

# EVENT MANAGER

DOKUMENTACJA

---

# Spis treści

<b>1.</b>	<b>Wymagania użytkownika .....</b>	<b>2</b>
<b>2.</b>	<b>Diagram przypadków użycia.....</b>	<b>4</b>
<b>3.</b>	<b>Diagram klas.....</b>	<b>5</b>
3.1	<i>Diagram klas analityczny .....</i>	<i>5</i>
3.2	<i>Diagram klas projektowy .....</i>	<i>6</i>
<b>4.</b>	<b>Scenariusz przypadku użycia Edycja Wydarzenia - rezerwacja sali.....</b>	<b>7</b>
<b>5.</b>	<b>Diagram aktywności dla przypadku użycia Edycja Wydarzenia - rezerwacja sali.....</b>	<b>8</b>
<b>6.</b>	<b>Diagram stanu dla klasy Sala .....</b>	<b>9</b>
<b>7.</b>	<b>Projekt GUI .....</b>	<b>9</b>
7.1	<i>Okno wyboru wydarzenia.....</i>	<i>9</i>
7.2	<i>Okno szczegółów wydarzenia .....</i>	<i>10</i>
7.3	<i>Okno wyboru Sali z listy dostępnych sal .....</i>	<i>11</i>
7.4	<i>Okno ostrzeżenia .....</i>	<i>11</i>
7.5	<i>Okno potwierdzenia .....</i>	<i>12</i>
<b>8.</b>	<b>Omówienie decyzji projektowych i skutków analizy dynamicznej.....</b>	<b>12</b>

## 1. Wymagania użytkownika

System ma na celu zarządzanie wydarzeniami organizowanymi w obiekcie, ich harmonogramami, a także kwestiami związanymi ze sprzedażą biletów oraz sponsorami wspierającymi wydarzenia.

System powinien przechowywać dane o pracownikach obiektu, wystawcach i uczestnikach. Każda z tych osób powinna posiadać dane osobowe – numer telefonu, adres email (opcjonalnie), adres (opcjonalnie) imię oraz nazwisko. Osoba może być równocześnie zatrudniona w obiekcie jak i być wystawcą, czy uczestnikiem wydarzenia.

Dodatkowo należy przechowywać informację, czy uczestnik wyraził lub nie wyraził zgody na informacje marketingowe oraz listę zakupionych biletów. Zakupy biletów realizowane są przez natychmiastowe przelewy bankowe systemu banku.

Pracownicy są delegowani do obsługi wydarzeń. Każdy pracownik posiada listę pełnionych funkcji (mogą to być ochrona, kelner, technik, serwis sprząający, recepcjonista, kierownik, asystent) oraz status (wolny lub zajęty). Pracownicy przydzieleni do wydarzeń są rozpoznawani na podstawie unikalnych identyfikatorów.

Wystawcy biorący udział w wydarzeniach mają możliwość wykupienia pakietu VIP i biletów lunchowych, których wartość jest doliczana do ich indywidualnego kosztu uczestnictwa. Podstawą kosztu uczestnictwa jest cena bazowa, która obecnie wynosi 300 zł i jest wspólna dla wszystkich wystawców. Również cena zestawu VIP (150 zł) oraz koszt biletu lunchowego (25 zł) jest wspólny dla wszystkich wystawców. Wszystkie te kwoty mogą ulec w przyszłości zmianie.

Dane osobowe każdej osoby zarejestrowanej w systemie mogą zostać wyświetlone, w zależności od osoby pokazywane są też dodatkowe dane – u pracownika jego id i lista pełnionych funkcji, u uczestnika zgoda marketingowa a u wystawcy indywidualny koszt uczestnictwa

Wydarzenia muszą posiadać unikalną nazwę, maksymalną liczbę uczestników, status ("zaplanowane", "odwołane", "zakończone", "trwające"), datę rozpoczęcia i zakończenia, ilość dni jaką trwają oraz opcjonalny opis (do 500 znaków).

Każdemu wydarzeniu mogą zostać przypisane sale. Każda z nich posiada swój unikalny numer, opcjonalną nazwę, maksymalną liczbę osób jaką może pomieścić, informację czy jest zainstalowany projektor oraz listę wydarzeń wraz z terminami, w jakich została na nie zarezerwowana.

Wydarzenie może mieć przypisany harmonogram na konkretny dzień, ale ilość przypisanych harmonogramów nie może przekroczyć liczby dni trwania wydarzenia. Na wydarzenie składa się co najmniej jeden panel, w skład którego wchodzi nazwa, godzina rozpoczęcia i zakończenia oraz opis.

Każde wydarzenie może otrzymać wsparcie od sponsorów. W ramach wsparcia możliwe jest zaoferowanie kwoty lub materiałów promocyjnych. W przypadku sponsorów przechowywana jest nazwa, NIP (opcjonalnie) oraz status sponsora (brązowy, srebrny, złoty).

Ilość biletów dostępnych w ramach konkretnego wydarzenia nie może przekroczyć maksymalnej ilości uczestników wydarzenia. Standardowa cena biletu jest wspólna dla wszystkich biletów i wynosi 200 zł. Cena może ulec w przyszłości zmianie. Bilet może być ulgowy (50% normalnej ceny) lub darmowy. Dodatkowo bilety dzielą się na bilety standard i premium. Bilet premium umożliwia dostęp do bufetu, strefy premium oraz zniżkę VIP (-30%) w sklepie obiektu.

System musi umożliwiać wykonywanie poniższych czynności:

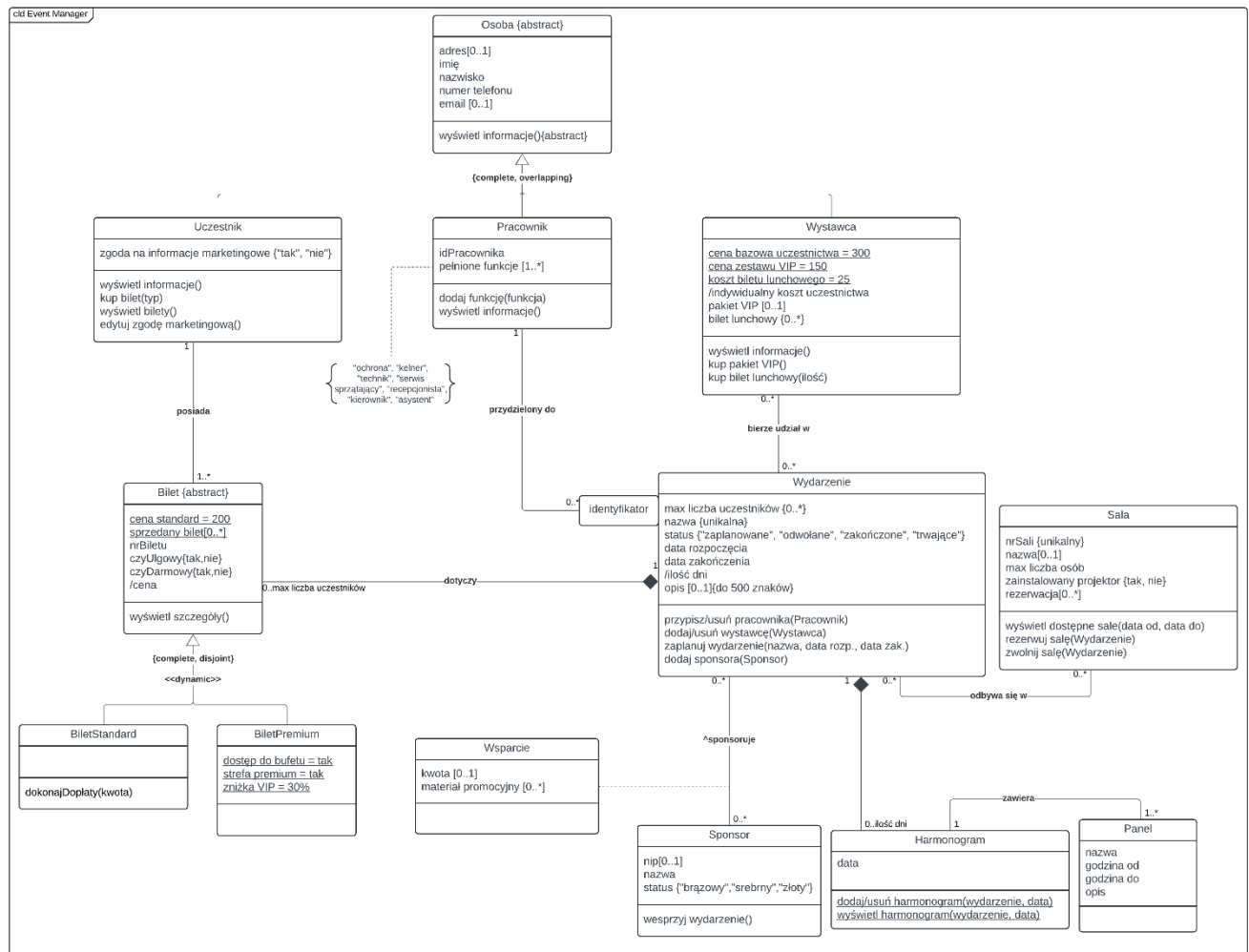
- wyświetlenie danych o osobie: Kierownik, asystent

- edycja zgody marketingowej: uczestnik
- zakup biletu: uczestnik
- dopłata do biletu premium: uczestnik
- wyświetlenie posiadanych biletów: uczestnik
- przypisanie/usunięcie pracownika do wydarzenia: Kierownik i asystent
- zaplanowanie wydarzenia: kierownik
- dodanie lub usunięcie sponsora: kierownik
- dodanie/usunięcie harmonogramu, wystawcy do wydarzenia: kierownik, asystent
- edycja informacji o pracowniku oraz dodanie/usunięcie funkcji: kierownik, asystent
- edycja szczegółów wydarzenia: asystent, kierownik
- rezerwacja sali dla wydarzenia po wcześniejszym wyświetleniu dostępnych: kierownik, asystent
- dokupienie pakietu VIP oraz biletów lunchowych: wystawca

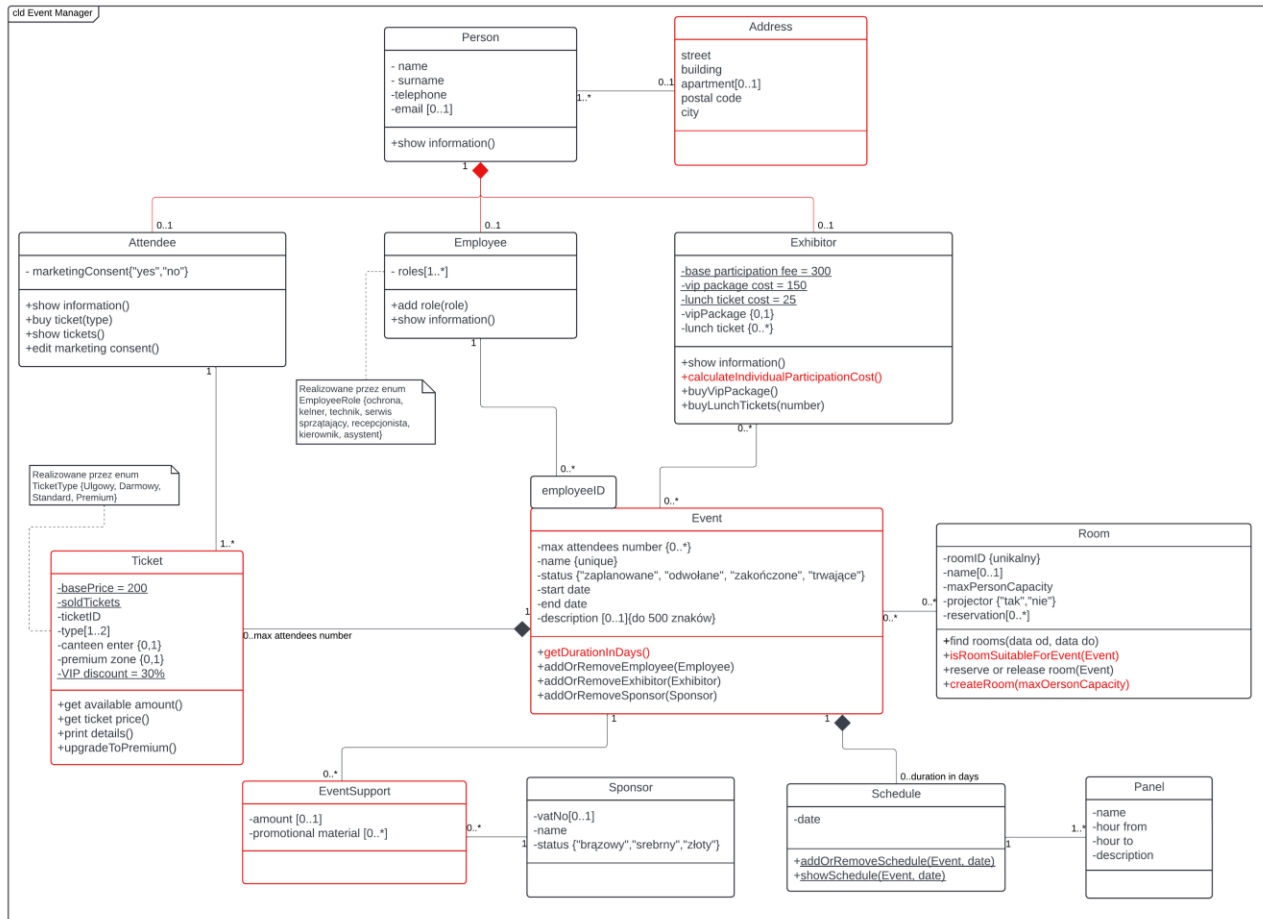


### 3. Diagram klas

#### 3.1 Diagram klas analityczny



### 3.2 Diagram klas projektowy

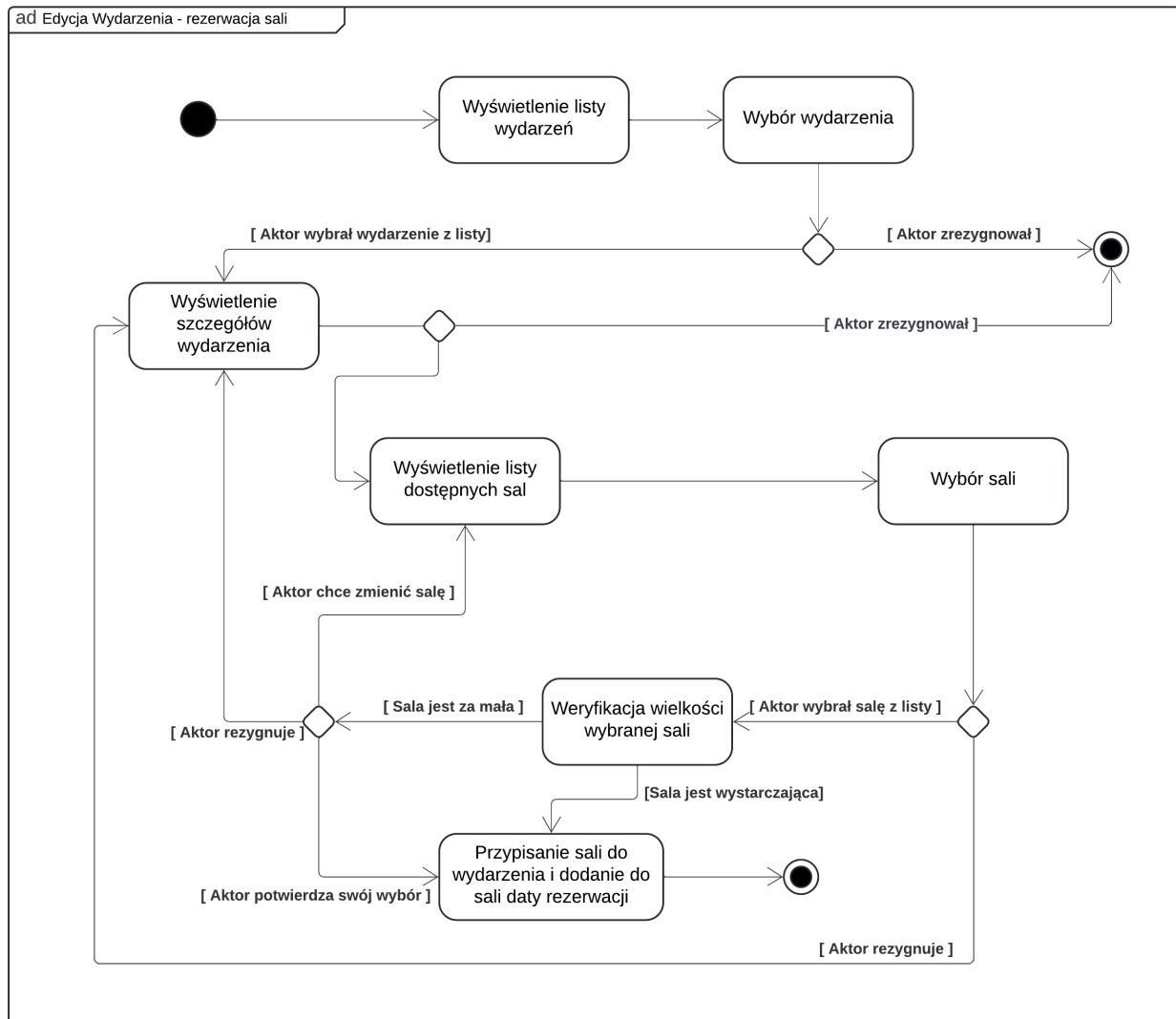


#### 4. Scenariusz przypadku użycia Edycja Wydarzenia - rezerwacja sali

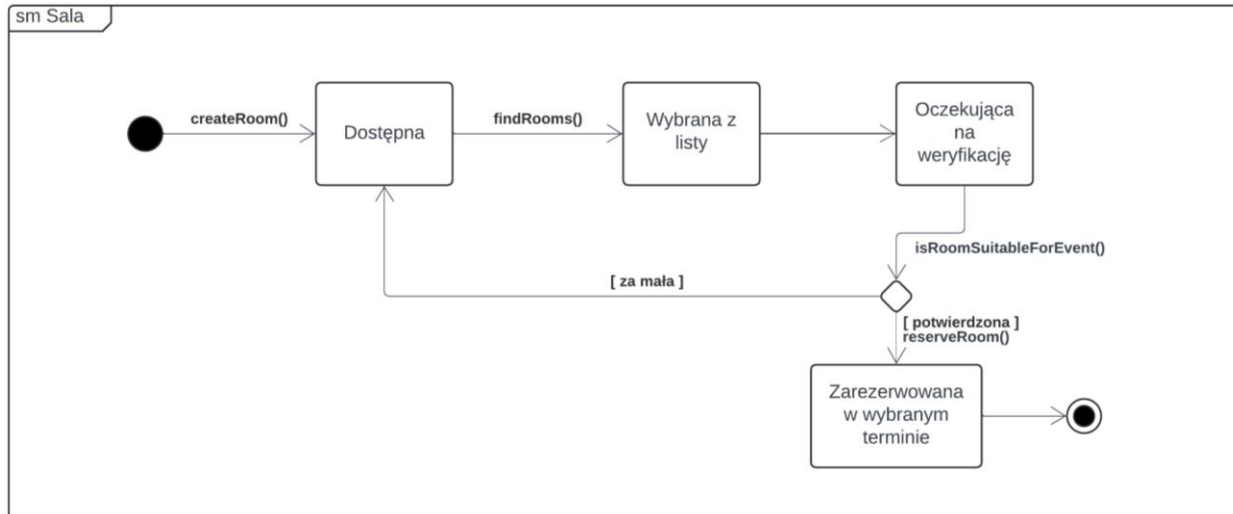
Sekcja	Treść
<b>Aktorzy</b>	Asystent
<b>Warunek początkowy</b>	Aktor jest zalogowany w systemie
<b>Scenariusz główny</b>	<ol style="list-style-type: none"> <li>1. Aktor uruchamia przypadek użycia "Edycja wydarzenia".</li> <li>2. System wyświetla listę wydarzeń i prosi o decyzję</li> <li>3. Aktor wybiera wydarzenie z listy i zatwierdza.</li> <li>4. System wyświetla szczegóły wydarzenia.</li> <li>5. Aktor wybiera opcję "Rezerwacja sali".</li> <li>6. System wyświetla listę dostępnych sal i prosi o podjęcie decyzji.</li> <li>7. Aktor wybiera salę z listy i zatwierdza.</li> <li>8. System weryfikuje poprawność danych.</li> <li>9. System przypisuje salę do wydarzenia oraz dodaje datę rezerwacji sali w bazie.</li> </ol>
<b>Scenariusz alternatywny</b>	<p>3a. Aktor rezygnuje. Koniec przypadku użycia</p> <p>6a. Aktor rezygnuje. Koniec przypadku użycia</p> <p>10a System zgłasza, że sala jest mniejsza niż maksymalna ilość osób przypisana do wydarzenia i prosi o podjęcie decyzji.</p> <p>10aa Aktor rezygnuje. Koniec przypadku użycia</p> <p>10ab Aktor chce poprawić dane. Przejście do pkt 7.</p> <p>10ac Aktor potwierdza wybraną salę. Przejście do pkt 10.</p>
<b>Warunek końcowy</b>	System poprawnie przypisał salę do wydarzenia i dodał datę rezerwacji sali w bazie.



## 5. Diagram aktywności dla przypadku użycia Edycja Wydarzenia - rezerwacja sali



## 6. Diagram stanu dla klasy Sala



## 7. Projekt GUI

### 7.1 Okno wyboru wydarzenia

#### Lista Wydarzeń

IdWydarzenia	Nazwa	Data Od	Data Do	Status	Akcja
1	Wydarzenie A	2024-06-15	2024-06-16	Zaplanowane	Wybierz
2	Wydarzenie B	2024-07-10	2024-07-12	Trwające	Wybierz
3	Wydarzenie C	2023-08-20	2023-08-22	Zakończone	Wybierz
4	Wydarzenie D	2024-09-01	2024-09-01	Zaplanowane	Wybierz

Rezygnuj

## 7.2 Okno szczegółów wydarzenia

### Szczegóły Wydarzenia

**Nazwa wydarzenia:** Wydarzenie 1

**Status:** Zaplanowane

**Max liczba uczestników:** 100

**Data rozpoczęcia:** 2024-06-10

**Data zakończenia:** 2024-06-12

**Przypisane sale:** BRAK

**Opis:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

**Lista sponsorów:** [Sponsor A](#), [Sponsor B](#), [Sponsor C](#)

**Harmonogram:** [Zobacz szczegóły](#)

**Bilety:** [Zobacz szczegóły](#)

Rezerwacja sali

Usunięcie sali

Dodaj wystawcę

Usuń wystawcę

Dodaj harmonogram

Usuń harmonogram

Rezygnuj

### 7.3 Okno wyboru Sali z listy dostępnych sal

## Lista Dostępnych Sal

Numer sali	Nazwa sali	Max liczba osób	Projektor	Akcja
101	Sala Konferencyjna A	50	Tak	Wybierz
102	Sala Konferencyjna B	30	Nie	Wybierz
103	Sala Warsztatowa	20	Tak	Wybierz
104	Sala Konferencyjna C	25	Nie	Wybierz

Rezygnuj

### 7.4 Okno ostrzeżenia

## Ostrzeżenie

Uwaga! Sala jest mniejsza niż maksymalna ilość osób przypisana do wydarzenia.

Podejmij działanie:

Rezygnuj

Popraw dane

Potwierdź salę

## 7.5 Okno potwierdzenia

### Potwierdzenie Rezerwacji Sali

**Szczegóły Rezerwacji****Numer sali:** 101**Nazwa sali:** Sala Konferencyjna A**Data od:** 2024-06-15**Data do:** 2024-06-16**Max liczba osób:** 50

Rezerwacja została zapisana.

Potwierdź

## 8. Omówienie decyzji projektowych i skutków analizy dynamicznej

Po analizie przedstawionych wymagań użytkownika, zdecydowano się na stworzenie następujących klas wraz z atrybutami:

**Person:**

- Atrybuty:
  - String phoneNumber
  - String email (opcjonalnie)
  - String firstName
  - String lastName
  - Address address (opcjonalnie, reprezentowany przez osobną klasę)
- Relacje: Klasa bazowa dla wszystkich osób w systemie.

**Address:**

- Atrybuty:
  - String street
  - String city
  - String postalCode
  - String country
- Relacje: Połączona relacją \* to \* z klasą Person.

**Attendee:**

- Kompozycja z klasą Person.
- Atrybuty:
  - boolean marketingConsent

- List<Ticket> purchasedTickets
- Metody:
  - void editMarketingConsent(boolean consent)
  - void purchaseTicket(Ticket ticket)
  - void upgradeToPremiumTicket(Ticket ticket)
  - List<Ticket> displayPurchasedTickets()

**Employee:**

- Kompozycja z klasą Person.
- Atrybuty:
  - String employeeId
  - List<Function> functions (enum Function: OCHRONA, KELNER, TECHNIK, SERWIS\_SPRZĄTAJĄCY, RECEPCJONISTA, KIEROWNIK, ASYSTENT)
  - String status (wolny lub zajęty)
  - String identifier
- Metody:
  - void assignToEvent(Event event)
  - void removeFromEvent(Event event)
  - void editEmployeeDetails(String phoneNumber, String email, Address address)
  - void addFunction(Function function)
  - void removeFunction(Function function)

**Exhibitor:**

- Kompozycja z klasą Person.
- Atrybuty:
  - double base participation fee = 300
  - double vip package cost = 150
  - double lunch ticket cost = 25
  - boolean vipPackage
  - int lunch ticket
- Metody:
  - void purchaseVipPackage()
  - void purchaseLunchTickets(int numberOfTickets)
  - void displayDetails()

**Ticket:**

- Atrybuty:
  - List <TicketType> (Ulgowy, Darmowy, Standard, Premium)
  - double basePrice = 200
  - Map<String, List<String>> soldTickets
  - String ticketID
  - List<TicketType> type[1..2]
  - boolean canteen enter
  - boolean premium zone
  - int VIP discount = 30%
- Relacje: Połączona z Attendee i Event

**Event:**

- Atrybuty:
  - String name
  - int maxAttendees
  - String status (planned, canceled, completed, ongoing)
  - Date startDate
  - Date endDate
  - int durationDays
  - String description (opcjonalnie)
- Relacje:
  - Kompozycja z Ticket.
  - Asocjacja z Exhibitor.
  - Kwalifikowana asocjacja z Employee (każdy pracownik może być przypisany do konkretnego wydarzenia poprzez identyfikator).
- Metody:
  - Int getDurationInDays()
  - Void addOrRemoveEmployee(Employee)
  - Void addOrRemoveExhibitor(Exhibitor)
  - Void addOrRemoveSponsor(Sponsor)

**Sponsor:**

- Atrybuty:
  - String name
  - String NIP (opcjonalnie)

- String status (bronze, silver, gold)
- Relacje:
  - Asocjacja 1 do \* z klasą pośredniczącą EventSupport.

**EventSupport:**

- Atrybuty:
  - double amount
  - List<String> promotionalMaterials
- Relacje:
  - Asocjacja \* to 1 z klasą Event.
  - Połączenie 1 do \* z klasą Sponsor.

**Schedule:**

- Atrybuty:
  - Date date
  - List<Panel> panels
- Relacje:
  - Kompozycja z Event.

**Panel:**

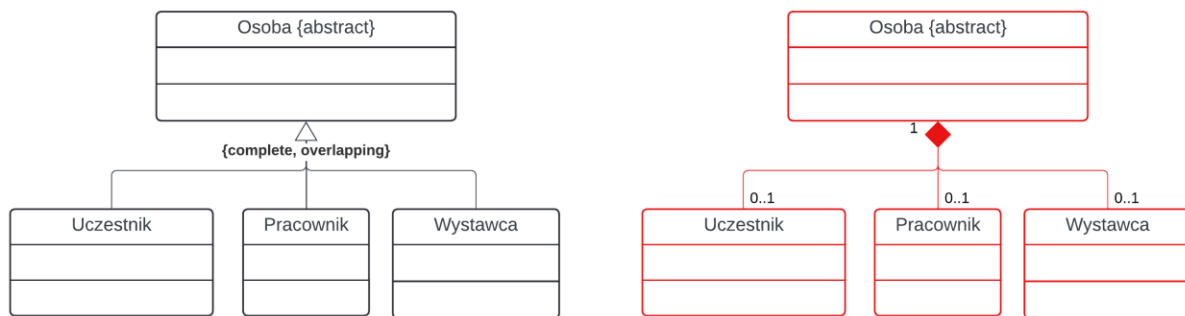
- Atrybuty:
  - String name
  - Time startTime
  - Time endTime
  - String description
- Relacje:
  - Asocjacja ze Schedule.

**Room:**

- Atrybuty:
  - String number
  - String name (opcjonalnie)
  - int maxCapacity
  - boolean hasProjector
  - List<Event> reservedEvents
- Relacje:
  - Asocjacja z Event.

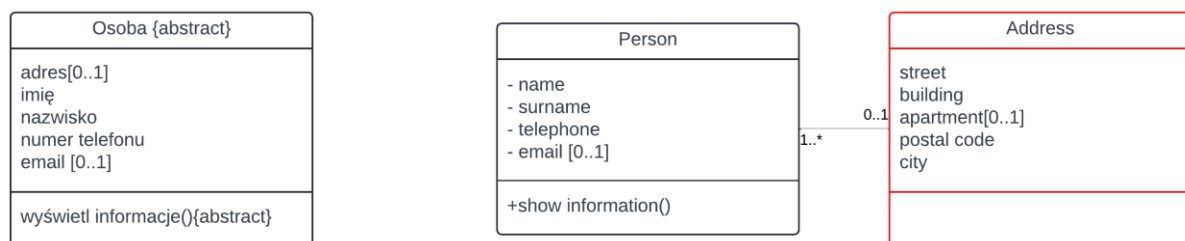


W systemie zarządzania wydarzeniami zdecydowano się na użycie różnych asocjacji i kompozycji, aby właściwie odwzorować relacje pomiędzy elementami systemu. Klasa *Person* stanowi bazę dla reprezentacji wszystkich osób w systemie, takich jak uczestnicy, pracownicy i wystawcy, ponieważ każda z tych grup posiada wspólne atrybuty, takie jak numer telefonu, adres email, imię i nazwisko oraz opcjonalny adres, który jest reprezentowany przez osobną klasę *Address* i połączony z *Person* relacją \*. Klasa *Attendee*, *Employee* i *Exhibitor* są połączone z klasą *Person* poprzez kompozycję, co jest rozwiązaniem wynikającym z braku mechanizmu overlapping w Java, umożliwiając reprezentację osoby jednocześnie jako pracownika, wystawcę i uczestnika.



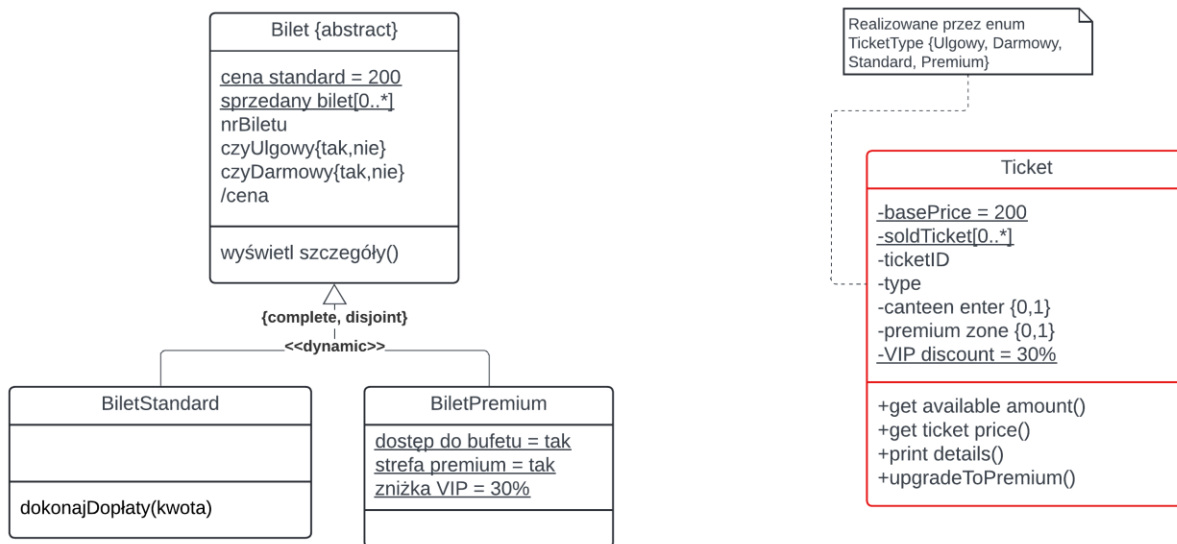
Grafika 1. Implementacja dziedziczenia overlapping

Atrybut adres z klasy *Osoba* został zaimplementowany jako osobna klasa, co pozwoli na lepsze zarządzanie danymi – edycję, dodawanie, usuwanie.



Grafika 2. Implementacja atrybutu złożonego Adres

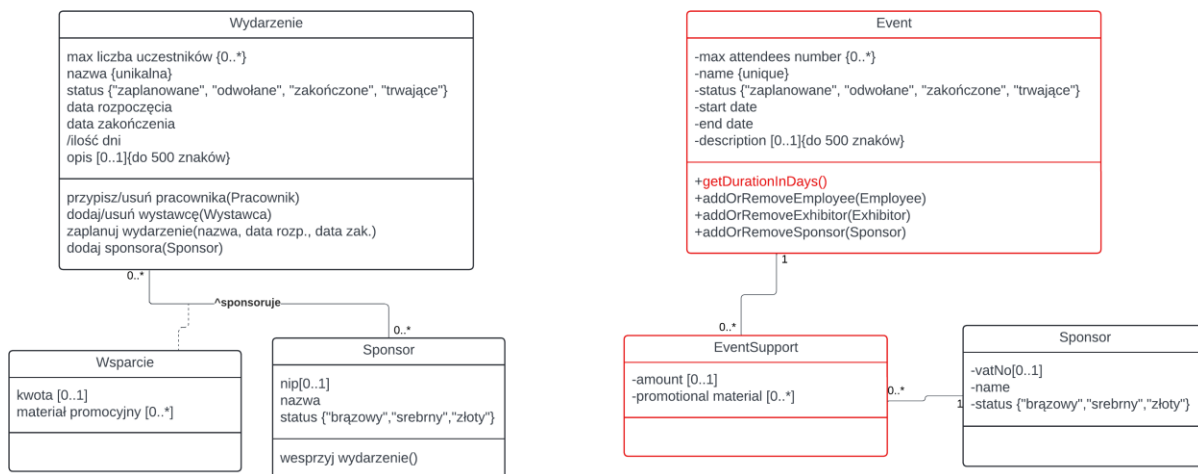
Klasa *Ticket* jest powiązana z *Attendee* poprzez asocjację, co umożliwia przypisanie biletów do uczestników. Z uwagi na brak mechanizmu dziedziczenia dynamicznego w Java, wszystkie rodzaje biletów są reprezentowane przez spłaszczoną klasę *Ticket*, a ich rodzaj jest rozróżniany poprzez enum *TicketType*. Dodatkowo atrybut cena został zaimplementowany za pomocą metody *get ticket price()* co usprawni działanie programu.



Grafika 3. Implementacja klasy Ticket i dziedziczenia dynamic

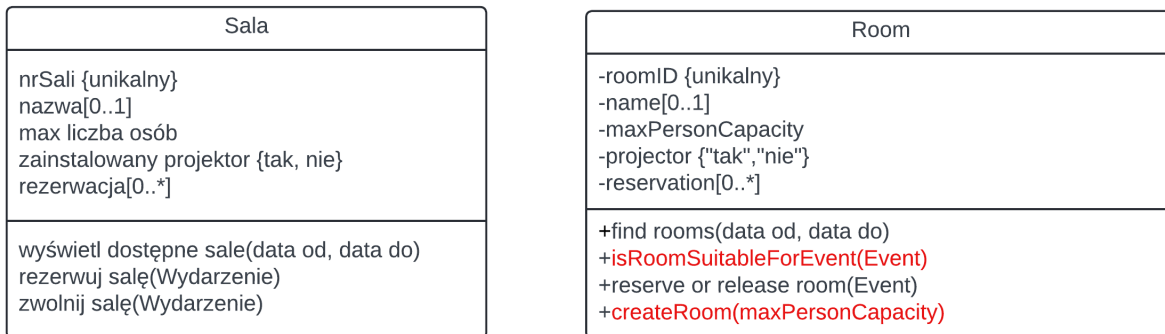
Klasa Event jest połączona kompozycją z Ticket, co oznacza, że bilety są integralną częścią wydarzeń oraz asocjacją z Exhibitor, pozwalającą na powiązanie wystawców z wydarzeniami. Ponadto, asocjacja kwalifikowana z Employee umożliwia przypisanie pracowników do konkretnych wydarzeń za pomocą unikalnych identyfikatorów.

Aby zarządzać wsparciem sponsorów dla wydarzeń, utworzono klasę Sponsor, która jest połączona relacją 1 do \* z klasą pośredniczącą EventSupport. Klasa EventSupport powstała z powodu braku w języku Java mechanizmu implementującego asocjację z atrybutem, co pozwala na przechowywanie informacji o wsparciu finansowym lub promocyjnym sponsorów dla wydarzeń. EventSupport jest połączona relacją \* do 1 z klasą Event, co oznacza, że jedno wydarzenie może mieć wiele różnych form wsparcia od różnych sponsorów. Dodatkowo atrybut ilość dni w klasie Wydarzenie został zaimplementowany poprzez metodę getDurationDays() co usprawni działanie programu.



Grafika 4. Implementacja asocjacji z atrybutem

Harmonogramy wydarzeń są reprezentowane przez klasę Schedule, która jest połączona kompozycją z Event, oznaczając, że harmonogram jest nierozdzielnie związany z wydarzeniem. Klasa Panel, będąca częścią harmonogramu, jest powiązana z Schedule poprzez zwykłą asocjację, co pozwala na dodawanie i zarządzanie różnymi panelami w ramach harmonogramu wydarzenia. Sale, reprezentowane przez klasę Room, są połączone zwykłą asocjacją z Event, co umożliwia rezerwację różnych pomieszczeń na potrzeby konkretnych wydarzeń. W wyniku analizy w klasie Room zaimplementowane zostały dodatkowe metody, których celem jest weryfikacja decyzji podejmowanych przez użytkownika oraz pomoc w administrowaniu obiektem.



Te relacje i decyzje projektowe umożliwiają kompleksowe zarządzanie wszystkimi aspektami organizacji wydarzeń, włączając w to rejestrację uczestników, przydzielanie pracowników, zarządzanie biletami, planowanie harmonogramów oraz koordynację wsparcia sponsorów, zapewniając elastyczność i możliwość rozszerzania funkcjonalności systemu w przyszłości.