System Lotniska

Dokumentacja projektowa aplikacji systemu lotniska

1. Wymagania użytkownika

Pod warszawą powstaje nowe lotnisko WIIZFREE, które planuje być najlepsze ze wszystkich. Zamówiło ono system wspomagający pracę lotniskowej bazy danych. System ma przechowywać dane o wszystkich lotach, pasażerach, pracownikach jak i pojazdach takich jak samoloty. Lotnisko posiada swoją nazwę, miejsce w którym się znajduję oraz powierzchnię jaką zajmuje teren lotniska. Należy pamiętać, że są to dane wymagane.

W systemie są przechowywane informacje na temat osób takie jak imię, naziwsko, PESEL (musi mieć 11 cyfr), mail, wiek (wyliczany na podstawie daty urodzenia, oraz nie może być ujemny). Osoby dzielimy na Pasażerów którzy mogą decydować na temat bagażu czy go biorą czy nie, oraz pracowników którzy odszymują określoną stawkę wyliczaną na podstawie liczby przepracowanych godzin oraz stanowiska jakie obejmują. Ponadto pracownik może być pasażerem jak i na odwrót. Klasy są kompletne.

Chcąc zarezerwać bilet należy wybrać lot na który chcemy polecieć, stąd wybieramy numer lotu (jest unikatowy dla danej podróży), który nas intersuję, lotnisko startowe, oraz lotnisko docelowe na którym będziemy lądować. W rezerwacji powinniśmy zawrzeć informacje na temat daty rezerwacji (której dane organizowane są w kolejności od najwcześniejszej do najpóźniejszej), informacje czego dotyczy oraz statusu (potwierdzona, oczekująca na płatność, zrezygnowano).

Na lotnisku znajdują się również pojazdy takie jak Samolot, pojazd zasilany elektrycznie czy pojazd zasilany paliwem. W pojazdach muszą być zawarte informacje takie jak nazwa, rok produkcji. Dane te są wymagane. W klasie samolot notujemy informację o typie samolotu, maksymalnym zasięgu i cenie paliwa lotniczego. Pojazdy zasilane paliwem mają informację na temat pojemności baku, typu paliwa jakim jeżdżą oraz jej ceny, natomiast pojazdy zasilane elektrycznie zawierają informacje na temat pojemności baterii oraz ceny prądu. Na podstawie danych zawrtych w pojazdach obliczany jest koszt operacyjny danego pojazdu.

Na lotnisku znajdują się terminale, które mają swoją nazwę oraz informacje kontaktowe. Lotnisko musi posiadać minmum 2 terminale by móc poprawnie funkcjonować.

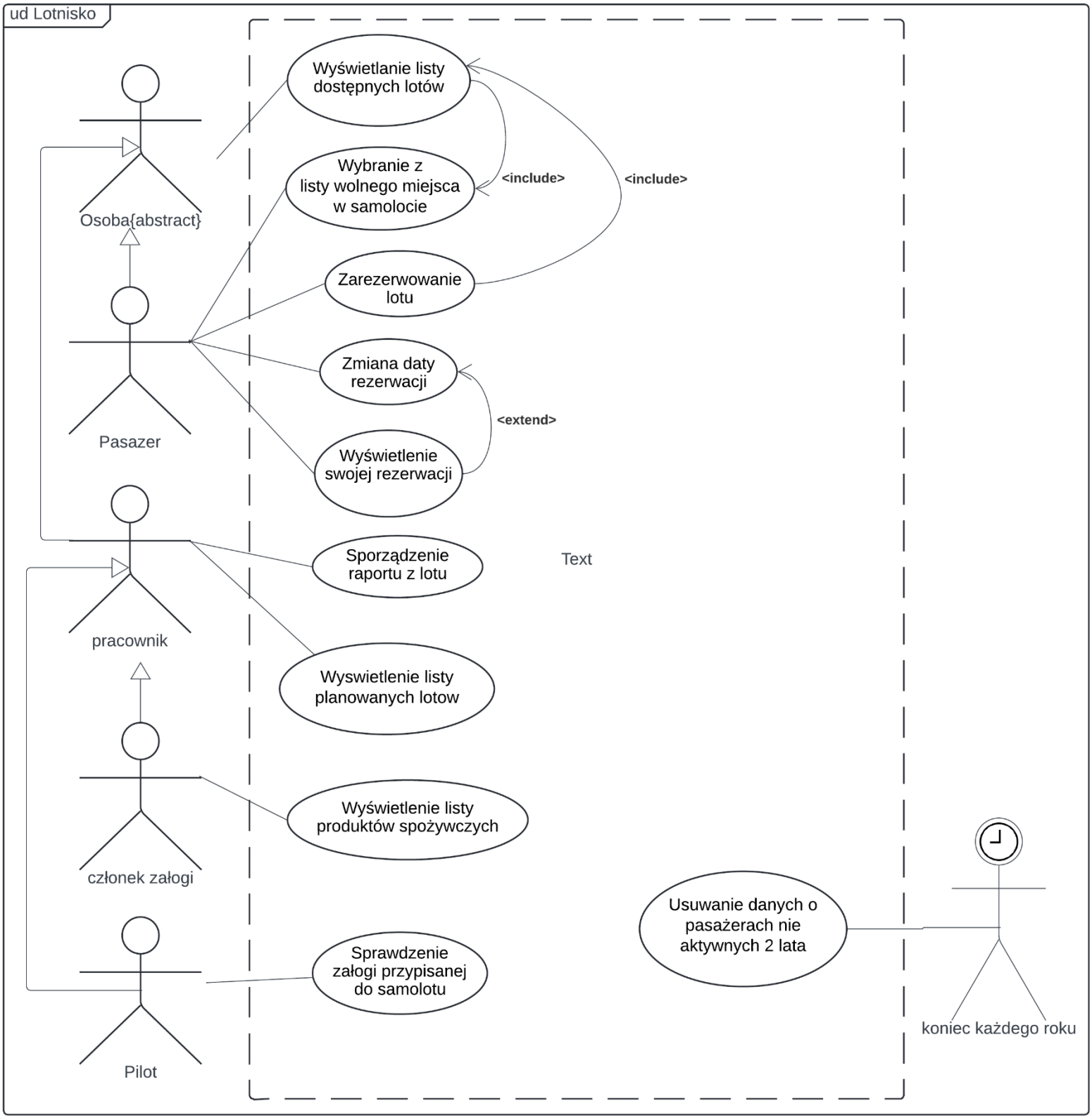
Pracowników dzielimy na członka załogi, który ma określone stanowisko oraz pilota, który do pracy potrzebuję mieć listę swoich certyfikatów. Ponadto członek załogi może awansować i stać się pilotem lub pilot zostać ustawiony na pozycji członka załogi.

System powinien umożliwiać:

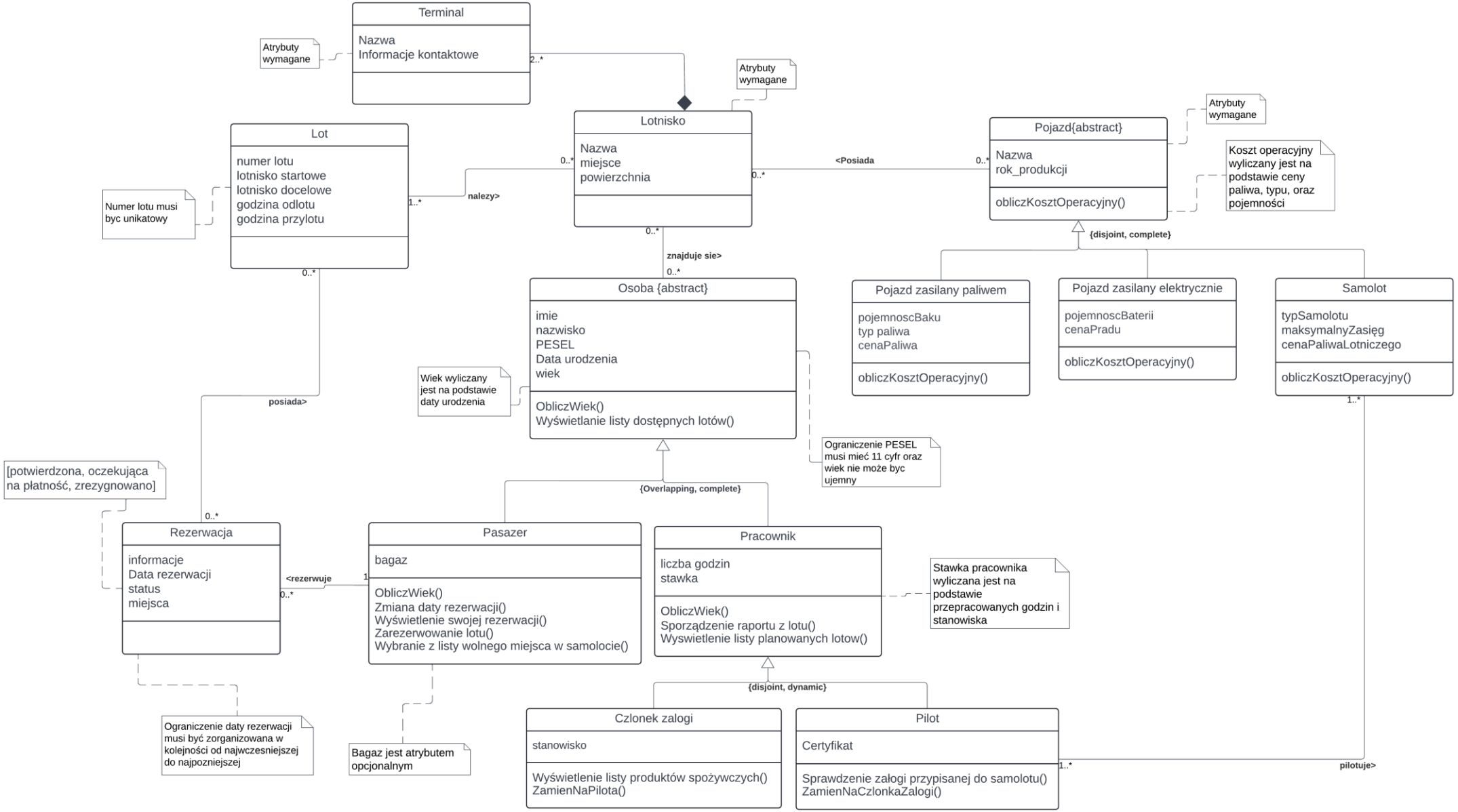
- Wyświetlanie listy dostępnych lotów
- Zarezerwowanie lotu
- Zmiana daty rezerwacji
- Wyświetlenie swojej rezerwacji, dla pasażera
- Sporządzenie raportu z lotu
- Wyświetlanie listy planowanych lotów
- Sprawdzenie załogi przypisanej do samolotu
- Wybranie z listy wolnego miejsca w samolocie
- Wyświetlenie listyproduktów spożywczych, sprzedawanych podczas lotu

Ponadto co 5 lat system będzie automatycznie usuwał dane o pasażerach nie aktywnych conajmniej 2 lata by zapobiec nadmiernemu składowaniu danych.

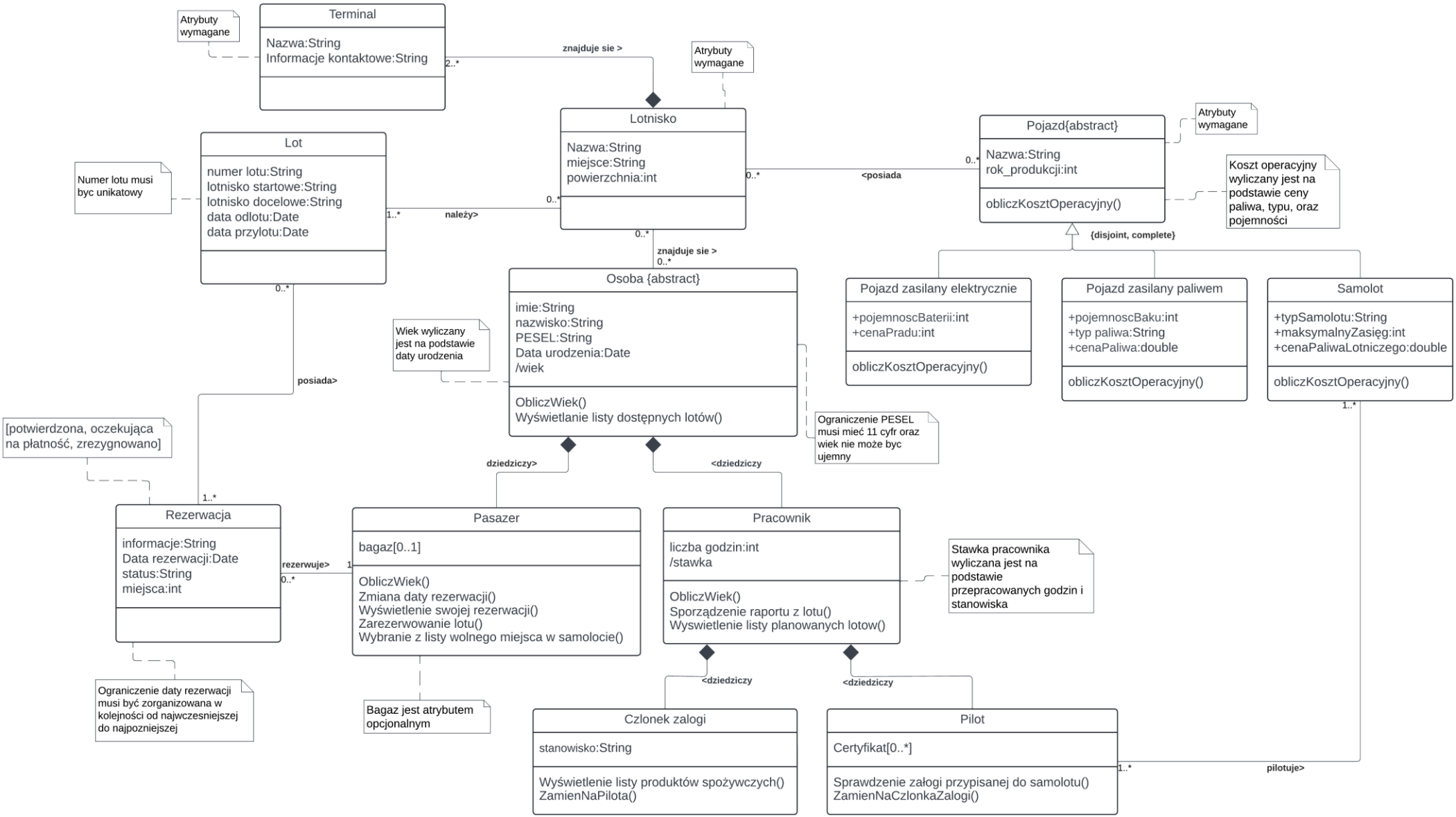
2. Diagram przypadków użycia



3. Diagram klas – analityczny



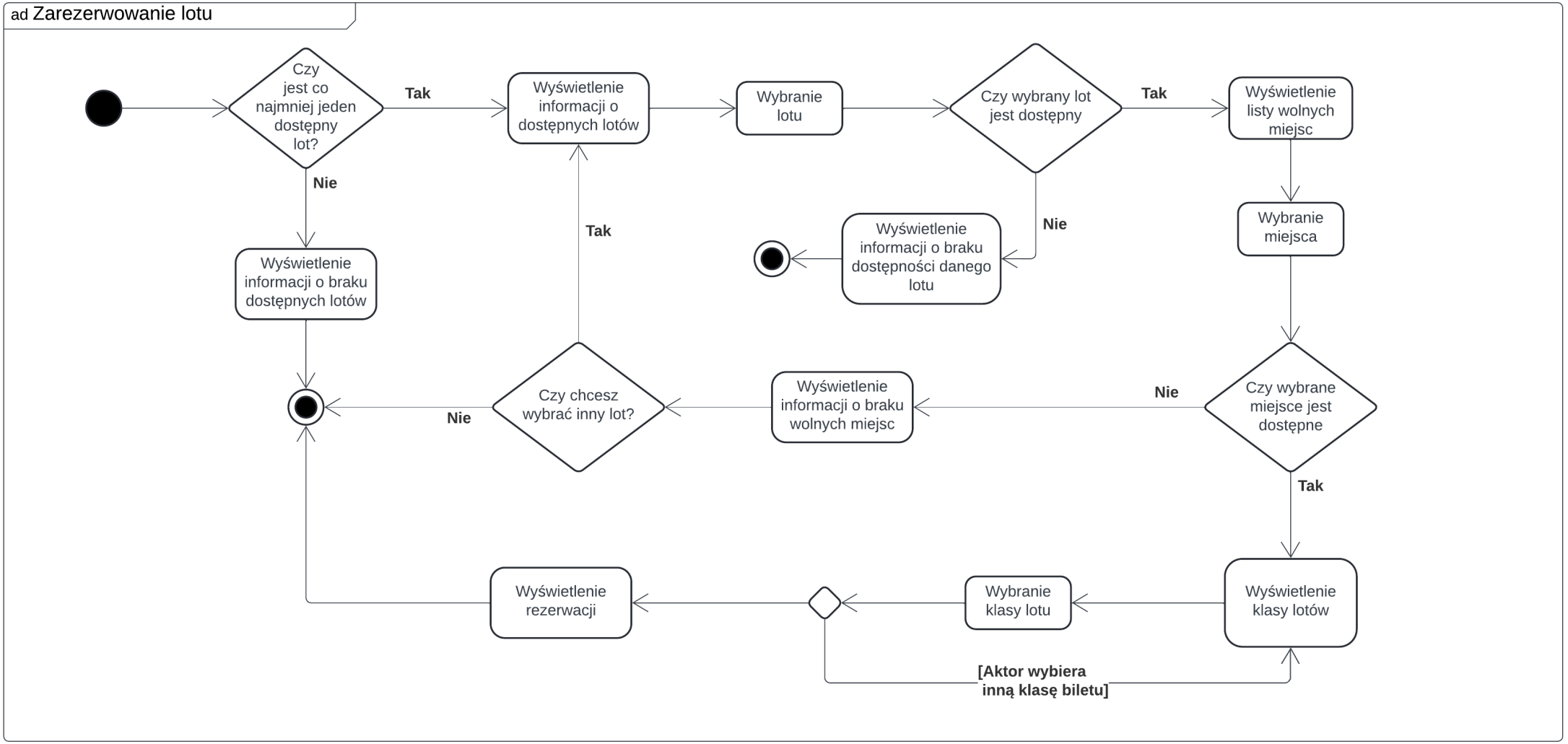
4. Diagram klas – projektowy



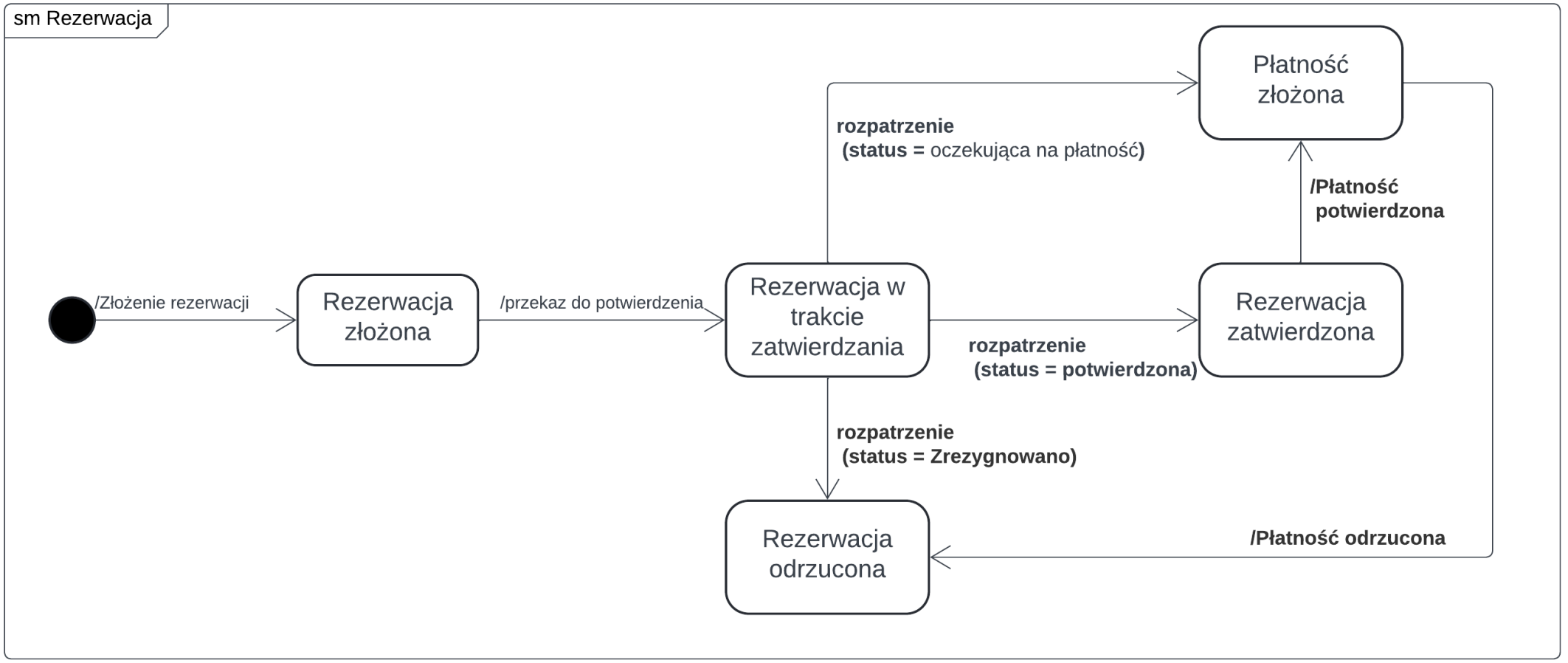
5. Scenariusz przypadku użycia

Tytuł przypadku użycia:	Zarezerwowanie lotu
Aktorzy:	Pasażer
Warunek początkowy:	Użytkownik jest zalogowany do systemu.
Główny przepływ zdarzeń:	
	1. Pasażer rozpoczyna rozpocznya przypadek użycia i prosi o wyświetlenie listy lotów.
	2. System wyświetla dostępną liczbę lotów.
	3. Pasażer wybiera lot którym jest zainteresowany.
	4.System wyświetla dostępną liczbę miejsc.
	5. pasażer wybiera miejsce.
	6. System prosi o o wybranie klasy lotu.
	7. Pasażer wybiera klasę.
	8. System wyświetla rezerwację i zapisuję ją.
	9. Koniec przypadku użycia.
Alternatywny przepływ zdarzeń:	
	1a. Brak dostępnych lotów. System wyświetla informacje o braku lotów.
	1aa. System, kończy przypadek użycia.
	4a. System wyświetla informacje o braku dostępnych miejsc i pyta o chęć wybrania innego lotu.
	4aa. Pasażer wybiera nowy lot, System przechodzi do kroku 2.
	4ab. Pasażer rezygnuje z wyboru, System kończy przypadek użycia.
	7a. Pasażer wprowadza błędną klasę lotu. System wyświetla komunikat i pyta o chęć ponownego wprowadzenia.
	7aa. Pasażer wprowadza nową klasę, System przchodzi do kroku 6.
	7ab. Pasażer rezygnuje z wyboru klasy lotu, System kończy przypadek użycia.

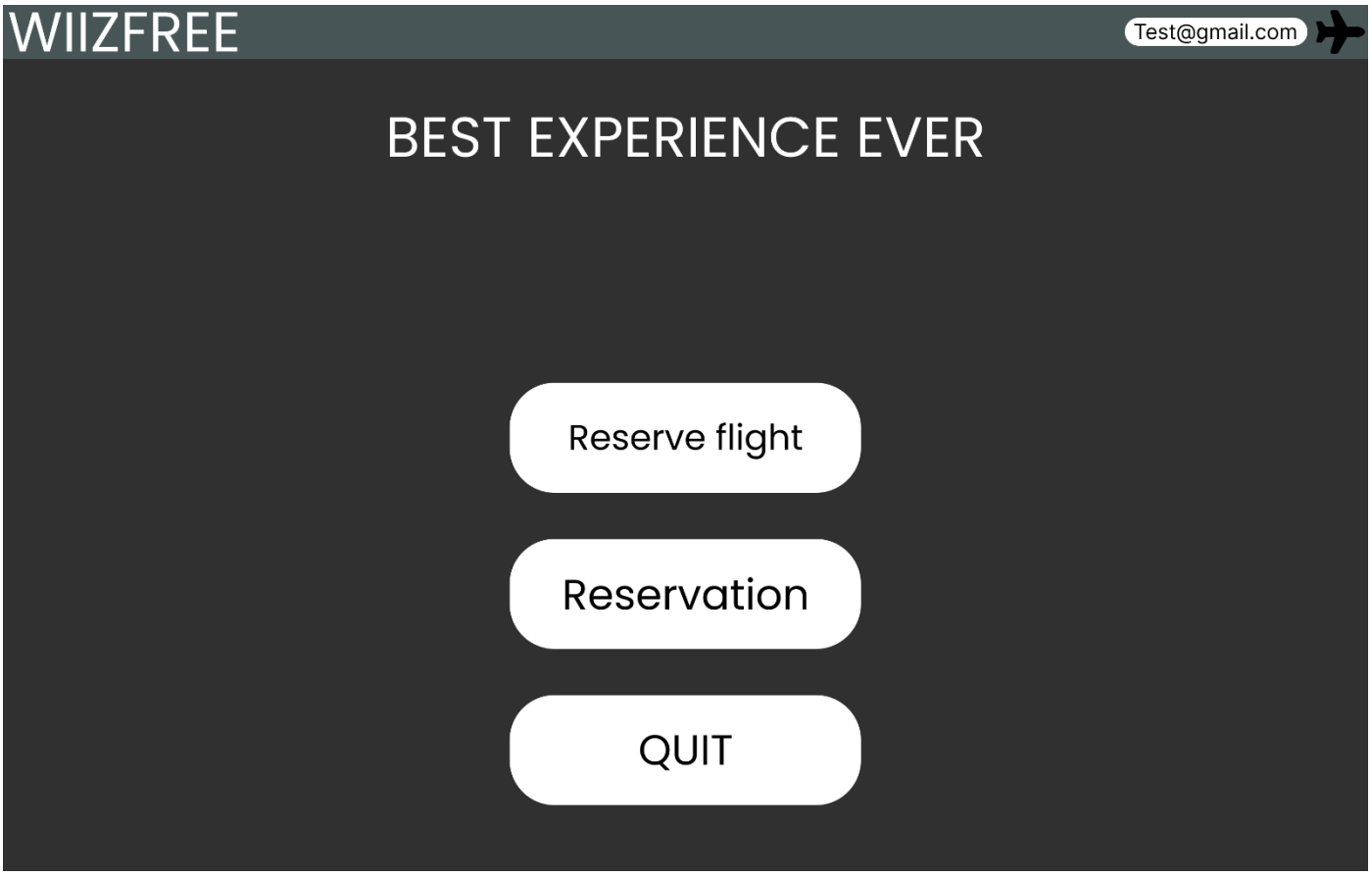
6. Diagram aktywności dla przypadku użycia

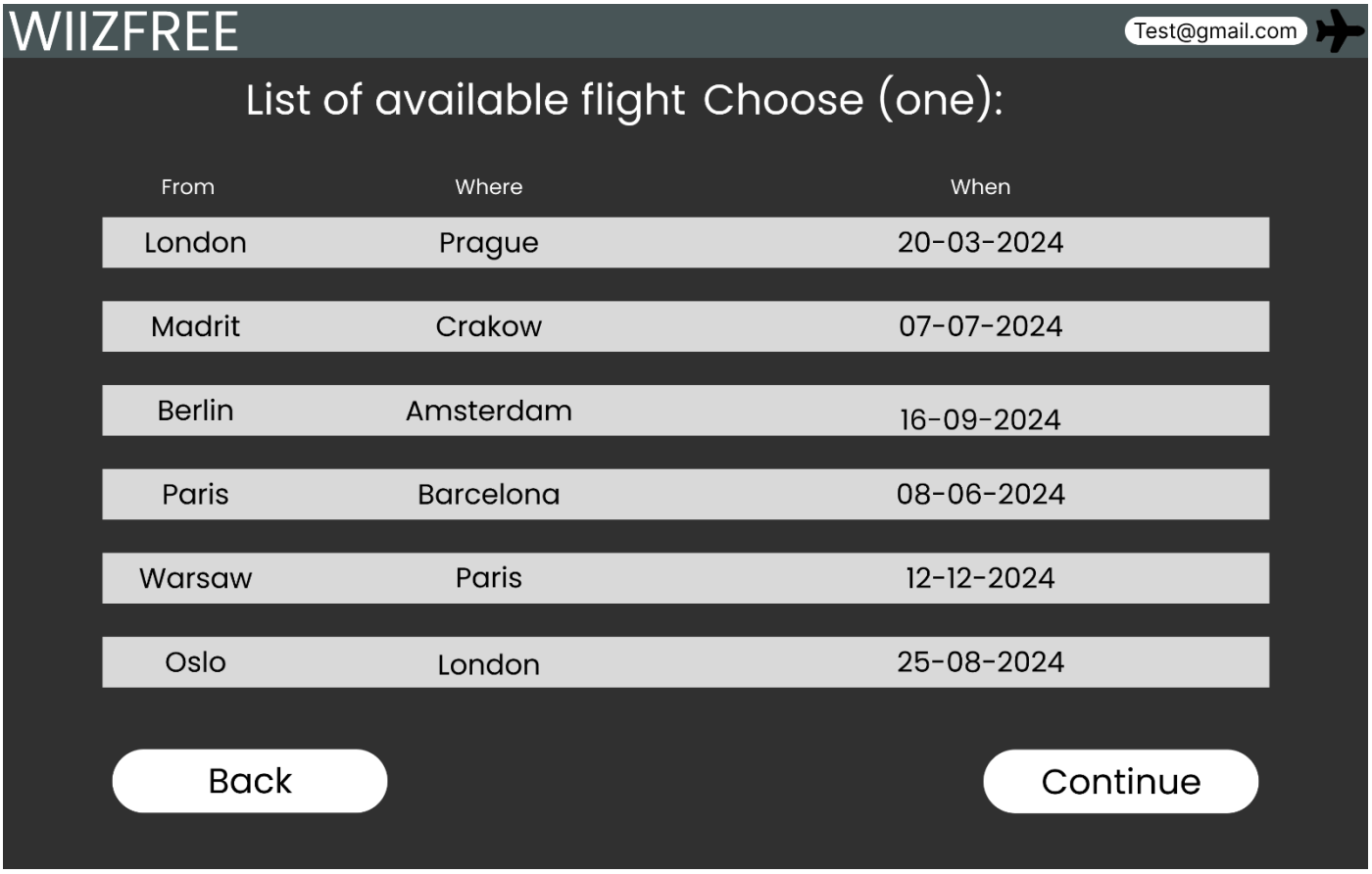


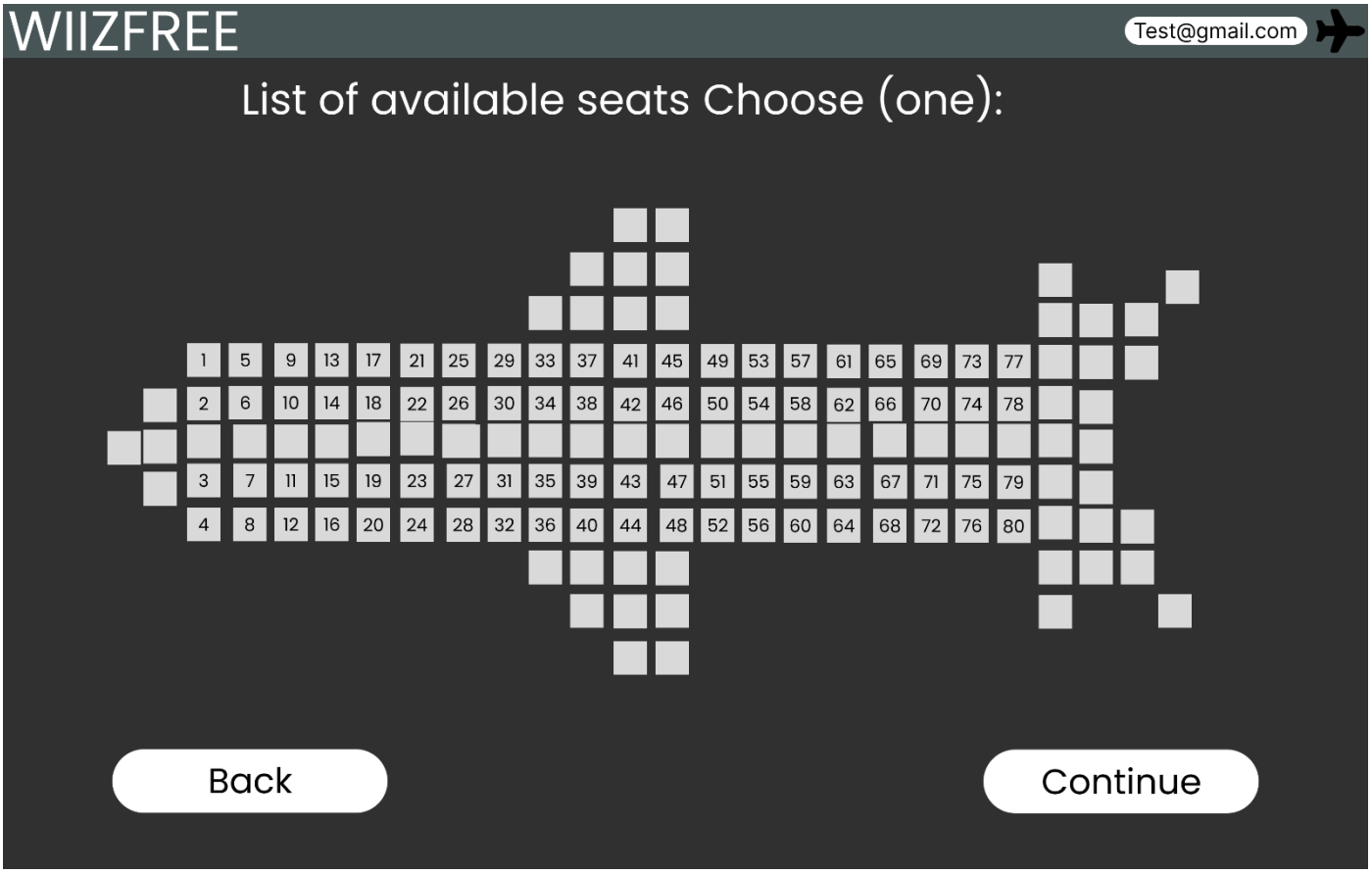
7. Diagram stanu klasy

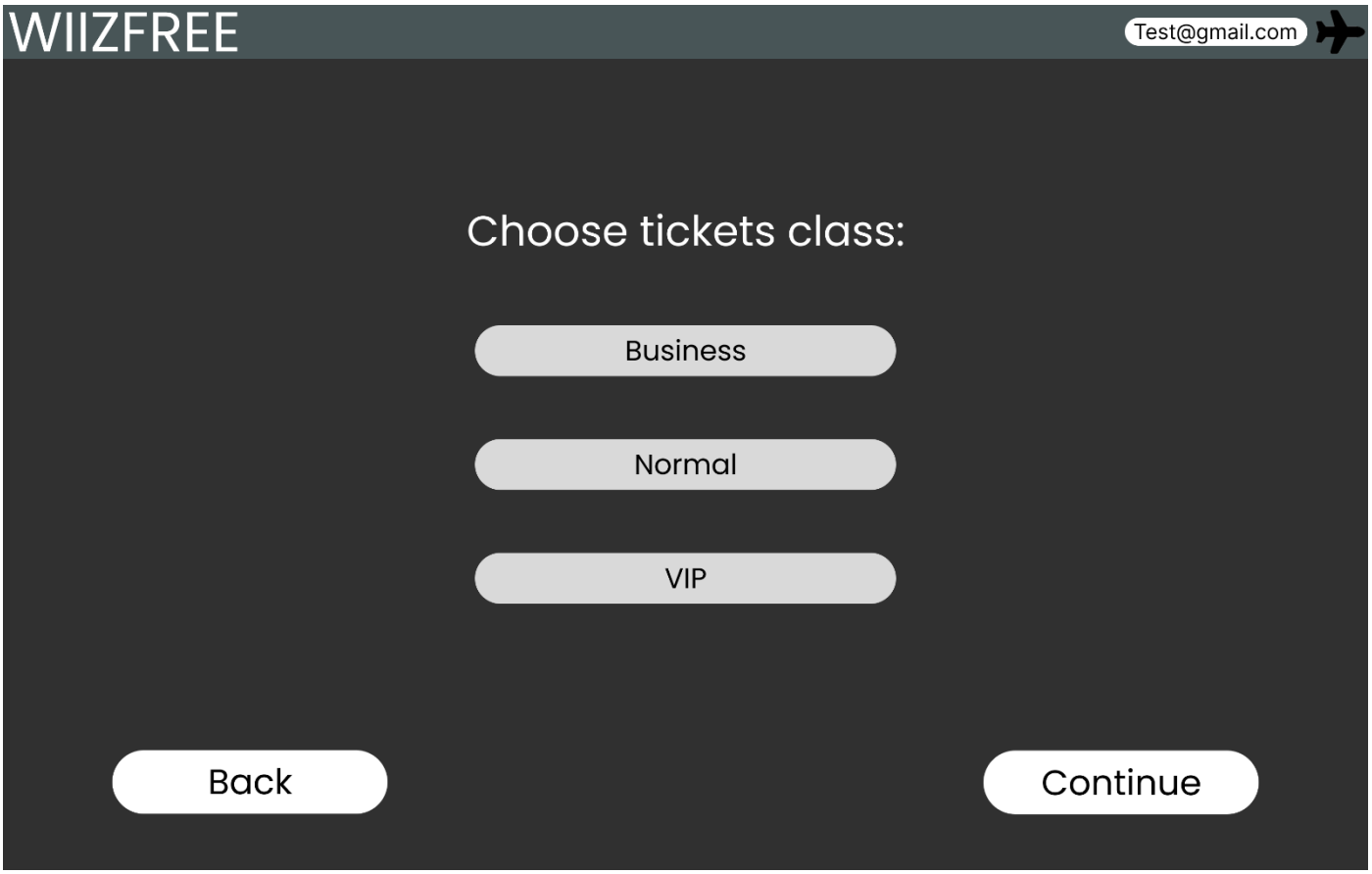


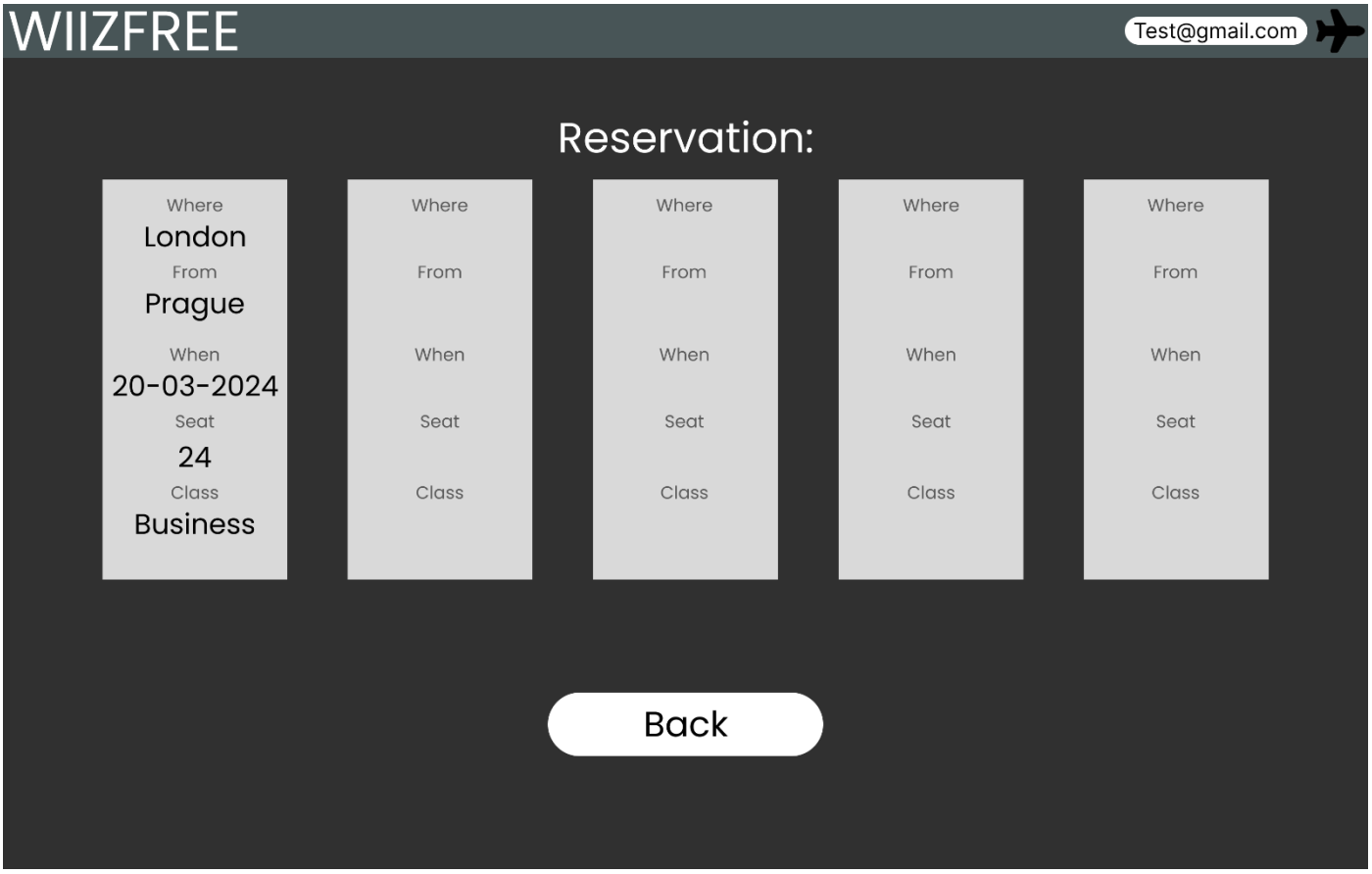
8. Projekt GUI






WIIZFREE
Test@gmail.com
List of available seats Choose (one):
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77
2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62 66 70 74 78
3 7 11 15 19 23 27 31 35 39 43 47 51 55 59 63 67 71 75 79
4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80
Back
Continue


WIIZFREE
Test@gmail.com
Choose tickets class:
Business
Normal
VIP
Back
Continue


WIIZFREE
Test@gmail.com
Reservation:
Where
London
From
Prague
When
20-03-2024
Seat
24
Class
Business
Where
From
When
Seat
Class
Where
From
When
Seat
Class
Where
From
When
Seat
Class
Where
From
When
Seat
Class
Back

9. Omówienie decyzji pprojektowych i skutków analizy dynamicznej.

Projekt zostanie zaimplementowany przy użyciu języka Java. Graficzny interfejs użytkownika (GUI) zostanie opracowany w Java FX z wykorzystaniem środowiska IntelliJ IDEA.

Dziedziczenie typu "overlapping, complete" dla klasy "Osoba" z podziałem na "Pasażer" i "Pracownik" zostanie zaimplementowane przez kompozycję. Dla każdej z ról zostaną stworzone oddzielne klasy, zawierające charakterystyczne składowe.

Dziedziczenie typu "disjoint, dynamic" dla klasy "Pracownik" z rozróżnieniem na "Członek załogi" i "Pilot" będzie realizowane poprzez kompozycję. Stworzone zostaną odpowiednie metody umożliwiające zmianę roli z jednej na drugą by zachować dynamiczność.(ZamienNaPilota(), ZamienNaCzlonkaZalogi()).

Dziedziczenie typu "disjoint, complete" dla klasy "Pojazd" z podziałem na "Pojazd zasilany paliwem", "Pojazd zasilany elektrycznie" oraz "Samolot" zostanie zrealizowane poprzez stworzenie trzech klas i ustanowienie klasy "Pojazd" jako klasy abstrakcyjnej.

W projekcie będzie wiele dozwolonych asocjacji ze względu na wymaganie przechowywania historii rezerwacji, oznaczone przez specyfikator {bag}.

Kompozycja typu "terminal – lotnisko" zostanie zaimplementowana przez referencje do całości (Lotnisko), przy czym na każdym lotnisku muszą być przynajmniej dwa terminale.

Zwykłe asocjacje między takimi klasami jak "Pasażer – Rezerwacja", "Pilot - Samolot”, "Rezerwacja – Lot", "Lot – Lotnisko" oraz "Lotnisko – Pojazd" będą zaimplementowane poprzez listy referencji lub referencje.