

Bazy Danych

CoronAir Traffic Statistics

Time to fly together

Łukasz Chwistek ind. 243662 Kinga Długosz ind. 249002
Krzysztof Ragan ind. 249026

Etap I: 19 kwietnia 2020
Etap I: 17 maja 2020

Spis treści

1	Założenia ogólne	2
1.1	Opis zasobów ludzkich	2
1.2	Przepisy i strategia	2
1.3	Dane techniczne	2
2	Wymagania	3
2.1	Wymagania funkcjonalne	3
2.2	Wymagania нефункционалне	3
2.3	Scenariusz funkcjonowania	4
2.4	Diagram	4
3	Analiza przypadków użycia	6
3.1	Definicja aktorów	6
3.2	Przypadki użycia	6
4	Analiza ilości instancji dla encji oraz określenie zmienności zawartości	8
5	Realizacja bazy danych	9
5.1	Tworzenie i wstawianie danych do tablicy	9
5.2	Przykładowe operacje	13
5.3	Polityka bezpieczeństwa	14
6	Połączenie ze stroną	15
6.1	PHP	15
6.2	Widok strony	15
7	Bezpieczeństwo	17
7.1	Bezpieczeństwo połączenia	17
7.2	Bezpieczeństwo bazy danych	18
8	Możliwości rozwoju	18

1 Założenia ogólne

Projekt ma na celu zrealizowanie bazy danych wykorzystywanej do prowadzenia linii lotniczej. Firma zatrudnia wielu pracowników oraz posiada dziesiątki różnych modeli samolotów. Do sprawnego działania tak dużego przedsiębiorstwa potrzebna jest baza danych. Firma ma główną siedzibę we Wrocławiu, gdzie również wykonywanych jest większość lotów.

1.1 Opis zasobów ludzkich

Linie lotnicze zatrudniają pracowników typu:

- piloci
- stewardessy
- pracownicy techniczni
- pracownicy obsługi klienta

Pracownicy otrzymują wynagrodzenie adekwatne do zajmowanego stanowiska w przedziale od 2300 zł do 18000 zł

Piloci odpowiedzialni są za bezpieczny transport pasażerów z punktu A do punktu B oraz odczytanie wskazań aparatury pomiarowej i informacjach przekazywanych mu przez kontrolera lotu.

Stewardessy zajmują się obsługą pasażerów podczas lotu, podawaniem posiłków i napojów, utrzymaniem czystości podczas trwania przelotu oraz zapewnieniem bezpieczeństwa podróżnych oraz lotu.

Pracownicy techniczni odpowiedzialni są za konserwację i przeglądy samolotów. Wprowadzają informacje o kontroli i stanie technicznym poszczególnych samolotów. Dodatkowo zatwierdzają gotowość jednostki latającej do wystartowania.

Pracownicy obsługi klienta odpowiadają za konsultację z klientem tj. pomoc przy zakupie biletów, czy ich zwrocie. Ponadto administrują dane personalne klienta i weryfikują rezerwacje.

1.2 Przepisy i strategia

W samolotach pasażerskich znajduje się od 104 (Boeing B737-100) do 660 (Boeing 747) miejsc. Ich ceny uzależnione są od destynacji lotu. Każde połączenie ma wcześniej ustaloną cenę miejsc.

1.3 Dane techniczne

Zaprojektowana baza danych obejmuje encje:

- materialne,
- niematerialne,
- ożywione,

- nieożywione,

Większość atrybutów jest możliwa do modyfikacji z poziomu administratora. Linie lotnicze gromadzą dane o wyposażeniu, pracownikach i klientach.

2 Wymagania

2.1 Wymagania funkcjonalne

2.1.1 Zarządzanie rezerwacjami

Strona internetowa udostępni funkcje zarządzania rezerwacjami takimi jak: dodanie, unieważnienie biletu, zmiana informacji o bagażu klienta.

2.1.2 Zarządzanie danymi klientów

W bazie będzie możliwość przechowywania danych gości korzystających z usług linii lotniczych. Umożliwia to prowadzenie historii rezerwacji klientów, w bazie będą informacje umożliwiające sprawdzenie kto danego dnia korzystał z danego lotu, a także umożliwi wprowadzenie opcji usług stałego klienta.

2.1.3 Zarządzanie danymi pracowników

W bazie będzie możliwość dodawania, usuwania oraz edycji danych zatrudnionych pracowników.

2.1.4 Zarządzanie danymi wyposażenia firmy

W bazie możliwa będzie zmiana informacji o poszczególnych jednostkach lotniczych.

2.1.5 Zarządzanie danymi lotów

W bazie przechowywane będą dane na temat miejsca przylotów oraz odlotów, numerów lotów, maszyn lotniczych przypisanych do danych tras oraz rozkład pracowników dla danego lotu.

2.2 Wymagania нефunkcjonalne

2.2.1 Zarządzanie rezerwacjami i danymi klienta przez administratora

Administrator odpowiada za dane dotyczące rezerwacji oraz dane klientów, dodatkowo może usuwać rekordy z bazy.

2.2.2 Zarządzanie rezerwacjami i danymi klienta przez pracownika obsługi klienta

Pracownik obsługi klienta również odpowiada za rezerwacje oraz dane klientów, przy czym jest odpowiedzialny za zarządzanie oczekującymi rezerwacjami i obsługę bieżących zleceń. **Nie może** on jednak usuwać rekordów z bazy.

2.2.3 Zarządzanie danymi o stanie maszyn lotniczych przez obsługę techniczną

Obsługa techniczna posiada uprawnienia do aktualizacji danych o stanie technicznym oraz przeprowadzonej konserwacji samolotów.

2.2.4 Dostęp do rezerwacji i swoich danych personalnych przez klienta

Klient ma dostęp do utworzenia rezerwacji. Edycję rezerwacji może przeprowadzić jedynie poprzez kontakt z obsługą klienta. Klient ma uprawnienia edycji swoich danych personalnych.

2.2.5 Dostęp przez stronę WWW

Wprowadzanie oraz edycja danych za pośrednictwem strony WWW.

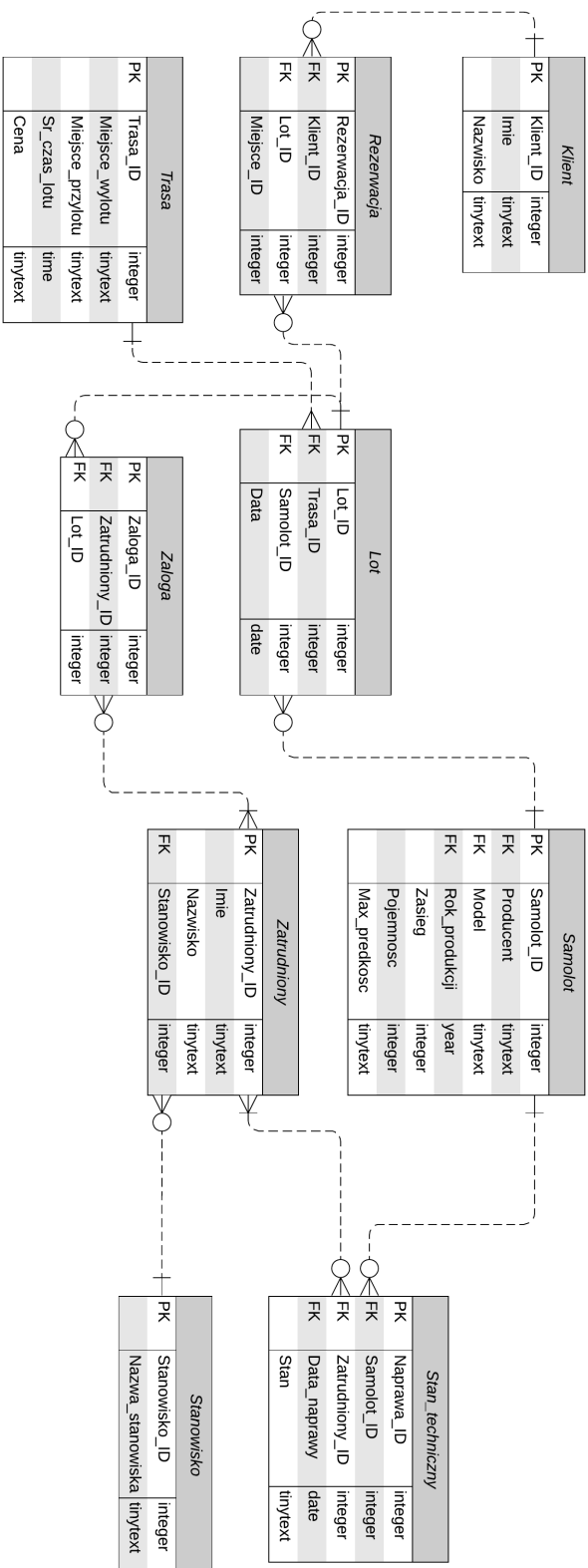
2.3 Scenariusz funkcjonowania

1. Klient przegląda wszystkie dostępne loty. Może podać swoje preferencje stosując odpowiednie filtry lub sortowanie i na tej podstawie wyszukuje w bazie wybrane przez siebie loty.
2. Klient podaje swoje dane i dokonuje zakupu konkretnego miejsca
3. Pracownik obsługi klienta weryfikuje i zatwierdza rezerwację
4. Klient znajduje się na pokładzie samolotu, wcześniej sprawdzonego przez obsługę techniczną

2.4 Diagram

CoronAir

Lukart | June 28, 2020



3 Analiza przypadków użycia

3.1 Definicja aktorów

3.1.1 Administrator

Administrator posiada możliwość zarządzania rezerwacjami oraz danymi klientów. Przydziela on również stanowiska pracownikom oraz może ich zatrudniać i zwalniać. Może on edytować oraz usuwać trasy lotów.

3.1.2 Pracownik obsługi klienta

Pracownik obsługi klienta ma kontakt z klientami. Ma dostęp do danych klientów oraz rezerwacji. Ma możliwość tworzenia rezerwacji dla klienta oraz odpowiada za ich akceptacje. Obsługuje bieżące zlecenia. Nie ma uprawnień żeby anulować rezerwacje.

3.1.3 Obsługa techniczna

Przeprowadzanie napraw i konserwacja maszyn, wprowadzanie informacji o stanie maszyny do bazy danych.

3.1.4 Klient

Klient jest podmiotem, który tworzy rezerwację jednocześnie podając swoje dane personalne, które może później edytować.

3.2 Przypadki użycia

3.2.1 Zatrudnienie pracownika

Cel : Zatrudnienie pracownika
Warunki początkowe : -
Warunki końcowe : Dodanie do bazy nowego pracownika.
Przebieg : Administrator uruchamia bazę, wprowadza dane osobowe pracownika, zatwierdza je, a następnie zamyka bazę.

3.2.2 Edycja pracownika

Cel : Edycja danych pracownika
Warunki początkowe : Pracownik jest zatrudniony
Warunki końcowe : Dane pracownika zostaną uaktualnione
Przebieg : Administrator uruchamia bazę, wyszukuje w niej pracownika, edytuje dane pracownika, akceptuje zmiany i zamyka bazę.

3.2.3 Zwolnienie pracownika

Cel : Zwolnienie pracownika
Warunki początkowe : Pracownik jest zatrudniony.
Warunki końcowe : Status pracownika w bazie zostaje zmieniony na: ZWOLNIONY.
Przebieg : Administrator uruchamia bazę, wyszukuje w niej pracownika, zmienia jego status zatrudnienia, akceptuje zmiany i zamyka bazę.

3.2.4 Dodanie maszyny

- Cel : Dodanie maszyny
- Warunki początkowe : -
- Warunki końcowe : Jednostka lotnicza zostaje dodana do bazy.
- Przebieg : Administrator uruchamia bazę, wprowadza dane maszyny, zatwierdza je, a następnie zamyka bazę.

3.2.5 Usunięcie maszyny

- Cel : Usunięcie maszyny
- Warunki początkowe : Maszyna znajduje się w bazie
- Warunki końcowe : Status maszyny w bazie zostaje zmieniony na WYCOFANA.
- Przebieg : Administrator uruchamia bazę, wyszukuje w niej maszynę, zmienia jej status na "wycofana", akceptuje zmiany i zamyka bazę.

3.2.6 Dodanie lotu

- Cel : Dodanie lotu
- Warunki początkowe : -
- Warunki końcowe : Lot figuruje w bazie i można zacząć rezerwację.
- Przebieg : Administrator uruchamia bazę, podaje dane lotu (destynacja, numer lotu, załoga, samolot[+pojemność]), zatwierdza zmiany i zamyka bazę.

3.2.7 Edycja lotu

- Cel : Edycja danych o locie
- Warunki początkowe : Lot istnieje w bazie
- Warunki końcowe : Dane lotu zostają uaktualnione.
- Przebieg : Administrator uruchamia bazę, Wyszukuje dany lot. Edytuje dane (destynacja, numer lotu, załoga, samolot[+pojemność]), zatwierdza zmiany i zamyka bazę.

3.2.8 Dodanie nowej trasy

- Cel : Dodanie nowej destynacji.
- Warunki początkowe : -
- Warunki końcowe : W bazie znajduje się nowa trasa.
- Przebieg : Administrator uruchamia bazę, dodaje nową pozycję połączenia.

3.2.9 Aktualizacja danych o stanie technicznym

- Cel : Aktualizacja danych technicznych maszyny
- Warunki początkowe : Maszyna znajduje się w bazie. Został dokonany przegląd techniczny maszyny.
- Warunki końcowe : W bazie znajdują się aktualne dane o stanie technicznym maszyny
- Przebieg : Pracownik obsługi technicznej, po wykonaniu przeglądu technicznego maszyny, uruchamia bazę. Wyszukuje w niej maszynę, zmienia dane na aktualne, akceptuje zmiany i zamyka bazę.

3.2.10 Stworzenie rezerwacji przez klienta

Cel	:	Stworzenie nowej rezerwacji.
Warunki początkowe	:	Istnieją połączenia lotnicze.
Warunki końcowe	:	Dane rezerwacji oraz dane klienta znajdują się w bazie.
Przebieg	:	Klient przegląda na stronie internetowej połączenia lotnicze, składając rezerwację wpierw podaje swoje dane personalne. Następnie (po skończeniu czynności przez klienta) w bazie tworzy się nowa pozycja rezerwacji ze statusem 'Oczekiwanie na potwierdzenie'.

3.2.11 Akceptacja rezerwacji

Cel	:	Potwierdzenie rezerwacji
Warunki początkowe	:	Istnieje pozycja nowej rezerwacji ze statusem 'Oczekiwanie na potwierdzenie'
Warunki końcowe	:	Rezerwacja ma status 'Zatwierdzona'
Przebieg	:	Pracownik obsługi klienta na bieżąco obsługuje bazę danych z pozycji strony internetowej, zmienia oczekujące rezerwacje na 'Zatwierdzono' / 'Odrzucono' i zatwierdza zmiany.

3.2.12 Stworzenie rezerwacji za pośrednictwem obsługi klienta

Cel	:	Stworzenie nowej rezerwacji.
Warunki początkowe	:	Istnieją połączenia lotnicze. Klient kontaktuje się z obsługą.
Warunki końcowe	:	Dane rezerwacji oraz dane klienta znajdują się w bazie.
Przebieg	:	Klient przegląda na stronie internetowej połączenia lotnicze, kontaktując się z obsługą klienta informuje o chęci rezerwacji wybranego lotu. Obsługa klienta informuje o dostępności. Przy pozytywnym rozpatrzeniu zlecenia obsługa wprowadza dane klienta poprzez stronę internetową. Po zapisaniu wniosku w bazie tworzy się nowa pozycja rezerwacji ze statusem 'Zatwierdzono'.

3.2.13 Usunięcie danych klienta z bazy

Cel	:	Usunięcie danych klienta z bazy na jego prośbę
Warunki początkowe	:	Dane klienta znajdują się w bazie. Prośba o ich usunięcie.
Warunki końcowe	:	Dane klienta zostają usunięte z bazy.
Przebieg	:	Klient wystosowuje prośbę o usunięcie danych. Administrator uruchamia bazę, wyszukuje w niej klienta, usuwa jego dane, zatwierdza zmiany i zamyka bazę.

4 Analiza ilości instancji dla encji oraz określenie zmienności zawartości

- klient
 - Baza przechowuje informacje o wszystkich pasażerach, którzy zakupili bilety przez okres dwóch lat. Szacuje się, że codziennie pojawia się 150 nowych klientów (nie są wliczani tu klienci, którzy zakupili bilet do dwóch lat wstecz) co daje ok. 55. tys. klientów rocznie. Jeden klient może posiadać więcej niż 1 rezerwację.

- rezerwacja
 - Liczba rezerwacji zależy od tego ile nowych klientów pojawi się w bazie oraz ile rezerwacji na lot zrobi dany klient.
- lot
 - Ilość lotów raczej pozostaje stała, ale może ulegać ewentualnym, nieznacznym zmianom.
- samolot
 - Liczba samolotów pozostaje stała. Zakłada się, że jeśli samolot będzie wymagał naprawy, to na jego miejsce wejdzie samolot zastępczy.
- stan techniczny
 - Ilość danych o stanie technicznym samolotu oraz zdolności do użytku będzie się zwiększała wraz z ilością wykonanych przeglądów oraz napraw. Zakłada się, że będzie ona regularnie zwiększana by dane były aktualne oraz by skutecznie planować logistykę.
- stanowisko
 - Liczba stanowisk nie ulega zmianom i wynosi ona 4.
- trasa
 - Ilość tras będzie sukcesywnie zwiększana wraz z rozwojem firmy.
- zatrudniony
 - Zakłada się, że ilość osób zatrudnionych jest stała. Jeśli pracownik zostanie zwolniony, na jego miejsce od razu pojawi się nowy.
- załoga
 - Załoga samolotu może się zmieniać z każdym lotem. Zakłada się, że baza będzie wzrastać wraz z ilością wykonanych lotów.

5 Realizacja bazy danych

5.1 Tworzenie i wstawianie danych do tablicy

5.1.1 Stanowiska

```

1 CREATE TABLE IF NOT EXISTS 'stanowisko' (
2   'Stanowisko_ID' int(11) NOT NULL AUTO_INCREMENT,
3   'Nazwa_stanowiska' text DEFAULT NULL,
4   PRIMARY KEY ('Stanowisko_ID')
5 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;
6
7 INSERT INTO 'stanowisko'
8   VALUES ('Administrator'),('Pilot'),('Steward'),('Pracownik techniczny'),('
   Obsługa klienta'),('Szef');

```

5.1.2 Pracownicy

```
1 CREATE TABLE IF NOT EXISTS 'zatrudniony' (  
2   'Zatrudniony_ID' int(11) NOT NULL AUTO_INCREMENT,  
3   'Imie' tinytext DEFAULT NULL,  
4   'Nazwisko' tinytext DEFAULT NULL,  
5   'Stanowisko_ID' int(11) DEFAULT NULL,  
6   'Wynagrodzenie' int(11) NULL DEFAULT NULL,  
7   PRIMARY KEY ('Zatrudniony_ID'),  
8   KEY 'stanowisko_fk6' ('Stanowisko_ID'),  
9   CONSTRAINT 'stanowisko_fk6'  
10  FOREIGN KEY ('Stanowisko_ID')  
11  REFERENCES 'stanowisko' ('Stanowisko_ID')  
12  ON DELETE NO ACTION  
13  ON UPDATE NO ACTION  
14 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;  
15  
16 INSERT INTO 'zatrudniony' ('Imie', 'Nazwisko', 'Stanowisko_ID', 'Wynagrodzenie')  
17 VALUES ('Krzysztof', 'Reagan', 1, 5000), ('Kinga', 'D Ćuga', 4, 4000), ('Ałkasz',  
    'Chwastek', 4, 4000), ('Rocky', 'Balboa', 2, 7000), ('Jesus', 'Bezklapek', 3, 3500), ('  
    Tomasz', 'Dejtrzy', 6, 10000), ('Joahim', 'Arenu', 5, 3000), ('Adam', 'Ma Ćysz', 2, 7000)  
    , ('Kaja', 'Bergovic', 3, 3500);
```

5.1.3 Zagłoga zamolotu

```
1 CREATE TABLE IF NOT EXISTS 'zaloga' (  
2   'Zaloga_ID' int(11) NOT NULL AUTO_INCREMENT,  
3   'Zatrudniony_ID' int(11) DEFAULT NULL,  
4   'Lot_ID' int(11) DEFAULT NULL,  
5   PRIMARY KEY ('Zaloga_ID'),  
6   KEY 'zatrudniony_fk5' ('Zatrudniony_ID'),  
7   KEY 'lot_fk6' ('Lot_ID'),  
8   CONSTRAINT 'lot_fk6'  
9   FOREIGN KEY ('Lot_ID')  
10  REFERENCES 'lot' ('Lot_ID')  
11  ON DELETE NO ACTION  
12  ON UPDATE NO ACTION,  
13  CONSTRAINT 'zatrudniony_fk5'  
14  FOREIGN KEY ('Zatrudniony_ID')  
15  REFERENCES 'zatrudniony' ('Zatrudniony_ID')  
16  ON DELETE NO ACTION  
17  ON UPDATE NO ACTION  
18 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;  
19  
20 INSERT INTO 'zaloga' ('Zatrudniony_ID', 'Lot_ID')  
21 VALUES (5, 12), (7, 12), (4, 12), (2, 12), (3, 16), (8, 16), (9, 16);
```

5.1.4 Samolot

```
1 CREATE TABLE IF NOT EXISTS 'samolot' (  
2   'Samolot_ID' int(11) NOT NULL AUTO_INCREMENT,  
3   'Producent' text DEFAULT NULL,  
4   'Model' text DEFAULT NULL,  
5   'Rok_produkcji' year(4) DEFAULT NULL,  
6   'Zasieg' int(11) DEFAULT NULL COMMENT '- km',  
7   'Pojemnosc' int(11) DEFAULT NULL COMMENT '- osob',  
8   'Max_predkosc' text DEFAULT '0 km/h' COMMENT 'km/h',  
9   PRIMARY KEY ('Samolot_ID')  
10 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;  
11  
12 INSERT INTO 'samolot' ('Producent', 'Model', 'Rok_produkcji', 'Zasieg', 'Pojemnosc'  
13   , 'Max_predkosc')  
VALUES ('Boeing', 'B737-100', 2011, 3440, 104, '876 km/h'), ('Boeing', '747-400',  
    2009, 14815, 660, '977 km/h');
```

5.1.5 Klienci

```
1 CREATE TABLE IF NOT EXISTS 'klient' (  
2   'Klient_ID' int(11) NOT NULL AUTO_INCREMENT,  
3   'Imie' tinytext DEFAULT NULL,  
4   'Nazwisko' tinytext DEFAULT NULL,  
5   PRIMARY KEY ('Klient_ID')  
6 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;  
7  
8 INSERT INTO 'klient' ('Imie', 'Nazwisko')  
9   VALUES ('Jozef', 'Zbagien'), ('Hatsune', 'Miku'), ('Andrzej', 'Zguba'), ('Janne', 'Ahonen'), ('Piotr', 'Nowak');
```

5.1.6 Trasy

```
1 CREATE TABLE IF NOT EXISTS 'trasa' (  
2   'Trasa_ID' int(11) NOT NULL AUTO_INCREMENT,  
3   'Miejsce_wylotu' text DEFAULT 'Wroclaw',  
4   'Miejsce_przylotu' text DEFAULT 'Bieszczady',  
5   'Sr_czas_lotu' time DEFAULT NULL,  
6   'Cena' int(11) DEFAULT 0,  
7   PRIMARY KEY ('Trasa_ID')  
8 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;  
9  
10 INSERT INTO 'trasa' ('Miejsce_wylotu', 'Miejsce_przylotu', 'Sr_czas_lotu', 'Cena')  
11   VALUES ('Wroclaw', 'Gdansk', '02:00:00', 169), ('Wroclaw', 'Rzeszow', '01:00:00', 289),  
   ('Wroclaw', 'Oslo', '03:03:42', 349), ('Gdansk', 'Wroclaw', '02:03:21', 289), ('Oslo',  
   'Wroclaw', '03:01:19', 349), ('Rzeszow', 'Wroclaw', '01:00:00', 169);
```

5.1.7 Loty

```
1 CREATE TABLE IF NOT EXISTS 'lot' (  
2   'Lot_ID' int(11) NOT NULL AUTO_INCREMENT,  
3   'Trasa_ID' int(11) NOT NULL,  
4   'Samolot_ID' int(11) NOT NULL,  
5   'Data' date DEFAULT NULL,  
6   PRIMARY KEY ('Lot_ID'),  
7   KEY 'trasa_fk1' ('Trasa_ID'),  
8   KEY 'samolot_fk1' ('Samolot_ID'),  
9   CONSTRAINT 'samolot_fk1'  
10     FOREIGN KEY ('Samolot_ID')  
11     REFERENCES 'samolot' ('Samolot_ID')  
12     ON DELETE NO ACTION  
13     ON UPDATE NO ACTION,  
14   CONSTRAINT 'trasa_fk1'  
15     FOREIGN KEY ('Trasa_ID')  
16     REFERENCES 'trasa' ('Trasa_ID')  
17     ON DELETE NO ACTION  
18     ON UPDATE NO ACTION  
19 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;  
20  
21 INSERT INTO 'lot' ('Trasa_ID', 'Samolot_ID', 'Data')  
22   VALUES (1,1, '2020-07-05'), (4,1, '2020-07-06'), (2,2, '2020-07-07'), (5,2, '2020-07-08'), (3,1, '2020-07-09'), (6,1, '2020-07-10');
```

5.1.8 Rezerwacje

```
1 CREATE TABLE IF NOT EXISTS 'rezerwacja' (  
2   'Rezerwacja_ID' int(11) NOT NULL AUTO_INCREMENT,  
3   'Klient_ID' int(11) DEFAULT NULL,  
4   'Lot_ID' int(11) DEFAULT NULL,  
5   'Miejsce_ID' int(11) DEFAULT NULL,  
6   PRIMARY KEY ('Rezerwacja_ID'),  
7   KEY 'klient_fk3' ('Klient_ID'),
```

```

8  KEY 'lot_fk3' ('Lot_ID'),
9  CONSTRAINT 'klient_fk3'
10 FOREIGN KEY ('Klient_ID')
11 REFERENCES 'klient' ('Klient_ID')
12 ON DELETE NO ACTION
13 ON UPDATE NO ACTION,
14 CONSTRAINT 'lot_fk3'
15 FOREIGN KEY ('Lot_ID')
16 REFERENCES 'lot' ('Lot_ID')
17 ON DELETE NO ACTION
18 ON UPDATE NO ACTION
19 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;
20
21 INSERT INTO 'rezerwacja' ('Klient_ID', 'Lot_ID', 'Miejsce_ID')
22 VALUES (1,12,1),(3,16,1),(1,19,2);

```

5.1.9 Stan techniczny

```


1  CREATE TABLE IF NOT EXISTS 'stan_techiczny' (
2  'Naprawa_ID' int(11) NOT NULL AUTO_INCREMENT,
3  'Samolot_ID' int(11) DEFAULT NULL,
4  'Zatrudniony_ID' int(11) DEFAULT NULL,
5  'Data_naprawy' date DEFAULT NULL,
6  'Stan' text DEFAULT 'W uzyciu',
7  PRIMARY KEY ('Naprawa_ID'),
8  KEY 'samolot_fk4' ('Samolot_ID'),
9  KEY 'zatrudniony_fk4' ('Zatrudniony_ID'),
10 CONSTRAINT 'samolot_fk4'
11 FOREIGN KEY ('Samolot_ID')
12 REFERENCES 'samolot' ('Samolot_ID')
13 ON DELETE NO ACTION
14 ON UPDATE NO ACTION,
15 CONSTRAINT 'zatrudniony_fk4'
16 FOREIGN KEY ('Zatrudniony_ID')
17 REFERENCES 'zatrudniony' ('Zatrudniony_ID')
18 ON DELETE NO ACTION
19 ON UPDATE NO ACTION
20 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;
21
22 INSERT INTO 'stan_techiczny' ('Naprawa_ID', 'Samolot_ID', 'Zatrudniony_ID', '
23 Data_naprawy', 'Stan')
24 VALUES (1,1,2,'2020-05-17','W uzyciu'),(2,2,2,'2020-05-17','W uzyciu'),(3,2,2,'
25 2020-05-13','Wycofany');

```

5.2 Przykładowe operacje

- Zapytanie zwracające numer identyfikacyjny samolotów, które są w stanie użytkowym, ich model oraz status.

```
1 SELECT
2     stan_techiczny.Samolot_ID, samolot.Model, stan_techiczny.Stan
3 FROM
4     stan_techiczny
5     LEFT JOIN samolot ON stan_techiczny.Samolot_ID=samolot.Samolot_ID
6 WHERE stan_techiczny.Stan='W użyciu';
```

 Samolot_ID	Model	Stan
1	B737-100	W użyciu
2	747-400	W użyciu

Rysunek 1: Wyświetlanie samolotów w stanie użytkowym

- Zliczanie ilości pasażerów w danych lotach z wyświetleniem miejsca docelowego lotu

```
1 SELECT
2     l.Samolot_ID AS NumerID_samolotu, COUNT(r.Miejsce_ID) AS Ilosc_pasazerow,
3     l.Trasa_ID AS Numer_trasy, t.Miejsce_przylotu AS Miejsce_docelowe
4 FROM
5     rezerwacja AS r, lot AS l, trasa AS t
6 WHERE
7     t.Trasa_ID=l.Trasa_ID AND l.Lot_ID=r.Lot_ID
8 GROUP BY
9     r.Lot_ID;
```

NumerID_samolotu	Ilosc_pasazerow	Numer_trasy	Miejsce_docelowe
1	3	1	Gdańsk
2	2	5	Wrocław

Rysunek 2: Wyświetlanie ilości pasażerów poszczególnych lotów

- Spis klientów w bazie

```
1 SELECT DISTINCT
2     klient.Klient_ID, klient.Imie AS Klient_Imie, klient.Nazwisko AS
3     Klient_Nazwisko
4 FROM
5     klient;
```

 Klient_ID	Klient_Imie	Klient_Nazwisko
1	Józef	Zbagien
2	Hatsune	Miku
3	Andrzej	Zguba
4	Janne	Ahonen
5	Piotr	Nowak

Rysunek 3: Spis klientów

- Wyświetlenie wszystkich dostępnych lotów na stronie głównej. Zawarcie informacji o trasie, dacie, czasie, cenie lotu oraz ilości wolnych miejsc.

```

1 SELECT
2     l.Trasa_ID AS Numer_trasy, t.Miejsce_wylotu AS Miejsce_wylotu, t.
3     Miejsce_przylotu AS Miejsce_docelowe, l.'Data' AS 'Data', t.Sr_czas_lotu,
4     t.Cena, COUNT(r.Miejsce_ID) AS Ilosc_pasazerow
5 FROM
6     rezerwacja AS r, lot AS l, trasa AS t
7 WHERE
8     t.Trasa_ID=l.Trasa_ID AND l.Lot_ID=r.Lot_ID
9 GROUP BY
10    r.Lot_ID;

```

Numer_trasy	Miejsce_wylotu	Miejsce_docelowe	Data	Sr_czas_lotu	Cena	Ilosc_pasazerow
1	Wrocław	Gdańsk	2020-07-05	02:00:00	169	3
5	Oslo	Wrocław	2020-07-08	03:01:19	349	2

Rysunek 4: Wyświetlanie samolotów w stanie użytkowym

5.3 Polityka bezpieczeństwa

Połączenie z bazą odbywa się poprzez sieć. Powoduje to narażenie danych na wiele niebezpieczeństw. Aby temu zapobiegać, docelowo, postawiona strona HTML będzie posiadać certyfikat SSL. Pozwoli to na zapewnienie większego bezpieczeństwa i ochronę przed działaniem osób trzecich.

Protokół SSL znajduje zastosowanie nawet w bankowości elektronicznej, co potwierdza jego skuteczność.

W bazie przechowywane będą dane klientów. W związku z ustawą o ochronie danych osobowych, każdy pracownik, mający dostęp do danych klientów będzie posiadał stosowne upoważnienie.

Aby zapewnić jeszcze większe bezpieczeństwo, dostęp do bazy będzie wymagał uwierzytelnienia. Dodatkowo wprowadzono ograniczenia, które wyszczególniono w punkcie 3.1.

6 Połączenie ze stroną

6.1 PHP

PHP jest skryptowym językiem programowania wykorzystywanym najczęściej do tworzenia stron internetowych. Po napotkaniu znacznika PHP, następuje analiza składniowa danych wejściowych z założeniem pewnej struktury tych danych i wyodrębnienie ich do postaci w której dane da się wykorzystać w oprogramowaniu. Skrypt wykonywany jest po stronie serwera, co oznacza, że jest niewidoczny dla użytkowników. Pozwala na m.in. dynamiczne generowanie strony, tworzenie i modyfikowanie plików na serwerze, ograniczanie dostępu do danych podstron na twojej stronie, szyfrowanie danych czy stosowanie ciasteczek. Kod PHP może być przeplatany z kodem HTML, jednak nie jest to dobra praktyka.

Najprostszym przykładem jego wykorzystania jest obsługa formularzy na stronach internetowych. PHP pozwala także na obsługę baz danych przy znajomości SQL.

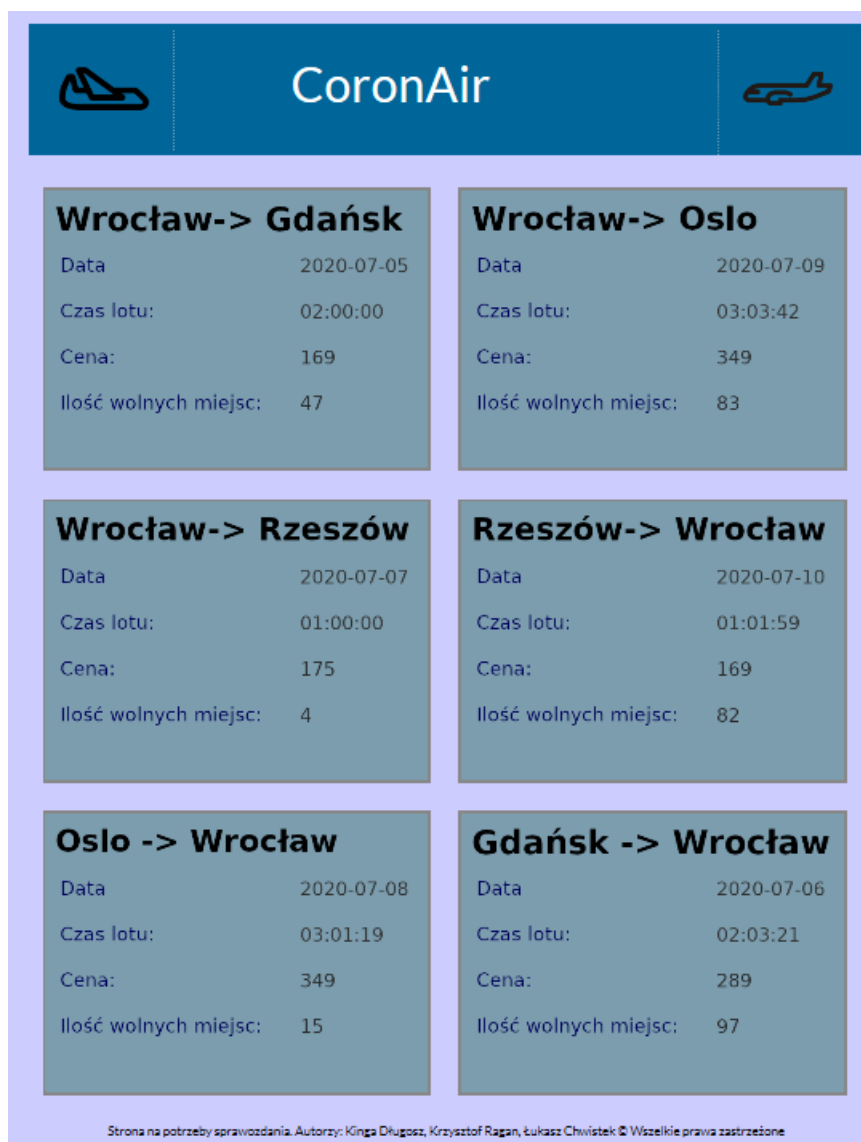
W PHP najczęściej tworzone są systemy zarządzania treścią, systemy forów dyskusyjnych, aplikacje pocztowe oraz klienty baz danych.

Kod strony połączono z bazą danych.

6.2 Widok strony

Na stronie głównej widnieją dostępne połączenia.

Klient po wybraniu konkretnej trasy widzi podsumowanie informacji o wybranym locie oraz miejsce na uzupełnienie swoich danych.



Rysunek 5: Zrzut ekranu strony startowej

The screenshot shows the CoronAir website interface. At the top is a blue header with the CoronAir logo and airplane icons. Below the header, the page is divided into two main sections. The left section, titled "SZCZEGÓŁY TWOJEGO LOTU:", displays flight details: "Miejsce wylotu: Wrocław", "Miejsce przylotu: Gdańsk", "Czas lotu: 02:00:00", "Cena: 289", and "Data: 2020-07-05". The right section, titled "Uzupełnij swoje dane:", contains input fields for "Imię:" and "Nazwisko:", followed by a "Prześlij" button. At the bottom of the page, a footer contains the text: "Strona na linii lotniczych CoronAir. Autorzy: Kinga Długosz, Krzysztof Ragan, Łukasz Chwistek © Wszelkie prawa zastrzeżone".

Rysunek 6: Zrzut ekranu podstrony, w której klient uzupełnia dane

Po uzupełnieniu danych, zostają one przesłane do weryfikacji.

The screenshot shows the CoronAir website interface after data submission. The header is the same as in Figure 6. The main content area is a grey box with the text: "Dziękujemy za skorzystanie z naszych usług." and "Twoja rezerwacja została przekazana do weryfikacji." At the bottom, the same footer text is present: "Strona na linii lotniczych CoronAir. Autorzy: Kinga Długosz, Krzysztof Ragan, Łukasz Chwistek © Wszelkie prawa zastrzeżone".

Rysunek 7: Zrzut ekranu podstrony po przesłaniu danych

7 Bezpieczeństwo

7.1 Bezpieczeństwo połączenia

Połączenie z bazą danych odbywa się za pomocą sieci przez stronę internetową. Taka aplikacja webowa niesie ze sobą wiele zagrożeń. Przez to że jest wrażliwa na działanie osób trzecich, aby prawidłowo funkcjonować powinna gwarantować odpowiedni poziom bezpieczeństwa, który możemy zapewnić dzięki systemowi logowania, szyfrowaniu danych na stronie oraz zabezpieczeniom przed wstrzykiwaniem zapytań SQL. Operacje wykonywalne na bazie danych będą możliwe dopiero po zalogowaniu na swoje konto użytkownika i stopień zmian będzie zależny od aktora, a jeśli dane

wprowadza klient, to muszą te zmiany zostać zweryfikowane przez odpowiedniego pracownika firmy.

7.2 Bezpieczeństwo bazy danych

W bazie danych przechowywane są dane wrażliwe firmy, pracowników oraz klientów. Aby je zabezpieczyć, wprowadzono ograniczony dostęp do bazy danych w postaci uwierzytelniania. Każda osoba mająca dostęp do bazy, będzie posiadała swoje konto z danymi do logowania, a także wprowadzone zostaną ograniczenia na wykonywane operacje przez poszczególnych aktorów zgodnie z ich definicją w analizie przypadków użycia. W przypadku próby wykonania niedozwolonej operacji zostanie ona anulowana a baza pozostanie bez zmian.

8 Możliwości rozwoju

Na ten moment realizacji projektu baza jest w stanie surowym. Zaawansowane operacje na bazie danych wymagają znajomości języka SQL oraz posiadania odpowiedniego oprogramowania by operować bezpośrednio na bazie danych (np. MySQLWorkbench, HeidiSQL). Istnieje jednak możliwość dopracowania interfejsu użytkownika do tego stopnia, by korzystanie z bazy dla przeciętnego użytkownika było wystarczająco intuicyjne.

Projekt umożliwi zrealizowanie postawionych założeń. Wszystkie elementy bazy są w pełni funkcjonalne. Użytkownicy mogą wykonywać operacje zgodnie z nadanymi im uprawnieniami.