

Państwowa Wyższa Szkoła Zawodowa w Nowym Sączu

Bazy Danych – dokumentacja projektowa

Temat projektu:

Rozkład lotów pasażerskiej linii lotniczej

Nazwisko i imię: Cetera Maciej Adamczyk Mateusz	Kierunek studiów: Informatyka Stosowana (stacjonarnie) Rok II
Data złożenia projektu: 15.06.2022	Grupa: P1

Spis treści

1. Tytuł Projektu	3
2. Nazwa Robocza	3
3. Cel i opis projektu.....	3
4. Zakres	3
5. Wymagania	4
5.1. Funkcjonalne	4
5.2. Niefunkcjonalne	5
6. Diagram ERD	6
7. Diagram przypadków użycia	7
8. Scenariusze przypadków użycia	8
9. Opis użytych technologii.....	23
10. Implementacja.....	26
11. Bilans oraz estymacja czasowa.....	27
11.1. Zrealizowane wymagania	27
11.2. Niezrealizowane wymagania	29
12. Repozytorium	29
13. Podsumowanie oraz wnioski	29
14. Bibliografia	29

1. Tytuł Projektu

Rozkład lotów pasażerskiej linii lotniczej

2. Nazwa Robocza

Linia lotnicza

3. Cel i opis projektu

Celem projektu było stworzenie aplikacji webowej wraz z bazą danych wspomagającej funkcjonowanie linii lotniczej w zakresie umożliwienia pasażerom kupna biletu lotniczego i organizacji harmonogramu lotów.

Aplikacja umożliwia bezproblemowy, szybki oraz wygodny dostęp do harmonogramu lotów, kupna biletu na wybrany rejs.

Projekt był tworzony w grupie 2-osobowej, postępy projektowe były na bieżąco sprawdzane na zajęciach projektowych.

4. Zakres

Aplikacja webowa obejmuje wszelkie funkcjonalności niezbędne do:

- zarządzania harmonogramem lotów lotniczych
- sprzedaży biletów lotniczych
- organizacji pracy dla pracowników

Aplikacja przewiduje 3 grupy użytkowników:

- pasażer
- pracownik (pilot, stewardessa)
- administrator

Dostęp do aplikacji webowej odbywa się poprzez nazwę domenową

<http://linia-lotnicza.test/> za pośrednictwem serwera HTTP Apache.

5. Wymagania

5.1. Funkcjonalne

Wspólne dla Pasażera, Pracownika i Administratora:

- możliwość logowania
- możliwość wylogowania
- edycję danych osobowych w bazie danych (imię, nazwisko, email, hasło)

Z punktu widzenia pasażera:

- możliwość rejestracji
- możliwość podglądu harmonogramu lotu (miejsce, data, godzina odlotu -> miejsce, data, godzina przylotu, model samolotu przewidzianego do lotu)
- możliwość kupna biletu lotniczego na dany lot (wybór klasy, miejsca siedzenia, widok ceny)
- możliwość zarządzania bagażem (dodania, edycji, usunięcia swojego bagażu z bazy danych)
- możliwość zarządzania dokonanymi rezerwacjami (eksport do pliku w formacie PDF, złożenie reklamacji, anulowanie rezerwacji)

Z punktu widzenia pracownika aplikacja umożliwia:

- podgląd otrzymanych wypłat (kwota, okres od, okres do)
- wyświetlenie harmonogramu pracy (widok przpisania do danego lotu)

Z punktu widzenia administratora aplikacja umożliwia:

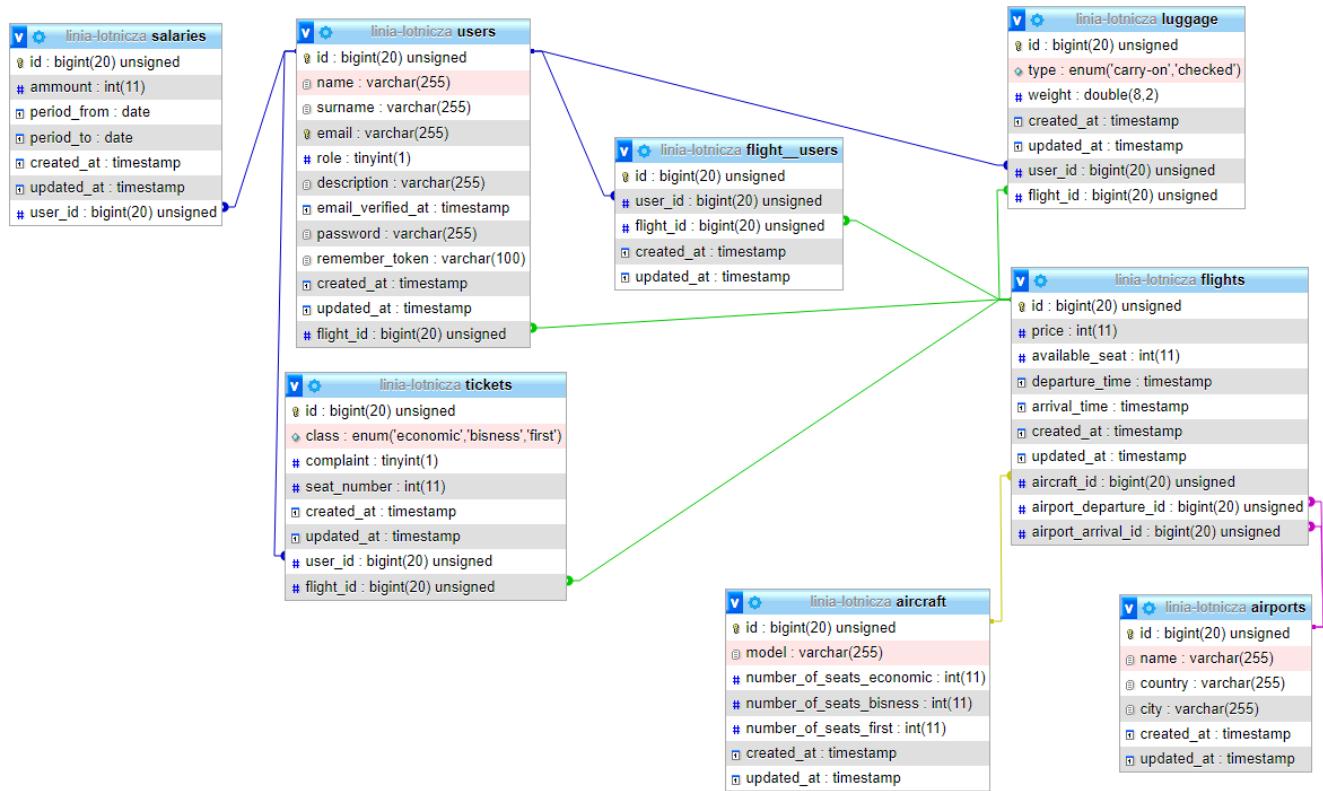
- zarządzanie pracownikami, na co składa się:
 - dodanie pracownika do bazy danych (imię, nazwisko, email, hasło, opis)
 - edycję danych osobowych pracownika w bazie danych (imię, nazwisko, email, hasło, opis)
 - usunięcie pracownika z bazy danych
- zarządzanie lotniskami, na co składa się:
 - dodanie lotniska do bazy danych (nazwa, kraj, miasto)
 - edycję danych lotniska w bazie danych (nazwa, kraj, miasto)
 - usunięcie lotniska z bazy danych
- zarządzanie samolotami, na co składa się:
 - dodanie samolotu do bazy danych (model, liczba siedzeń w danej klasie)
 - edycję danych samolotu w bazie danych (model, liczba siedzeń w danej klasie)
 - usunięcie samolotu z bazy danych

- zarządzanie wypłatami, na co składa się:
 - dodanie wypłat pracowników do bazy danych (kwota, okres od, okres do, pracownik)
 - edycję danych wypłat pracowników w bazie danych (kwota, okres od, okres do, pracownik)
 - usunięcie wypłat pracowników z bazy danych
- zarządzanie harmonogramem lotu, na co składa się:
 - dodanie lotu do bazy danych (samolot, lotnisko odlotu, data, godzina odlotu -> lotnisko przylotu, data, godzina przylotu, cena biletu)
 - edycję danych lotu w bazie danych (lotnisko odlotu, data, godzina odlotu -> lotnisko przylotu, data, godzina przylotu, cena biletu)
 - usunięcie lotu z bazy danych
- zarządzanie harmonogramem pracy, na co składa się:
 - dodanie pracownika do danego lotu do bazy danych
 - edycję danych dodania pracownika do danego lotu w bazie danych
 - usunięcie pracownika z danego lotu z bazy danych

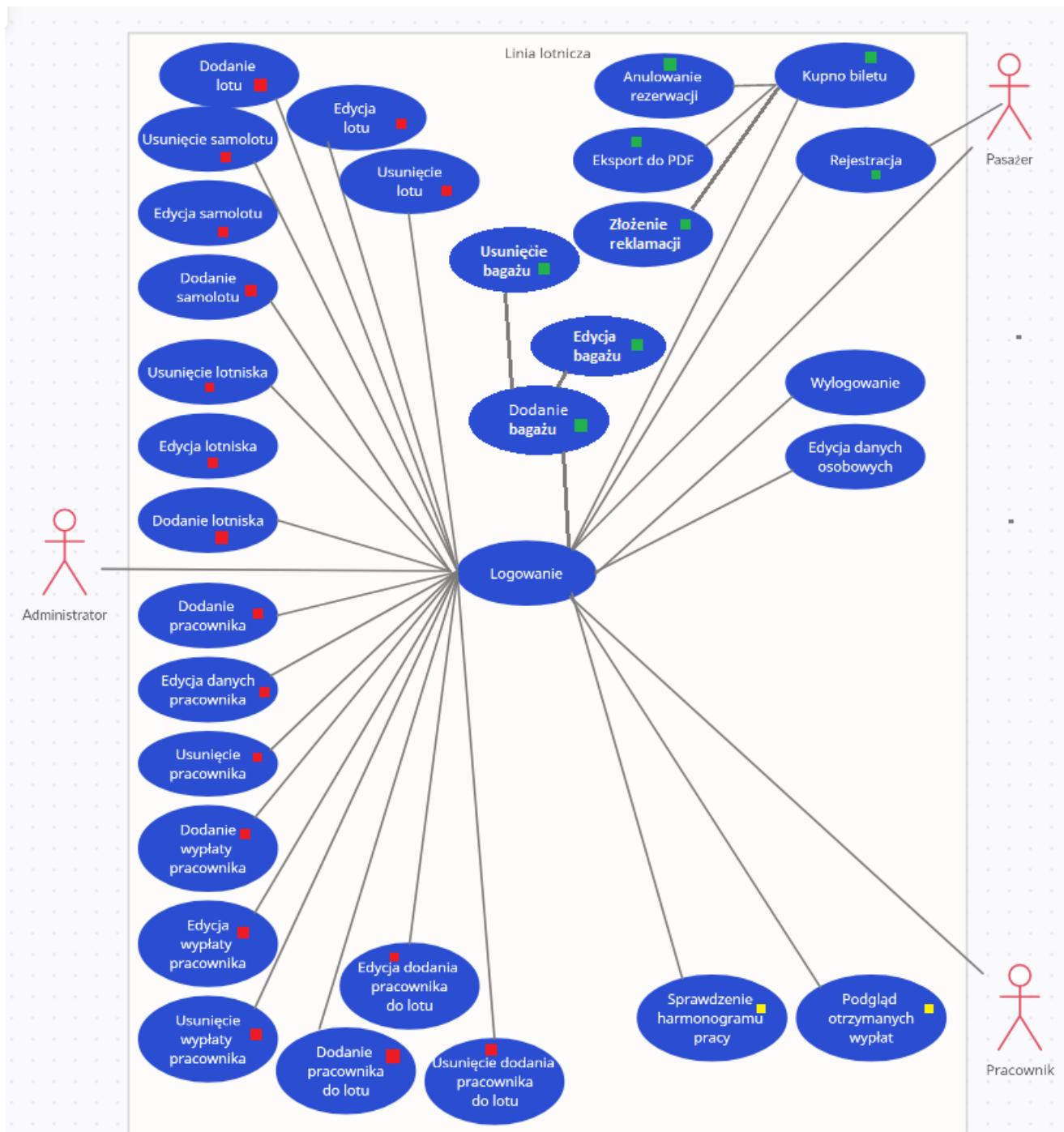
5.2. Niefunkcjonalne

- wygląd aplikacji webowej
 - wygląd interfejsów
 - pasażera:
 - panel logowania
 - panel rejestracji
 - panel zarządzania (profil, kupno biletu, zarządzanie rezerwacją)
 - pracownika:
 - panel logowania
 - panel zarządzania (profil, podgląd wypłat, podgląd harmonogramu pracy)
 - administratora:
 - panel logowania
 - panel zarządzania (profil, zarządzanie pracownikami, zarządzanie lotniskami, zarządzanie samolotami, zarządzanie harmonogramem lotów, zarządzanie wypłatami, zarządzanie harmonogramem pracy)
 - wygląd strony głównej
- szybkość ładowania się strony
- szybkość responsywności strony

6. Diagram ERD



7. Diagram przypadków użycia



Legenda:

█ Dla Administratora

█ Dla Pracownika

█ Dla Pasażera

[brak] Dla Wszystkich

8. Scenariusze przypadków użycia

PU01: Scenariusz dla przypadku użycia: Rejestracja	
Nazwa	<i>Rejestracja</i>
Numer	<i>01</i>
Aktorzy	<i>Pasażer</i>
Krótki opis	<i>Rejestracja pasażera.</i>
Warunki wstępne	<i>Pasażer nie jest zarejestrowany.</i>
Warunki końcowe	<i>Pasażer został zarejestrowany lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer kliką „Rejestracja”. 2. Pasażer wpisuje dane do formularza (imię, nazwisko, email, hasło). 3. Pasażer po wypełnieniu formularza kliką w „Register”. 4. System sprawdza poprawność danych. 5. Pasażer zostaje zarejestrowany. 6. Pasażer zostaje przekierowany do panelu powitalnego.
Alternatywne przepływy zdarzeń	<p>5b) Jeżeli w kroku 2 pasażer wprowadzi niepoprawne dane, system informuje o niepoprawnych danych i potrzebie ich poprawy.</p>

PU02: Scenariusz dla przypadku użycia: Logowanie	
Nazwa	<i>Logowanie</i>
Numer	<i>02</i>
Aktorzy	<i>Pasażer lub Pracownik lub Administrator</i>
Krótki opis	<i>Zalogowanie Pasażera, Pracownika lub Administratora.</i>
Warunki wstępne	<i>Pasażer jest zarejestrowany lub Pracownik został dodany przez Administratora lub Administrator został przypisany do bazy danych.</i>
Warunki końcowe	<i>Pasażer lub Pracownik lub Administrator został zalogowany lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer lub Pracownik lub Administrator kliką „logowanie”. 2. Pasażer lub Pracownik lub Administrator wpisuje dane do formularza (email, hasło). 3. Pasażer lub Pracownik lub Administrator po wypełnieniu formularza kliką w „Login”. 4. System sprawdza poprawność danych.

	<p>5. Pasażer lub Pracownik lub Administrator zostaje zarejestrowany.</p> <p>6. Pasażer lub Pracownik lub Administrator zostaje przekierowany do panelu zarządzania.</p>
Alternatywne przepływy zdarzeń	<p>5b) Jeżeli w kroku 3 Pasażer lub Pracownik lub Administrator wprowadzi niepoprawne dane, system informuje o niepoprawnych danych i potrzebie ich poprawy.</p>

PU03: Scenariusz dla przypadku użycia: Wylogowanie

Nazwa	Wylogowanie
Numer	03
Aktorzy	Pasażer lub Pracownik lub Administrator
Krótki opis	Wylogowanie Pasażera, Pracownika lub Administratora.
Warunki wstępne	Pasażer lub Pracownik lub Administrator jest zalogowany.
Warunki końcowe	Pasażer lub Pracownik lub Administrator został wylogowany lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.
Główny przepływ zdarzeń	<p>1. Pasażer lub Pracownik lub Administrator kliką „Logout”.</p> <p>2. Pasażer lub Pracownik lub Administrator zostaje wylogowany.</p> <p>3. System przekierowuje Pasażera lub Pracownika lub Administratora do strony głównej.</p>
Alternatywne przepływy zdarzeń	<p>3b) Jeżeli w kroku 2 wystąpi błąd stan systemu nie ulegnie zmianie.</p>

PU04: Scenariusz dla przypadku użycia: Kupno biletu lotniczego

Nazwa	Kupno biletu lotniczego
Numer	04
Aktorzy	Pasażer
Krótki opis	Pasażer zakupuje bilet lotniczy na lot.
Warunki wstępne	Pasażer jest zalogowany.
Warunki końcowe	Pasażer zakupił bilet lotniczy na dany lot lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.
Główny przepływ zdarzeń	<p>1. Pasażer z poziomu swojego panelu zarządzania kliką „Kup bilet”.</p> <p>2. Pasażer przy odpowiednim locie kliką „Zobacz bilety”.</p> <p>3. Pasażer przy odpowiedniej klasie biletu kliką „Zakup”.</p> <p>4. Pasażer wybiera miejsce siedzenia oraz kliką „Zatwierdź”</p>

	<i>transakcje”.</i> 5. System zmniejsza o 1 liczbę dostępnych miejsc.
Alternatywne przepływy zdarzeń	3b) Jeżeli w kroku 2 liczba dostępnych wolnych miejsc wynosi 0 opcja „Zakup” jest niedostępna .

PU05: Scenariusz dla przypadku użycia: Eksport biletu lotniczego do pliku w formacie PDF	
Nazwa	<i>Eksport biletu lotniczego do pliku w formacie PDF</i>
Numer	<i>05</i>
Aktorzy	<i>Pasażer</i>
Krótki opis	<i>Bilet lotniczy zostaje wyeksportowany do pliku w formacie PDF.</i>
Warunki wstępne	<i>Pasażer jest zalogowany oraz posiada zakupiony bilet lotniczy.</i>
Warunki końcowe	<i>Bilet lotniczy zostaje wyeksportowany do pliku w formacie PDF lub został wyświetlony komunikat o błędzie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer z poziomu swojego panelu zarządzania kliką „Rezerwacje”. 2. Pasażer przy odpowiednim bilecie kliką „PDF”. 3. System eksportuje bilet lotniczy do pliku w formacie PDF, a następnie pobiera go.
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> 4. Jeżeli w kroku 3 wystąpi błąd bilet lotniczy nie zostanie wyeksportowany i pobrany.

PU06: Scenariusz dla przypadku użycia: Edycja danych osobowych	
Nazwa	<i>Edycja danych osobowych</i>
Numer	<i>06</i>
Aktorzy	<i>Pasażer lub Pracownik lub Administrator</i>
Krótki opis	<i>Pasażer lub Pracownik lub Administrator edytuje swoje dane osobowe z bazy danych.</i>
Warunki wstępne	<i>Pasażer lub Pracownik lub Administrator jest zalogowany.</i>
Warunki końcowe	<i>Dane osobowe zostały zmienione lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer lub Pracownik lub Administrator z poziomu swojego panelu zarządzania kliką „Profile”. 2. Pasażer lub Pracownik lub Administrator kliką „Edycja”. 3. Pasażer lub Pracownik lub Administrator wypełnia formularz odpowiednimi danymi.

	4. Pasażer lub Pracownik lub Administrator kliką „Zmień”. 5. System sprawdza poprawność danych. 6. Dane zostają zmienione.
Alternatywne przepływy zdarzeń	6b). Jeżeli w kroku Pasażer lub Pracownik lub Administrator dane nie ulegną zmianie.

PU07: Scenariusz dla przypadku użycia: Podgląd otrzymanych wypłat

Nazwa	<i>Podgląd otrzymanych wypłat</i>
Numer	<i>07</i>
Aktorzy	<i>Pracownik</i>
Krótki opis	<i>Pracownik sprawdza swoje wypłaty.</i>
Warunki wstępne	<i>Pracownik jest zalogowany.</i>
Warunki końcowe	<i>Lista wypłat Pracownika zostaje wyświetlona lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	1. Pracownik z poziomu swojego panelu zarządzania kliką „Twoje wypłaty”. 2. System wyświetla listę wypłat Pracownika.
Alternatywne przepływy zdarzeń	<i>3. Jeżeli w kroku 2 wystąpi błąd system wyświetli komunikat o błędzie.</i>

PU08: Scenariusz dla przypadku użycia: Sprawdzenie harmonogramu pracy

Nazwa	<i>Sprawdzenie harmonogramu pracy</i>
Numer	<i>08</i>
Aktorzy	<i>Pracownik</i>
Krótki opis	<i>Pracownik sprawdza swój harmonogram pracy.</i>
Warunki wstępne	<i>Pracownik jest zalogowany.</i>
Warunki końcowe	<i>Harmonogram pracy pracownika został wyświetlony lub został wyświetlony komunikat o błędzie.</i>
Główny przepływ zdarzeń	1. Pracownik z poziomu swojego panelu zarządzania kliką „Harmonogram pracy”. 2. System wyświetla harmonogram pracy Pracownika.
Alternatywne przepływy zdarzeń	<i>3. Jeżeli w kroku 2 wystąpi błąd system wyświetli komunikat o błędzie.</i>

PU09: Scenariusz dla przypadku użycia: Dodanie pracownika	
Nazwa	<i>Dodanie pracownika</i>
Numer	<i>09</i>
Aktorzy	<i>Administrator, Pracownik</i>
Krótki opis	<i>Administrator dodaje pracownika do bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Pracownik zostaje dodany do bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie pracownikami”. 2. Administrator kliką „Dodaj”. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Dodaj”. 5. System sprawdza poprawność danych. 6. Pracownik zostaje dodany do bazy danych.
Alternatywne przepływy zdarzeń	<i>6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane Pracownik nie zostanie dodany do bazy danych.</i>

PU10: Scenariusz dla przypadku użycia: Edycja danych pracownika przez administratora	
Nazwa	<i>Edycja danych pracownika przez administratora</i>
Numer	<i>10</i>
Aktorzy	<i>Administrator, Pracownik</i>
Krótki opis	<i>Administrator edytuje dane pracownika z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Dane Pracownika zostały zmienione w bazie danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie pracownikami”. 2. Administrator kliką „Edycja” przy odpowiednim Pracowniku. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Zmień”. 5. System sprawdza poprawność danych. 6. Dane Pracownika zostają zmienione w bazie danych.
Alternatywne przepływy zdarzeń	<i>6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane, dane Pracownika nie zostaną zmienione w bazie danych.</i>

PU11: Scenariusz dla przypadku użycia: Usunięcie pracownika	
Nazwa	<i>Usunięcie pracownika</i>
Numer	11
Aktorzy	<i>Administrator, Pracownik</i>
Krótki opis	<i>Administrator usuwa pracownika z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Pracownik został usunięty z bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie pracownikami”. Administrator kliką „Edycja” przy odpowiednim Pracowniku. Administrator kliką „Usuń”. System usuwa Pracownika z bazy danych.
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> Jeżeli w kroku 4 wystąpi błąd Pracownik nie zostanie usunięty z bazy danych.

PU12: Scenariusz dla przypadku użycia: Dodanie lotniska	
Nazwa	<i>Dodanie lotniska</i>
Numer	12
Aktorzy	<i>Administrator</i>
Krótki opis	<i>Administrator dodaje lotnisko do bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Lotnisko zostaje dodane do bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie lotniskami”. Administrator kliką „Dodaj”. Administrator wypełnia formularz odpowiednimi danymi. Administrator kliką „Dodaj”. System sprawdza poprawność danych. Lotnisko zostaje dodane do bazy danych.
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> Jeżeli w kroku 3 Administrator poda niepoprawne dane lotnisko nie zostanie dodane do bazy danych.

PU13: Scenariusz dla przypadku użycia: Edycja danych lotniska	
Nazwa	<i>Edycja danych lotniska</i>

Numer	13
Aktorzy	<i>Administrator</i>
Krótki opis	<i>Administrator edytuje dane lotniska z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Dane lotniska zostały zmienione w bazie danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie lotniskami”. 2. Administrator kliką „Edycja” przy odpowiednim lotnisku. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Zmień”. 5. System sprawdza poprawność danych. 6. Dane lotniska zostają zmienione w bazie danych.
Alternatywne przepływy zdarzeń	<i>6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane, dane lotniska nie zostaną zmienione w bazie danych.</i>

PU14: Scenariusz dla przypadku użycia: Usunięcie lotniska

Nazwa	<i>Usunięcie lotniska</i>
Numer	14
Aktorzy	<i>Administrator</i>
Krótki opis	<i>Administrator usuwa lotnisko z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Lotnisko zostało usunięte z bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie lotniskami”. 2. Administrator kliką „Edycja” przy odpowiednim lotnisku. 3. Administartor kliką „Usuń”. 4. System usuwa lotnisko z bazy danych.
Alternatywne przepływy zdarzeń	<i>5. Jeżeli w kroku 4 wystąpi błąd lotnisko nie zostanie usunięte z bazy danych.</i>

PU15: Scenariusz dla przypadku użycia: Dodanie samolotu

Nazwa	<i>Dodanie samolotu</i>
Numer	15

Aktorzy	<i>Administrator</i>
Krótki opis	<i>Administrator dodaje samolot do bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Samolot zostaje dodany do bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie samolotami”. 2. Administrator kliką „Dodaj”. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Dodaj”. 5. System sprawdza poprawność danych. 6. Samolot zostaje dodany do bazy danych.
Alternatywne przepływy zdarzeń	<i>6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane samolot nie zostanie dodany do bazy danych.</i>

PU16: Scenariusz dla przypadku użycia: Edycja danych samolotu	
Nazwa	<i>Edycja danych samolotu</i>
Numer	<i>16</i>
Aktorzy	<i>Administrator</i>
Krótki opis	<i>Administrator edytuje dane samolotu z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Dane samolotu zostały zmienione w bazie danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie samolotami”. 2. Administrator kliką „Edycja” przy odpowiednim samolocie. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Zmień”. 5. System sprawdza poprawność danych. 6. Dane samolotu zostają zmienione w bazie danych.
Alternatywne przepływy zdarzeń	<i>6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane, dane samolotu nie zostaną zmienione w bazie danych.</i>

PU17: Scenariusz dla przypadku użycia: Usunięcie samolotu	
Nazwa	<i>Usunięcie samolotu</i>
Numer	<i>17</i>

Aktorzy	<i>Administrator</i>
Krótki opis	<i>Administrator usuwa samolot z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Samolot został usunięty z bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania klika „Zarządzanie samolotami”. 2. Administrator klika „Edycja” przy odpowiednim samolocie. 3. Administartor klika „Usuń”. 4. System usuwa samolot z bazy danych.
Alternatywne przepływy zdarzeń	<i>5. Jeżeli w kroku 4 wystąpi błąd samolot nie zostanie usunięty z bazy danych.</i>

PU18: Scenariusz dla przypadku użycia: Dodanie wypłaty pracownika	
Nazwa	<i>Dodanie wypłaty pracownika</i>
Numer	<i>18</i>
Aktorzy	<i>Administrator, Pracownik</i>
Krótki opis	<i>Administrator dodaje wypłatę pracownika do bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Wypłata pracownika zostaje dodana do bazy danych lub zostanie wyświetlony komunikat o błędzie, a stan systemu nie ulegnie zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania klika „Zarządzanie wypłatami”. 2. Administrator klika „Dodaj”. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor klika „Dodaj”. 5. System sprawdza poprawność danych. 6. Wypłata pracownika zostaje dodana do bazy danych.
Alternatywne przepływy zdarzeń	<i>6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane wypłata pracownika nie zostanie dodana do bazy danych.</i>

PU19: Scenariusz dla przypadku użycia: Edycja danych wypłaty pracownika	
Nazwa	<i>Edycja danych wypłaty pracownika</i>
Numer	<i>19</i>
Aktorzy	<i>Administrator, Pracownik</i>

Krótki opis	<i>Administrator edytuje dane wypłaty pracownika z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Dane wypłaty pracownika zostały zmienione w bazie danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie wypłatami”. 2. Administrator kliką „Edycja” przy odpowiedniej wypłacie. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Zmień”. 5. System sprawdza poprawność danych. 6. Dane wypłaty pracownika zostają zmienione w bazie danych.
Alternatywne przepływy zdarzeń	<i>6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane, dane wypłaty nie zostaną zmienione w bazie danych.</i>

PU20: Scenariusz dla przypadku użycia: Usunięcie wypłaty pracownika

Nazwa	<i>Usunięcie wypłaty pracownika</i>
Numer	<i>20</i>
Aktorzy	<i>Administrator, Pracownik</i>
Krótki opis	<i>Administrator usuwa wypłatę pracownika z bazy danych.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Wypłata pracownika została usunięta z bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie wypłatami”. 2. Administrator kliką „Edycja” przy odpowiedniej wypłacie. 3. Administartor kliką „Usuń”. 4. System usuwa wypłatę pracownika z bazy danych. 5. Jeżeli w kroku 4 wystąpi błąd wypłata pracownika nie zostanie usunięta z bazy danych.
Alternatywne przepływy zdarzeń	<i>5. Jeżeli w kroku 4 wystąpi błąd wypłata pracownika nie zostanie usunięta z bazy danych.</i>

PU21: Scenariusz dla przypadku użycia: Dodanie lotu do harmonogramu lotów

Nazwa	<i>Dodanie lotów do harmonogramu lotów</i>
Numer	<i>21</i>
Aktorzy	<i>Administrator</i>
Krótki opis	<i>Administrator dodaje lot do harmonogramu lotów.</i>

Warunki wstępne	Administrator jest zalogowany.
Warunki końcowe	<i>Lot zostaje dodana do harmonogramu lotów lub zostanie wyświetlony komunikat o błędzie, a stan systemu nie ulegnie zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie harmonogramem lotów”. 2. Administrator kliką „Dodaj”. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Dodaj”. 5. System sprawdza poprawność danych. 6. Lot zostaje dodany do harmonogramu lotów.
Alternatywne przepływy zdarzeń	6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane lot nie zostanie dodany do harmonogramu lotów.

PU22: Scenariusz dla przypadku użycia: Edycja danych lotu w harmonogramie lotów	
Nazwa	<i>Edycja danych lotu w harmonogramie lotów</i>
Numer	22
Aktorzy	Administrator
Krótki opis	<i>Administrator edytuje dane lotu z harmonogramu lotów.</i>
Warunki wstępne	Administrator jest zalogowany.
Warunki końcowe	<i>Dane lotu zostały zmienione w harmonogramie lotów lub zostanie wyświetlony komunikat o błędzie, a stan systemu nie ulegnie zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie harmonogramem lotów”. 2. Administrator kliką „Edycja” przy odpowiednim locie. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Zmień”. 5. System sprawdza poprawność danych. 6. Dane lotu pracownika zostają zmienione w harmonogramie lotów.
Alternatywne przepływy zdarzeń	6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane, dane lotu nie zostaną zmienione w harmonogramie lotów.

PU23: Scenariusz dla przypadku użycia: Usunięcie lotu z harmonogramu lotów	
Nazwa	<i>Usunięcie lotu z harmonogramu lotów</i>
Numer	23
Aktorzy	Administrator

Krótki opis	<i>Administrator usuwa lot z harmonogramu lotów.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Lot został usunięty z harmonogramu lotów lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie harmonogramem lotów”. 2. Administrator kliką „Edycja” przy odpowiednim locie. 3. Administartor kliką „Usuń”. 4. System usuwa lot z harmonogramu lotów.
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> 5. Jeżeli w kroku 4 wystąpi błąd lot nie zostanie usunięta z harmonogramu lotów.

PU24: Scenariusz dla przypadku użycia: Dodanie pracownika do lotu

Nazwa	<i>Dodanie pracownika do lotu</i>
Numer	<i>24</i>
Aktorzy	<i>Administrator, Pracownik</i>
Krótki opis	<i>Administrator dodaje pracownika do lotu do harmonogramu pracy.</i>
Warunki wstępne	<i>Administrator jest zalogowany.</i>
Warunki końcowe	<i>Pracownik zostaje dodany do lotu lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania kliką „Zarządzanie harmonogramem pracy”. 2. Administrator kliką „Dodaj”. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor kliką „Dodaj”. 5. System sprawdza poprawność danych. 6. Pracownik zostaje dodany do lotu.
Alternatywne przepływy zdarzeń	<ol style="list-style-type: none"> 6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane pracownik nie zostanie dodany do lotu.

PU25: Scenariusz dla przypadku użycia: Edycja dodania pracownika do lotu

Nazwa	<i>Edycja danych dodania pracownika do lotu</i>
Numer	<i>25</i>
Aktorzy	<i>Administrator, Pracownik</i>
Krótki opis	<i>Administrator edytuje dodanie pracownika do lotu z harmonogramu pracy.</i>

Warunki wstępne	Administrator jest zalogowany.
Warunki końcowe	Dodanie pracownika do lotu zostało zmienione w harmonogramie pracy lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania klika „Zarządzanie harmonogramem pracy”. 2. Administrator klika „Edycja” przy odpowiednim dodanym do lotu pracowniku. 3. Administrator wypełnia formularz odpowiednimi danymi. 4. Administartor klika „Zmień”. 5. System sprawdza poprawność danych. 6. Dodanie pracownika do lotu zostaje zmienione w harmonogramie pracy.
Alternatywne przepływy zdarzeń	6b) Jeżeli w kroku 3 Administrator poda niepoprawne dane, dodanie pracownika do lotu nie zostanie zmienione w harmonogramie pracy.

PU26: Scenariusz dla przypadku użycia: Usunięcie dodania pracownika do lotu	
Nazwa	Usunięcie dodania pracownika do lotu
Numer	26
Aktorzy	Administrator, Pracownik
Krótki opis	Administrator usuwa dodanie pracownika do lotu z harmonogramu pracy.
Warunki wstępne	Administrator jest zalogowany.
Warunki końcowe	Dodanie pracownika do lotu zostało usunięte z harmonogramu pracy lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Administrator z poziomu swojego panelu zarządzania klika „Zarządzanie harmonogramem pracy”. 2. Administrator klika „Edycja” przy odpowiednim dodaniu pracownika do lotu. 3. Administartor klika „Usuń”. 4. System usuwa dodanie pracownika do lotu z harmonogramu pracy.
Alternatywne przepływy zdarzeń	5. Jeżeli w kroku 4 wystąpi błąd dodanie pracownika do lotu nie zostanie usunięte z harmonogramu pracy.

PU27: Scenariusz dla przypadku użycia: Dodanie bagażu	
Nazwa	Dodanie bagażu

Numer	27
Aktorzy	Pasażer
Krótki opis	Pasażer dodaje bagaż do bazy danych.
Warunki wstępne	Pasażer jest zalogowany.
Warunki końcowe	Bagaż zostaje dodany do bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer z poziomu swojego panelu zarządzania kliką „Bagaże”. 2. Pasażer kliką „Dodaj”. 3. Pasażer wypełnia formularz odpowiednimi danymi. 4. Pasażer kliką „Dodaj”. 5. System sprawdza poprawność danych. 6. Bagaż zostaje dodany do bazy danych.
Alternatywne przepływy zdarzeń	6b) Jeżeli w kroku 3 Pasażer poda niepoprawne dane bagaż nie zostanie dodany do bazy danych.

PU28: Scenariusz dla przypadku użycia: Edycja danych bagażu

Nazwa	Edycja danych bagażu
Numer	28
Aktorzy	Pasażer
Krótki opis	Pasażer edytuje dane bagażu z bazy danych.
Warunki wstępne	Pasażer jest zalogowany.
Warunki końcowe	Dane bagażu zostały zmienione w bazie danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer z poziomu swojego panelu zarządzania kliką „Bagaże”. 2. Pasażer kliką „Edycja” przy odpowiednim bagażu. 3. Pasażer wypełnia formularz odpowiednimi danymi. 4. Pasażer kliką „Zmień”. 5. System sprawdza poprawność danych. 6. Dane bagażu zostają zmienione w bazie danych.
Alternatywne przepływy zdarzeń	6b) Jeżeli w kroku 3 Pasażer poda niepoprawne dane, dane bagażu nie zostaną zmienione w bazie danych.

PU29: Scenariusz dla przypadku użycia: Usunięcie bagażu

Nazwa	Usunięcie bagażu
--------------	------------------

Numer	29
Aktorzy	<i>Pasażer</i>
Krótki opis	<i>Pasażer usuwa bagaż z bazy danych.</i>
Warunki wstępne	<i>Pasażer jest zalogowany.</i>
Warunki końcowe	<i>Bagaż został usunięty z bazy danych lub został wyświetlony komunikat o błędzie, a stan systemu nie uległ zmianie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer z poziomu swojego panelu zarządzania kliką „Bagaże”. 2. Pasażer kliką „Edycja” przy odpowiednim bagażu. 3. Pasażer kliką „Usuń”. 4. System usuwa bagaż z bazy danych.
Alternatywne przepływy zdarzeń	<i>5. Jeżeli w kroku 4 wystąpi błąd bagaż nie zostanie usunięty z bazy danych.</i>

PU30: Scenariusz dla przypadku użycia: Złożenie reklamacji biletu lotniczego

Nazwa	<i>Złożenie reklamacji biletu lotniczego</i>
Numer	30
Aktorzy	<i>Pasażer</i>
Krótki opis	<i>Reklamacja zostaje aktywowana.</i>
Warunki wstępne	<i>Pasażer jest zalogowany oraz posiada zakupiony bilet lotniczy.</i>
Warunki końcowe	<i>Bilet lotniczy zostaje poddany reklamacji lub został wyświetlony komunikat o błędzie.</i>
Główny przepływ zdarzeń	<ol style="list-style-type: none"> 1. Pasażer z poziomu swojego panelu zarządzania kliką „Rezerwacje”. 2. Pasażer przy odpowiednim bilecie kliką „reklamacja”. 3. System zmienia status reklamacji bilet lotniczy na aktywny.
Alternatywne przepływy zdarzeń	<i>4. Jeżeli w kroku 3 wystąpi błąd bilet lotniczy nie zostanie poddany reklamacji.</i>

PU31: Scenariusz dla przypadku użycia: Anulowanie rezerwacji

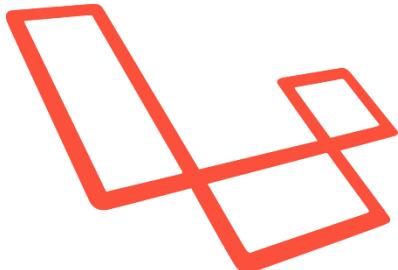
Nazwa	<i>Anulowanie rezerwacji</i>
Numer	31
Aktorzy	<i>Pasażer</i>
Krótki opis	<i>Rezerwacja zostaje anulowana.</i>
Warunki wstępne	<i>Pasażer jest zalogowany oraz posiada zakupiony bilet lotniczy.</i>

Warunki końcowe	<i>Bilet lotniczy zostaje anulowany lub został wyświetlony komunikat o błędzie.</i>
Główny przepływ zdarzeń	<i>1. Pasażer z poziomu swojego panelu zarządzania kliką „Rezerwacje”. 2. Pasażer przy odpowiednim bilecie kliką „anuluj”. 3. System anuluje bilet lotniczy. 4. Jeżeli w kroku 3 wystąpi błąd bilet lotniczy nie zostanie anulowany.</i>
Alternatywne przepływy zdarzeń	

9. Opis użytych technologii

- Laravel

Framework do aplikacji internetowych napisany w języku PHP bazujący na wzorcu architektonicznym Model-View-Controller.



Laravel udostępnia m.in. modułowy system budowania aplikacji z dedykowanym menedżerem zależności, różne sposoby dostępu do relacyjnych baz danych, narzędzia pomagające we wdrażaniu i utrzymaniu aplikacji oraz jego ukierunkowaniem na cukier syntaktyczny

- PHP

Interpretowany, skryptowy język programowania zaprojektowany do generowania stron internetowych i budowania aplikacji webowych w czasie rzeczywistym.

PHP jest najczęściej stosowany do tworzenia skryptów po stronie serwera WWW, ale może być on również używany do przetwarzania danych z poziomu wiersza poleceń, a nawet do pisania programów pracujących w trybie graficznym (np. za pomocą biblioteki GTK+, używając rozszerzenia PHP-GTK). Implementacja PHP wraz z serwerem WWW Apache oraz serwerem baz danych MySQL określana jest jako platforma AMP (w środowisku Linux – LAMP, w Windows – WAMP).



- HTTP (ang. Hypertext Transfer Protocol)

Protokół przesyłania dokumentów hipertekstowych to protokół sieci WWW (ang. World Wide Web). Obecną definicję HTTP stanowi RFC 2616. Za pomocą protokołu HTTP przesyła się żądania udostępnienia dokumentów WWW i informacje o

kliknięciu odnośnika oraz informacje z formularzy. Zadaniem stron WWW jest publikowanie informacji – natomiast protokół HTTP właśnie to umożliwia.

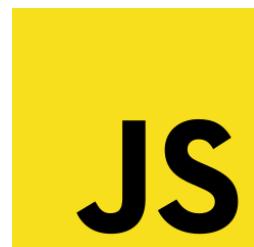


Protokół HTTP jest użyteczny, ponieważ udostępnia znormalizowany sposób komunikowania się komputerów ze sobą. Określa on formę żądań klienta (tj. np. przeglądarki www) dotyczących danych oraz formę odpowiedzi serwera na te żądania. Jest zaliczany do protokołów bezstanowych (ang.

stateless) z racji tego, że nie zachowuje żadnych informacji o poprzednich transakcjach z klientem (po zakończeniu transakcji wszystko "przepada"). Pozwala to znacznie zmniejszyć obciążenie serwera, jednak jest kłopotliwe w sytuacji, gdy np. trzeba zapamiętać konkretny stan dla użytkownika, który wcześniej łączył się już z serwerem. Najczęstszym rozwiązaniem tego problemu jest wprowadzenie mechanizmu ciasteczek. Inne podejście to m.in. sesje po stronie serwera, ukryte parametry (gdy aktualna strona zawiera formularz) oraz parametry umieszczone w URL-u (jak np. /index.php?userid=3). HTTP standardowo korzysta z portu nr 80 (TCP).

- JavaScript, w skrócie JS

Skryptowy oraz wieloparadygmatowy język programowania, stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych. Twórcą JavaScriptu jest Brendan Eich. W połowie lat 90. XX wieku organizacja ECMA wydała na podstawie JavaScriptu standard języka skryptowego o nazwie ECMAScript, aktualnie rozwijaniem tego standardu zajmuje się komisja TC39.



- SQL (ang. Structured Query Language)



Strukturalny oraz deklaratywny język zapytań. Jest to język dziedzinowy używany do tworzenia, modyfikowania relacyjnych baz danych oraz do umieszczania i pobierania danych z tych baz.

Decyzję o sposobie przechowywania i pobrania danych pozostawia się systemowi zarządzania bazą danych (DBMS).

- Kaskadowe arkusze stylów (ang. Cascading Style Sheets, w skrócie CSS)

Język służący do opisu formy prezentacji (wyświetlania) stron WWW. CSS został opracowany przez organizację W3C w 1996 r. jako potomek języka DSSSL

przeznaczony do używania w połączeniu z SGML-em. Pierwszy szkic CSS zaproponował w 1994 r. Håkon Wium Lie.

Arkusz stylów CSS to lista dyrektyw (tzw. reguł) ustalających w jaki sposób ma zostać wyświetlana przez przeglądarkę internetową zawartość wybranego elementu (lub elementów) (X)HTML lub XML. Można w ten sposób opisać wszystkie pojedyncze elementy odpowiedzialne za prezentację dokumentów internetowych, takie jak rodzina czcionek, kolor tekstu, marginesy, odstęp międzywierszowy lub nawet pozycja danego elementu względem innych elementów bądź okna przeglądarki. Wykorzystanie arkuszy stylów daje znacznie większe możliwości pozycjonowania elementów na stronie, niż oferuje sam (X)HTML.



- [phpMyAdmin](#)

Narzędzie służące do łatwego zarządzania bazą danych MySQL, napisane w języku PHP. Oprogramowanie wydawane jest na licencji GNU General Public License i umożliwia między innymi tworzenie/usuwanie baz danych, dodawanie/kasowanie relacji oraz edycję ich struktury i zawartości. Wszystkie operacje mogą być wykonywane z poziomu przeglądarki internetowej, w graficznym środowisku, bez konieczności pracy z domyślnym interfejsem tekstowym.



- [GitHub](#)

Hostingowy serwis internetowy przeznaczony do projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang.

Serwis działa od kwietnia 2008 roku. GitHub udostępnia darmowy hosting programów open source i prywatnych repozytoriów (część funkcji w ramach prywatnych repozytoriów jest płatna).



10. Implementacja

Implementacja została wykonana przy użyciu framework'a laravel, javascript, html, css, sql, php.

Przykłady implementacji języka sql:

```
$flights = DB::table('flights')
    ->join('aircraft','flights.aircraft_id', '=', 'aircraft.id')
    ->join('airports AS d','flights.airport_departure_id', '=', 'd.id')
    ->join('airports AS f','flights.airport_arrival_id', '=', 'f.id')
    ->select('flights.id','aircraft.model','d.name as
o','flights.departure_time','f.name as
p','flights.arrival_time','flights.created_at','flights.updated_at')
    ->get();
```

Złączenie kilku table i wyświetlenie wybranych kolumn.

```
$pracas = DB::table('flight_users')
    ->join('flights','flight_users.flight_id', '=', 'flights.id')
    ->join('users','flight_users.user_id', '=', 'users.id')
    -
    ->select('flight_users.id','flight_users.created_at','flight_users.updated_a
t','users.name','users.surname','users.description','flights.departure_time','
flights.arrival_time','flights.airport_departure_id','flights.airport_arrival_
id','flights.aircraft_id')
    ->paginate(3);
```

Złączenie kilku table i wyświetlenie wybranych kolumn.

```
$employees = User::where('role', '=', 3)->orderBy('description', 'ASC')->get();
```

Sortowanie alfabetyczne według wybranej kolumny pod warunkiem, że jest to pracownik.

```
$dt1 = \DateTime::createFromFormat('Y-m-d H:i:s', $flight->departure_time);
        $date1 = $dt1->format('Y-m-d\TH:i:s');
```

Zmiana formatu daty pobranej z bazy do innej zmiennej.

```
$pracas = DB::table('flight_users')
    ->join('flights','flight_users.flight_id', '=', 'flights.id')
    ->join('users','flight_users.user_id', '=', 'users.id')
    ->whereRaw("flight_users.user_id = '".Auth()->user()->id."")
    ->select(\DB::raw('
        DATE_FORMAT(flights.departure_time, "%Y.%m.%d %W") as data,
        flight_users.user_id as idu,
        DATE_FORMAT(flights.departure_time, "%H:%i") as od_time,
        DATE_FORMAT(flights.arrival_time, "%H:%i") as do_time,
        flights.airport_departure_id,
        flights.airport_arrival_id,
        flights.aircraft_id
    '))
    ->orderBy('flights.departure_time', 'ASC')->get();
```

Złączenie kilku table, wyświetlenie wybranych kolumn, sortowanie według wybranej kolumny pod warunkiem że jest to aktualnie zalogowany użytkownik.

```
$max = DB::table('users')->max('id');
```

Przypisanie do zmiennej maksymalnego id z tabeli users.

11. Bilans oraz estymacja czasowa

Znaczna większość założonych na początku funkcjonalności została przez nas zrealizowana. Poniżej przedstawienie zrealizowanych i niezrealizowanych wymagań wraz z estymacją czasową pracy nad projektem.

11.1. Zrealizowane wymagania

Przygotowanie dokumentacji – 10h

Wspólne dla Pasażera, Pracownika i Administratora:

- możliwość logowania – 3h (wliczona rozbudowa)
- możliwość wylogowania – 1h
- edycję danych osobowych w bazie danych (imię, nazwisko, email, hasło) -8h

Z punktu widzenia pasażera:

- możliwość rejestracji - 3h (wliczona rozbudowa)
- możliwość podglądu harmonogramu lotu (miejsce, data, godzina odlotu -> miejsce, data, godzina przylotu, model samolotu przewidzianego do lotu) – 8h
- możliwość kupna biletu lotniczego na dany lot (wybór klasy, miejsca siedzenia, widok ceny) – 10h

- możliwość zarządzania dokonanymi rezerwacjami (eksport do pliku w formacie PDF) - 10h

Z punktu widzenia pracownika aplikacja umożliwia:

- podgląd otrzymanych wypłat (kwota, okres od, okres do) – 4h
- wyświetlenie harmonogramu pracy (widok przpisania do danego lotu) – 8h

Z punktu widzenia administratora aplikacja umożliwia:

- zarządzanie pracownikami, na co składa się:
 - dodanie pracownika do bazy danych (imię, nazwisko, email, hasło, opis) – 5h
 - edycję danych osobowych pracownika w bazie danych (imię, nazwisko, email, hasło, opis) - 5h
 - usunięcie pracownika z bazy danych – 3h
- zarządzanie lotniskami, na co składa się:
 - dodanie lotniska do bazy danych (nazwa, kraj, miasto) – 5h
 - edycję danych lotniska w bazie danych (nazwa, kraj, miasto) -5h
 - usunięcie lotniska z bazy danych - 3h
- zarządzanie samolotami, na co składa się:
 - dodanie samolotu do bazy danych (model, liczba siedzeń w danej klasie) - 5h
 - edycję danych samolotu w bazie danych (model, liczba siedzeń w danej klasie) - 5h
 - usunięcie samolotu z bazy danych – 3h
- zarządzanie wypłatami, na co składa się:
 - dodanie wypłat pracowników do bazy danych (kwota, okres od, okres do, pracownik) – 5h
 - edycję danych wypłat pracowników w bazie danych (kwota, okres od, okres do, pracownik) – 5h
 - usunięcie wypłat pracowników z bazy danych – 3h
- zarządzanie harmonogramem lotu, na co składa się:
 - dodanie lotu do bazy danych (samolot, lotnisko odlotu, data, godzina odlotu -> lotnisko przylotu, data, godzina przylotu, cena biletu) – 5h
 - edycję danych lotu w bazie danych (lotnisko odlotu, data, godzina odlotu -> lotnisko przylotu, data, godzina przylotu, cena biletu) - 5h
 - usunięcie lotu z bazy danych – 3h
- zarządzanie harmonogramem pracy, na co składa się:
 - dodanie pracownika do danego lotu do bazy danych – 5h
 - edycję danych dodania pracownika do danego lotu w bazie danych – 5h
 - usunięcie pracownika z danego lotu z bazy danych – 3h

11.2. Niezrealizowane wymagania

Z punktu widzenia pasażera:

- możliwość zarządzania bagażem (dodania, edycji, usunięcia swojego bagażu z bazy danych) – 2h (niedokończone)
- złożenie reklamacji, anulowanie rezerwacji – 2h (niedokończone)

Łączny czas spędzony nad projektem: 147h

12. Repozytorium

Projekt został wyposażony w system kontroli wersji oraz został umieszczony na zdalnym repozytorium.

<https://github.com/matada2000/linia-lotnicza.git>

13. Podsumowanie oraz wnioski

Podczas realizacji projektu poznaliśmy nowy dla nas framework Laravel. Po dłuższym spędzeniu z nim czasu możemy powiedzieć, że jest przyjaznym środowiskiem do pracy nad tego typu projektami.

14. Bibliografia

<https://laravel.com/docs/9.x>

<https://pl.wikipedia.org/wiki/Laravel>

<https://pl.wikipedia.org/wiki/PHP>

https://pl.wikipedia.org/wiki/Hypertext_Transfer_Protocol

<https://pl.wikipedia.org/wiki/JavaScript>

<https://pl.wikipedia.org/wiki/SQL>

https://pl.wikipedia.org/wiki/Kaskadowe_arkusze_styl%C3%B3w

<https://pl.wikipedia.org/wiki/PhpMyAdmin>

<https://pl.wikipedia.org/wiki/GitHub>