Dokumentacja Techniczna Projektu

System Rezerwacji Lotów

Autor	Piotr Majchrzak
Nazwa projektu	Majchrzak_WDP_Projekt
Język programowania	C++
Środowisko programistyczne	XCODE
Repozytorium GitHub	Link do repozytorium

18 grudnia 2025

Spis treści

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 (całkowita liczba linii wynosi ok. 300).
- **Styl kodu:** Zgodnie z wytycznymi, w projekcie nie użyto dyrektywy `using namespace std;`. Wszystkie funkcje biblioteczne wywoływane są w sposób bezpieczny dla przestrzeni nazw lub oparte o standardowe biblioteki C inkorporowane do C++.
- **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	Typ/Def	Plik	Funkcja/Opis
`tryb`	int	main.cpp	Stan aplikacji: 0 (Klient), 1 (Administrator).
`wybor`	int	main.cpp	Zmienna sterująca instrukcją `switch` w pętli głównej.
`HASLO`	macro	main.cpp	Hasło administratora: "admin123".
`BAZA_LOTY`	macro	rezerwacja.h	Ścieżka do pliku bazy lotów.
`BAZA_REZERWACJE`	macro	rezerwacja.h	Ścieżka do pliku bazy rezerwacji.

6 Opis Funkcjonalności (API)

6.1 Zarządzanie Lotami

- **dodajLot**: Weryfikuje unikalność ID oraz poprawność danych (wartości dodatnie). Zapisuje nowy rekord na końcu pliku (tryb `append`).

- **usunLot**: Realizuje usuwanie logiczne poprzez przepisanie pliku. Tworzy plik tymczasowy, kopiuje do niego rekordy nieusuwane, a następnie podmienia plik bazy.
- **edytujLot**: Modyfikuje istniejący rekord. Wykorzystuje funkcję `fseek` do odnalezienia odpowiedniego offsetu w pliku i nadpisuje dane.
- **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.
- **listaPasazerow**: Funkcja administracyjna. Iteruje przez plik rezerwacji i wyświetla pasażerów przypisanych do konkretnego `id_lotu`.

7 Instrukcja Obsługi i Kompilacja

7.1 Plik Makefile

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

```
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)
```

Listing 1: Zawartość pliku Makefile

7.2 Kompilacja i Uruchomienie (macOS)

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 macOS, 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`.