcie jest metoda o znajdź najlepiej ocenioną pracę. Jej implementacja będzie polegała na znalezieniu w ekstynkcji klasy obiektu o najwyższym atrybucie ocena Końcowa.

OPERACJE POLIMORFICZNE

. Jedyną metodą tego typu znajdującą się w projekcie jest metoda. uzupełnij CV. Mentor i Ewaluator muszą podać swój krótki życiorys, natomiast zwykły uczestnik nie musi tego robić. Tę operację polimorficzną zrealizowano poprzez dodanie ograniczenia na minimalną długość CV w przypadku wywoływania tej operacji przez Mentora i Ewaluatora.

```
// W klasie Ewaluator i Mentor
private void setkrotkieCV(String krotkiecv) throws Exception {
    if(krotkiecv.length() < 50) {
        throw new Exception("Długość CV nie może być mniejsza od 50");
    }
    this.krotkiecv = krotkiecv;
}

// W klasie Uczestnik
private void setkrotkieCV(String krotkiecv) throws Exception {
    this.krotkiecv = krotkiecv;
}
```

ASOCJACJE, AGREGACJE, KOMPOZYCJE

ASOCJACJE I AGREGACJE

Asocjacje i agregacje zostały zrealizowane poprzez referencje. Przykładem asocjacji jeden do wielu jest asocjacja zadaje pomiędzy klasami Osoba i Pytanie. Przykładem asocjacji wielu do wielu jest asocjacja pomiędzy klasami Help Desk a Pytanie. Ich implementacja wyglądają w następujący sposób:

```
public abstract class Osoba {
//…
    public ArrayList<Pytanie> pytania;
//…
}

public class Pytanie {
//…
      public Osoba autor;
      public ArrayList<help_desk> helpdeski;
//…
}

public class help_desk extends Osoba {

//…
      public ArrayList<Pytanie> pytania;
//…
```

```
}
```

Klasa Organizator i Ewaluator były połączone ze sobą dwiema asocjacjami, jedna odpowiadała za potwierdzanie rejestracji druga za przypisywanie prac. W celu implementacji tych dwóch asocjacji wystaczy jedna nazwana nadzoruje.

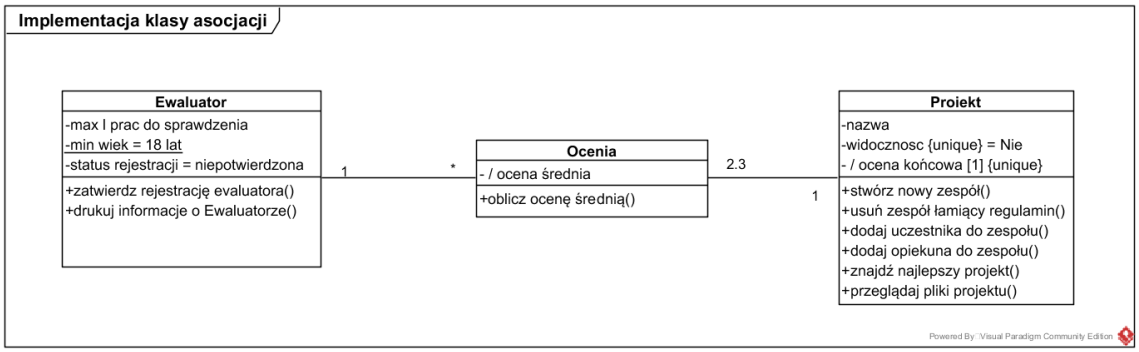
KOMPOZYCJE

Kompozycje. Zostały wykorzystane między klasą Projekt i Zespół oraz między Plik i Projekt. Tę pierwszą agregację zrealizowano łącząc klasę Projekt z Zespołem. Ponieważ jeden zespół może zrealizować tylko jeden projekt to tego typu połączenie nie spowoduje straty informacji. Implementację kompozycji między klasą Plik a Projekt została zrealizowana w następujący sposób:

```
public class Projekt {
private Vector<Plik> pliki = new Vector<Plik>();
//…
    public void dodajPlik(Plik plik) {
        if(!pliki.contains(plik)) {
            pliki.add(plik);
        }
    }
}

public class Plik {
      public String nazwa;
      private Projekt projekt;
      private Plik(Projekt projekt, String nazwa) {
            this.nazwa = nazwa;
            this.projekt = projekt;
      }

public static Plik utworzPlik(Projekt projekt, String nazwa) throws Exception {
       if(projekt == null) {
            throw new Exception("Projekt nie istnieje!");
       }
       Plik p = new Plik(projekt, nazwa);
       projekt.dodajPlik(p);
       return p;
}

}
```

KLASA ASOCJACJI

Klasa asocjacji Oceniaj została zaimplementowana poprzez zamienienie jej na zwykłą klasę połączoną asocjacjami z klasami Ewaluator i Projekt. Implementacja asocjacji została zrealizowana w sposób analogiczny do przedstawionego powyżej.



OGRANICZENIA

OGRANICZENIA STATYCZNE:

Ograniczenie liczbowe określające minimalną lub maksymalną wartość pewnego argumentu zostaną zrealizowane w następujący sposób:

```
public void setRozmiar(float rozmiar) throws Exception {
if(rozmiar > maxRozmiar) {
          throw new Exception("Plik musi być mniejszy od" + rozmiar);
}
}
```

W analogiczny sposób zostały zrealizowane ograniczenia dotyczące minimalnego wieku Ewaluatorów, Mentorów i Uczestników.

Ograniczenie polegając na przyjmowaniu określonych wartości przez atrybut zostały zrealizowane przez zmienne typu enum. Przykładem może być atrybut płeć w klasie Osoba

```
public enum plec {mezczyzna, kobieta};
```

lub atrybut status rejestracji w klasie Mentor i Ewaluator

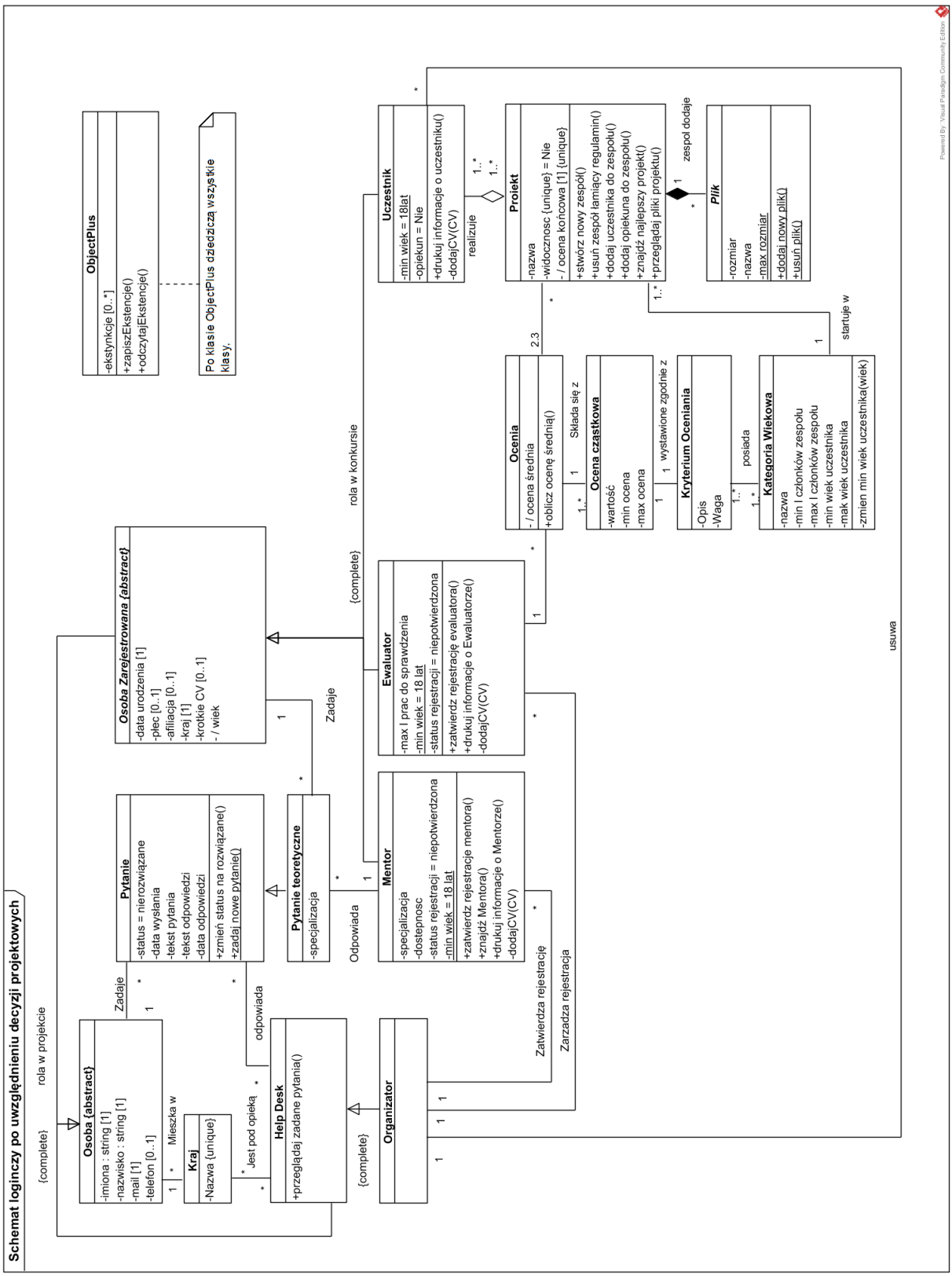
```
public enum statusRejestracji {potwierdzona, niepotwierdzona};
```

Osobnym przypadkiem jest ograniczenie związane z atrybutem kraj w klasie Osoba. Istnieje skończona liczba krajów na świecie, jednak jest ona na tyle duża by zastosowanie atrybutu typu enum było mało praktyczne. Problem ten rozwiązano dodając nową klasę Kraj. A wspomniany wyżej argument zastąpiono asocjacją.

OGRANICZENIA DYNAMICZNE

Ograniczenie dynamiczne związane jest z atrybutem minimalny wiek uczestnika w klasie kategoria wiekowa. Możliwe jest, że w trakcie konkursu organizatorzy postanowią dopuścić do udziału w danej kategorii również osoby młodsze niż wcześniej zakładano.

```
    private void setminLCzlonkowZespolu(int minWUczestnika) throws Exception {
        if(minWUczestnika > this.minWUczestnika) {
            throw new Exception("Minimalny wiek uczestnika nie może wzrosnąć");
        }

      this.minWUczestnika = minWUczestnika;

      }
```


Schemat loginczy po uwzględnieniu decyzji projektowych
ObjectPlus
-ekstynkcje [0..*]
+zapiszEkstencje()
+odczytajEkstencje()
Po klasie ObjectPlus dziedziczą wszystkie klasy;
Osoba {abstract}
-imiona : string [1]
-nazwisko : string [1]
-mail [1]
-telefon [0..1]
Osoba Zarejestrowana {abstract}
-data urodzenia [1]
-płec [0..1]
-afiliacja [0..1]
-kraj [1]
-krotkie CV [0..1]
- / wiek
Uczestnik
-min wiek = 18lat
-opiekun = Nie
+drukuj informacje o uczestniku()
-dodajCV(CV)
Projekt
-nazwa
-widocznosc {unique} = Nie
- / ocena końcowa [1] {unique}
+stwórz nowy zespół()
+usuń zespół łamiący regulamin()
+dodaj uczestnika do zespołu()
+dodaj opiekuna do zespołu()
+znajdź najlepszy projekt()
+przeglądaj pliki projektu()
Plik
-rozmiar
-nazwa
-max rozmiar
+dodaj nowy plik()
+usuń plik()
Pytanie
-status = nierozwiązane
-data wysłania
-tekst pytania
-tekst odpowiedzi
-data odpowiedzi
+zmień status na rozwiązane()
+zadaj nowe pytanie()
Pytanie teoretyczne
-specjalizacja
Ewaluator
-max l prac do sprawdzenia
-min wiek = 18 lat
-status rejestracji = niepotwierdzona
+zatwierdz rejestracje ewaluatora()
+drukuj informacje o Ewaluatorze()
-dodajCV(CV)
Mentor
-specjalizacja
-dostepnosc
-status rejestracji = niepotwierdzona
-min wiek = 18 lat
+zatwierdz rejestracje mentora()
+znajdź Mentora()
+drukuj informacje o Mentorze()
-dodajCV(CV)
Ocenia
- / ocena średnia
+oblicz ocenę średnią()
Ocena cząstkowa
-wartość
-min ocena
-max ocena
Kryterium Oceniania
-Opis
-Waga
Kategoria Wiekowa
-nazwa
-min l członków zespołu
-max l członków zespołu
-min wiek uczestnika
-mak wiek uczestnika
-zmien min wiek uczestnika(wiek)
Kraj
-Nazwa {unique}
Help Desk
+przeglądaj zadane pytania()
Organizator
{complete}
rola w projekcie
Zadaje
Mieszka w
Jest pod opieką
odpowiada
Odpowiada
Zadaje
{complete}
rola w konkursie
Składa się z
wystawione zgodnie z
posiada
zespol dodaje
startuje w
usuwa
Zatwierdza rejestrację
Zarządza rejestracją
realizuje
Powered By Visual Paradigm Community Edition