

# Bazy danych I

## Airline reservation system

Bartosz Piętka

# Spis treści

1.	Projekt koncepcji, założenia .....	2
1.1.	Zdefiniowanie tematu projektu .....	2
1.2.	Analiza wymagań .....	2
1.3.	Zaprojektowanie funkcji .....	3
2.	Projekt diagramu .....	3
2.1.	Budowa diagramu ERD .....	3
2.2.	Zdefiniowanie encji .....	4
2.3.	Zaprojektowanie relacji pomiędzy encjami .....	4
3.	Projekt logiczny .....	4
3.1.	Projektowanie tabel, kluczy, indeksów .....	4
3.2.	Operacje na danych .....	7
4.	Projekt funkcjonalny .....	7
4.1.	Edycja i obsługa danych .....	7
4.2.	Wizualizacja danych .....	7
4.3.	Zdefiniowanie panelu obsługi aplikacji .....	8
5.	Dokumentacja .....	9
5.1.	Wprowadzanie danych .....	9
5.2.	Dokumentacja użytkownika .....	9
5.3.	Dokumentacja techniczna .....	9
5.4.	Literatura .....	10

## 1. Projekt koncepcji, założenia

### 1.1. Zdefiniowanie tematu projektu

Tematem projektu w ramach przedmiotu Bazy Danych I jest stworzenie serwisu umożliwiającego wyszukiwanie oraz dokonywanie rezerwacji lotów lotniczych. Inspiracją do stworzenia projektu był serwis Skyscanner <https://www.skyscanner.pl/>.

### 1.2. Analiza wymagań

Baza danych powinna posiadać takie funkcjonalności jak:

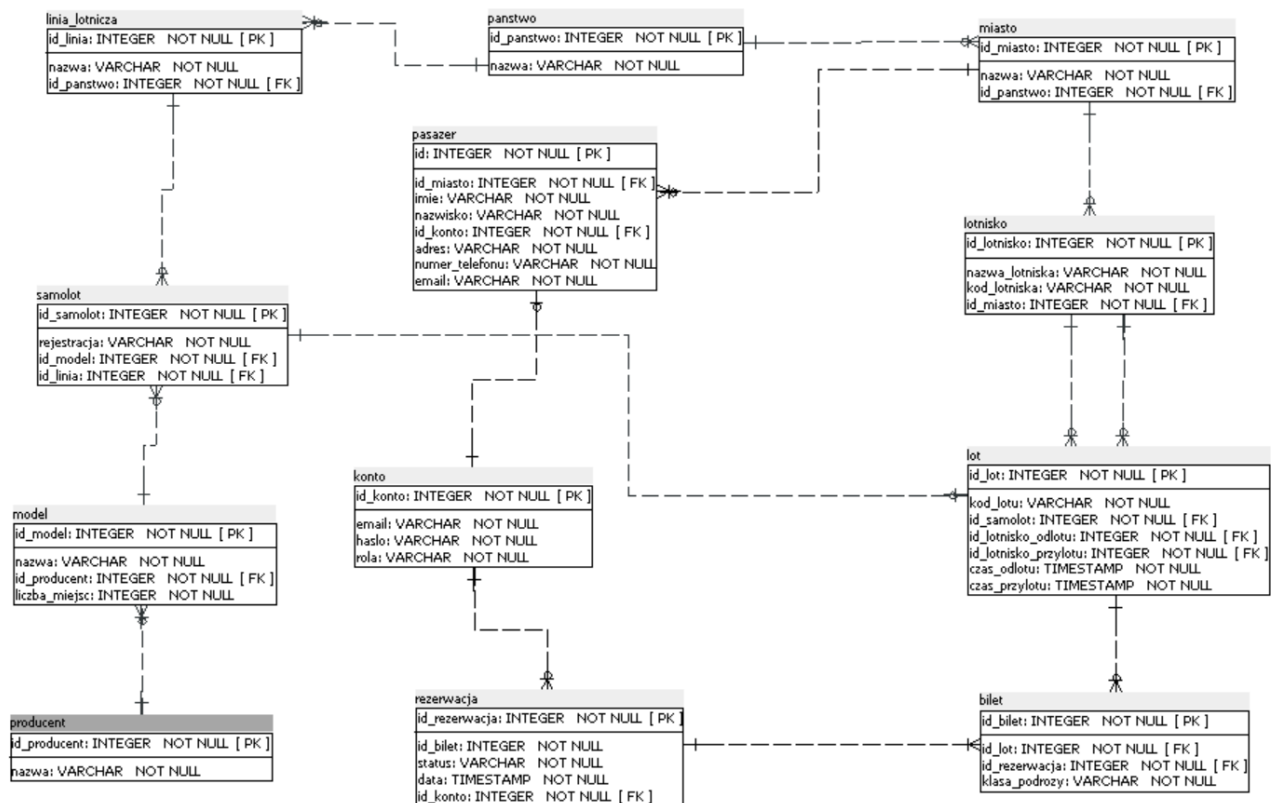
- Wyszukiwanie lotów na podstawie zadanych kryteriów
- Rezerwowanie biletów (po zalogowaniu)
- Podgląd wykonanych rezerwacji (po zalogowaniu)
- Zakładanie konta
- Usuwanie wybranego użytkownika (admin)
- Dodawanie rekordów do tabel oraz wyświetlanie danych z każdej tabeli (admin)

### 1.3. Zaprojektowanie funkcji

Podstawową funkcją serwisu jest wyszukiwanie rekordów z odpowiednich tabel. Serwis powinien zapewniać możliwość zakładania konta, logowania oraz dokonywania rezerwacji. Edycja danych w bazie możliwa tylko z poziomu administratora.

## 2. Projekt diagramu

### 2.1. Budowa diagramu ERD



## 2.2. Zdefiniowanie encji

W stworzonej bazie danych istnieją poniższe encje:

- Lot
- Lotnisko
- Miasto
- Państwo
- Pasażer
- Rezerwacja
- Bilet
- Linia lotnicza
- Samolot
- Model
- Producent

## 2.3. Zaprojektowanie relacji pomiędzy encjami

Wszystkie relacje między tabelami są z założenia typu 1:N. W projekcie przyjęto relację 1:1 tabel konto:pasazer.

## 3. Projekt logiczny

### 3.1. Projektowanie tabel, kluczy, indeksów

<i>lot</i>		
<i>KLUCZ</i>	<i>TYP</i>	<i>OPIS</i>
<i>id_lot[PK]</i>	<i>INT</i>	<i>id lotu, klucz główny</i>
<i>kod_lotu</i>	<i>VARCHAR</i>	<i>Unikalny kod lotu</i>
<i>id_samolot[FK]</i>	<i>INT</i>	<i>Numer id samolotu</i>
<i>id_lotnisko_odlotu[FK]</i>	<i>INT</i>	<i>id lotniska startowego</i>
<i>id_lotnisko_przylotu[FK]</i>	<i>INT</i>	<i>id lotniska docelowego</i>
<i>czas_odlotu</i>	<i>TIMESTAMP</i>	<i>Format YYYY-MM-DD HH:MM:SS</i>
<i>czas_przylotu</i>	<i>TIMESTAMP</i>	<i>Format YYYY-MM-DD HH:MM:SS</i>

<i>bilet</i>		
<i>KLUCZ</i>	<i>TYP</i>	<i>OPIS</i>
<i>id_bilet[PK]</i>	<i>INT</i>	<i>unikalny numer biletu</i>
<i>id_lot[FK]</i>	<i>INT</i>	<i>unikalny numer lotu</i>
<i>id_rezerwacja[FK]</i>	<i>INT</i>	<i>unikalny numer rezerwacji</i>
<i>id_pasażer</i>	<i>VARCHAR</i>	<i>unikalny numer pasażera</i>
<i>klasa_podróży</i>	<i>VARCHAR</i>	<i>klasa podróży</i>

lotnisko		
KLUCZ	TYP	OPIS
<i>id_lotniska</i> [PK]	INT	unikalny numer lotniska
<i>kod_lotniska</i>	VARCHAR	unikalny kod lotniska (3 znaki)
<i>nazwa_lotniska</i>	VARCHAR	pełna nazwa lotniska
<i>id_miasto</i> [FK]	INT	id miasta w którym jest lotnisko

rezerwacja		
KLUCZ	TYP	OPIS
<i>id_rezerwacja</i> [PK]	INT	unikalny numer rezerwacji
<i>id_bilet</i> [FK]	INT	unikalny numer biletu
<i>id_konto</i> [FK]	INT	Unikalny numer konta na którym dokonany
<i>status</i>	VARCHAR	aktualny status rezerwacji
<i>id_pasazer</i> [FK]	INT	unikalny numer pasażera
<i>data</i>	TIMESTAMP	data złożenia rezerwacji, format YYYY-MM-DD HH:MM:SS

konto		
KLUCZ	TYP	OPIS
<i>id_konto</i> [PK]	INT	Unikalny numer konta
<i>email</i>	VARCHAR(40)	Email na który założone jest konto
<i>haslo</i>	VARCHAR(65)	Hasło zdefiniowane dla konta
<i>rola</i>	VARCHAR	Rola użytkownika, domyślnie 'user', możliwy również 'admin'

pasazer		
KLUCZ	TYP	OPIS
<i>id</i> [PK]	INT	unikalny numer pasażera
<i>Imie</i> (50)	VARCHAR	imię pasażera
<i>Nazwisko</i> (50)	VARCHAR	nazwisko pasażera
<i>id_konto</i> [FK]	INT	Unikalny numer konta
<i>id_miasto</i> [FK]	INT	unikalny numer miasta w którym mieszka pasażer
<i>Adres</i> (200)	VARCHAR	adres pasażera
<i>Email</i> (40)	VARCHAR	Adres email pasazera
<i>numer_telefonu</i>	VARCHAR	numer telefonu kontaktowy

miasto		
KLUCZ	TYP	OPIS
<i>id_miasto</i> [PK]	INT	unikalny numer miasta
<i>id_panstwo</i> [FK]	INT	unikalny numer państwa w którym leży miasto
<i>nazwa</i>	VARCHAR	nazwa miasta

<i>panstwo</i>		
<i>KLUCZ</i>	<i>TYP</i>	<i>OPIS</i>
<i>id_panstwo[PK]</i>	<i>INT</i>	<i>unikalny numer państwa</i>
<i>nazwa</i>	<i>VARCHAR</i>	<i>nazwa państwa</i>

<i>linia lotnicza</i>		
<i>KLUCZ</i>	<i>TYP</i>	<i>OPIS</i>
<i>id_linia[PK]</i>	<i>INT</i>	<i>unikalny numer linii</i>
<i>nazwa</i>	<i>VARCHAR</i>	<i>nazwa linii</i>
<i>id_panstwo[FK]</i>	<i>INT</i>	<i>id państwa w którym zarejestrowana jest linia</i>

<i>samolot</i>		
<i>KLUCZ</i>	<i>TYP</i>	<i>OPIS</i>
<i>id_samolot[PK]</i>	<i>INT</i>	<i>Unikalny numer samolotu</i>
<i>rejestracja</i>	<i>VARCHAR</i>	<i>Numer rejestracyjny samolotu</i>
<i>id_linia[FK]</i>	<i>INT</i>	<i>Unikalny numer linii do której należy samolot</i>
<i>id_model[FK]</i>	<i>INT</i>	<i>Unikalny numer modelu samolotu</i>

<i>model</i>		
<i>KLUCZ</i>	<i>TYP</i>	<i>OPIS</i>
<i>id_model[PK]</i>	<i>INT</i>	<i>unikalny numer modelu</i>
<i>nazwa</i>	<i>VARCHAR</i>	<i>nazwa modelu</i>
<i>liczba_miejsc</i>	<i>INT</i>	<i>Liczba miejsc w modelu</i>
<i>id_producent[FK]</i>	<i>INT</i>	<i>id producenta danego modelu</i>

<i>producent</i>		
<i>KLUCZ</i>	<i>TYP</i>	<i>OPIS</i>
<i>id_producent[PK]</i>	<i>INT</i>	<i>unikalny numer producenta</i>
<i>nazwa</i>	<i>VARCHAR</i>	<i>nazwa producenta</i>

### 3.2. Operacje na danych

Do modyfikowania i wyszukiwania danych zostały utworzone zapytania oraz funkcje, które dostępne są w przesłanych plikach w dołączonym folderze sql.

Administrator ma możliwość dodawania z poziomu serwisu danych do tabel niezwiązanych z rezerwacją, poprawność oraz logika danych wprowadzonych przez administratora nie jest sprawdzana. Do tabel związanych z rezerwacją i kontem użytkownika dane dodawane są automatycznie w celu zachowania spójności danych, poprawność danych z formularza jest sprawdzana z poziomu aplikacji. Podczas rejestracji pobrane hasło jest poddawane kodowaniu poprzez haszowanie za pomocą BCryptPasswordEncoder, a następnie odkodowane podczas logowania. W bazie przechowywane są zakodowane hasła. Sesje są zabezpieczone przed włamaniem. Administrator posiada również możliwość usunięcia użytkownika wraz z jego rezerwacjami, biletami i kontem realizowane poprzez wywołanie funkcji sql. W celu bezpieczeństwa administratora można dodać tylko z poziomu bazy danych.

## 4. Projekt funkcjonalny

### 4.1. Edycja i obsługa danych

Możliwość obsługi danych została umożliwiona w następującej hierarchii:

- użytkownik niezalogowany: wyszukiwanie lotu, wyszukiwanie informacji o locie, zakładanie konta, logowanie
- użytkownik zalogowany: dokonywanie rezerwacji, przegląd rezerwacji, wyszukiwanie i dodawanie biletów powrotnych do rezerwacji
- administrator: przegląd danych z tabeli, dodawanie rekordów, usuwanie danych wybranych użytkowników

### 4.2. Wizualizacja danych

Dane pobierane są za pomocą prostych formularzy tekstowych, przykład:

Miasto startowe
<input type="text"/>
Miasto docelowe
<input type="text"/>
Szukaj lotów

W celu wizualizacji danych zostały utworzone specjalne klasy imitujące widoki. Dane na stronie wyświetlają się w formie sformatowanych tabel w wersjach dla użytkowników:

a) niezalogowanych, przykład:

Kod	Lotnisko odlotu	Lotnisko docelowe	Data odlotu	Data przylotu
QB878	Warsaw Chopin Airport	Los Angeles International Airport	2020-09-08 15:49:06.0	2020-09-10 00:15:21.0
TZ741	Warsaw Chopin Airport	Los Angeles International Airport	2021-01-05 12:31:55.0	2021-01-06 03:06:37.0
LO9	Warsaw Chopin Airport	Los Angeles International Airport	2021-07-12 12:00:00.0	2021-07-13 02:25:00.0

b) zalogowanych, wraz z odsyłaczami powodującymi określone działania, przykład:

Kod	Lotnisko odlotu	Lotnisko docelowe	Data odlotu	Data przylotu		
NC388	Frankfurt Airport	Los Angeles International Airport	2020-09-09 13:40:53.0	2020-09-10 22:47:13.0	Rezerwuj klasa ekonomiczna	Rezerwuj klasa biznes
JT415	Frankfurt Airport	Los Angeles International Airport	2022-05-02 02:18:00.0	2022-05-03 10:37:20.0	Rezerwuj klasa ekonomiczna	Rezerwuj klasa biznes
GH734	Frankfurt Airport	Los Angeles International Airport	2022-05-02 04:26:09.0	2022-05-03 08:49:22.0	Rezerwuj klasa ekonomiczna	Rezerwuj klasa biznes
HX716	Frankfurt Airport	Los Angeles International Airport	2022-05-02 19:04:03.0	2022-05-03 18:56:36.0	Rezerwuj klasa ekonomiczna	Rezerwuj klasa biznes

### 4.3. Zdefiniowanie panelu obsługi aplikacji

Dostępne są trzy panele obsługi klienta:

a) Użytkownik niezalogowany:

Wyszukaj loty	Wyszukaj wszystkie loty z miasta	Wyszukaj loty danej linii	Informacje o locie	Zaloguj	Załącz konto
---------------	----------------------------------	---------------------------	--------------------	---------	--------------

b) Użytkownik zalogowany:

Zarezerwuj lot	Moje rezerwacje	Wyloguj
----------------	-----------------	---------

c) Administrator:

Dane z tabel	Dodaj dane do tabel	Usuń użytkownika	Wyloguj
--------------	---------------------	------------------	---------

Pola dane z tabel oraz dodaj dane do tabel rozwijają się po najechaniu kursorem:

Dane z tabel	Dodaj dane do tabel							Usuń użytkownika	Wyloguj
	Państwo	Miasto	Producent	Samolot	Model	Lot	Lotnisko	Linia lotnicza	



## 5. Dokumentacja

### 5.1. Wprowadzanie danych

Wprowadzenie danych startowych odbyło się za pomocą skryptów napisanych w języku sql znajdującymi się w załączonym folderze sql.

### 5.2. Dokumentacja użytkownika

Strona dostępna jest na Amazon Web Services pod adresem:

<http://ars-env-2.eba-95hxx22h.us-east-2.elasticbeanstalk.com/>

Wszystkie funkcje opisane są na stronie w języku polskim. Aby zalogować się do panelu administratora należy użyć danych:

Email: admin@admin.pl

Hasło:

### 5.3. Dokumentacja techniczna

Technologie wykorzystane do projektu:

- język Java 8
- Spring Boot 2.4.1
- Thymeleaf 2.2.6
- Spring Security

Baza danych założona jest w serwisie Elephantsql.

Struktura katalogów w przesłanym projekcie:

app ->

    java ->

        controller -> pliki obsługujące endpointy

        DAO -> pliki obsługujące operacje na bazie danych

        exceptions-> wyjątki

        models -> klasy modelowe dla wszystkich tabeli w bazie

        security -> konfiguracje zabezpieczeń www

        service -> obsługa logowania i sesji

        utils -> walidacja danych z formularzy

        wrappers -> klasy imitujące widoki

resources ->

    static -> plik css oraz zdjęcie tła

    templates -> szablony stron www

sql\_utils -> plik .architect z schematem erd projektu

    sql -> zawiera skrypty sql używane w projekcie

doc -> zawiera plik z dokumentacją

## 5.4. Literatura

- materiały dr Antoniego Dydejczyka - <https://newton.fis.agh.edu.pl/~antek/index.php>
- strony <https://spring.io/> oraz <https://www.baeldung.com/>
- dokumentacja języka HTML – <https://www.w3schools.com/html/>
- dokumentacja CSS - <https://www.w3schools.com/cssref/>