

Tomasz Kołodziejak

gr. 614 s5206

Imprezogenerator.

1. CEL PROJEKTU

Celem tego projektu jest stworzenie systemu dla firmy organizującej różnego rodzaju imprezy okolicznościowe, oraz uzyskanie zaliczenia z PRI.

2. ZAKRES

1. Wyliczanie pensji miesięcznej pracownika.
2. Sprawdzanie dostępności organizatora(nie może organizować więcej niż jednej imprezy na raz).
3. Zamawianie, anulowanie, edytowanie imprezy.
4. Wyliczanie ceny imprezy.
5. Wyświetlani listy realizowanych imprez(tylko dla organizatorów).

3. KONTEKST

W systemie wyróżniamy:

- Klienta – osobę składającą zamówienie na realizację imprezy na wybraną przez siebie okoliczność. Może też taką imprezę odwołać(jest wtedy usuwana z systemu).

- Organizatora – osobę odpowiedzialną za realizację imprezy zgodnie z oczekiwaniami klienta. Może on edytować dane w systemie dotyczące imprezy i postępów w jej realizacji oraz mieć wgląd w dane imprez.

- Szefa – właściciela firmy organizującej imprezy. Zarządza on klientami (dodaje ich do systemu), zarządza imprezami (dodaje, usuwa, podgląda, wylicza cenę), zarządza pracownikami (sprawdza czy są wolni, przydziela do imprez wylicza pensje).

- Czasowy – system co 3 miesiące sprawdza, czy klienci dokonywali w tym czasie zamówienie. Jeśli nie to usuwa ich z systemu.

4. WYMAGANIA FUNKCJONALNE:

- W systemie mają być przechowywane dane osobowe zarówno pracowników (osoba przydzielona do organizacji imprezy, kelner) oraz dane o kliencie. Klient, który przez 3 miesiące nie organizował imprezy jest usuwany z systemu. Pracownik może być jednocześnie klientem.
- Dla każdego pracownika system ma przechowywać wysokość jego pensji, która wyliczana jest na podstawie ilości godzin przepracowanych w miesiącu. Dodatkowo dla organizatora znana jest jego specjalizacja (jakie imprezy organizuje). Dla kelnera pamiętamy stawkę godzinową oraz, czy jest certyfikowanym kelnerem (certyfikowani kelnerzy mają wyższą pensję o 15%).
- W systemie jest przechowywana informacja o imprezach. Każda impreza ma przypisane nr (unikalny), miejsce, datę rozpoczęcia i datę zakończenia, ilość osób, rodzaj preferowanej kuchni oraz opcjonalnie, czy ma być na niej uwzględniony alkohol i aktualny stan (czy oczekuje na realizację, w trakcie, czy już zrealizowana).
- Firma zajmuje się organizowaniem imprez takich jak: śluby, imprezy noworoczne, chrzciny itp. Dla ślubów przechowywane mają być informacje o kościele w którym ślub nastąpi, informacje o marce samochodu którym młoda para będzie się przemieszczała, oraz o zespole który zagra podczas wesela. Na imprezę noworoczną klient może wybrać ilość fajerwerków, repertuarze muzycznym i opcjonalnie o kostiumach. Przy pogrzebie klient wybiera rodzaj pochówku, oraz cmentarz. Cena imprezy zależy od jej indywidualnych atrybutów (np. od samochodu).
- Organizator dobierany jest do przygotowania imprezy na podstawie swojej specjalizacji.
- Przy dłuższych imprezach ma być przechowywana godzina rozpoczęcia i zakończenia pracy przez kelnera. (Na dłuższej imprezie, np. kilkudniowej, może pracować wielu kelnerów, w różnych dniach i godzinach).
- System powinien umożliwiać realizowanie następujących funkcji:
 1. Wyliczanie pensji miesięcznej pracownika.
 2. Sprawdzanie dostępności organizatora (nie może organizować więcej niż jednej imprezy na raz).
 3. Zamawianie, anulowanie, edytowanie imprezy.
 4. Wyliczanie ceny imprezy.
 5. Wyświetlenie listy realizowanych imprez (tylko dla organizatorów).
 6. Przydzielenie organizatora do imprezy.
 7. Dodawanie klienta do systemu.

5. WYMAGANIA NIEFUNKCJONALNE

- System powinien być niewymagający sprzętowo, tak aby użytkownik nie musiał ponosić dużych kosztów w związku z zakupem hardware go obsługującego.
- System powinien móc pracować w środowisku Windows/Linux/MacOs
- System powinien mieć nieskomplikowany interfejs graficzny, obsługiwany za pomocą myszki lub klawiatury.
- Dane powinny być uporządkowane, reprezentowane najlepiej w postaci tabel.
- System powinien być chroniony hasłem.

6. EWOLUCJA SYSTEMU

W przyszłości system powinien uwzględniać podział na klientów zwykłych i stałych. Po osiągnięciu statusu klienta stałego, klientowi przysługiwałyby zniżki. System po sprawdzeniu statusu klienta wysyłałby automatycznie maile do stałych klientów z promocyjnymi ofertami.

7. OPIS PRZYPADKÓW UŻYCIA

Aktor	Przypadek użycia	Opis
Klient	Złóż zamówienie	Klient składa zamówienie na organizację imprezy na wybraną przez siebie okoliczność.
	Anuluj zamówieni	Klient anuluje zamówieni, co skutkuje usunięciem go z systemu.
Szef	Dodaj imprezę	Dodanie zamówienia do systemu.
	Przydziel organizatora	Przydzielenie wolnego organizatora o odpowiedniej specjalizacji do organizacji imprezy.
	Dodaj klienta	Rejestracja klienta w systemie.
	Policz cenę	Liczy cenę za organizację imprezy.
	Policz pensję pracownika	Wylicza pensję pracownika za ostatni miesiąc.
	Usuń imprezę	Usunięcie zamówienia z systemu.
	Pokaż imprezę	System pokazuje na jakim etapie jest przygotowywana impreza (co zostało już załatwione i czy w ogóle zostało).
Organizator	Edytuj imprezę	Zmiana atrybutów dotyczących zamówienia.
	Pokaż imprezę	System pokazuje na jakim etapie jest przygotowywana impreza (co zostało już załatwione i czy w ogóle zostało).
System	Usuń klienta	Jeśli klient przez ostatnie 3 miesiące nie złożył zamówienia zostaje usunięty z systemu.

Diagram PRI

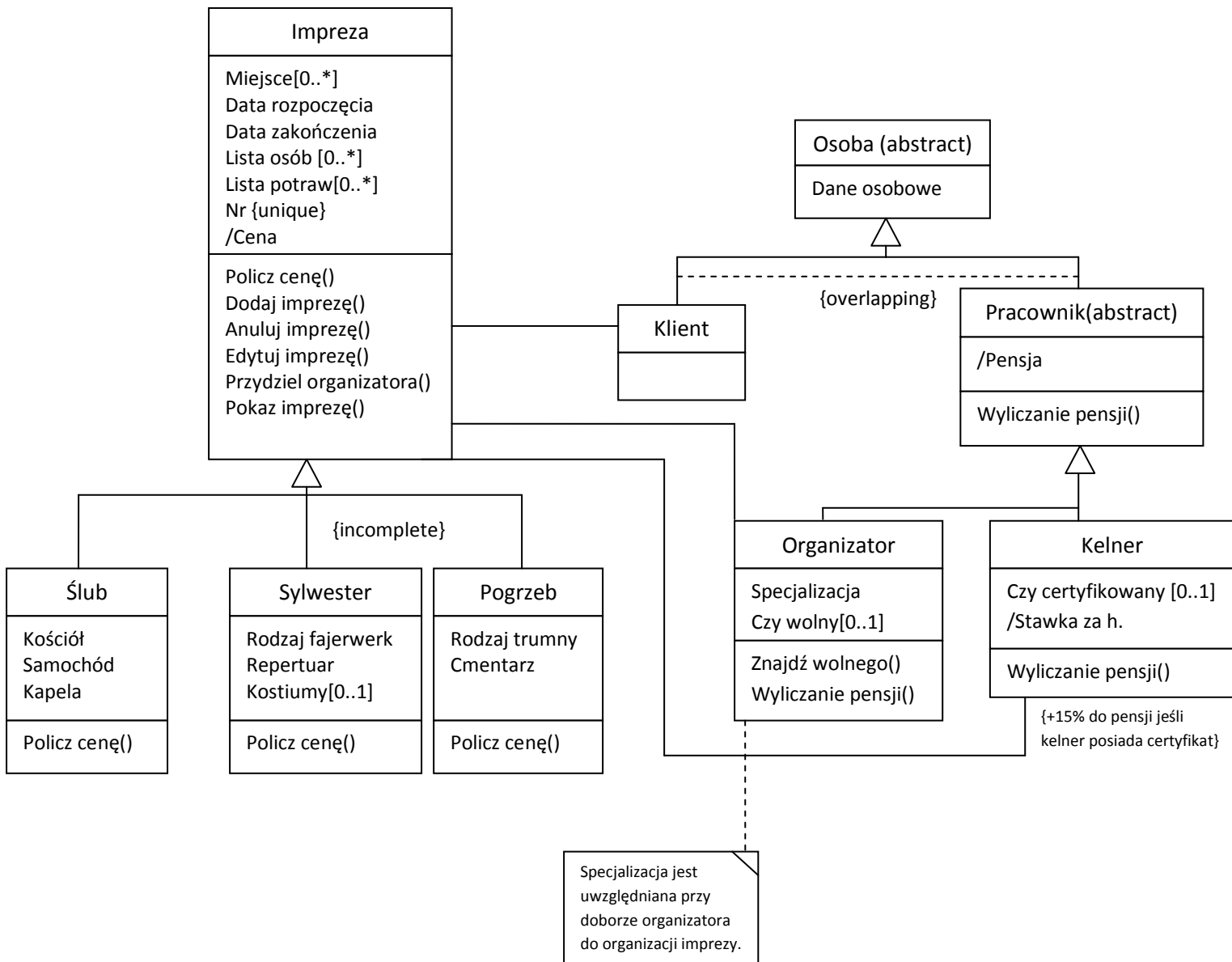
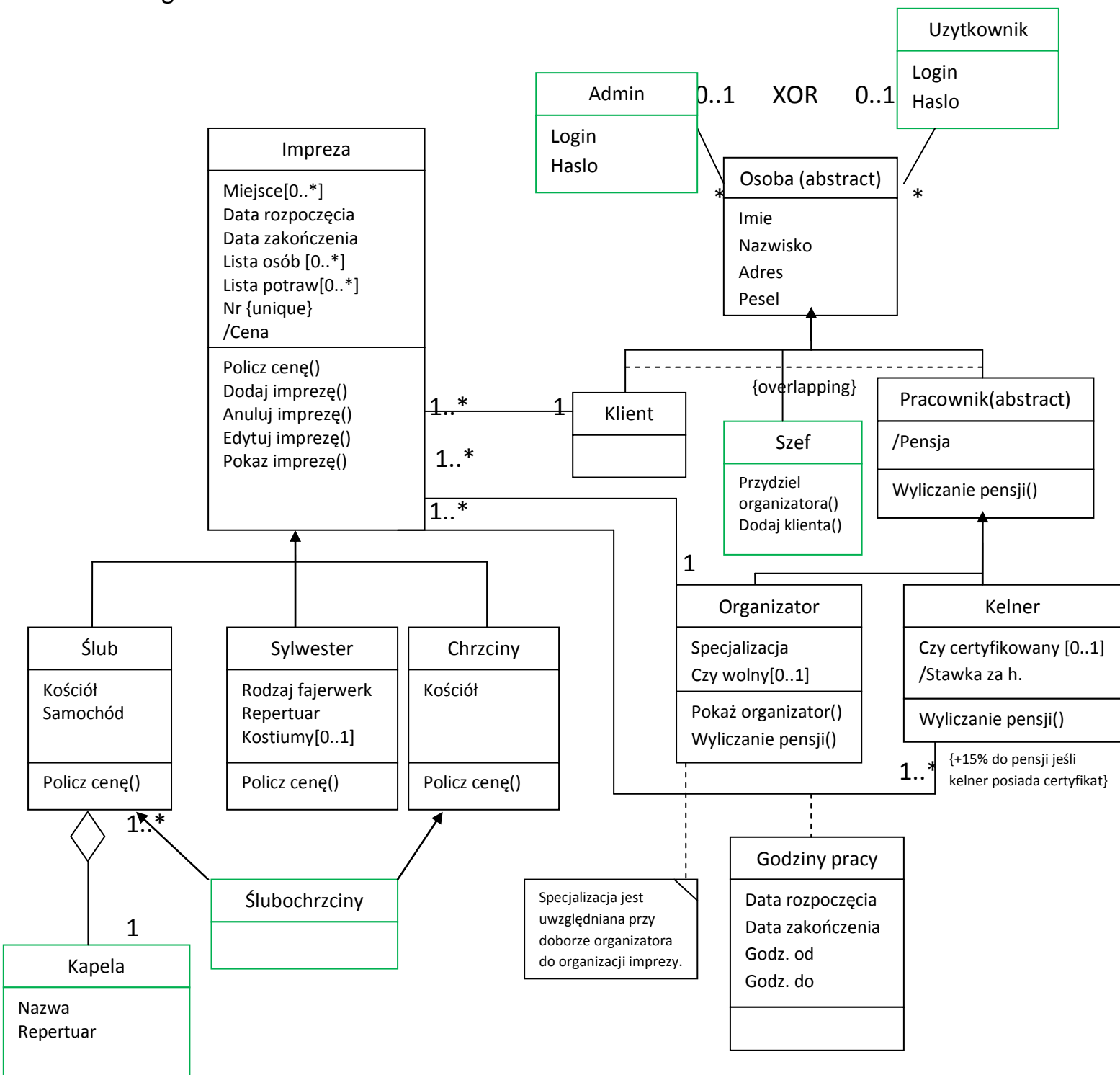


Diagram MAS



Implementacja:

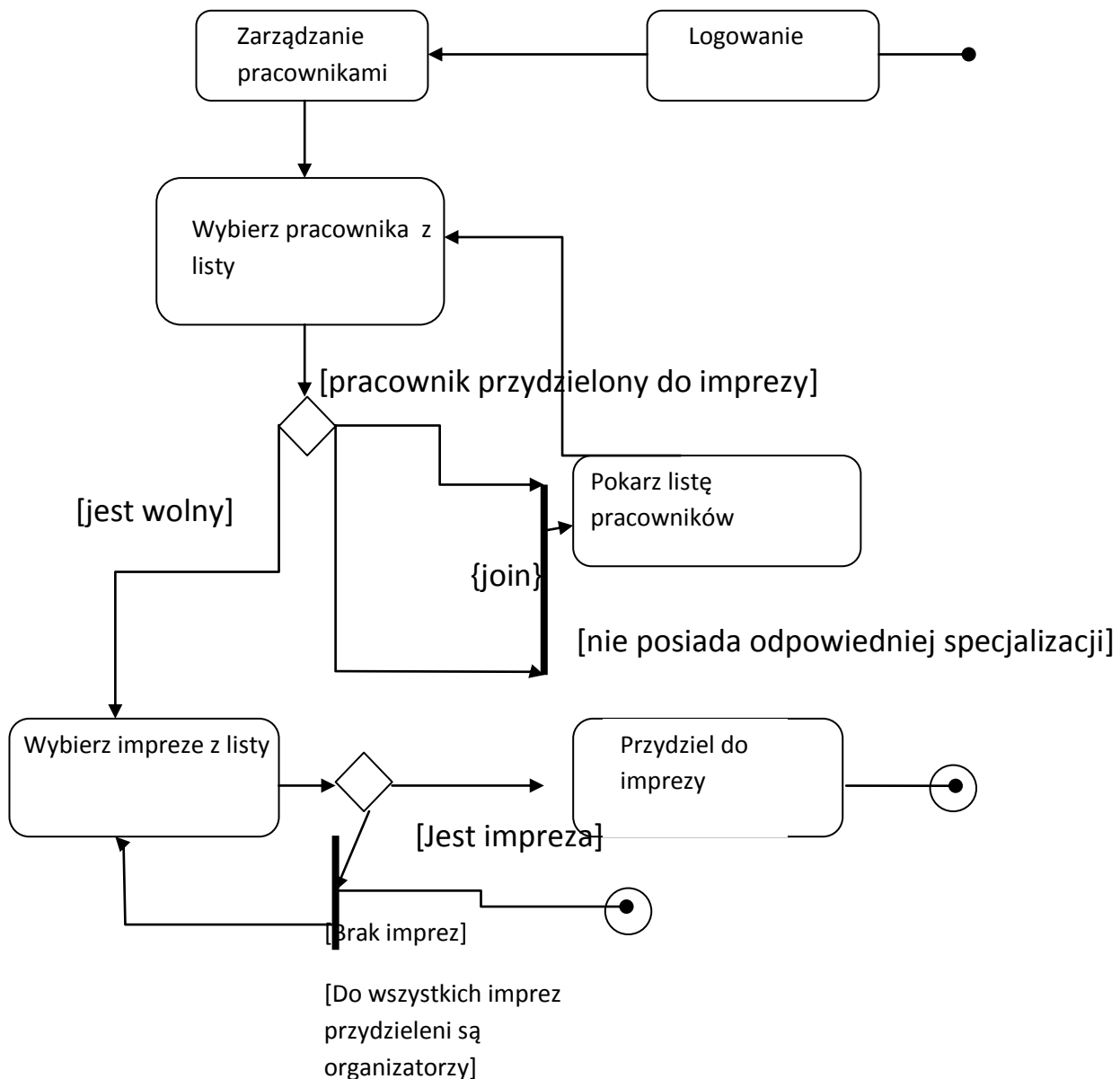
Na potrzeby przechowywania informacji o godzinach pracy kelnera na konkretnej imprezie niezbędne było wprowadzenie asocjacji z atrybutem. Java nie oferuje tego rozwiązania bezpośrednio. Wykorzystany model relacyjny w bazie danych traktuje klasę „Godziny pracy” jako asocjację pomiędzy imprezą a kelnerem.

Ekstensje oraz dziedziczenie realizowane są poprzez bazę danych MySQL z którą Java łączy się dzięki użyciu lżejszej od Hybernata biblioteki Jenkov Butterfly. Na potrzeby tego projektu okazuje się być wystarczająca.

W projekcie dodatkowo została umieszczona wielodziedziczenie realizowane przez klasę ” Slubochrzciny”.

Aby nadać użytkownikom odmienne uprawnienia stworzeni zostali administrator i użytkownik. Ograniczenie XOR zapewnia , że ich kompetencje będą różne.

Kapela dzięki zastosowaniu agregacji została trwale związana z organizowanym ślubem

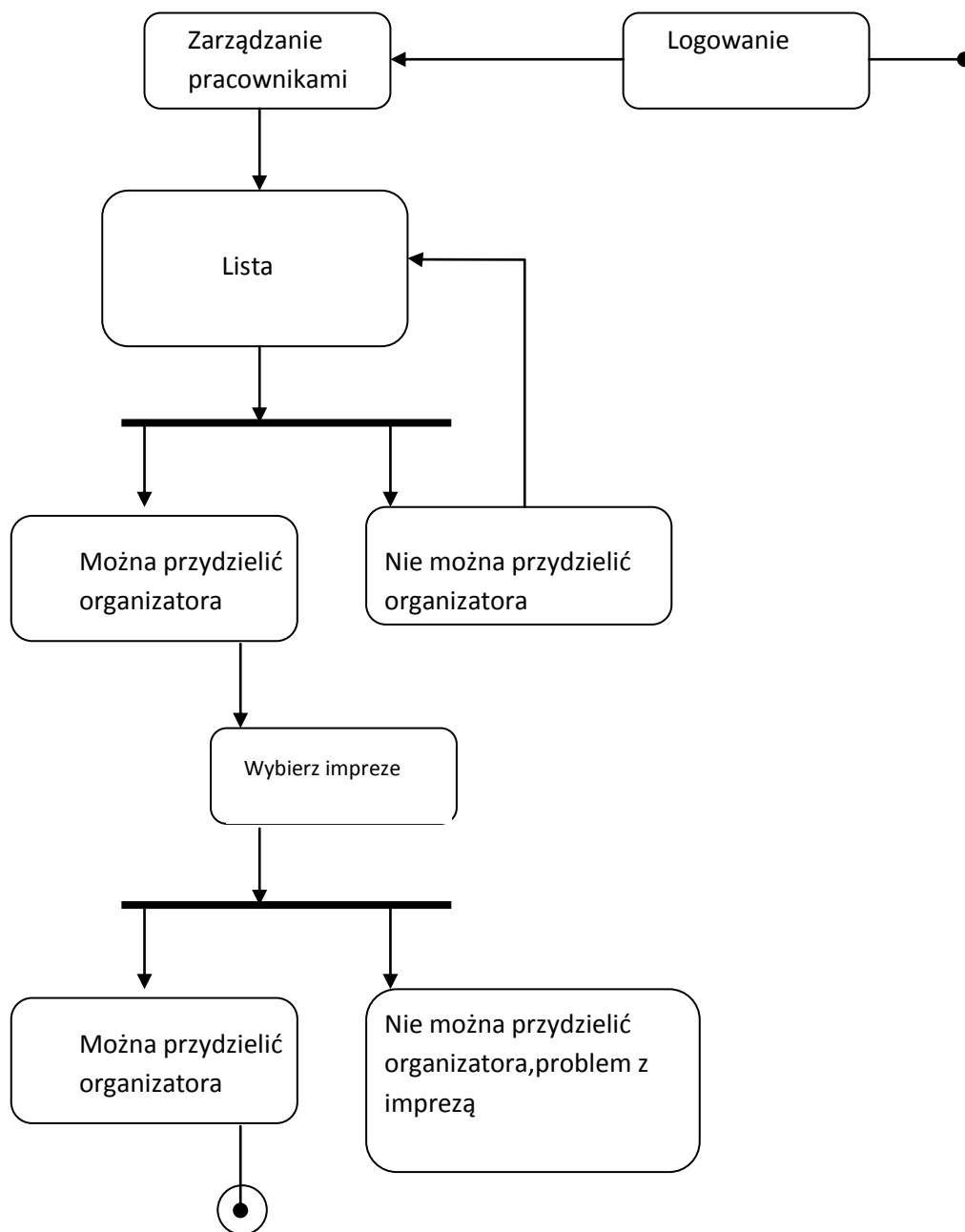


Scenariusz:

1. Przypadek użycia rozpoczyna się z momentem logowania się szefa do systemu.
2. System prosi o podanie loginu i hasła.
3. Po poprawnym zalogowaniu wyświetlany jest panel programu.
4. Użytkownik wybiera zakładkę „Lista pracowników”
5. Użytkownik wybiera pracownika z listy .
6. Użytkownik wybiera impreze.
7. Użytkownik przydziela użytkownika do imprezy
8. Przypadek użycia kończy się.

Alternatywa:

- 5a. Wybrany pracownik jest już przydzielony do jakiegoś zadania
- 5b. Wybrany pracownik nie posiada odpowiedniej specjalizacji
- 6a. Brak organizowanej imprezy bez organizatora
- 6b. Brak organizowanych imprez, powrót do głównego okna



Analizując powyższe diagramy możemy zauważyć następujące wnioski
 Przypadek użycia nie rozpocznie się do momentu aż użytkownik z nadanymi uprawnieniami nie zaloguje się do systemu. W momencie wyboru pracownika może on być już przydzielony do jakiegoś konkretnego zadania co uniemożliwia przypisanie go do nowego. W momencie gdy specjalizacja organizatora, pomimo jego dostępności, nie odpowiada rodzajowi imprezy, przypadek użycia również nie zostanie zakończony pomyślnie. Dopiero po spełnieniu 2 warunków, dostępności i odpowiedniej specjalizacji, można przejść do wyboru imprezy do której może zostać przypisany. W tym momencie organizatora można przydzielić jedynie jeśli w systemie jest impreza której jeszcze nikt nie organizuje. W przypadku braku imprezy, bądź obsadzonych wszystkich stanowisk przydzielenie jest niemożliwe i przypadek użycia kończy się.

Konceptcja GUI

Okno aplikacji	
Logowanie	
Login	<input type="text"/>
Haslo	<input type="text"/>

Okno aplikacji		
Menu		
Zarządzanie pracownikami		
<div style="border: 1px solid black; width: 200px; height: 100px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">Lista</div>		
<input type="button" value="Wybierz impreze"/>	<input type="button" value="Wylucz pensje"/>	<input type="button" value="Wróć"/>

