

# Dokumentacja Techniczna Projektu System Rezerwacji Lotów

Julia Litawa

1 stycznia 2026

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Wymagania i Specyfikacja Projektu</b>	<b>2</b>
2.1	Spełnione wymagania techniczne . . . . .	2
<b>3</b>	<b>Struktura Projektu</b>	<b>2</b>
<b>4</b>	<b>Struktury Danych</b>	<b>2</b>
4.1	Struktura Lot . . . . .	3
4.2	Struktura Rezerwacja . . . . .	3
<b>5</b>	<b>Znaczące Zmienne i Stałe</b>	<b>3</b>
<b>6</b>	<b>Opis Funkcjonalności (API)</b>	<b>4</b>
6.1	Zarządzanie Lotami . . . . .	4
6.2	System Rezerwacji . . . . .	4
<b>7</b>	<b>Instrukcja Obsługi i Kompilacja</b>	<b>4</b>
7.1	Plik Makefile . . . . .	4
7.2	Kompilacja i Uruchomienie . . . . .	5
7.3	Logowanie Administratora . . . . .	5

# 1 Wstęp

Celem projektu jest stworzenie aplikacji konsolowej w języku C++, umożliwiającej zarządzanie systemem rezerwacji lotów. Aplikacja została zaprojektowana tak, aby umożliwić trwałość danych bez konieczności użycia zewnętrznych serwerów bazodanowych, co czyni ją lekkim i przenośnym rozwiązaniem.

## 2 Wymagania i Specyfikacja Projektu

Projekt został zrealizowany zgodnie z narzuconą specyfikacją techniczną oraz wymaganiami funkcjonalnymi.

### 2.1 Spełnione wymagania techniczne

- **Język programowania:** C++.
- **Objętość kodu:** Projekt przekracza wymagane minimum 100 wierszy kodu. Plik `main.cpp` zawiera 182 wiersze kodu, `rezerwacja.cpp` 358 wierszy kodu, a `rezerwacja.h` 34 wiersze, co daje łącznie 574 wiersze kodu.
- **Styl kodu:** Zgodnie z wytycznymi, w projekcie nie użyto dyrektywy `using namespace std;`. Zamiast tego korzystano z jawnych nazw np. `std::cout`, co jasno wskazuje pochodzenie elementów i zapobiega kolizjom, zmniejszając ryzyko trudnych do wykrycia błędów.
- **Kompilacja:** Przygotowano plik `Makefile` umożliwiający automatyczną kompilację z wymaganymi flagami: `-Wall -std=c++11`.
- **Baza danych:** Zrezygnowano z zewnętrznych serwerów bazodanowych (np. MySQL) na rzecz autorskiego rozwiązania opartego na plikach binarnych (`.bin`), co jest zgodne z dopuszczeniem rozwiązań typu SQLite lub własnych implementacji plikowych.

## 3 Struktura Projektu

Kod źródłowy został podzielony na moduły, co ułatwia zarządzanie projektem i jego kompilację.

- `main.cpp` – Główny plik programu, zawiera funkcję `main`, pętlę obsługi zdarzeń oraz interfejs użytkownika (menu tekstowe).
- `rezerwacja.h` – Plik nagłówkowy. Definiuje stałe (np. hasła, nazwy plików), struktury danych oraz deklaracje funkcji.
- `rezerwacja.cpp` – Plik źródłowy implementujący logikę biznesową, w tym operacje wejścia/wyjścia na plikach.
- `Makefile` – Skrypt automatyzujący proces budowania aplikacji.

## 4 Struktury Danych

Dane są utrwalane w plikach binarnych, a ich reprezentacja w pamięci opiera się na dwóch głównych strukturach.

## 4.1 Struktura Lot

Odzwierciedla rekord w pliku `bazadb.bin`.

Typ	Nazwa pola	Opis
int	id_lotu	Unikalny numer identyfikacyjny lotu. Klucz główny.
char[]	miasto_odlotu	Bufor znakowy dla miasta wylotu (maks. 30 znaków).
char[]	miasto_przylotu	Bufor znakowy dla miasta docelowego (maks. 30 znaków).
int	dostepne_miejsca	Liczba wolnych miejsc.
double	cena	Cena biletu.

## 4.2 Struktura Rezerwacja

Odzwierciedla rekord w pliku `rezerwacjedb.bin`.

Typ	Nazwa pola	Opis
int	id_rezerwacja	Numer rezerwacji (auto-inkrementacja).
int	id_lotu	Klucz obcy powiązany z ID lotu.
char[]	imie	Imię pasażera (maks. 20 znaków).
char[]	nazwisko	Nazwisko pasażera (maks. 30 znaków).

## 5 Znaczące Zmienne i Stałe

Nazwa Funkcja/Opis	Typ/Def	Plik
tryb Stan aplikacji: 0 (Klient), 1 (Administrator).	int	main.cpp
wybor Zmienna sterująca instrukcją switch w pętli głównej.	int	main.cpp
HASLO Hasło administratora: "admin123".	macro	main.cpp
BAZA_LOTY Ścieżka do pliku bazy lotów.	macro	rezerwacja.h
BAZA_REZERWACJE Ścieżka do pliku bazy rezerwacji.	macro	rezerwacja.h

## 6 Opis Funkcjonalności (API)

### 6.1 Zarządzanie Lotami

- **dodajLot:** Weryfikuje unikalność ID oraz poprawność danych (wartości dodatnie). Zapisuje nowy rekord na końcu pliku (*std::ios::app*).
- **usunLot:** Realizuje usuwanie logiczne poprzez przepisanie pliku. Tworzy plik tymczasowy, kopiuje do niego rekordy nieusuwane, a następnie podmienia plik bazy. Dodatkowo usuwa wszystkie rezerwacje powiązane z danym lotem.
- **edytujLot:** Modyfikuje istniejący rekord. Wykorzystuje pozycjonowanie w pliku *seekg/seekp* do odnalezienia odpowiedniego offsetu rekordu, a następnie nadpisuje dane. Zapewnia spójność danych poprzez sprawdzenie liczby istniejących rezerwacji.
- **wyswietlLoty:** Wyświetla sformatowaną tabelę wszystkich dostępnych połączeń.

### 6.2 System Rezerwacji

- **rezerwacjaBiletu:** Sprawdza dostępność miejsc. Jeśli miejsce jest dostępne: dekrementuje licznik miejsc w pliku lotów i dopisuje nową strukturę do pliku rezerwacji. W przypadku błędu zapisu przywraca poprzedni stan danych (mechanizm cofania zmian).
- **listaPasazerow:** Funkcja administracyjna. Iteruje przez plik rezerwacji i wyświetla pasażerów przypisanych do konkretnego *id\_lotu*. Jeżeli brak rezerwacji dla danego lotu, informuje o tym użytkownika

## 7 Instrukcja Obsługi i Kompilacja

### 7.1 Plik Makefile

Zgodnie z wymaganiami projekt zawiera plik **Makefile**, który definiuje reguły kompilacji.

Listing 1: Zawartość pliku Makefile

```
CC = g++
CFLAGS = -Wall -std=c++11
OBJ = main.o rezerwacja.o
TARGET = system_rezerwacji

all: $(TARGET)

$(TARGET): $(OBJ)
    $(CC) $(CFLAGS) -o $(TARGET) $(OBJ)

main.o: main.cpp rezerwacja.h
    $(CC) $(CFLAGS) -c main.cpp

rezerwacja.o: rezerwacja.cpp rezerwacja.h
    $(CC) $(CFLAGS) -c rezerwacja.cpp

clean:
```

```
rm -f *.o $(TARGET)
```

## 7.2 Kompilacja i Uruchomienie

Aby zbudować projekt przy użyciu narzędzia **make**, należy w terminalu (w katalogu projektu) wywołać polecenie:

```
make
```

Po pomyślnej kompilacji zostanie utworzony plik wykonywalny. Aby uruchomić program w terminalu systemowym, należy użyć polecenia:

```
./system_rezerwacji
```

## 7.3 Logowanie Administratora

Dostęp do funkcji edycji i usuwania lotów wymaga autoryzacji.

1. W menu głównym wybierz opcję 3. Przełącz na tryb **ADMIN**.
2. Wprowadź hasło zdefiniowane w stałej **HASLO**: **admin123**.