



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: Technologie internetowe i mobilne

Patryk Maziarz
Nr albumu studenta w67160

System Rezerwacji Miejsc w Samolotach

Prowadzący: mgr inż. Ewa Żesławska

Projekt Programowanie Obiektowe C#

Rzeszów 2023

Spis treści

Wstęp	4
1 Opis Projektu	5
1.1 Założenia Aplikacji	5
1.1.1 Opis założenia Projektu	5
1.1.2 Cele Projektu	5
1.1.3 Wymagania Funkcjonalne	5
1.1.4 Wymagania Niefunkcjonalne	6
2 Opis Bazy danych	7
2.1 Wstęp do Dokumentacji Bazy Danych RMwS	7
2.1.1 Cel Bazy danych	7
2.1.2 Zakres	7
2.1.3 Encje Bazy Danych	7
2.1.4 Relacje i Krotności	7
2.2 Sktuktura danych	8
2.2.1 Encje i Atrybuty	8
2.2.2 Związki i Krotności	8
2.2.3 Diagram ERD	9
3 Struktura projektu	10
3.1 Klasa DatabaseManager	10
3.1.1 Wstęp	10
3.1.2 Struktura	10
3.1.3 Opis	11
3.1.4 Diagram	11
3.2 Klasa Flight	11
3.2.1 Wstęp	11
3.2.2 Struktura	11
3.2.3 Opis	12
3.2.4 Diagram	12
3.3 Klasa Aircrafts	13
3.3.1 Wstęp	13
3.3.2 Struktura	13
3.3.3 Opis	13
3.3.4 Diagram	14
3.4 Klasa Reservations	14
3.4.1 Wstęp	14
3.4.2 Struktura	14
3.4.3 Opis	15
3.4.4 Diagram	15
3.5 Klasa Program	15

3.5.1	Wstęp	15
3.5.2	Struktura	16
3.5.3	Opis	16
3.5.4	Diagram	16
3.6	Techniczny Opis Projektu	17
4	Prezentacja projektu	18
4.1	Menu główne	18
4.2	Wyświetl Loty	18
4.3	Dokonaj Rezerwacji	19
4.4	Nasza Flota	21
5	Podsumowanie Projektu	22
5.1	Diagram Ganta	22
5.2	Repozytorium	23
5.3	Podsumowanie	23
	Bibliografia	25
	Spis rysunków	26

Wstęp

W obliczu szybko rozwijającego się rynku lotniczego i rosnących oczekiwań pasażerów, projekt "System Rezerwacji Miejsc w Samolotach" został zaprojektowany, by sprostać wyzwaniom nowoczesnej podróży. Celem systemu jest zapewnienie prostego, ale potężnego narzędzia, które umożliwia intuicyjną i efektywną rezerwację miejsc na pokładzie samolotów, oferując użytkownikom łatwy dostęp do informacji o dostępnych lotach i wolnych miejscach. Dzięki integracji z bazą danych linii lotniczych pasażerowie zyskują pewność, że otrzymują aktualne informacje, które pozwalają im na świadome podejmowanie decyzji dotyczących ich podróży. Podobnie jak w życiu codziennym, gdzie prostota i porządek obowiązków domowych przekładają się na komfort i dobre samopoczucie, tak system rezerwacji ma na celu przekształcenie potencjalnie stresującego procesu planowania podróży w przyjemne doświadczenie. Małe niedociągnięcia i zaniedbania mogą znacząco wpłynąć na jakość doświadczeń, zarówno w domu, jak i w pracy. Dlatego system został stworzony z myślą o dbałości o szczegóły, które są kluczowe dla wygodnych i bezproblemowych rezerwacji podróży dla naszych użytkowników. System rezerwacji jest odpowiedzią na potrzebę harmonijnego i efektywnego zarządzania podróżami. Każdy aspekt korzystania z naszego systemu - od wyboru miejsca, przez rezerwację - był równie przyjemny. W ten sposób, "System Rezerwacji Miejsc w Samolotach" nie tylko ułatwia podróżowanie, ale także przyczynia się do budowania pozytywnej atmosfery i wspiera dobre samopoczucie pasażerów - na ziemi i w powietrzu.

Rozdział 1

Opis Projektu

1.1 Założenia Aplikacji

1.1.1 Opis założenia Projektu

Projekt zakłada stworzenie aplikacji w języku C#, która umożliwi użytkownikom rezerwację miejsc w samolotach. Aplikacja powinna być intuicyjna w obsłudze, zapewniać szybki dostęp do informacji o dostępnych lotach i umożliwiać zarządzanie rezerwacjami. Baza danych zostanie zrealizowana za pomocą SQL.

1.1.2 Cele Projektu

- **Uproszczenie Procesu Rezerwacji:** Tworzenie łatwej i szybkiej metody rezerwacji miejsc w samolotach.
- **Integracja z Systemami Lotnisk:** Możliwość współpracy z różnymi systemami lotnisk i linii lotniczych.
- **Bezpieczeństwo i Prywatność:** Zapewnienie ochrony danych osobowych i finansowych użytkowników.
- **Skalowalność i Rozszerzalność:** Budowa aplikacji z myślą o łatwej rozbudowie i skalowaniu.

1.1.3 Wymagania Funkcjonalne

- **Rezerwacja Miejsca:** Użytkownicy mogą przeglądać dostępne loty, wybierać miejsca i dokonywać rezerwacji.
- **Zarządzanie Rezerwacjami:** Możliwość modyfikowania lub anulowania istniejących rezerwacji.
- **Wyszukiwanie Lotów:** Funkcja wyszukiwania pozwala filtrować loty według różnych kryteriów.
- **Autentykacja Użytkownika:** Rejestracja i logowanie do systemu.
- **Płatności:** Integracja bezpiecznego systemu płatności.
- **Pobieranie Danych:** Wykorzystanie interfejsów do pobierania danych z różnych źródeł.
- **Obsługa Wyjątków:** Zapewnienie stabilności aplikacji poprzez obsługę potencjalnych błędów i wyjątków.
- **Walidacja Danych:** Sprawdzanie poprawności wprowadzanych danych.

1.1.4 Wymagania Niefunkcjonalne

- **Bezpieczeństwo:** Szyfrowanie danych i zabezpieczenia przed nieautoryzowanym dostępem.
- **Skalowalność:** Możliwość obsługi rosnącej liczby użytkowników.
- **Wydajność:** Optymalizacja pod kątem szybkiego przetwarzania zapytań.
- **Dostępność:** Dostępność systemu 24/7 z minimalnym czasem przestoju.
- **Kompatybilność:** Wsparcie dla różnych urządzeń i systemów operacyjnych.
- **Użyteczność:** Intuicyjny interfejs użytkownika, zapewniający łatwość obsługi.

Rozdział 2

Opis Bazy danych

2.1 Wstęp do Dokumentacji Bazy Danych RMwS

2.1.1 Cel Bazy danych

Celem dokumentacji jest zapewnienie kompleksowego przeglądu struktury i funkcjonalności Bazy Danych Systemu Rezerwacji Lotniczych. Zawiera szczegółowe opisy encji, atrybutów, relacji oraz ograniczeń, które składają się na model danych.

2.1.2 Zakres

Dokumentacja koncentruje się na strukturze bazy danych, która wspiera działanie systemu rezerwacji lotniczych. Opisuje zależności pomiędzy tabelami, które przechowują dane o lotach, samolotach, miejscach, rezerwacjach i lotniskach. Przedstawia również logikę, która leży u podstaw działania systemu, oraz zasady integralności danych.

2.1.3 Encje Bazy Danych

System składa się z następujących głównych encji:

1. **Lotniska (Airports):** Przechowują informacje o lotniskach, w tym kody IATA, miasta oraz pełne nazwy lotnisk.
2. **Samoloty (Aircrafts):** Zawierają dane dotyczące modeli samolotów, ich wymiarów oraz całkowitej liczby miejsc.
3. **Loty (Flights):** Opisują poszczególne loty, w tym numery lotów, informacje o lotniskach odlotu/przylotu oraz czasy odlotu/przylotu.
4. **Miejsca (Seats):** Zawierają informacje o miejscach w samolotach, ich numeracji, klasie oraz dostępności.
5. **Rezerwacje (Reservations):** Rejestrują rezerwacje dokonane przez pasażerów, wskazując na konkretne loty i miejsca.

2.1.4 Relacje i Krotności

Każda z encji jest powiązana w sposób, który odzwierciedla realne procesy związane z planowaniem i zarządzaniem rezerwacjami lotniczymi. Dokumentacja szczegółowo opisuje relacje typu "jeden do wielu" i "wiele do jednego", które występują między encjami.

2.2 Sktuktura danych

2.2.1 Encje i Atrybuty

1. Aircrafts

- AircraftId (Primary Key): Unikalny identyfikator samolotu.
- AircraftModel: Model samolotu.
- TotalSeats: Łączna liczba miejsc w samolocie.
- Dimensions: Wymiary samolotu.

2. Seats

- SeatId (Primary Key): Unikalny identyfikator miejsca.
- AircraftId (Foreign Key): Identyfikator samolotu, do którego należy miejsce.
- SeatNumber: Numer miejsca.
- ClassType: Klasa miejsca (np. ekonomiczna, biznesowa).
- IsReserved: Wskaźnik, czy miejsce jest zarezerwowane.

3. Reservations

- ReservationId (Primary Key): Unikalny identyfikator rezerwacji.
- PassengerName: Imię i nazwisko pasażera.
- FlightNumber: Numer lotu.
- SeatId (Foreign Key): Identyfikator miejsca, które jest rezerwowane.

4. Flights

- FlightId (Primary Key): Unikalny identyfikator lotu.
- FlightNumber (Unique): Unikalny numer lotu.
- DepartureAirport (Foreign Key): Kod IATA lotniska odlotu.
- ArrivalAirport (Foreign Key): Kod IATA lotniska przylotu.
- DepartureTime: Czas odlotu.
- ArrivalTime: Czas przylotu.

5. Airports

- IATA_Code (Primary Key): Trzyliterowy kod IATA lotniska.
- City: Miasto, w którym lotnisko się znajduje.
- Airport_Name: Nazwa lotniska.

2.2.2 Związki i Krotności

1. Reservations ↔ Seats

- Każda rezerwacja jest powiązana z dokładnie jednym miejscem (1:1).
- Jedno miejsce może być zarezerwowane dla wielu rezerwacji (1:N), co prezentuje, że miejsca mogą być rezerwowane i zwalniane w czasie.

2. Seats ↔ Flights

- Miejsce odnosi się do konkretnego lotu poprzez referencję FlightNumberRef, co prezentuje związek typu wiele do jednego (N:1), gdzie wiele miejsc jest powiązanych z jednym lotem.

3. Flights ↔ Aircrafts

- Lot jest powiązany z samolotem przez klucz obcy AircraftId w tabeli miejsc (Seats), co prezentuje, że jeden samolot może obsługiwać wiele lotów (1:N), a każdy lot jest obsługiwany przez jeden samolot.

4. Flights ↔ Airports

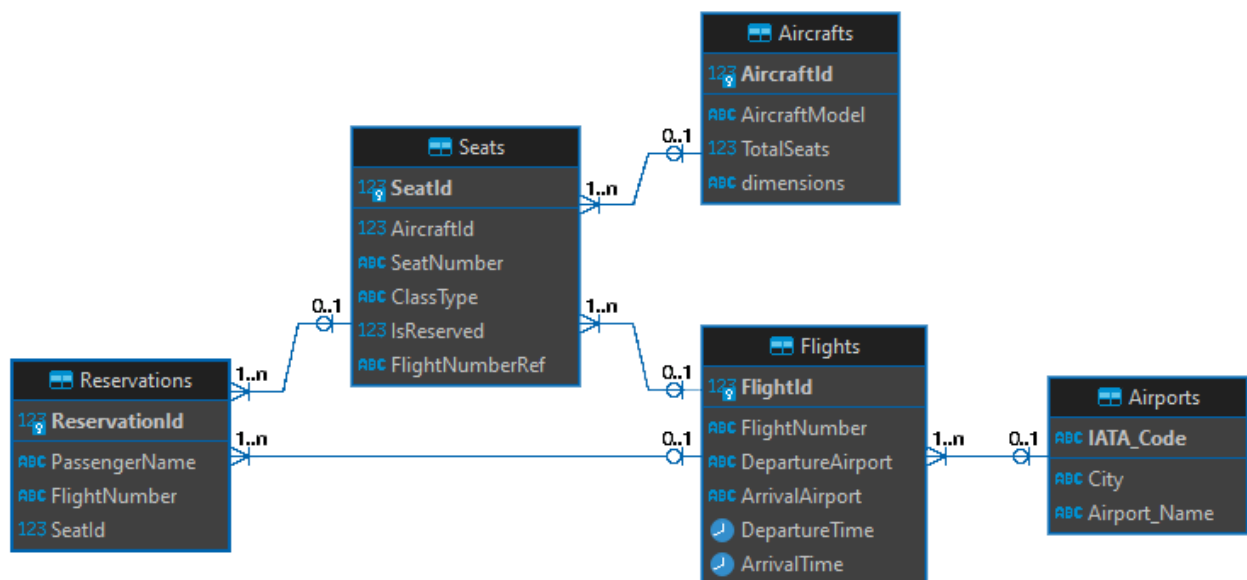
- Każdy lot ma określone lotnisko wylotu (DepartureAirport) i przylotu (ArrivalAirport). To oznacza, że każdy lot jest powiązany z dwoma lotniskami.

5. Aircrafts ↔ Seats

- Każdy samolot ma wiele miejsc (1:N), gdzie jeden samolot zawiera wiele miejsc, a każde miejsce jest przypisane do jednego samolotu.

2.2.3 Diagram ERD

Na przedstawionym diagramie encja-związek (ERD) wyświetlone są struktury danych dla Systemu Rezerwacji Miejs w Samolotach, z precyzyjnie zdefiniowanymi relacjami między Lotniskami, Samolotami, Lotami, Miejscami i Rezerwacjami. Diagram ukazuje zarówno strukturę bazodanową, jak i relacje "jeden do wielu" oraz "wiele do wielu", które umożliwiają śledzenie właściwości bazy danych.



Rysunek 2.1: Diagram ERD

Rozdział 3

Struktura projektu

3.1 Klasa DatabaseManager

3.1.1 Wstęp

Klasa DatabaseManager zdefiniowana w przestrzeni nazw Projekt jest narzędziem do zarządzania bazą danych w celu wykonywania zapytań SQL do bazy danych SQL Server. Poniżej znajduje się zarys klasy i jej członków.

3.1.2 Struktura

- **Namespace:** Projekt
- **Zależności:** System, System.Collections.Generic, System.Linq, System.Text, System.Threading.Tasks, System.Data.SqlClient

Klasa prywatna

- **connectionString (string):** Jest to prywatne pole tylko do odczytu, które przechowuje ciąg połączenia wymagany do połączenia z bazą danych.

Konstruktor

- **DatabaseManager(string connectionString):** Jest to konstruktor, który inicjalizuje DatabaseManager z podanym ciągiem połączenia.

Metody

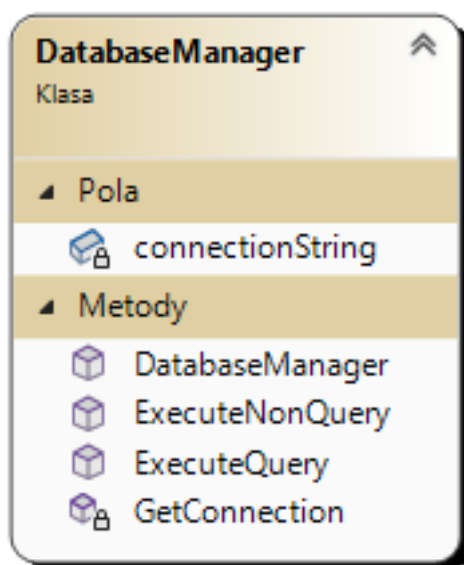
- **private SqlConnection GetConnection():** Ta prywatna metoda tworzy i zwraca nowy obiekt *SqlConnection* zainicjalizowany za pomocą *connectionString*.
- **public SqlDataReader ExecuteQuery(string query):** Ta metoda pobiera zapytanie SQL jako ciąg znaków, wykonuje je za pomocą polecenia *SqlCommand* i zwraca *SqlDataReader* zawierający wyniki zapytania. Otwiera połączenie, odczytuje z bazy danych i zapewnia zamknięcie połączenia.
- **public void ExecuteNonQuery(string query):** Ta metoda służy do wykonywania poleceń SQL, które nie zwracają żadnych danych (np. INSERT, UPDATE, DELETE). Otwiera połączenie, wykonuje polecenie niezwiązane z zapytaniem, a następnie zamyka połączenie.

3.1.3 Opis

Klasa `DatabaseManager` jest klasą narzędziową zaprojektowaną do obsługi zapytań do bazy danych. Zawiera logikę połączenia i zapewnia metody wykonywania zapytań i poleceń. Jej głównym celem jest uproszczenie operacji na bazie danych poprzez zapewnienie czystego i prostego kodu API dla reszty aplikacji. Klasa zapewnia, że wszystkie połączenia z bazą danych są prawidłowo otwierane i zamykane po zakończeniu operacji, co ma kluczowe znaczenie dla zarządzania zasobami i unikania wycieków pamięci w aplikacji.

3.1.4 Diagram

Na diagramie klas przedstawimy klasę `DatabaseManager` wraz z jej polami i ich metodami:



Rysunek 3.1: Diagram klasy `DatabaseManager`

3.2 Klasa `Flight`

3.2.1 Wstęp

Klasa `Flight` w przestrzeni nazw `Projekt` zawiera dane i operacje wyświetlające loty w systemie. Poniżej znajduje się zarys klasy:

3.2.2 Struktura

- **Namespace:** `Projekt`
- **Zależności:** `System`, `System.Collections.Generic`, `System.Linq`, `System.Text`, `System.Threading.Tasks`, `System.Data.SqlClient`

Konstruktor

- **`Flight(string flightNumber, string departureAirport, string arrivalAirport, DateTime departureTime, DateTime arrivalTime)`:** Inicjuje nowe wystąpienie klasy **`Flight`** ze szczegółami dotyczącymi numeru lotu, lotniska odlotu i przylotu oraz godziny.

Właściwości

- **FlightNumber** (publiczny ciąg znaków): Unikalny identyfikator lotu.
- **DepartureAirport** (publiczny ciąg znaków): Kod lotniska, z którego odlatuje lot.
- **ArrivalAirport** (publiczny ciąg znaków): Kod lotniska, do którego przylatuje lot.
- **DepartureTime** (publiczny DateTime): Planowany czas odlotu lotu.
- **ArrivalTime** (publiczny DateTime): Planowany czas przylotu lotu.

Metody

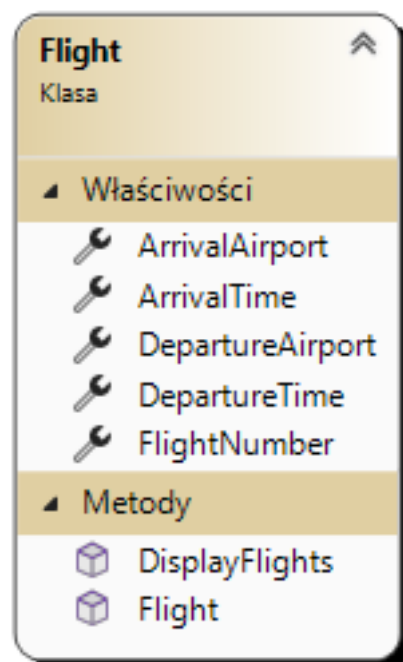
- **public static void DisplayFlights(DatabaseManager dbManager)**: Ta statyczna metoda jest odpowiedzialna za pobieranie i wyświetlanie informacji o lotach z bazy danych przy użyciu *DatabaseManager*. Uruchamia zapytanie SQL, aby wybrać wszystkie loty i iteruje po wynikach, drukując je na konsoli.

3.2.3 Opis

Klasa *Flight* to sposób reprezentacji lotów w systemie, która zawiera ważne informacje o harmonogramach lotów. Jej właściwości odpowiadają kolumnom w tabeli lotów w bazie danych. Klasa posiada statyczną metodę do wyświetlania informacji o lotach, która komunikuje się z bazą danych poprzez *DatabaseManager*. Taki projekt pozwala na oddzielenie zadań, gdzie klasa *Flight* skupia się na przedstawianiu danych o lotach, a *DatabaseManager* zajmuje się operacjami na bazie danych. Dzięki temu klasa *Flight* jest łatwa do zarządzania.

3.2.4 Diagram

Na diagramie klas przedstawimy klasę *Flight* wraz z jej polami i ich metodami:



Rysunek 3.2: Diagram klasy *Flight*

3.3 Klasa Aircrafts

3.3.1 Wstęp

Klasa Aircrafts w przestrzeni nazw Projekt została zaprojektowana w celu prezentacji informacji o samolotach w systemie.

3.3.2 Struktura

- **Namespace: Projekt**
- **Zależności:** W zakresie interakcji z bazą danych opiera się głównie na klasie **DatabaseManager** z tej samej przestrzeni nazw.

Konstruktor

- **Aircrafts (int aircraftId, string aircraftModel, int totalSeats, string dimensions):** Ten konstruktor inicjuje odwołanie do klasy *Aircrafts* z podanym identyfikatorem, modelem, całkowitą liczbą miejsc i wymiarami.

Właściwości

- **AircraftId** (publiczny int): Identyfikator samolotu.
- **AircraftModel** (publiczny string): Model samolotu.
- **TotalSeats** (public int): Całkowita liczba miejsc dostępnych w samolocie.
- **Dimensions** (publiczny string): Wymiary samolotu.

Metody

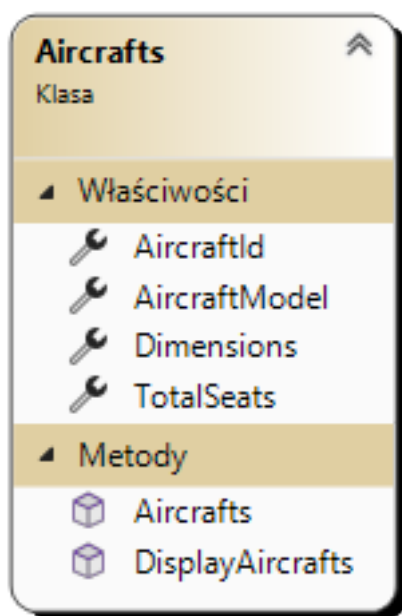
- **public static void DisplayAircrafts(DatabaseManager dbManager):** Ta statyczna metoda wykorzystuje *DatabaseManager* do pobierania i wyświetlania informacji o samolotach z bazy danych. Wykonuje instrukcję SQL SELECT i wyświetla szczegóły samolotu na konsoli.

3.3.3 Opis

Klasa *Aircrafts* służy do przedstawiania samolotów w systemie zarządzania lotnictwem. Zawiera informacje jak numer identyfikacyjny samolotu, model, liczbę miejsc oraz wymiary. Statyczna metoda *DisplayAircrafts* jest odpowiedzialna za połączenie z bazą danych w celu pobrania i wyświetlenia listy wszystkich samolotów. Ta separacja problemów pozwala klasie *Aircrafts* skupić się na reprezentacji danych samolotu, jednocześnie przekazując operacje bazy danych do *DatabaseManager*. Dzięki temu klasa Aircrafts jest łatwiejsza w utrzymaniu i dostosowaniu do zmian w systemie, takich jak modyfikacje atrybutów samolotu lub schematu bazy danych.

3.3.4 Diagram

Na diagramie klas przedstawimy klasę Aircrafts wraz z jej polami i ich metodami:



Rysunek 3.3: Diagram klasy Aircrafts

3.4 Klasa Reservations

3.4.1 Wstęp

Klasa Reservation w przestrzeni nazw Projekt służy do zarządzania rezerwacjami lotów.

3.4.2 Struktura

- **Namespace: Projekt**
- **Wykorzystuje klasę DatabaseManager do interakcji z bazą danych.**

Konstruktor

- **Reservation(string reservationId, string passengerName, string flightNumberRef, string seatId):** Ten konstruktor inicjuje odwołanie do klasy *Reservations* z szczegółami dotyczącymi identyfikatora rezerwacji, nazwiska pasażera, numeru lotu i identyfikatora miejsca.

Właściwości

- **AircraftId (publiczny int):** Identyfikator samolotu.
- **AircraftModel (publiczny string):** Model samolotu.
- **TotalSeats (public int):** Całkowita liczba miejsc dostępnych w samolocie.
- **Dimensions (publiczny string):** Wymiary samolotu.

Metody

- **public static void MakeReservation(DatabaseManager dbManager):** Ta statyczna metoda obsługuje proces dokonywania rezerwacji. Łączy się z *DatabaseManager* w celu pobrania szczegółów lotu, sprawdzenia dostępności lotów i miejsc, prosi użytkownika o wprowadzenie danych i aktualizuje bazę danych, aby uwzględnić nową rezerwację i status miejsca.

3.4.3 Opis

Klasa Reservation dokonuje rezerwacji w systemie zarządzania liniami lotniczymi. Zapewnia właściwości, które odzwierciedlają atrybuty rekordu rezerwacji w bazie danych, takie jak identyfikator rezerwacji, imię i nazwisko pasażera oraz powiązane miejsce i lot. Statyczna metoda MakeReservation to główna funkcjonalność tej klasy, która prowadzi użytkownika przez proces dokonywania rezerwacji.

Gromadzenie danych wejściowych: Metoda poprosi i zbiera niezbędne dane wejściowe od użytkownika, takie jak dane pasażera i preferencje dotyczące lotu.

Sprawdza dostępności lotów i miejsc: Zapytanie do bazy danych w celu sprawdzenia dostępnych lotów i miejsc na podstawie danych wprowadzonych przez użytkownika.

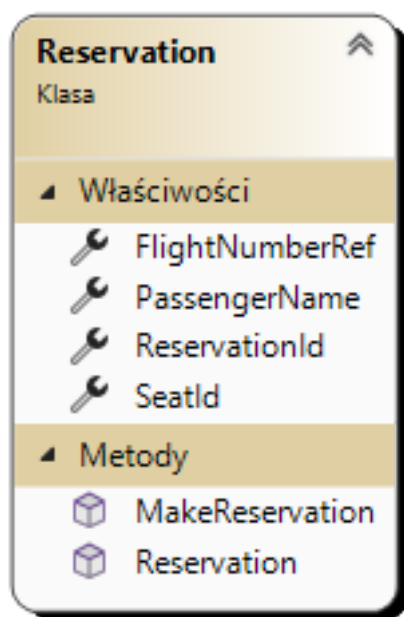
Potwierdzenie szczegółów rezerwacji: Metoda ta sprawdza dane wpisane przez użytkownika, upewniając się, że wybrany lot i miejsce są dostępne do wyboru.

Aktualizacja bazy danych: Na koniec metoda aktualizuje bazę danych, aby utworzyć nowy rekord rezerwacji i oznaczyć wybrane miejsce jako zarezerwowane.

Proces ten obejmuje kompleksowy przepływ pracy rezerwacji lotu od interfejsu użytkownika systemu aż do aktualizacji bazy danych, zapewniając spójność danych.

3.4.4 Diagram

Na diagramie klas przedstawimy klasę Reservations wraz z jej polami i ich metodami:



Rysunek 3.4: Diagram klasy Reservations

3.5 Klasa Program

3.5.1 Wstęp

Klasa Program w przestrzeni nazw Projekt jest głównym punktem wejścia aplikacji. Klasa ta, zdefiniowana w aplikacji konsolowej C#, obsługuje podstawową pętlę interaktywną, umożliwiając użytkownikowi wykonywanie zestawu predefiniowanych operacji.

3.5.2 Struktura

- **Namespace: Projekt**
- **Biblioteki systemowe i klasy DatabaseManager, Flight, Reservation, Aircrafts.**

Główna Metoda

- **Main(string[] args):** Punkt wejścia aplikacji. Inicjalizuje główną pętlę i obsługuje interakcje użytkownika.

Działanie

1. **Konfiguracja połączenia z bazą danych:** Inicjuje połączenie z bazą danych przy użyciu DatabaseManager.
2. **Interfejs użytkownika:** Wyświetla menu z opcjami przeglądania lotów, dokonywania rezerwacji, przeglądania floty samolotów lub wyjścia z programu.
3. **Wybór operacji:**
 - **Wyświetl loty:** Wywołuje Flight.DisplayFlights, aby wyświetlić listę wszystkich lotów.
 - **Make a Reservation:** Wywołuje Reservation.MakeReservation w celu zarezerwowania miejsca na lot.
 - **Wyświetl flotę samolotów:** Używa Aircrafts.DisplayAircrafts do wyświetlenia szczegółów samolotów.
 - **Wyjście:** Powoduje zamknięcie aplikacji.

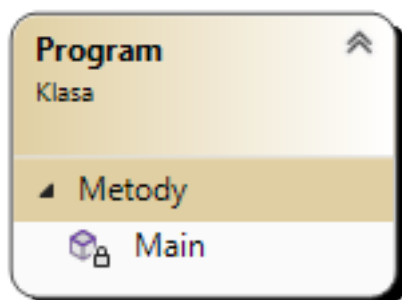
Każda opcja jest opakowana w blok try-catch, aby obsłużyć wszelkie potencjalne wyjątki, które mogą wystąpić podczas operacji na bazie danych, zapewniając, że aplikacja może z powodzeniem obsługiwać błędy i kontynuować działanie.

3.5.3 Opis

Klasa *Program* jest odpowiedzialna za koordynację aplikacji. Używa interfejsu konsolowego do interakcji z użytkownikiem, przetwarza jego wybory i przekazuje do odpowiednich klas (*Flight*, *Reservation*, *Aircrafts*). Solidna obsługa błędów jest zapewniona dzięki zastosowaniu bloków try-catch dla każdej operacji, przyczyniając się do stabilności aplikacji. Pętla ciągle działa, umożliwiając wykonywanie wielu operacji. Użytkownik może opuścić aplikację w dowolnym momencie, dzięki czemu jest ona przyjazna dla użytkownika i interaktywna.

3.5.4 Diagram

Na diagramie klas przedstawimy klasę Program wraz z jej polami i ich metodami:



Rysunek 3.5: Diagram klasy Program

3.6 Techniczny Opis Projektu

Język i Narzędzia:

- Język: C#
- Środowisko programistyczne: Visual Studio
- System zarządzania bazą danych: SQL Server

Minimalne Wymagania Sprzętowe:

- Procesor: 1 GHz lub szybszy
- Pamięć RAM: 512 MB
- Dysk twardy: 20 GB wolnego miejsca
- System operacyjny: Windows 7 lub nowszy

Zarządzanie danymi i Baza Danych:

- Dane są przechowywane i zarządzane przy użyciu SQL Server.
- Aplikacja wykorzystuje klasy ADO.NET do interakcji z bazą danych, umożliwiając operacje takie jak wprowadzanie, aktualizacja i wyświetlanie danych.
- Struktura bazy danych jest zaprojektowana do obsługi danych związanych z lotami, samolotami, rezerwacjami oraz klientami.

Rozdział 4

Prezentacja projektu

4.1 Menu główne

W rozdziale "Prezentacja projektu - Menu główne" zaprezentuje kluczowy element interakcji użytkownika z aplikacją. Menu główne jest pierwszym z interfejsów, z którym użytkownik ma do czynienia, a jego struktura i użyteczność są niezbędne dla płynnego przepływu pracy. Oto szczegółowy opis warstwy użytkowej aplikacji.

Menu główne składa się z 4 opcji:

```
1. Wyświetl loty
2. Dokonaj rezerwacji
3. Nasza flota
4. Wyjście
Wybierz opcję:
```

Rysunek 4.1: Menu główne aplikacji konsolowej

4.2 Wyświetl Loty

Wybierając tę opcję, użytkownik zobaczy listę dostępnych lotów z informacjami takimi jak numer lotu, lotnisko wylotu, lotnisko przylotu, czas wylotu i przylotu. Dane te będą pobierane z bazy danych i prezentowane w przejrzystej formie tabelarycznej.

```

1. Wyświetl loty
2. Dokonaj rezerwacji
3. Nasza flota
4. Wyjście
Wybierz opcję: 1
Lot 1: Nr. Lotu FR1234, Cel JFK, Odłot 15.01.2024 14:30:00, Przybycie 15.01.2024 20:00:00
Lot 2: Nr. Lotu FR5678, Cel LAX, Odłot 16.01.2024 10:00:00, Przybycie 16.01.2024 18:00:00
Lot 3: Nr. Lotu FR9101, Cel DXB, Odłot 17.01.2024 22:00:00, Przybycie 18.01.2024 06:00:00
Lot 4: Nr. Lotu FR1121, Cel HKG, Odłot 18.01.2024 11:00:00, Przybycie 18.01.2024 19:00:00
Lot 5: Nr. Lotu FR3141, Cel NRT, Odłot 19.01.2024 13:00:00, Przybycie 19.01.2024 21:00:00
Lot 6: Nr. Lotu FR2234, Cel BER, Odłot 20.01.2024 15:00:00, Przybycie 21.01.2024 07:00:00
Lot 7: Nr. Lotu FR3334, Cel LAX, Odłot 21.01.2024 16:00:00, Przybycie 21.01.2024 22:00:00
Lot 8: Nr. Lotu FR4434, Cel CDG, Odłot 22.01.2024 17:00:00, Przybycie 23.01.2024 08:00:00
Lot 9: Nr. Lotu FR5534, Cel LHR, Odłot 23.01.2024 14:00:00, Przybycie 23.01.2024 16:30:00
Lot 10: Nr. Lotu FR6634, Cel DXB, Odłot 24.01.2024 20:00:00, Przybycie 25.01.2024 05:00:00

1. Wyświetl loty
2. Dokonaj rezerwacji
3. Nasza flota
4. Wyjście
Wybierz opcję: _

```

Rysunek 4.2: Opcja - Wyświetl Loty

4.3 Dokonaj Rezerwacji

W tej sekcji użytkownik zostanie poproszony o podanie swoich danych osobowych oraz wybranie lotu i miejsca z dostępnych opcji. Po wprowadzeniu danych i potwierdzeniu wyboru, rezerwacja zostanie zapisana w bazie danych, a miejsce oznaczone jako zajęte.

```

1. Wyświetl loty
2. Dokonaj rezerwacji
3. Nasza flota
4. Wyjście
Wybierz opcję: 2
Nr. Lotu: FR1234, Odłot: Warszawa, Przylot: Nowy Jork, Czas odłotu: 15.01.2024 14:30:00, Czas przylotu: 15.01.2024 20:00:00
Nr. Lotu: FR5678, Odłot: Paryż, Przylot: Los Angeles, Czas odłotu: 16.01.2024 10:00:00, Czas przylotu: 16.01.2024 18:00:00
Nr. Lotu: FR9101, Odłot: Londyn, Przylot: Dubaj, Czas odłotu: 17.01.2024 22:00:00, Czas przylotu: 18.01.2024 06:00:00
Nr. Lotu: FR1121, Odłot: San Francisco, Przylot: Hongkong, Czas odłotu: 18.01.2024 11:00:00, Czas przylotu: 18.01.2024 19:00:00
Nr. Lotu: FR3141, Odłot: Amsterdam, Przylot: Tokio, Czas odłotu: 19.01.2024 13:00:00, Czas przylotu: 19.01.2024 21:00:00
Nr. Lotu: FR2234, Odłot: Singapur, Przylot: Berlin, Czas odłotu: 20.01.2024 15:00:00, Czas przylotu: 21.01.2024 07:00:00
Nr. Lotu: FR3334, Odłot: Sydney, Przylot: Los Angeles, Czas odłotu: 21.01.2024 16:00:00, Czas przylotu: 21.01.2024 22:00:00
Nr. Lotu: FR4434, Odłot: Toronto, Przylot: Paryż, Czas odłotu: 22.01.2024 17:00:00, Czas przylotu: 23.01.2024 08:00:00
Nr. Lotu: FR5534, Odłot: Istanbul, Przylot: Londyn, Czas odłotu: 23.01.2024 14:00:00, Czas przylotu: 23.01.2024 16:30:00
Nr. Lotu: FR6634, Odłot: Berlin, Przylot: Dubaj, Czas odłotu: 24.01.2024 20:00:00, Czas przylotu: 25.01.2024 05:00:00

Podaj Nr. Lotu do dokonania rezerwacji:
_

```

Rysunek 4.3: Opcja - Rezerwacja

Następnym krokiem w dokonaniu rezerwacji będzie wybór konkretnego lotu przez użytkownika. Po wyświetleniu listy dostępnych lotów, jak na załączonym obrazku, użytkownik musi podać numer lotu, który chce zarezerwować. W odpowiedzi na pytanie "Podaj Nr. Lotu do dokonania rezerwacji:", należy wprowadzić numer lotu widoczny na liście (np. FR1234) i nacisnąć Enter. System przejdzie do kolejnego etapu procesu rezerwacji.

```

1. Wyświetl loty
2. Dokonaj rezerwacji
3. Nasza flota
4. Wyjście
Wybierz opcję: 2
Nr. Lotu: FR1234, Odlot: Warszawa, Przylot: Nowy Jork, Czas odlotu: 15.01.2024 14:30:00, Czas przylotu: 15.01.2024 20:00:00
Nr. Lotu: FR5678, Odlot: Paryż, Przylot: Los Angeles, Czas odlotu: 16.01.2024 10:00:00, Czas przylotu: 16.01.2024 18:00:00
Nr. Lotu: FR9101, Odlot: Londyn, Przylot: Dubaj, Czas odlotu: 17.01.2024 22:00:00, Czas przylotu: 18.01.2024 06:00:00
Nr. Lotu: FR1121, Odlot: San Francisco, Przylot: Hongkong, Czas odlotu: 18.01.2024 11:00:00, Czas przylotu: 18.01.2024 19:00:00
Nr. Lotu: FR3141, Odlot: Amsterdam, Przylot: Tokio, Czas odlotu: 19.01.2024 13:00:00, Czas przylotu: 19.01.2024 21:00:00
Nr. Lotu: FR2234, Odlot: Singapore, Przylot: Berlin, Czas odlotu: 20.01.2024 15:00:00, Czas przylotu: 21.01.2024 07:00:00
Nr. Lotu: FR3334, Odlot: Sydney, Przylot: Los Angeles, Czas odlotu: 21.01.2024 16:00:00, Czas przylotu: 21.01.2024 22:00:00
Nr. Lotu: FR4434, Odlot: Toronto, Przylot: Paryż, Czas odlotu: 22.01.2024 17:00:00, Czas przylotu: 23.01.2024 08:00:00
Nr. Lotu: FR5534, Odlot: Istanbul, Przylot: Londyn, Czas odlotu: 23.01.2024 14:00:00, Czas przylotu: 23.01.2024 16:30:00
Nr. Lotu: FR6634, Odlot: Berlin, Przylot: Dubaj, Czas odlotu: 24.01.2024 20:00:00, Czas przylotu: 25.01.2024 05:00:00

Podaj Nr. Lotu do dokonania rezerwacji:
FR1234

Numer Siedzenia: 2A, Klasa: Ecoomy
Numer Siedzenia: 2B, Klasa: Ecoomy
Numer Siedzenia: 3A, Klasa: Ecoomy
Numer Siedzenia: 3B, Klasa: Ecoomy
Numer Siedzenia: 4A, Klasa: Ecoomy
Numer Siedzenia: 4B, Klasa: Ecoomy
Numer Siedzenia: 5A, Klasa: Ecoomy

Wybierz numer siedzenia:

```

Rysunek 4.4: Opcja - Rezerwacja - wybór siedzenia

Kolejnym krokiem w procesie rezerwacji będzie wybór przez użytkownika konkretnego miejsca z dostępnych opcji. System wyświetlił listę miejsc z ich numerami oraz klasą (np. Economy). Aby kontynuować, użytkownik powinien wpisać numer siedzenia, które chce zarezerwować (na przykład "3A") i zatwierdzić swój wybór, naciskając Enter.

```

2. Dokonaj rezerwacji
3. Nasza flota
4. Wyjście
Wybierz opcję: 2
Nr. Lotu: FR1234, Odlot: Warszawa, Przylot: Nowy Jork, Czas odlotu: 15.01.2024 14:30:00, Czas przylotu: 15.01.2024 20:00:00
Nr. Lotu: FR5678, Odlot: Paryż, Przylot: Los Angeles, Czas odlotu: 16.01.2024 10:00:00, Czas przylotu: 16.01.2024 18:00:00
Nr. Lotu: FR9101, Odlot: Londyn, Przylot: Dubaj, Czas odlotu: 17.01.2024 22:00:00, Czas przylotu: 18.01.2024 06:00:00
Nr. Lotu: FR1121, Odlot: San Francisco, Przylot: Hongkong, Czas odlotu: 18.01.2024 11:00:00, Czas przylotu: 18.01.2024 19:00:00
Nr. Lotu: FR3141, Odlot: Amsterdam, Przylot: Tokio, Czas odlotu: 19.01.2024 13:00:00, Czas przylotu: 19.01.2024 21:00:00
Nr. Lotu: FR2234, Odlot: Singapore, Przylot: Berlin, Czas odlotu: 20.01.2024 15:00:00, Czas przylotu: 21.01.2024 07:00:00
Nr. Lotu: FR3334, Odlot: Sydney, Przylot: Los Angeles, Czas odlotu: 21.01.2024 16:00:00, Czas przylotu: 21.01.2024 22:00:00
Nr. Lotu: FR4434, Odlot: Toronto, Przylot: Paryż, Czas odlotu: 22.01.2024 17:00:00, Czas przylotu: 23.01.2024 08:00:00
Nr. Lotu: FR5534, Odlot: Istanbul, Przylot: Londyn, Czas odlotu: 23.01.2024 14:00:00, Czas przylotu: 23.01.2024 16:30:00
Nr. Lotu: FR6634, Odlot: Berlin, Przylot: Dubaj, Czas odlotu: 24.01.2024 20:00:00, Czas przylotu: 25.01.2024 05:00:00

Podaj Nr. Lotu do dokonania rezerwacji:
FR1234

Numer Siedzenia: 2A, Klasa: Ecoomy
Numer Siedzenia: 2B, Klasa: Ecoomy
Numer Siedzenia: 3A, Klasa: Ecoomy
Numer Siedzenia: 3B, Klasa: Ecoomy
Numer Siedzenia: 4A, Klasa: Ecoomy
Numer Siedzenia: 4B, Klasa: Ecoomy
Numer Siedzenia: 5A, Klasa: Ecoomy

Wybierz numer siedzenia:
3A
Podaj swoje imię i nazwisko:

```

Rysunek 4.5: Opcja - Rezerwacja - Podanie imienia i nazwiska

Po wybraniu miejsca, następnym krokiem jest podanie danych osobowych przez użytkownika. System poprosi o wpisanie imienia i nazwiska. Użytkownik powinien wprowadzić te informacje w wyznaczonym miejscu i zatwierdzić je, naciskając klawisz Enter. Po podaniu danych osobowych, system przejdzie do etapu potwierdzenia rezerwacji.

```

Wybierz opcję: 2
Nr. Lotu: FR1234, Odlot: Warszawa, Przylot: Nowy Jork, Czas odlotu: 15.01.2024 14:30:00, Czas przylotu: 15.01.2024 20:00:00
Nr. Lotu: FR5678, Odlot: Paryż, Przylot: Los Angeles, Czas odlotu: 16.01.2024 10:00:00, Czas przylotu: 16.01.2024 18:00:00
Nr. Lotu: FR9101, Odlot: Londyn, Przylot: Dubaj, Czas odlotu: 17.01.2024 22:00:00, Czas przylotu: 18.01.2024 06:00:00
Nr. Lotu: FR1121, Odlot: San Francisco, Przylot: Hongkong, Czas odlotu: 18.01.2024 11:00:00, Czas przylotu: 18.01.2024 19:00:00
Nr. Lotu: FR3141, Odlot: Amsterdam, Przylot: Tokio, Czas odlotu: 19.01.2024 13:00:00, Czas przylotu: 19.01.2024 21:00:00
Nr. Lotu: FR2234, Odlot: Singapore, Przylot: Berlin, Czas odlotu: 20.01.2024 15:00:00, Czas przylotu: 21.01.2024 07:00:00
Nr. Lotu: FR3334, Odlot: Sydney, Przylot: Los Angeles, Czas odlotu: 21.01.2024 16:00:00, Czas przylotu: 21.01.2024 22:00:00
Nr. Lotu: FR4434, Odlot: Toronto, Przylot: Paryż, Czas odlotu: 22.01.2024 17:00:00, Czas przylotu: 23.01.2024 08:00:00
Nr. Lotu: FR5534, Odlot: Istanbul, Przylot: Londyn, Czas odlotu: 23.01.2024 14:00:00, Czas przylotu: 23.01.2024 16:30:00
Nr. Lotu: FR6634, Odlot: Berlin, Przylot: Dubaj, Czas odlotu: 24.01.2024 20:00:00, Czas przylotu: 25.01.2024 05:00:00

Podaj Nr. Lotu do dokonania rezerwacji:
FR1234

Numer Siedzenia: 2A, Klasa: Ecoomy
Numer Siedzenia: 2B, Klasa: Ecoomy
Numer Siedzenia: 3A, Klasa: Ecoomy
Numer Siedzenia: 3B, Klasa: Ecoomy
Numer Siedzenia: 4A, Klasa: Ecoomy
Numer Siedzenia: 4B, Klasa: Ecoomy
Numer Siedzenia: 5A, Klasa: Ecoomy

Wybierz numer siedzenia:
3A
Podaj swoje imię i nazwisko:
Patryk Maziarz

Rezerwacja została dokonana.

```

Rysunek 4.6: Opcja - Rezerwacja - Potwierdzenie rezerwacji

Po wprowadzeniu przez użytkownika swojego imienia i nazwiska, system przetwarza dane i dokonuje rezerwacji. Na ekranie pojawia się potwierdzenie: "Rezerwacja została dokonana". Oznacza to, że miejsce na wybranym locie zostało zarezerwowane na imię podane przez użytkownika, a proces rezerwacji został pomyślnie zakończony. Automatycznie cofnie do Menu głównego po 5 sekundach.

4.4 Nasza Flota

Opcja "Nasza flota" w menu głównym aplikacji pozwala użytkownikowi na przeglądanie informacji o samolotach należących do linii lotniczej. Po wybraniu tej opcji, użytkownikowi przedstawiana jest lista samolotów z kluczowymi informacjami, takimi jak model samolotu, liczba dostępnych miejsc, klasy podróży oraz ewentualne dodatkowe udogodnienia czy specyfikacje techniczne. Ta sekcja ma na celu zapewnienie przejrzystych i szczegółowych danych dotyczących floty.

```

Nr. 1, Model samolotu: Boeing 737, Ilość siedzien: 180,
Wymiary dla ciekawych: Długość: 39.5m, Rozpiętość: 43.4m, Wysokość: 12.5m, Prędkość: 840km/h
Nr. 2, Model samolotu: Airbus A320, Ilość siedzien: 160,
Wymiary dla ciekawych: Długość: 37.5m, Rozpiętość: 34.1m, Wysokość: 11m, Prędkość: 828km/h
Nr. 3, Model samolotu: Boeing 777, Ilość siedzien: 300,
Wymiary dla ciekawych: Długość: 73.8m, Rozpiętość: 64.8m, Wysokość: 18.5m, Prędkość: 920km/h
Nr. 4, Model samolotu: Airbus A380, Ilość siedzien: 500,
Wymiary dla ciekawych: Długość: 72.1m, Rozpiętość: 79m, Wysokość: 24m, Prędkość: 903km/h
Nr. 5, Model samolotu: Boeing 787 Dreamliner, Ilość siedzien: 240,
Wymiary dla ciekawych: Długość: 56.7m, Rozpiętość: 60m, Wysokość: 16.9m, Prędkość: 903km/h
Nr. 6, Model samolotu: Embraer E190, Ilość siedzien: 100,
Wymiary dla ciekawych: Długość: 36.2m, Rozpiętość: 28.7m, Wysokość: 10.6m, Prędkość: 830km/h
Nr. 7, Model samolotu: Boeing 747, Ilość siedzien: 410,
Wymiary dla ciekawych: Długość: 70.6m, Rozpiętość: 64.4m, Wysokość: 19.4m, Prędkość: 920km/h
Nr. 8, Model samolotu: Airbus A330, Ilość siedzien: 290,
Wymiary dla ciekawych: Długość: 58.8m, Rozpiętość: 60.3m, Wysokość: 17.4m, Prędkość: 880km/h
Nr. 9, Model samolotu: Bombardier CRJ900, Ilość siedzien: 90,
Wymiary dla ciekawych: Długość: 36.4m, Rozpiętość: 24.9m, Wysokość: 7.5m, Prędkość: 830km/h
Nr. 10, Model samolotu: Airbus A350, Ilość siedzien: 325,
Wymiary dla ciekawych: Długość: 66.8m, Rozpiętość: 64.8m, Wysokość: 17.1m, Prędkość: 910km/h

```

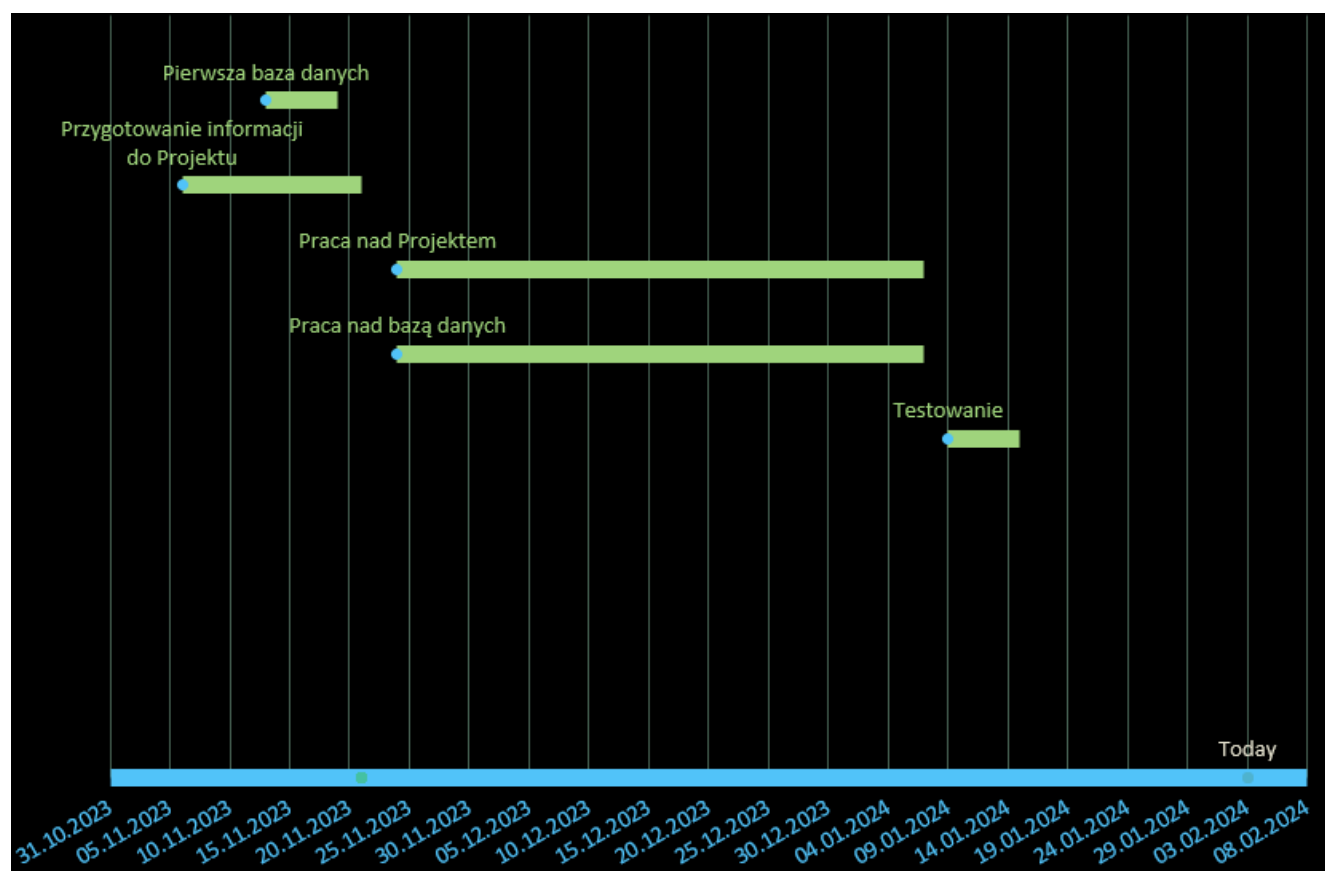
Rysunek 4.7: Opcja - Nasz flota

Rozdział 5

Podsumowanie Projektu

5.1 Diagram Ganta

Diagram Gantta jest kluczowym narzędziem w zarządzaniu projektami, który wizualizuje czas trwania poszczególnych etapów pracy nad projektem. Przedstawia on harmonogram i pozwala śledzić postęp w realizacji zadanych celów.



Rysunek 5.1: Diagram Gantta

1. **Pierwsza baza danych:** Ten etap, rozpoczynający się 05.11.2023, symbolizuje początek pracy nad bazą danych, która jest fundamentem dla całego projektu.
2. **Przygotowanie informacji do Projektu:** Pokazuje proces zbierania i organizowania niezbędnych informacji.
3. **Praca nad Projektem:** Jest to główna faza rozwoju projektu, rozciągająca się na większą część harmonogramu, co wskazuje na znaczący zakres prac.
4. **Praca nad bazą danych:** Przedstawia równoległe działania związane z rozwojem bazy danych, które mają kluczowe znaczenie dla funkcjonalności projektu.
5. **Testowanie:** Ostatnia faza projektu, zaplanowana po zakończeniu prac programistycznych, jest krótsza i skoncentrowana na weryfikacji działania projektu.

5.2 Repozytorium

Projekt korzysta z systemu kontroli wersji Git, co ułatwia zarządzanie zmianami. Repozytorium Git służy nam również jako archiwum wszystkich wersji projektu, dając pełną historię zmian.

Link Do GitHub

5.3 Podsumowanie

Projekt obejmuje rozbudowany system rezerwacji miejsc w samolotach, który zapewnia użytkownikom możliwość przeglądania dostępnych lotów, dokonywania rezerwacji miejsc, a także wyświetlania informacji o flotach samolotów. W ramach projektu zaimplementowano podstawowe komponenty aplikacji, w tym klasy reprezentujące główne encje systemu: *Aircraft*, *Flight*, *Reservation* oraz *DatabaseManager* do zarządzania połączeniami z bazą danych.

W trakcie realizacji projektu skoncentrowano się na zapewnieniu funkcjonalności takich jak wyświetlanie dostępnych lotów, rezerwacja miejsc z uwzględnieniem aktualnego stanu rezerwacji, oraz zarządzanie informacjami o samolotach. Dzięki zastosowaniu obiektowego podejścia do projektowania oprogramowania, udało się stworzyć skalowalną i modułową architekturę, która umożliwia łatwe rozszerzanie funkcjonalności systemu.

Podsumowując zrealizowane prace:

- Stworzono strukturę bazodanową oraz skrypty SQL do zarządzania danymi.
- Zaimplementowano klasy odpowiedzialne za logikę aplikacji.
- Zapewniono interakcję z użytkownikiem poprzez prosty interfejs konsolowy.
- Zastosowano najlepsze praktyki zarządzania połączeniami z bazą danych, w tym obsługę wyjątków i bezpieczne zamykanie połączeń.

Planowane dalsze prace rozwojowe:

1. **Rozbudowa interfejsu użytkownika:** Stworzenie bardziej zaawansowanego interfejsu użytkownika, możliwie w formie aplikacji desktopowej, dla lepszej interakcji i doświadczenia użytkownika.
2. **Wprowadzenie systemu autentykacji i autoryzacji:** Aby zwiększyć bezpieczeństwo i personalizację, planowane jest dodanie funkcjonalności logowania i zarządzania kontami użytkowników.
3. **Optymalizacja i skalowanie systemu:** Przeprowadzenie dokładnej analizy wydajności systemu, optymalizacja zapytań SQL oraz skalowanie rozwiązania w celu obsługi większej liczby jednocześnie użytkowników i zapytań.

Projekt jest na dobrej drodze do stania się kompleksowym rozwiązaniem, zapewniającym szerokie możliwości zarządzania rezerwacjami lotów, i będzie kontynuowany w celu dalszego rozwoju i doskonalenia oferowanych funkcji.

Bibliografia

- [1] Jacek Matulewski, *C#: lekcje programowania: praktyczna nauka programowania dla platform .NET i .NET Core*, Helion, Gliwice 2021.
- [2] Joseph Albahari, Eric Johannessen, *C# 8.0 w pigułce*, Helion, Gliwice 2021.
- [3] R. S. Miles, *C: zacznij programować!*, Helion, Gliwice 2020.
- [4] <https://dotnet.microsoft.com/en-us/learn/csharp>
- [5] <https://www.w3schools.com/cs/index.php>

Spis rysunków

2.1	Diagram ERD	9
3.1	Diagram klasy DatabaseManager	11
3.2	Diagram klasy Flight	12
3.3	Diagram klasy Aircrafts	14
3.4	Diagram klasy Reservations	15
3.5	Diagram klasy Program	16
4.1	Menu główne aplikacji konsolowej	18
4.2	Opcja - Wyświetl Loty	19
4.3	Opcja - Rezerwacja	19
4.4	Opcja - Rezerwacja - wybór siedzenia	20
4.5	Opcja - Rezerwacja - Podanie imienia i nazwiska	20
4.6	Opcja - Rezerwacja - Potwierdzenie rezerwacji	21
4.7	Opcja - Nasz flota	21
5.1	Diagram Grantta	22