

Polsko-Japońska Akademia Technik Komputerowych
Wydział Informatyki

Projekt indywidualny pt. „Muzeum”

Warszawa, 11.06.2023r.

Spis treści

1. Informacje ogólne	3
1. Dziedzina problemowa	3
2. Cel	3
3. Zakres odpowiedzialności systemu	3
4. Użytkownicy systemu	3
5. Wymagania użytkownika	3
6. Wymagania niefunkcjonalne	5
7. Opis przyszłej ewolucji systemu	5
8. Słownik pojęć z dziedziny problemowej	5
2. Diagram przypadków użycia	6
3. Diagram klas – analityczny	7
4. Diagram klas – projektowy	8
5. Scenariusz do przypadku użycia „Kupienie biletu”	9
6. Diagram aktywności do przypadku użycia „Kupienie biletu”	10
7. Diagram stanu dla klasy „Bilet”	11
8. Diagram interakcji (sekwencji) dla przypadku użycia „Kupienie biletu”	12
9. Projekt GUI	13
10. Omówienie decyzji projektowych i skutków analizy dynamicznej	20

1. Informacje ogólne

1. Dziedzina problemowa

Projekt został zamówiony przez dyrektora muzeum regionalnego.

2. Cel

System informatyczny ma za zadanie ułatwić nawigację w kompleksie muzealnym. W ten sposób można rozdzielić pracę, przechowywać informacje o dziełach sztuki i zachęcić gości do rozwijania zainteresowań z zakresu kultury i sztuki nowoczesnej oraz dawnej.

3. Zakres odpowiedzialności systemu

System powinien umożliwić zarządzanie informacjami dotyczącymi:

- Pracowników
- Zwiedzających i grup, w które wchodzi
- Uczestników
- Placówek muzealnych
- Ekspонатów i ich autorów
- Wystaw
- Warsztatów

Należy zapewnić właściwy dostęp do przechowywanych danych oraz zmian w organizacji miejsc pracy.

Ważna dla systemu jest lista pracowników, ich stażu, dokonań oraz kompetencji.

Od systemu nie oczekuje się trwałego przechowywania danych zwiedzających, w przeciwieństwie do danych na temat dzieł sztuki i wystaw stałych oraz czasowych. Dane na ich temat powinny być szczegółowe i dostępne do wglądu pracownikom oraz gościom.

Zarządzanie zarejestrowanymi gośćmi strony internetowej, których dane nie są systematycznie usuwane nie wchodzi w skład projektowanego systemu.

4. Użytkownicy systemu

Potencjalnymi użytkownikami systemu będą zwiedzający muzeum, pracownicy (przewodnicy, kierownicy i prowadzący), uczestnicy warsztatów oraz automatyczny system oraz podsystem czasu.

5. Wymagania użytkownika

1. Do każdej placówki muzeum w systemie należy pamiętać jej nazwę, adres, nr telefonu, godziny otwarcia i zamknięcia oraz zwiedzania. Aktualnie wszystkie placówki są czynne w godzinach 10:00-18:00.
2. W systemie należy przechowywać dane kontaktowe zwiedzających, czyli imię, nazwisko, datę urodzenia, nr telefonu i email, a w przypadku pracowników dodatkowo datę zatrudnienia, staż pracy podany w latach i pensję. Dla uczestników zapamiętujemy ich wiek. Uczestnik może być równocześnie zwiedzającym. Pracownicy dzielą się na przewodników, kierowników i prowadzących. Podział jest kompletny. Dla każdego

pracownika ważny jest jego staż pracy w obecnym i poprzednim miejscu pracy - w przypadku specjalistów nie mniejszy niż 5 lat.

3. Zwiedzający mogą wchodzić w skład grupy, do której jest przydzielony 1 przewodnik o unikatowym id. Należy zapamiętać typ biletu zakupionego przez zwiedzającego – „ulgowy”, „normalny”, określić licznosc grupy (obecnie maksymalna liczba uczestników to 20 osób) i jej status – „zaplanowana”, „w trakcie zwiedzania”, „zakończono zwiedzanie”. Trzeba pamiętać datę i godzinę wejścia każdej grupy. Na początku każdego miesiąca dane dotyczące grupy o ukończonym zwiedzaniu są automatycznie usuwane.
4. W trakcie zapisania się zwiedzającego do nowej grupy powinny zostać podane informacje: placówka, data, rodzaj biletu, cena biletu i rodzaj wystawy (czasowa, do której potrzeba specjalnego biletu lub stała). Należy również przydzielić przewodnika do grupy. Grupa w danej placówce może nie być dostępna dla podanej ilości biletów. W takim przypadku użytkownik systemu może spróbować ponownie, albo nie zapisać się do żadnej grupy. Wtedy, w tej samej cenie, będzie mógł wejść z biletem do placówki muzeum. Jeżeli wszystkie dane zapisu są zaakceptowane przez klienta (łącznie z kosztem wyliczonym przez system), zapis zostaje zapamiętany w systemie. W tym samym momencie mogą zwiedzać maksymalnie 2 grupy.
5. Do każdej placówki muzeum może należeć od 1 do 2 kierowników. Każdy kierownik może przynależeć do tylko jednej placówki.
6. Uczestnicy mogą zapisać się na warsztaty. Dla każdego warsztatu należy zapamiętać minimalny i maksymalny wiek uczestnika, datę rozpoczęcia, zakończenia i dni, w których warsztaty odbywają się w placówce. Do każdego warsztatu może należeć od 1 do 2 prowadzących, który również je prowadzi.
7. Do każdej wystawy przynależą eksponaty, które posiadają nazwę, opis, datę powstania, rodzaj i styl. Każdy z nich może mieć kilku autorów, dla których zapamiętujemy ich imię, nazwisko, datę urodzenia i ewentualnie datę śmierci.
8. Wystawy dzieli się na stałe i czasowe. Dla wystaw czasowych pamięta się ich nazwę, początek, koniec oraz ich cenę.
9. W muzeum prowadzi się również warsztaty, dla których pamiętana jest tematyka zajęć, przedział wiekowy uczestnika oraz dni, w których odbywają się w placówce. W danej placówce w jednym momencie mogą być organizowane tylko jedno takie zajęcia.
10. System powinien umożliwić dyrektorowi muzeum zrealizowanie poniższych funkcjonalności:
 - 10.1. Zorganizowanie nowej wystawy (kierownik)
 - 10.2. Zorganizowanie warsztatów wiążące się z przypisaniem prowadzącego do warsztatów (kierownik)
 - 10.3. Przypisanie przewodnika do danej grupy (podsystem zarządzania bazą danych)
 - 10.4. Wyświetlenie listy placówek z możliwością wyświetlenia wystaw (zwiedzający, prowadzący, kierownik, przewodnik, uczestnik)
 - 10.5. Wyświetlenie listy wystaw z możliwością wyświetlenia listy eksponatów (zwiedzający, uczestnik, przewodnik, prowadzący, kierownik)
 - 10.6. Wyświetlenie listy warsztatów (zwiedzający, prowadzący, kierownik, przewodnik, uczestnik) z możliwością zapisania się na nie (uczestnik)
 - 10.7. Dodanie eksponatu (kierownik)
 - 10.8. Kupienie biletu wiążące się z wyświetleniem listy placówek i z możliwością zapisania się do grupy (zwiedzający)

10.9. Usunięcie danych grupy (podsystem czasu)

6. Wymagania niefunkcjonalne

Ograniczenie	Miara
Dostępność	W jakim czasie powinien być dostępny system: przez cały czas, bez przerw
Wszechstronność interfejsu użytkownika	Interfejs powinien być dostępny w języku polskim, niemieckim i angielskim.
Komfort wyszukiwania	Powinien istnieć podział dzieł sztuki ze względu na epoki powstania, autorów, kraj pochodzenia.
Dostęp dla wielu użytkowników	System nie powinien być przeciążony i obsługiwać równocześnie setki użytkowników, bez względu na sezon, w weekendy i święta.
Zabezpieczenia	Powinno istnieć zabezpieczenie przed autoryzacją danych ze strony gości.
Przechowywanie danych w bazie danych	Serwerem bazy danych powinien być Oracle Server X8 lub nowszy

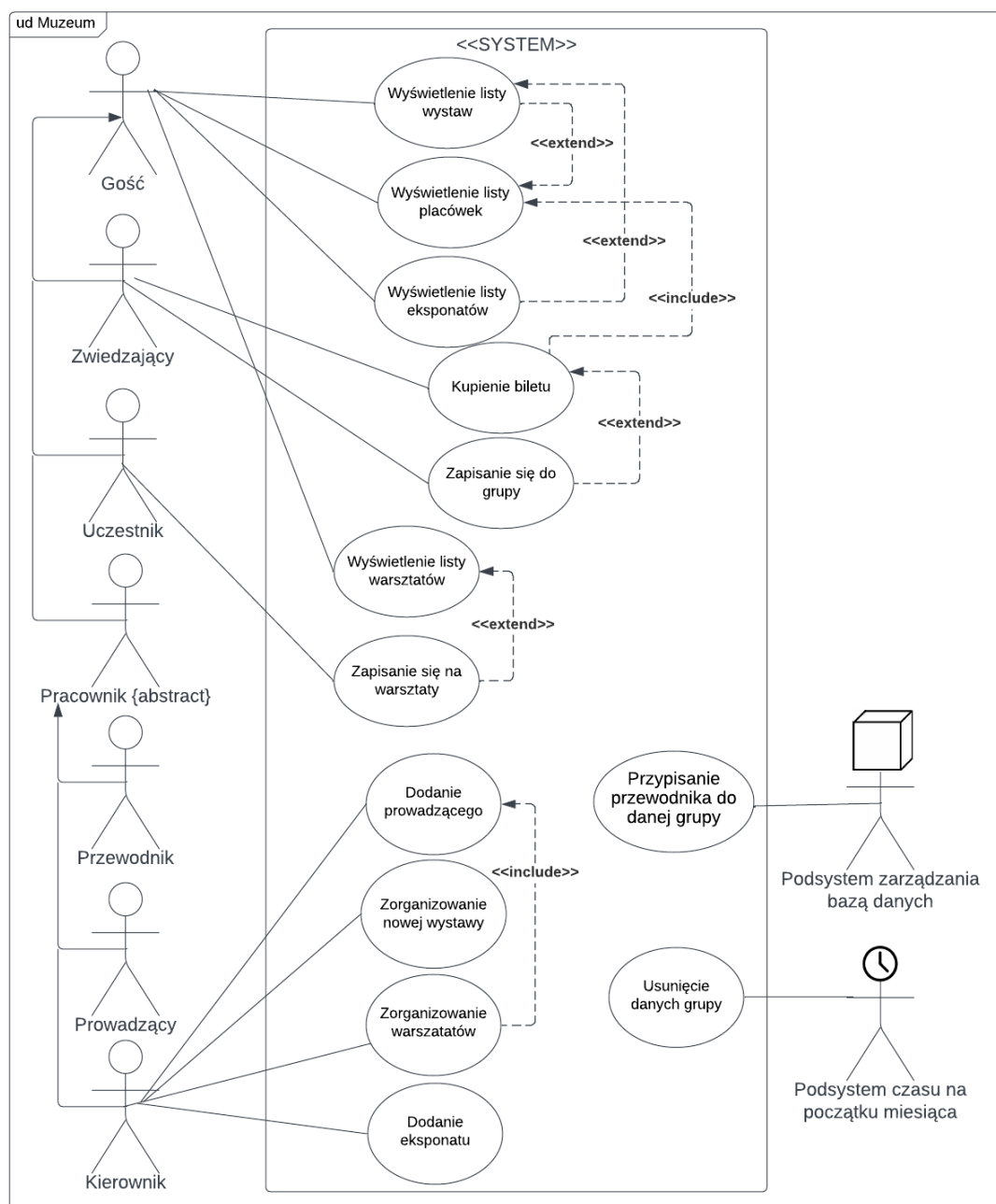
7. Opis przyszłej ewolucji systemu

W przyszłości planuje się wprowadzenie danych dotyczących kalendarium imprez oraz dnia bezpłatnego.

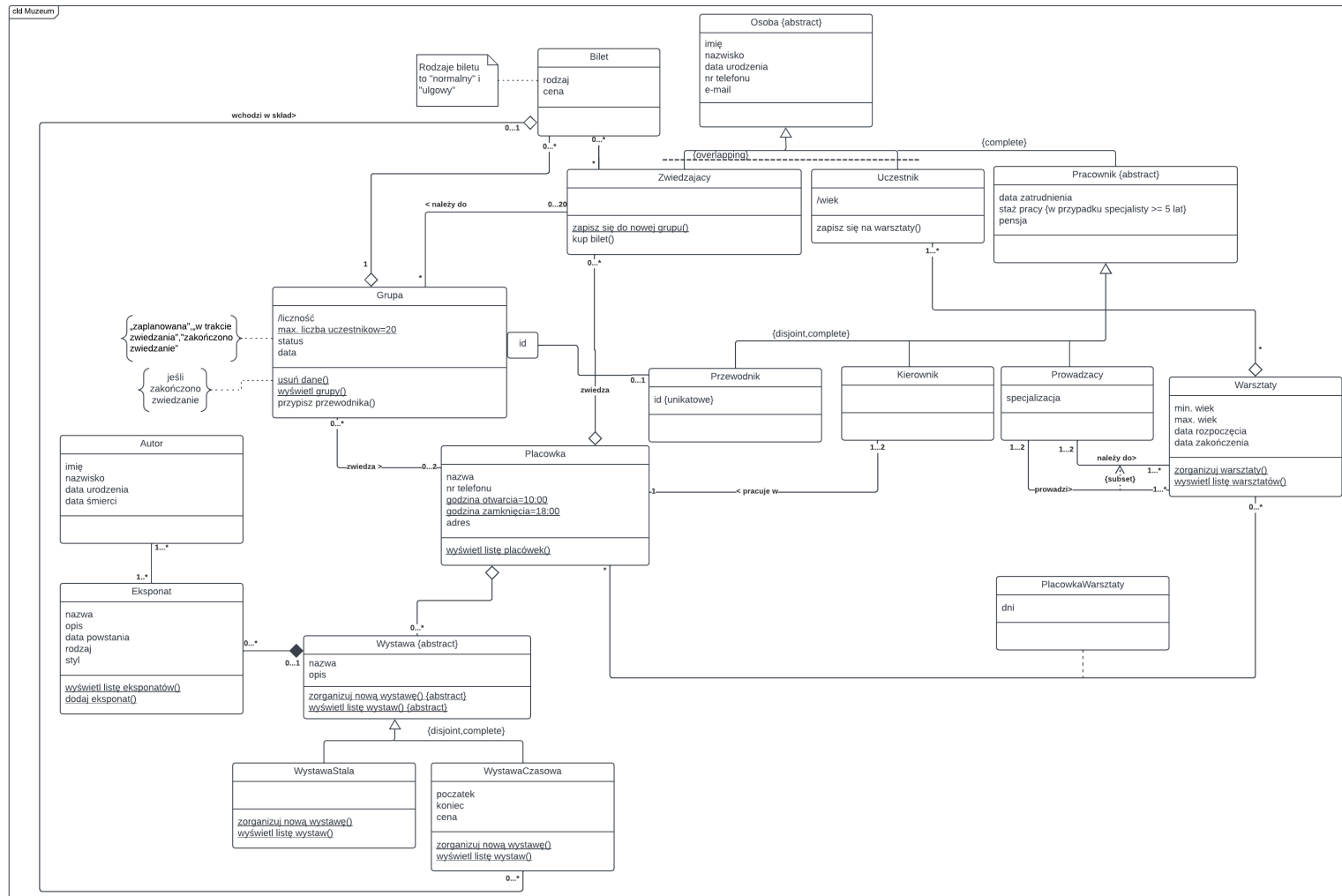
8. Słownik pojęć z dziedziny problemowej

Pojęcie (termin)	Synonim pojęcia	Objaśnienie
Grupa		Do 20 zwiedzających z przewodnikiem
Zwiedzający	Gość, klient	Osoba, która zapisała się na zwiedzanie, bądź dopiero się zapisze (bądź osoba uczęszczająca na warsztaty)
Placówka	Budynek, ośrodek muzealny	Obiekt kompleksu muzealnego

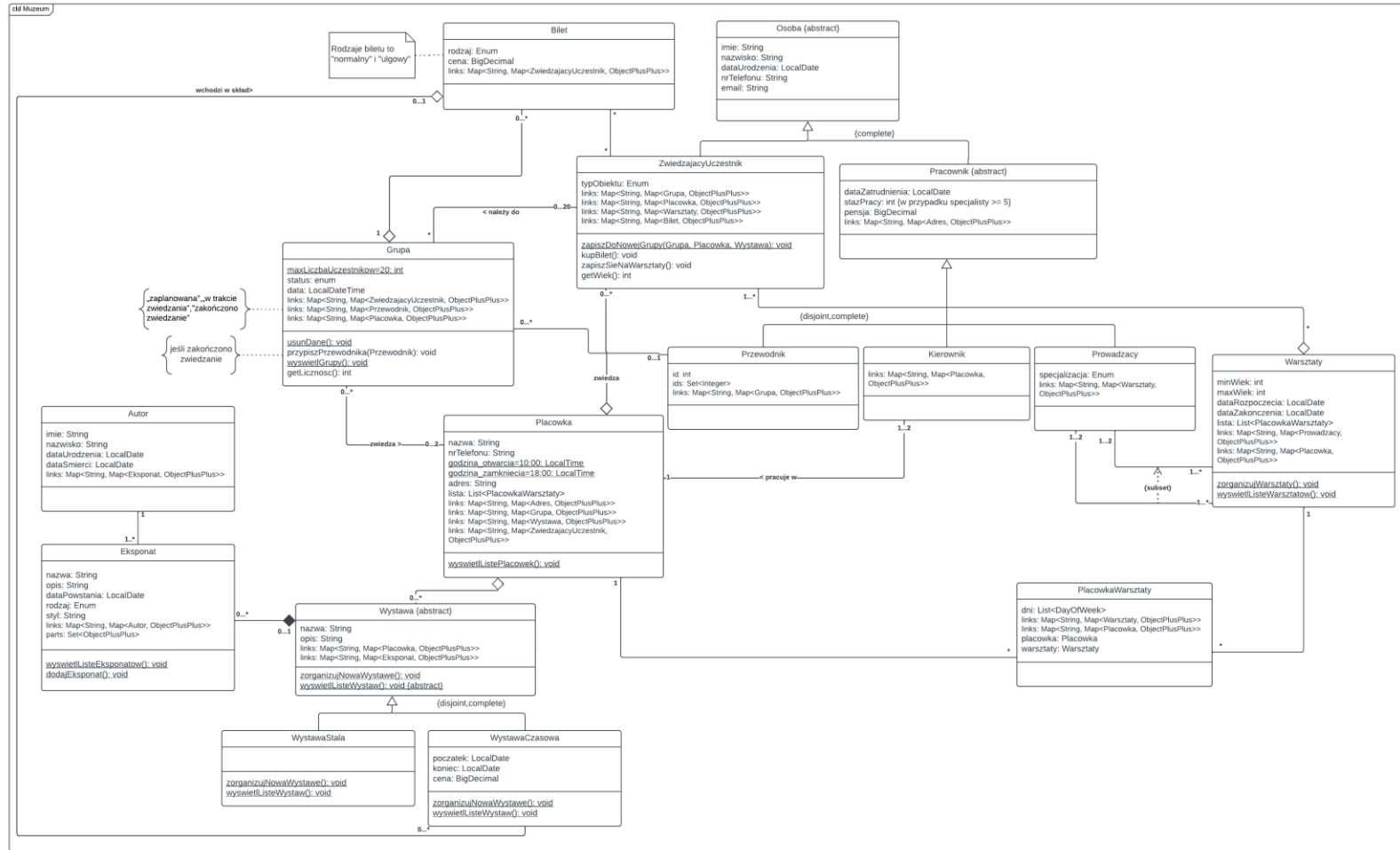
2. Diagram przypadków użycia



3. Diagram klas – analityczny



4. Diagram klas – projektowy



5. Scenariusz do przypadku użycia „Kupienie biletu”

Warunek początkowy: Gość wybrał opcję „Kup bilet”

Warunek końcowy: Gość kupił bilet i uiścił opłatę

Aktorzy: Zwiedzający

Do czego służy: Pozwala klientowi kupić bilet w wybranej placówce

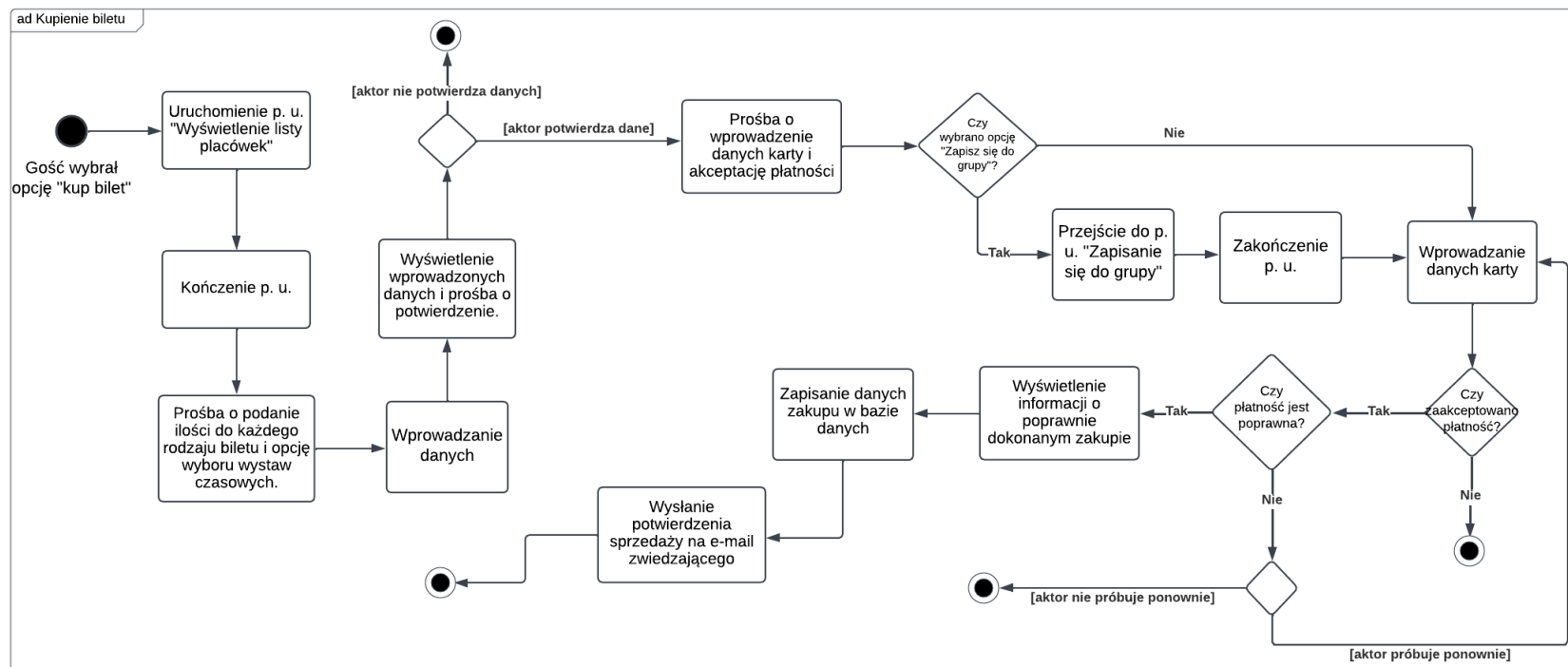
Główny przepływ zdarzeń:

1. Zwiedzający uruchamia p. u. „Kupienie biletu”.
2. Zwiedzający uruchamia p. u. „Wyświetlenie listy placówek”.
3. Zwiedzający kończy p. u. „Wyświetlenie listy placówek”.
4. System prosi o podanie ilości do każdego rodzaju biletu i opcję wyboru wystaw czasowych.
5. Zwiedzający wprowadza dane.
6. System wyświetla wprowadzone dane i prosi o potwierdzenie.
7. Zwiedzający potwierdza dane.
8. System prosi o podanie danych karty.
9. Zwiedzający wprowadza dane karty
10. Zwiedzający akceptuje płatność.
11. System wyświetla informację o poprawnym dokonaniu zakupu biletu.
12. System zapisuje dane zakupu.
13. System wysyła potwierdzenie sprzedaży na e-mail zwiedzającego.

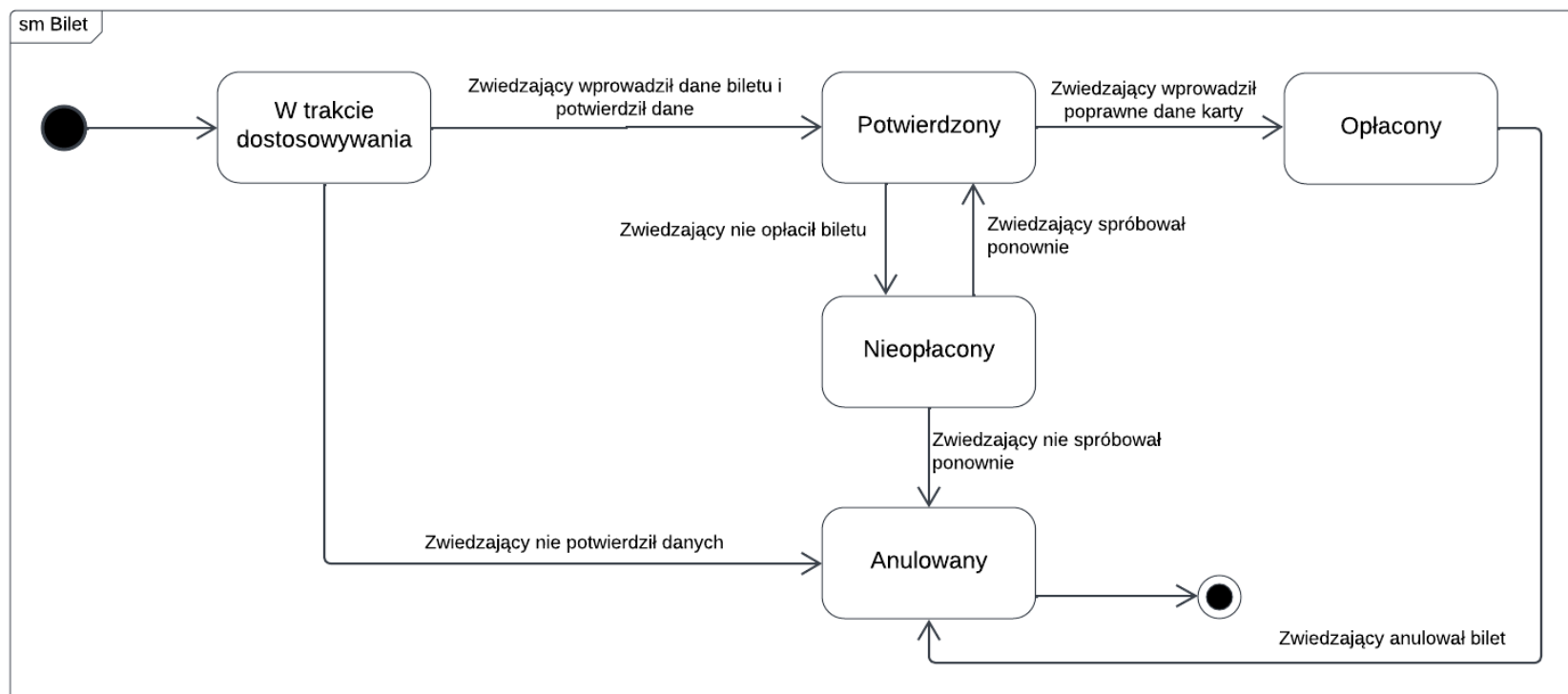
Alternatywny przepływ zdarzeń:

- 7a. Zwiedzający nie potwierdza – kończymy przypadek
- 8a. Zwiedzający dodatkowo włącza p. u. „Zapisanie się do grupy”
 - 8a1. System przechodzi do p. u.
 - 8a2. Zwiedzający kończy p. u.
 - 8a3. System wraca do 8
- 10a. Zwiedzający nie akceptuje płatności – kończymy przypadek.
- 11a. System wyświetla informację o niepoprawnym dokonaniu zakupu i prosi o spróbowanie jeszcze raz.
 - 11a1. System wraca do 13
 - 11a2. Zwiedzający nie próbuje jeszcze raz – kończymy przypadek

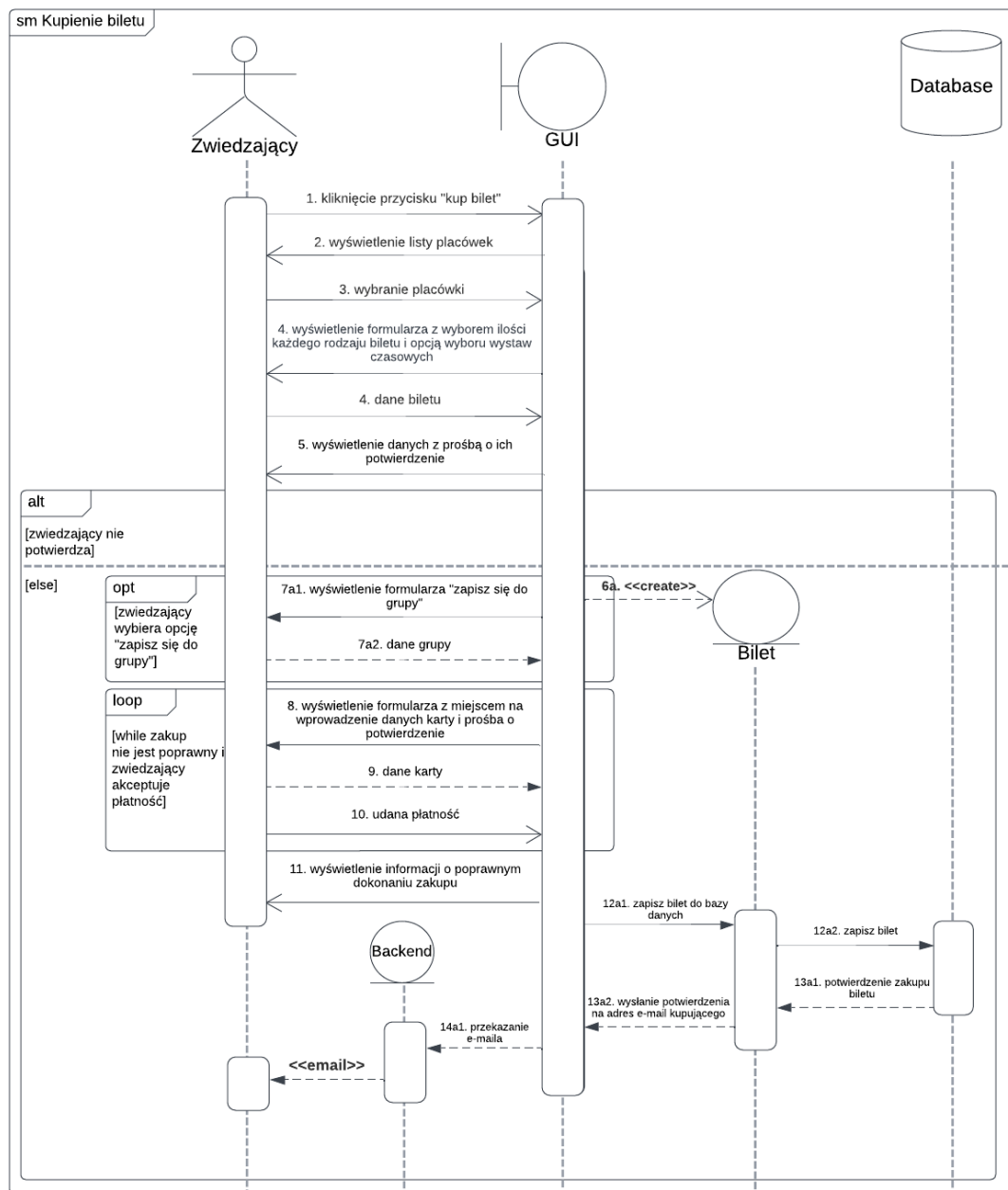
6. Diagram aktywności do przypadku użycia „Kupienie biletu”



7. Diagram stanu dla klasy „Bilet”

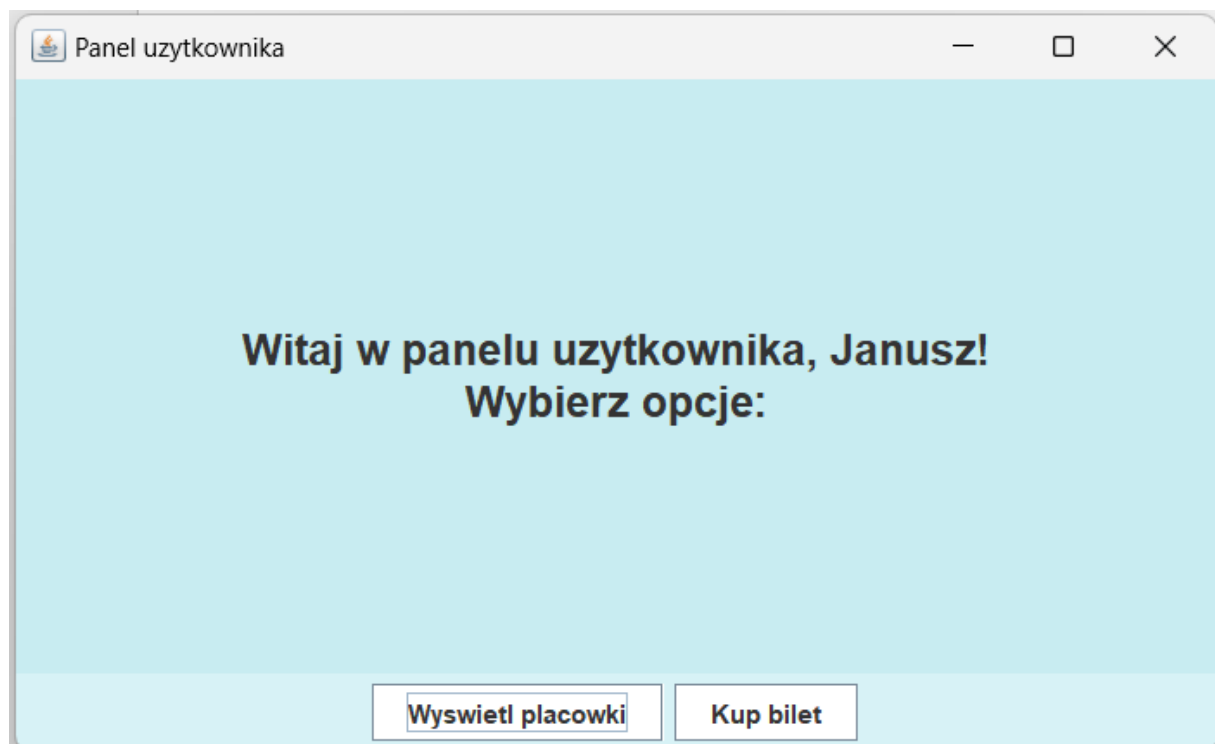


8. Diagram interakcji (sekwencji) dla przypadku użycia „Kupienie biletu”

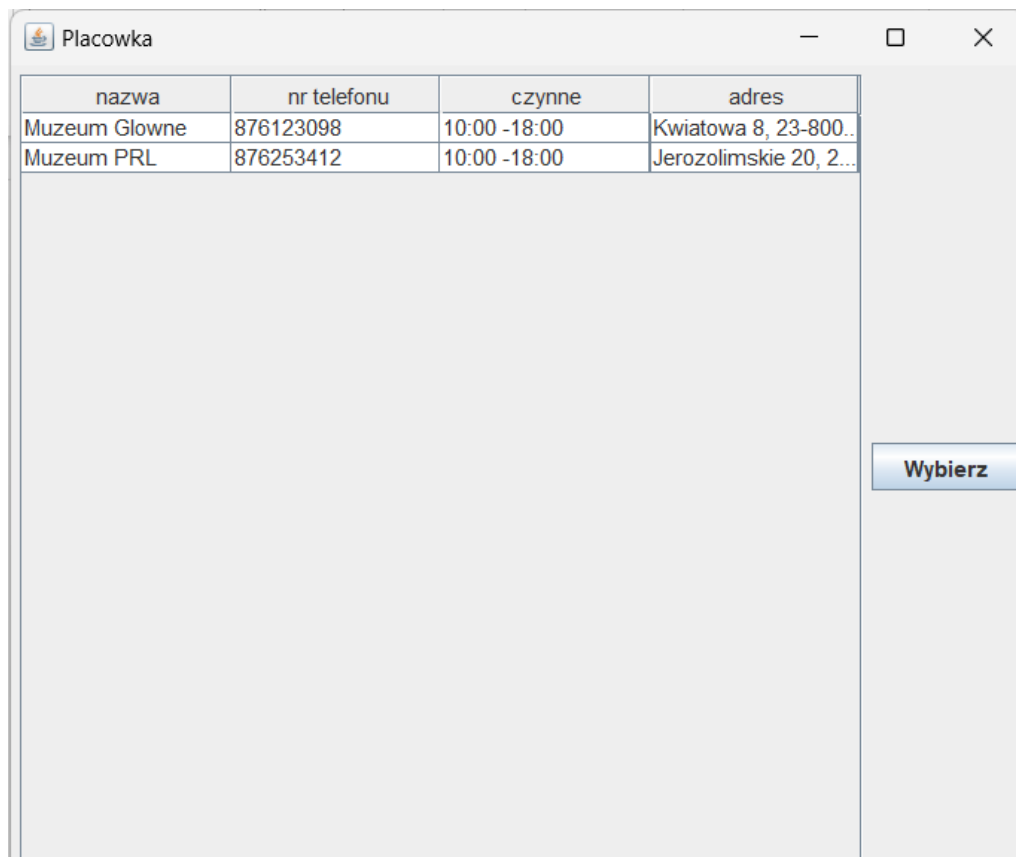


9. Projekt GUI

Panel użytkownika



Wybór placówki



Opcje biletu

Opcje biletu

Wybierz opcje biletu:

ulgowy:

normalny:

wystawa czasowa:

Nazwa	Wybierz
Sztuka uzytkowa	<input checked="" type="checkbox"/>
Sztuka XIX wieku	<input type="checkbox"/>

Anulowanie

Opcje biletu

Wybierz opcje biletu:


ulgowy:

normalny:

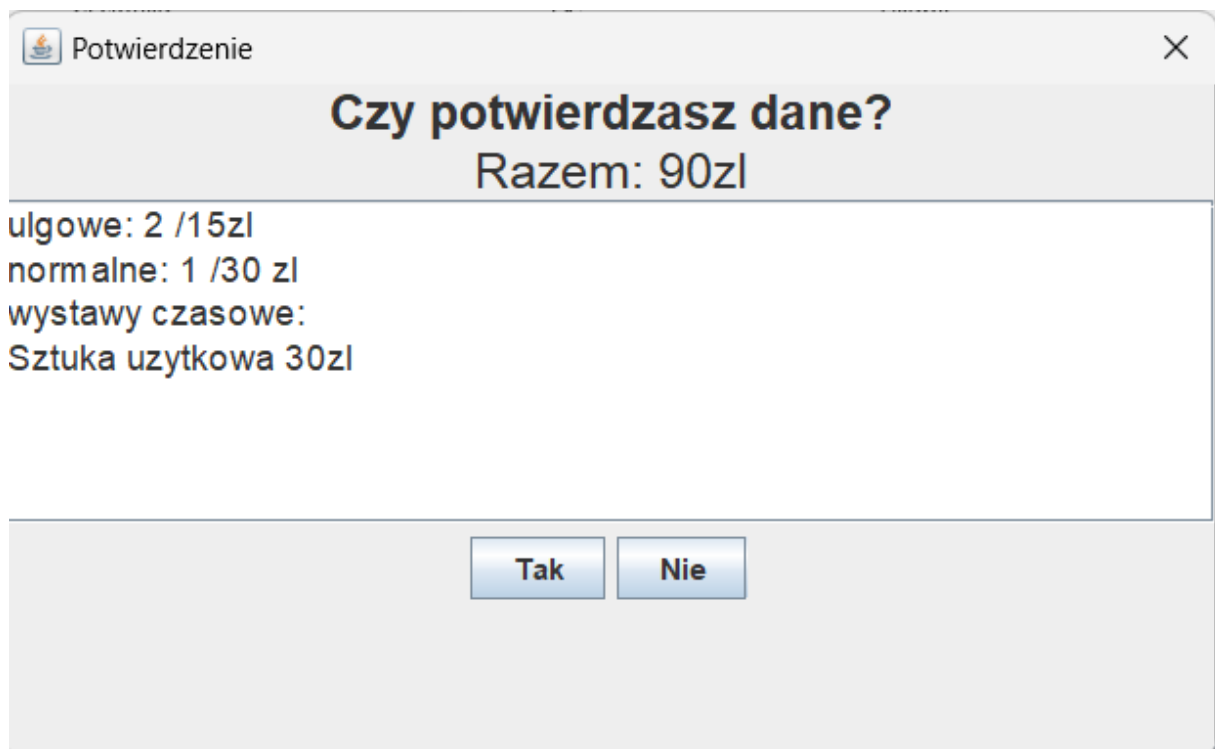
wysta

Nazwa	Wybierz
Sztuka uzytkowa	<input type="checkbox"/>
Sztuka XIX wieku	<input type="checkbox"/>

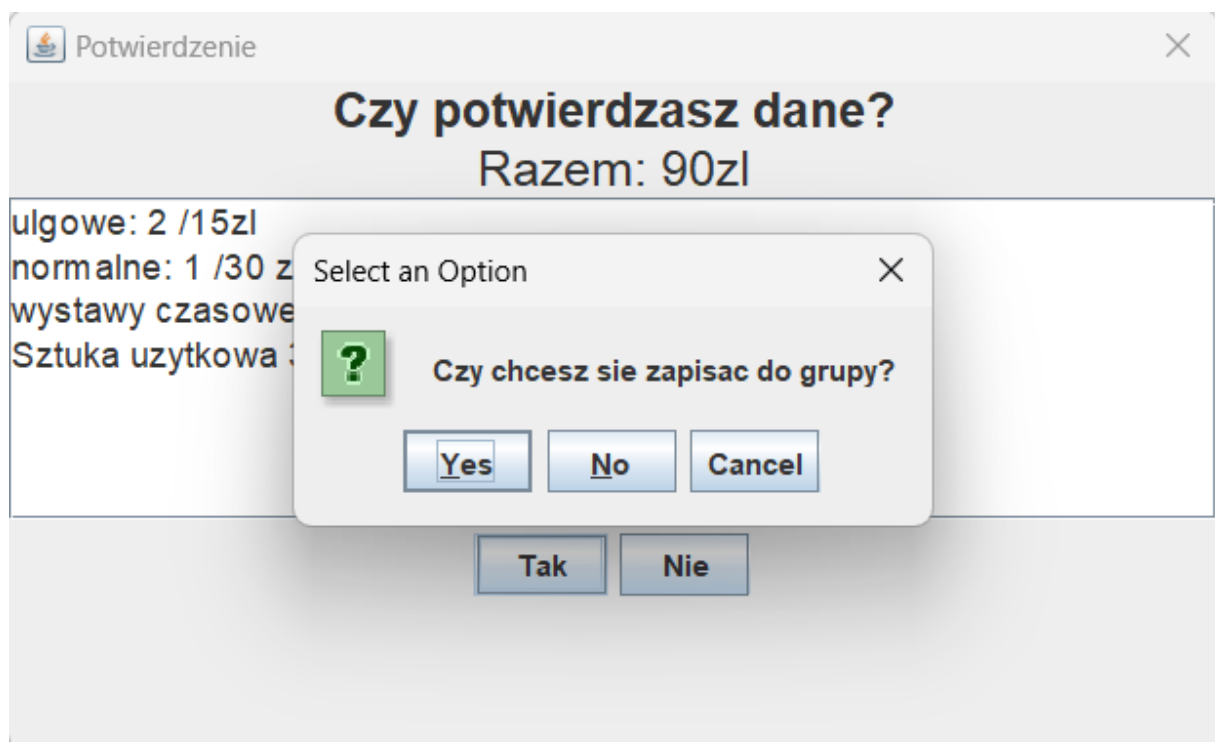
Select an Option

 **Na pewno chcesz anulowac?**


Potwierdzenie danych



Opcja z zapisaniem się do grupy



Wybór grupy

 Zapisz się do grupy

data	miejsca
1.07.2023 o 10:30	20
1.07.2023 o 10:00	18
2.07.2023 o 11:30	20


Wybierz

Poprawne zapisanie się do grupy

Zapisz się do grupy

data	miejsca
1.07.2023 o 10:30	20
1.07.2023 o 10:00	18
2.07.2023 o 11:30	20

Informacja

 Udalo się zapisac do grupy

OK


Wybierz

Brak miejsc w grupie

Zapisz sie do grupy

data	miejsca
1.07.2023 o 10:30	20
1.07.2023 o 10:00	18
2.07.2023 o 11:30	20

Informacja

 Nie ma miejsca w tej grupie

OK

Wybierz

Wprowadzenie danych karty

Platnosc

Podaj dane platnosci:

Do zaplacenia: 60 zl

Podaj nr karty:

Podaj cvv karty:

Podaj termin karty:

Potwierdz

Anuluj

Poprawne dane karty

Platnosc

Podaj dane platnosci:

Podaj nr k

Podaj cvv

Podaj termin karty: 07/23

Potwierdz Anuluj

Informacja

Platnosc udana

OK

Niepoprawne dane karty

Platnosc

Podaj dane platnosci:

Podaj nr k

Podaj cvv

Podaj termin karty: 08/23

Potwierdz Anuluj

Informacja

Platnosc niepoprawna

OK

10. 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 overlapping – gość może należeć do klas Zwiedzający, Uczestnik albo do obu naraz. Dlatego zastosowałam typ wyliczeniowy Enum pod postacią klasy umieszczonej w obiekcie ZwiedzającyUczestnik, dziedziczącej po Osobie. Konstrukcje warunkowe będą kontrolowały dostęp użytkownika do metod.
- Asocjacja kwalifikowana – w klasie ObjectPlusPlus znajduje się przeciążona metoda addLink(), która pobiera nazwy połączeń między obiektami, same obiekty oraz zmienną Object qualifier, pod którą może kryć się każdy typ będący kwalifikatorem. Można się do niego odwołać podczas przeszukiwania asocjacji.
- Asocjacja z atrybutem – w klasach Placówka i Warsztaty znajduje się lista wskazująca na obiekt PlacowkaWarsztaty, w którym asocjacje do obu klas dodawane są do Map<String, Map<Object, ObjectPlusPlus>>.
- Kompozycja – w klasie Ekspонат mieści się Set<ObjectPlusPlus>, a w Wystawie Map<String, Map<Ekspонат, ObjectPlusPlus>>, oba obiekty z klasy ObjectPlusPlus.
- Unique – id prowadzącego będzie dodawane do Set<Integer>, aby się nie powtórzyło.

Wszystkie klasy ukazane na diagramie dziedziczą po obiekcie ObjectPlus4, dziedziczącym po ObjectPlusPlus, dziedziczącym po ObjectPlus. W ten sposób można zaimplementować zwykłe asocjacje przechowywane w mapie oraz subset, dzięki metodzie addLink_subset(). Metoda ta sprawdza, czy istnieje wymagana asocjacja dla obiektu i dodaje nową.