

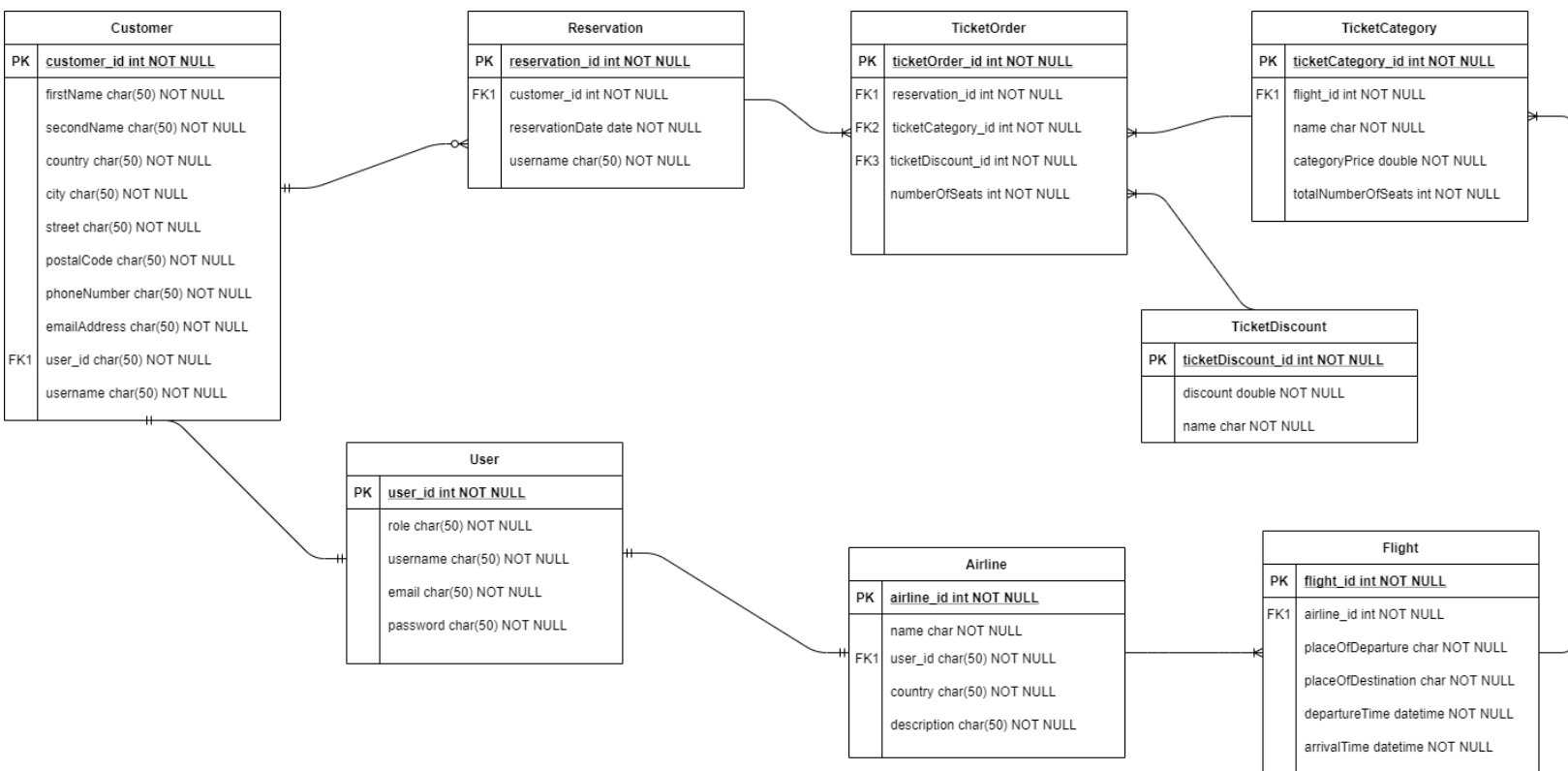
Technologie obiektowe 2020/2021

Projekt - grupa czwartek 16:15

Bartosz Kaszuba, Tomasz Kozyra, Kacper Rosiak, Amadeusz Szabala

1. Model danych

1.1. Schemat modelu danych



1.2. Opis struktur modelu

Airline

Tabela przechowuje informacje o liniach lotniczych. Pole description może pozostać puste. Kluczem głównym jest airline_id. Klucz obcy to user_id.

- Airline_id - Jest to pole przechowujące ID unikalne dla każdej linii lotniczej w bazie.
- Name - Nazwa linii lotniczej.
- Country - kraj.
- Description - Opcjonalny opis linii lotniczej.
- Username - Nazwa użytkownika przypisanego danej linii lotniczej.

Flight

Tabela przechowuje informacje o lotach. Klucz główny to Fligh_ID. Klucz obcy to airline_id. Każda linia lotnicza może posiadać wiele lotów. Wszystkie pola nie mogą być puste.

- Flight_id - Pole przechowujące ID lotu. Dla każdego lotu ID jest unikalne.
- Airline_id - ID linii lotniczej obsługującej dany lot.
- Departure - Miejsce wylotu.
- Destination - Miejsce przylotu.
- departureDate - Dzień wylotu.
- departureTime - Czas wylotu.
- arrivalDate - Dzień wylotu.
- arrivalTime - Czas przylotu.

User

Tabela przechowującą użytkowników zarejestrowanych w naszym systemie. Kluczem głównym jest user_id. Wszystkie pola nie mogą być puste.

- User_id - Unikalny klucz przypisany każdemu użytkownikowi.
- Role - Uprawnienia użytkownika w systemie.
- Username - Nazwa użytkownika.
- Password - Hasło użytkownika, przechowywane jako zaszyfrowane.
- Email - Adres email użytkownika.

Customer

Tabela przechowująca dane prywatne użytkowników. Kluczem głównym jest customer_id. Wszystkie pola nie mogą być puste.

- customer_ID - Unikalny klucz przypisany każdemu prywatnemu użytkownikowi, który może dokonywać rezerwacji lotów. Różnica między customer_ID, a user_ID jest taka, że każdy Customer ma user_ID, ale nie każdy User ma customer_ID. Może być User, który jest odpowiedzialny za linię lotniczą i ma airline_ID. Customer może mieć kilka rezerwacji.
- user_ID - jest to ID użytkownika wskazujące na jego konto oraz uprawnienia.

- Username - nazwa użytkownika przypisana danemu klientowi
- Pozostałe pola to są dane użytkownika, jego adres oraz mail i numer telefonu.

Reservation

Tabela przechowująca informacje o rezerwacjach klienta. Klucz główny to reservation_id, a klucz obcy customer_id. Wszystkie pola nie mogą być puste.

- Reservation_id - Unikalny klucz rozróżniający rezerwacje w systemie.
- Customer_id - Pole przechowujące ID klienta, który dokonał rezerwacji.
- reservationDate - Pole przechowujące datę dokonanej rezerwacji.
- Username - Nazwa użytkownika składającego rezerwację.

TicketOrder

Tabela przechowuje informacje o naszej rezerwacji. Dla każdej rezerwacji wybieramy sobie klasę biletów np biznes i ilość rezerwowanych biletów. Jeżeli np chcemy zarezerwować 2 bilety klasy biznes oraz 3 klasy ekonomicznej to dostaniemy dwa wpisy w TicketOrder. Kluczem głównym jest ticketOrder_ID, a kluczami obcymi reservation_id, ticketCategory_id oraz ticketDiscount_id.

Wszystkie pola nie mogą być puste.

- ticketOrder_id - Unikalny klucz dla każdej rezerwacji.
- reservation_ID - ID rezerwacji. Do jednego ID może być przypisane kilka TicketOrder.
- ticketCategory_ID - ID kategorii, dla której rezerwujemy bilety.
- ticketDiscount ID - ID zniżki, którą wybraliśmy.
- numberOfSeats - ilość miejsc, które rezerwujemy.

TicketDiscount

Tabela przechowująca informacje o wszystkich zniżkach. Wszystkie pola nie mogą być puste.

- ticketDiscount_ID - Unikalny klucz dla każdej zniżki.
- Discount - Wartość procentowa zniżki.
- Name - Nazwa zniżki.

TicketCategory

Tabela przechowuje informacje o klasach biletów dostępnych dla danego lotu. Każdy lot może mieć przypisane kilka wpisów w TicketCategory w zależności od ilości dostępnych klas. Tabela ta przechowuje również bazową cenę biletów dla danej kategorii oraz ilość miejsc przeznaczonych na daną klasę w danym locie. Wszystkie pola nie mogą być puste. Kluczem głównym jest ticketCategory_Id, a kluczem obcym flight_ID.

- ticketCategory_ID - Unikalny klucz dla każdej kategorii.
- flight_ID - Klucz lotu, dla którego opisywane są klasy.
- Name - Nazwa kategorii.
- categoryPrice - Bazowa cena biletu w danej klasie. Na jej podstawie obliczane są zniżki.
- totalNumberOfSeats - Ilość miejsca przeznaczonych na daną klasę w danym locie.

2. Struktura projektu

2.1. Model

W katalogu *model* znajdują się klasy definiujące model danych w aplikacji. Klasy zostały przygotowane do podłączenia do bazy danych (pola mają potrzebne adnotacje, klucze obce, definicje relacji itd.).

2.2. Repository

W katalogu *repository* znajdują się klasy odpowiedzialne za pobieranie danych z bazy danych. Klasy te pozwalają na wykonywanie na podłączonej bazie danych operacji CRUDowych za pomocą klas modelu. Ponadto, repozytoria pozwalają na definiowanie metod służących do pobierania danych spełniających jakieś wymagania (np. metoda *Airline findByName(String name)* w klasie *AirlineRepository*, pozwalająca na pobranie linii lotniczej o podanej nazwie).

2.3. Service

W katalogu *service* znajdują się klasy, które są pośrednikami pomiędzy modelem a repozytoriami.

- *AirlineService*,
- *CustomerService*,
- *FlightService*,
- *ReservationService*,
- *TicketCategoryService*,
- *TicketDiscountService*,
- *TicketOrderService*,
- *UserPrincipalService*.

2.4. Controller

W katalogu *controller* znajdują się klasy odpowiedzialne za pośrednictwo pomiędzy widokami a modelem. Klasy *controller* obsługują logikę wyświetlanego widoku, służą do walidacji pól formularza, routingu między poszczególnymi widokami, a także do interakcji z użytkownikiem (obsługiwanie naciśnięcia przycisków).

2.5. View

2.5.1. Widoki dla użytkownika niezalogowanego

- Widok główny (*AnonymousMainView*)

- Zawiera loty polecane na podstawie ich popularności (liczby złożonych na nie rezerwacji)
- Widok listy przewoźników (AnonymousAirlinesView)
 - Użytkownik niezalogowany może jedynie przeglądać przewoźników
- Widok listy lotów (AnonymousFlightView)
 - Użytkownik niezalogowany może jedynie przeglądać loty - nie może dokonywać na nie rezerwacji

2.5.2. Widoki dla użytkownika zalogowanego jako klient

- Widok główny (MainView)
 - Zawiera loty polecane na podstawie ich popularności (liczby złożonych na nie rezerwacji) oraz najpopularniejsze kierunki podróży
- Widok listy przewoźników (UserAirlinesView)
 - Klient może jedynie przeglądać przewoźników
- Widok listy klientów (UserCustomersView)
 - Klient widzi tylko dane klienta powiązanego z jego kontem, może edytować swoje dane.
- Widok listy lotów (UserFlightView)
 - Użytkownik zalogowany jako klient może przeglądać loty i składać na nie rezerwacje.
- Widok listy rezerwacji (UserReservationView)
 - Użytkownik zalogowany może przeglądać i edytować jedynie swoje rezerwacje
- Widok formularza do dodawania/edycji rezerwacji (AddReservationView)
 - Użytkownik zalogowany może dokonywać rezerwacji na loty

2.5.3. Widoki dla użytkownika zalogowanego jako linia lotnicza i administratora

- Widok główny (MainView)
 - Zawiera loty polecane na podstawie ich popularności (liczby złożonych na nie rezerwacji) oraz najpopularniejsze kierunki podróży
- Widok listy przewoźników (AirlinesView)
 - Administrator może dodawać, edytować i usuwać przewoźników
 - Usunięcie przewoźnika skutkuje usunięciem wszystkich lotów tego przewoźnika, i w konsekwencji wszystkich rezerwacji na loty tego przewoźnika
 - Użytkownik zalogowany jako linia lotnicza może edytować szczegóły linii lotniczej przypisanej do jego konta
- Widok listy lotów (FlightView)
 - Administrator może dodawać, edytować i usuwać loty
 - Usunięcie lotu przez administratora oznacza usunięcie wszystkich rezerwacji na ten lot

- Użytkownik zalogowany jako linia lotnicza może przeglądać, edytować i usuwać loty linii lotniczej przypisanej do jego konta
- Widok szczegółów lotu (FlightDetailsView)
 - Użytkownik zalogowany jako linia lotnicza może przeglądać szczegóły lotu, które zawierają też statystyki:
 - Liczby rezerwacji złożonych na lot
 - Liczby wolnych i zajętych miejsc na lot
 - Przychodów z rezerwacji na dany lot
- Widok formularza do dodawania i edycji lotów (addFlightView)
 - Użytkownik zalogowany jako linia lotnicza może dodawać i edytować loty linii lotniczej przypisanej do jego konta
- Widok listy rezerwacji (ReservationListView)
 - Użytkownik zalogowany jako linia lotnicza usuwać rezerwacje (jest to równoważne z anulowaniem rezerwacji)
- Widok listy klientów (CustomersView)
 - Użytkownik zalogowany jako linia lotnicza i administrator mogą przeglądać klientów

2.5.4. Widoki dotyczące rejestracji i logowania

- Widok logowania (LoginView)
- Widok rejestracji jako klient (RegistrationView)
- Widok rejestracji jako linia lotnicza (AirlineRegistrationView)

Za zarządzanie widokami odpowiedzialne są klasy znajdujące się w katalogu *controller*

2.6. Utils

W katalogu *utils* znajdują się klasy pomocnicze służące do obsługi pewnych powtarzalnych funkcjonalności w różnych fragmentach kodu (zgodnie z regułą dry). Na chwilę obecną znajduje się tam:

- Enumerator UserRole
- Klasa Validator - która zawiera różne metody walidacji pól formularza, wykorzystywana w formularzach dodawania użytkownika i linii lotniczej
- GenericFilter - klasa zawiera metody pozwalające na realizację filtrowania danych w widokach
- EmailHandler - klasa zawiera metody pozwalające na realizację wysyłania powiadomień email.

3. Warstwa persystencji i baza danych

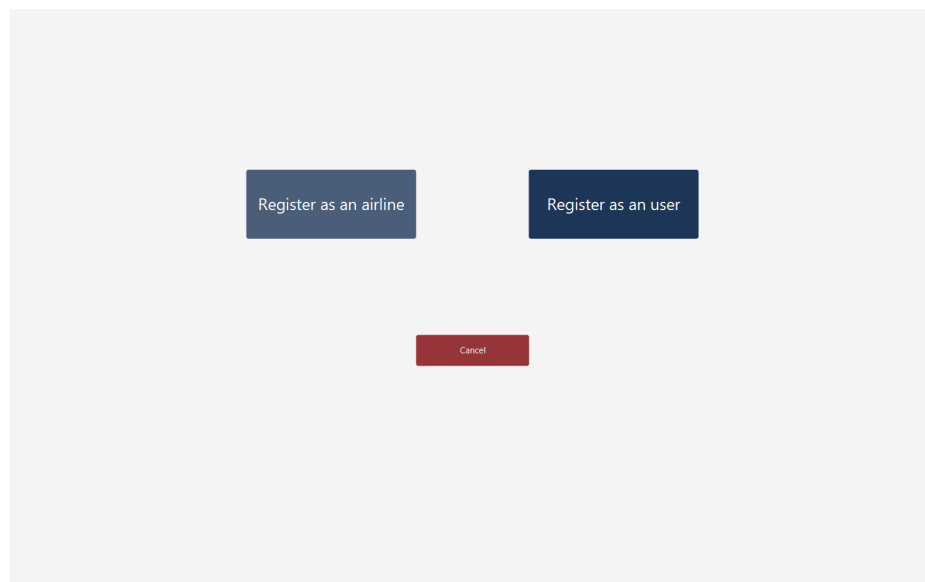
W projekcie korzystamy z bazy danych MySQL uruchamianej na lokalnym serwerze. W celu umożliwienia mapowania obiektowo-relacyjnego wykorzystaliśmy Spring Data JPA, bardzo

podobny do Hibernate. Spring Data JPA zajmuje się m.in. tworzeniem tabel w bazie danych na podstawie kodu napisanego w Javie. Udostępnia również wiele metod poprzez repozytoria zdefiniowane w projekcie (metody takie jak `save()`, `saveAll()`, `findAll()`, `delete()`, `deleteAll()`).

4. Autentykacja i autoryzacja

Autentykacja i autoryzacja są realizowane z wykorzystaniem *Spring Security*. Główna konfiguracja znajduje się w klasie `SecurityConfiguration` w katalogu *config*. Dane użytkowników są przechowywane w bazie danych. Wszystkie hasła przechowywane w bazie danych są zakodowane.

Aplikacja pozwala na rejestrację jako linia lotnicza (airline) albo jako klient (user). Stosownego wyboru dokonuje się po naciśnięciu przycisku Register.



Enter airline details

Airline Name:

Country:

Description:

Email Address:

Username:

Password:

Enter customer details

First Name:

Last Name:

Country:

City:

Street Name:

Postal code:

Phone Number:

Email Address:

Username:

Password:

Formularze do rejestracji wymagają wypełnienia wszystkich pól. Sprawdza się dodatkowo czy e-mail ma poprawną formę oraz czy w bazie danych nie istnieje już użytkownik o danej nazwie użytkownika.

Po rejestracji użytkownik zostaje przeniesiony do ekranu głównego i jeśli wprowadził poprawne dane, może się zalogować do aplikacji.

Log in

username

password

W systemie, dla użytkownika zdefiniowane są 4 role.

Użytkownik niezalogowany (anonymous)

Użytkownik niezalogowany może jedynie przeglądać polecane loty, najpopularniejsze kierunki lotów, listę wszystkich lotów i listę wszystkich linii lotniczych.

Użytkownik zalogowany jako klient (user)

Użytkownik zalogowany ma wszystkie możliwości użytkownika niezalogowanego, ponadto może:

- Edytować swoje dane użytkownika
- Składać rezerwacje na loty
- Przeglądać swoje rezerwacje
- Edytować i usuwać swoje rezerwacje

Użytkownik zalogowany jako linia lotnicza (airline)

Użytkownik zalogowany jako linia lotnicza ma wszystkie możliwości użytkownika niezalogowanego, ponadto może:

- Przeglądać klientów
- Dodawać, edytować i usuwać loty jego linii lotniczej (wraz z kategoriami biletów)
- Dodawać, edytować i usuwać zniżki na loty
- Przeglądać i usuwać rejestracje złożone na loty jego linii lotniczej
- Przeglądać szczegóły lotów, w których mieszczą się również statystyki dotyczące:
 - Liczby rezerwacji złożonych na lot
 - Liczby wolnych i zajętych miejsc na lot
 - Przychodów z rezerwacji na dany lot

Użytkownik zalogowany jako administrator (admin)

Użytkownik zalogowany jako administrator może:

- Przeglądać, edytować i usuwać klientów (usunięcie klienta usuwa również użytkownika z nim powiązanego, zabierając dostęp do aplikacji)
- Przeglądać, edytować i usuwać linie lotnicze (usunięcie linii lotniczej usuwa również użytkownika z nią powiązanego, zabierając dostęp do aplikacji)
- Przeglądać, dodawać, usuwać i edytować loty i szczegóły lotów, w których mieszczą się również statystyki dotyczące:
 - Liczby rezerwacji złożonych na lot

- Liczby wolnych i zajętych miejsc na lot
- Przychodów z rezerwacji na dany lot
- Przeglądać, usuwać i edytować rezerwacje wszystkich

5. Podział pracy

5.1. Etap m1

5.1.1. Praca wspólna

- Skonstruowanie modelu - diagram

5.1.2. Bartosz Kaszuba

- Dodanie klas TicketDiscount, User, Reservation do modelu aplikacji
- Stworzenie relacji pomiędzy elementami modelu (klucze obce itd.)
- Stworzenie widoku formularza do dodawania przewoźników
- Zaprogramowanie dodawania przewoźników

5.1.3. Tomasz Kozyra

- Inicjalizacja projektu
- Dodanie klas Airline i Flight do modelu aplikacji
- Stworzenie widoku przewoźników
- Zaprogramowanie widoku przewoźników tak aby wyświetlał listę przewoźników

5.1.4. Kacper Rosiak

- Dodanie klas TicketCategory, TicketOrder do modelu aplikacji
- Stworzenie widoku klientów
- Zaprogramowanie widoku klientów tak aby wyświetlał listę klientów
- Stworzenie opisu struktury projektu (punkt 2 dokumentacji)

5.1.5. Amadeusz Szabała

- Dodanie klas Customer, Review do modelu aplikacji
- Stworzenie widoku formularza do dodawania klientów
- Zaprogramowanie dodawania klientów
- Stworzenie opisu tabel w dokumentacji (punkt 1.2 dokumentacji)

5.2. Etap m2

5.2.1. Bartosz Kaszuba

- Filtrowanie przewoźników, lotów, rezerwacji
- Stworzenie widoku formularza do dodawania rezerwacji
- CRUD dla rezerwacji + persystencja
- Rozwiązywanie konfliktów przy tworzeniu rezerwacji

5.2.2. Tomasz Kozyra

- Aktualizacja wyglądu aplikacji
- CRUD dla przewoźników i klientów + persystencja
- Dodanie zapelniania bazy danych przykładowymi danymi przy starcie aplikacji
- Naprawa pomniejszych błędów
- Dokumentacja

5.2.3. Kacper Rosiak

- Autentykacja i autoryzacja
- Stworzenie dodatkowych widoków dla zalogowanego użytkownika
- Podział widoków na klasy dla użytkownika niezalogowanego/zalogowanego/administradora
- Naprawa pomniejszych błędów

5.2.4. Amadeusz Szabała

- Stworzenie widoku dla lotów oraz dodawania lotów
- CRUD dla lotów + persystencja
- Rozwiązywanie konfliktów przy tworzeniu lotów
- Dokumentacja

5.3. Etap m3

5.3.1. Praca wspólna

- Aktualizacja dokumentacji

5.3.2. Bartosz Kaszuba

- Aktualizacja(ulepszenie) CRUD dla rezerwacji
 - Dodanie możliwości tworzenia i przeglądania zamówień na określone klasy biletów
 - Dodanie możliwości tworzenia i przeglądania zniżek dla biletów
- Dodanie CRUD dla kategorii biletów

5.3.3. Tomasz Kozyra

- Dodanie statystyk dotyczących lotów
- Dodanie polecanych lotów
- Naprawa błędów

5.3.4. Kacper Rosiak

- Umożliwienie rejestracji jako klient
- Umożliwienie rejestracji i logowania jako linia lotnicza
- Naprawa błędów

5.3.5. Amadeusz Szabala

- Dodanie powiadomień wysyłanych na adres email użytkownika
 - Powiadomienia o złożeniu rezerwacji
 - Powiadomienia o zbliżającym się locie

6. Wykorzystane wzorce projektowe

- Inversion of Control realizowane przez wstrzykiwanie zależności. Wykorzystaliśmy do tego celu framework Spring. Obsługuje on m.in. do wstrzykiwanie właściwych implementacji kontrolerów.
- MVC - podział struktury aplikacji na kontrolery, widoki i model danych. Pozwala na niezależne rozwijanie poszczególnych elementów aplikacji, enkapsulację logiki aplikacji w obszarze kontrolerów.
- Repository - dostęp do danych. Dodatkowa warstwa abstrakcji dodana nad klasami modelu, realizowana przy pomocy Spring Data JPA.

7. Uruchamianie projektu

Do uruchomienia i korzystania z aplikacji niezbędna jest baza danych MySQL stworzona według instrukcji i uruchomiona lokalnie.

Aby stworzyć lokalną bazę danych potrzebną do uruchomienia projektu, należy:

1. Pobrać i zainstalować MySQL
2. Uruchomić lokalny serwer bazy danych na porcie 3306
3. Zalogować się do MySQL z linii poleceń komendą:
 - `mysql -u [username] -p`
 - Gdzie [username] to nazwa użytkownika root lokalnego serwera MySQL
 - Hasło o które system poprosi, jest hasłem przypisanym do tego konta
4. Utworzyć bazę danych ciągiem poleceń:
 - `create database yourflights;`
 - `create user 'yourFlightsUser'@'%' identified by '123';`
 - `grant all on yourflights.* to 'yourFlightsUser'@'%';`
5. Wgrać do bazy danych plik *yourflights.sql* znajdujący się w katalogu głównym projektu. Wchodzimy do katalogu głównego projektu i wykonujemy komendę:
 - `Mysql -u [username] -p yourflights < yourflights.sql`
 - Gdzie [username] to nazwa konta root lokalnego serwera MySQL
 - Hasło o które system poprosi, jest hasłem przypisanym do tego konta

Uruchomienie aplikacji:

7.1. Sposób 1 - Gradle

1. Przechodzimy do katalogu projektu
2. Uruchamiamy komendę "gradle run"

7.2. Sposób 2 - jar

Projekt uruchamiamy przez uruchomienie pliku **yourflights-3.0.0-m3.jar** znajdującego się w katalogu głównym projektu.

1. Przechodzimy do katalogu głównego projektu
2. Wywołujemy komendę "java -jar yourflights-3.0.0-m3.jar"