**Dokumentacja projektu**

**Zaawansowane Techniki Programistyczne**



**MusicPlayer**

**Skład zespołu:**

* Magdalena Kalisz
* Adam Bajguz
* Michał Kierzkowski

**Data oddania:** 22.01.2019 r.

**Prowadzący:** mgr inż. Daniel Reska

# Opis projektu

**Odtwarzacz muzyczny na komputer PC z systemem Windows dla jednego użytkownika działający jako aplikacja Universal Windows Platform:**

* obsługa podstawowych formatów plików dźwiękowych (przynajmniej WAV i MP3);
* playlisty odtwarzanych plików z możliwością edycji (dodawanie, usuwanie, zmiana kolejności);
* zapis playlist i ich eksport (do formatu XML i/lub JSON);
* biblioteka utworów z prezentacją w widokach względem nazwy artysty, jego albumów i ścieżek w albumie;
* możliwość sortowania widoków biblioteki przynajmniej po nazwie, roku wydania, długości ścieżki;
* grupowanie utworów w albumy (jeden utwór może znajdować się w kilku albumach);
* każdy utwór ma mieć przypisany dokładnie jeden gatunek (lista gatunków jest ustalana przez użytkownika, tzn. użytkownik dodaje, usuwa i edytuje dostępne gatunki);
* każdy utwór ma mieć możliwość dodania własnej grafiki, jeśli jej nie ma wyświetlana jest domyślna grafika zapisana w aplikacji lub wyświetlana jest okładka albumu o ile istnieje;
* każdy utwór w albumie ma przypisany numer ścieżki;
* przypisywanie albumu do artysty (artysta może posiadać wiele albumów);
* przypisywanie artysty do zespołu (artysta może być tylko w jednym zespole);
* album, utwór oraz artysta mogą posiadać dokładnie jedno zdjęcie/okładkę;
* oprócz playlist powinna istnieć również kolejka odtwarzania zawierająca wszystkie utwory do odtworzenia;
* użytkownik ma mieć możliwość dodania albumu/playlisty lub pojedynczego utworu do kolejki odtwarzania.

# Wykorzystane technologie

|  |  |
| --- | --- |
|  |  |
|  | Znalezione obrazy dla zapytania c# |

# Diagramy klas

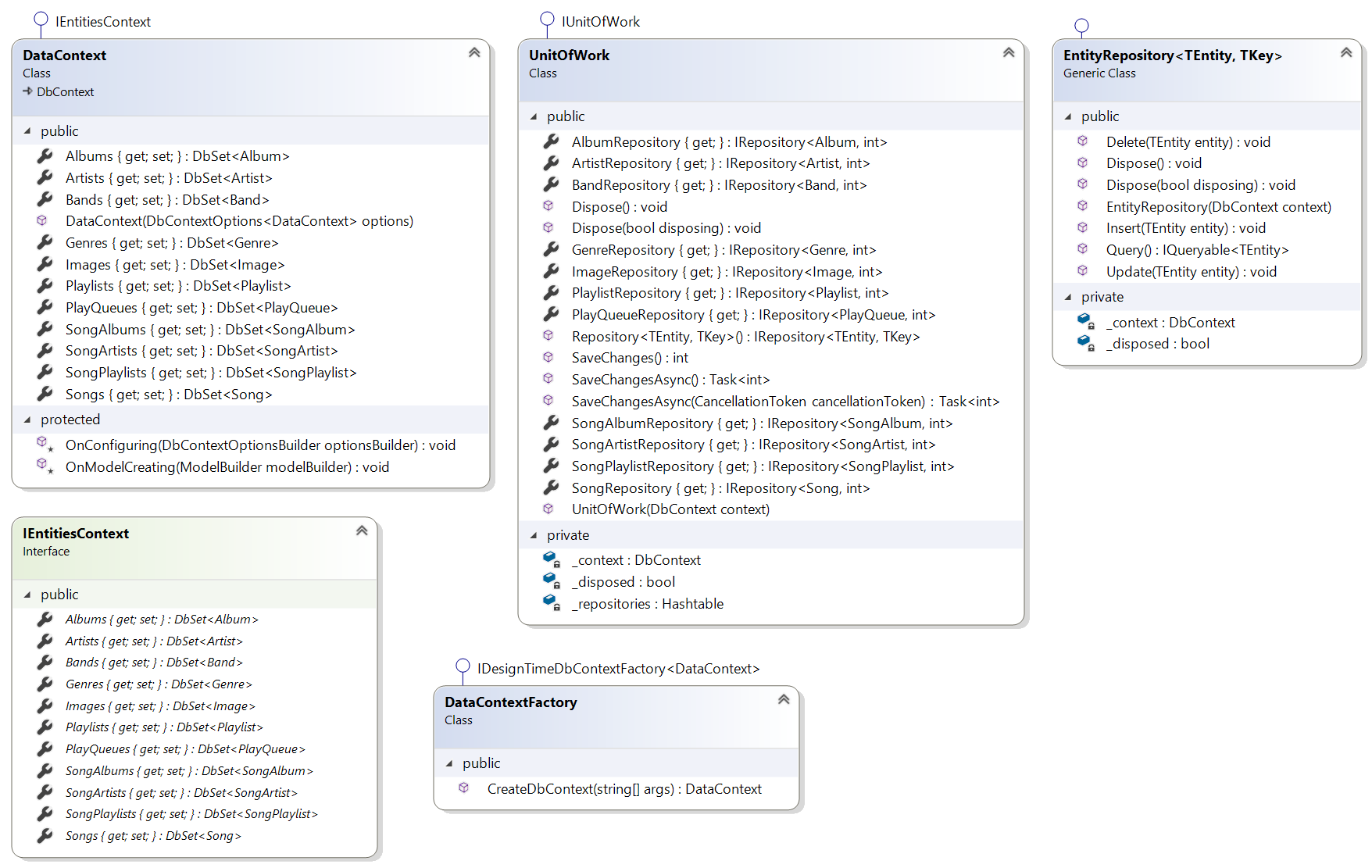
W skład kodu aplikacji (rozwiązania – solution) wchodzą nastęujące projekty:

* **MusicPlayer.Core** – projekt tworzący bibliotekę MusicPlayer.Core.dll
* **MusicPlayer.Data** – projekt tworzący bibliotekę MusicPlayer.Data.dll
* **MusicPlayer.Migrations.Startup** – projekt uruchamiania migracji
* **MusicPlayer.UWP** – projekt tworzący aplikację MusicPlayer.UWP.exe

## MusicPlayer.Core



## MusicPlayer.Data



## MusicPlayer.UWP

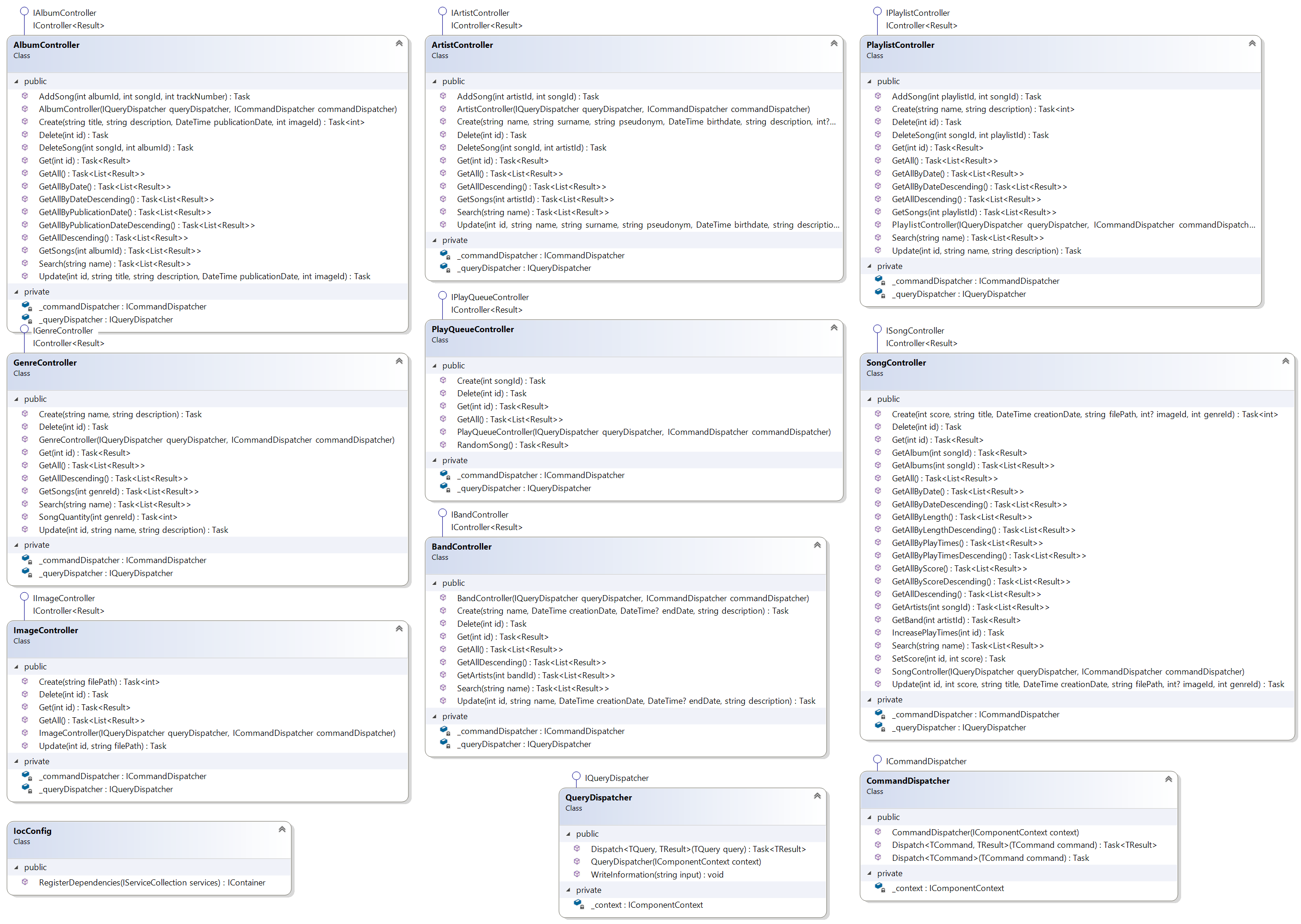
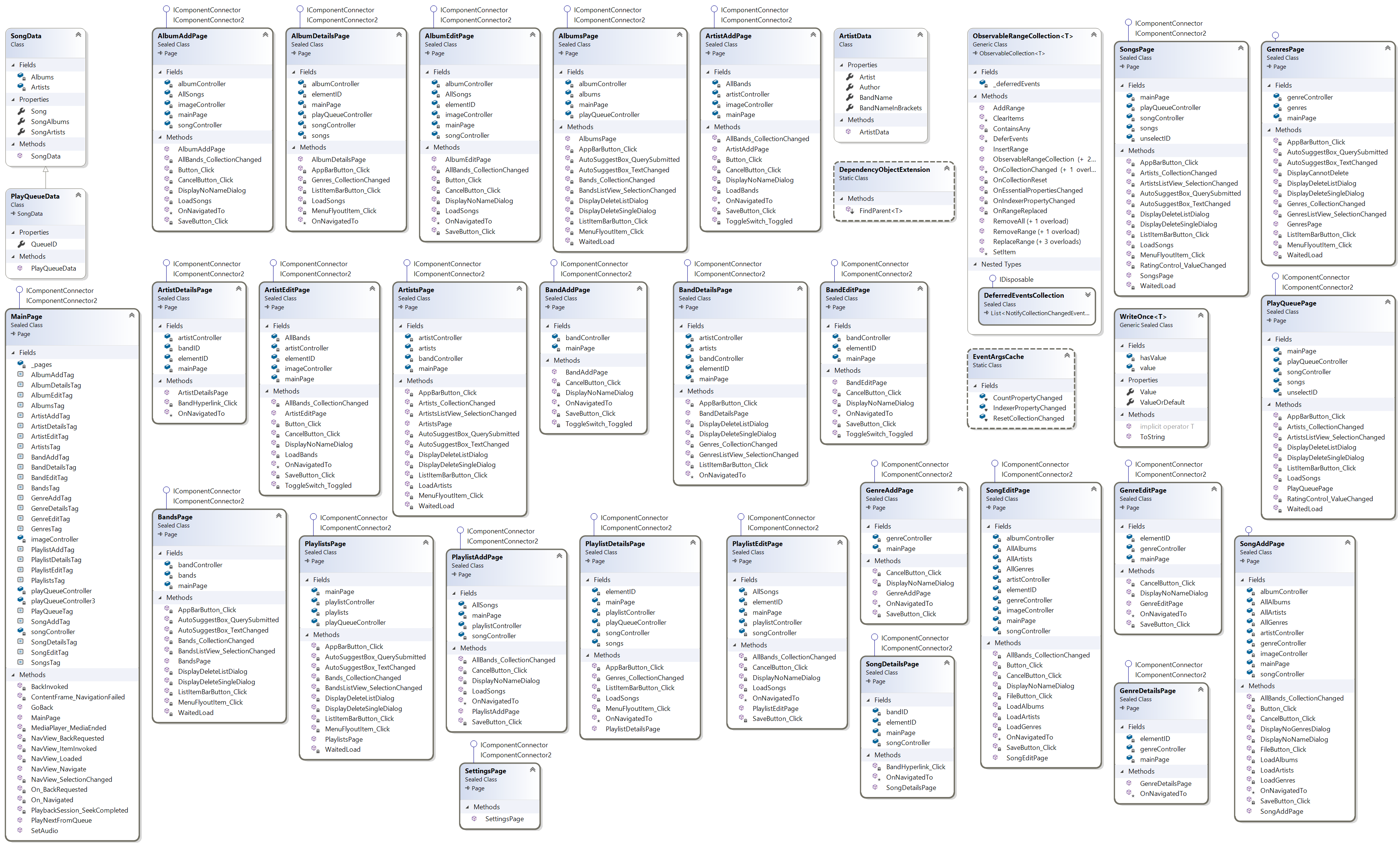
****



Diagram przestawia interfejsy i Klasy związane z prezenterem Albumu zawarte w MusicPlayer.UWP. Pozostałym modelom bazy odpowiadają zbliżone (mniej lub bardziej rozbudowane) prezentery.



# Mapy kodu według wzorców projektowych

## Legenda oznaczeń na diagramach

Obraz zawierający monitor, ekran, telefon komórkowy, telefon

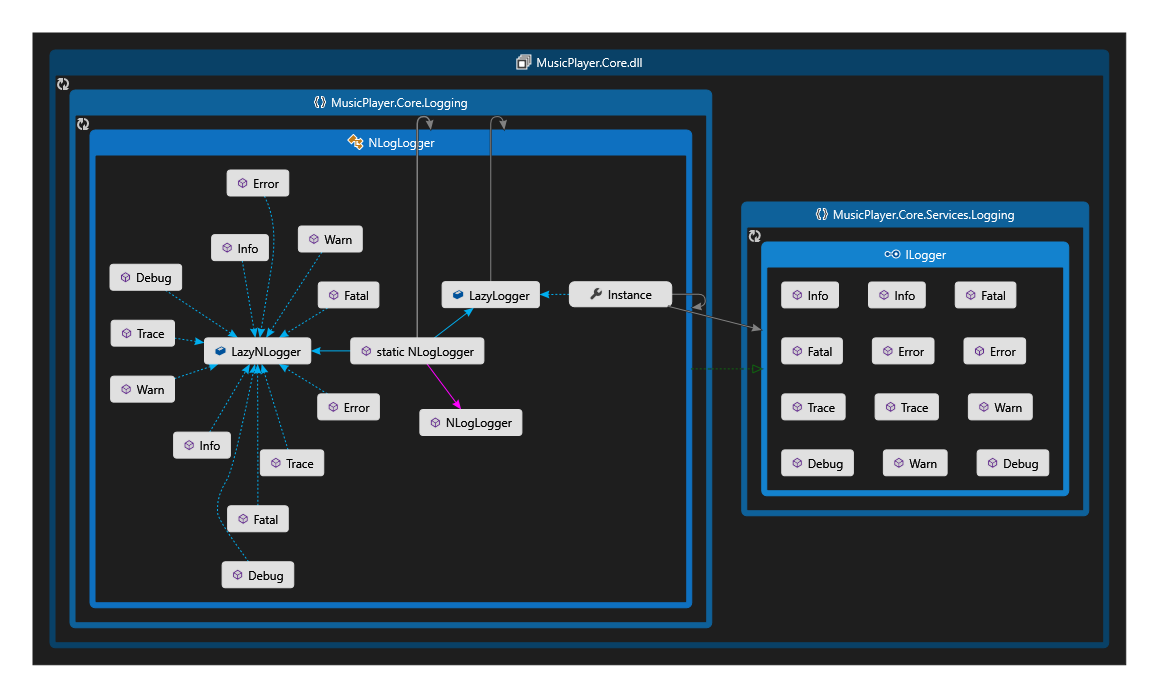
Opis wygenerowany automatycznie

## Zależności stworzonych bibliotek w projekcie

Obraz zawierający zrzut ekranu, monitor, ściana, ekran

Opis wygenerowany automatycznie

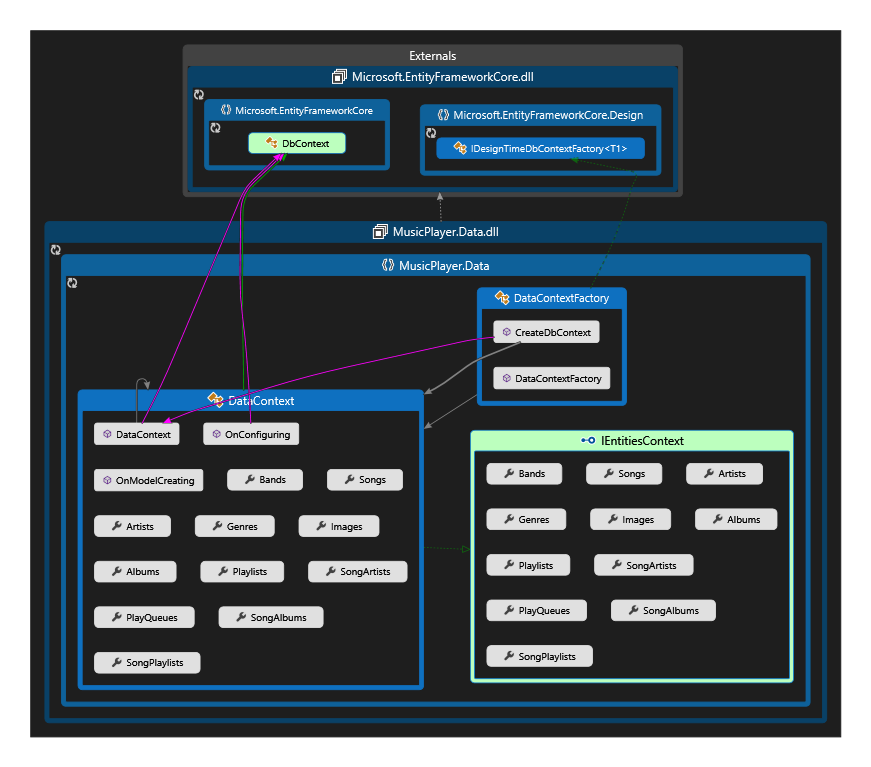
## Wzorzec #1 (kreacyjny): Singleton



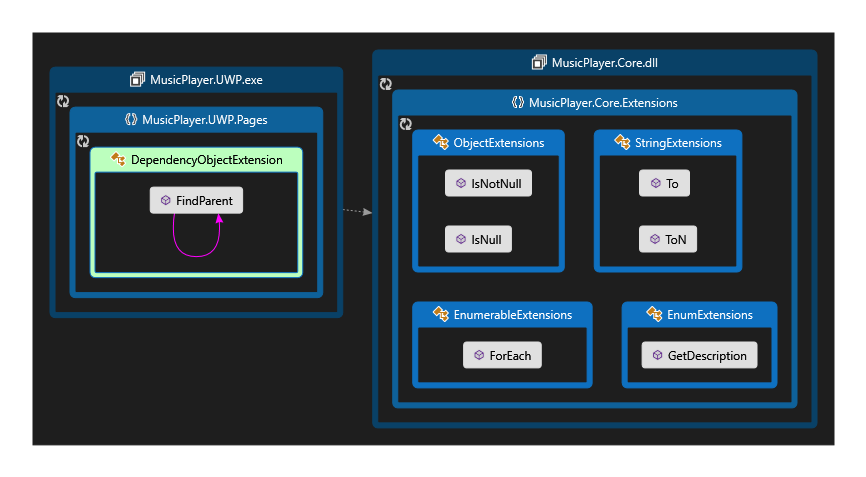
## Wzorzec #2 (kreacyjny): Dependency injection



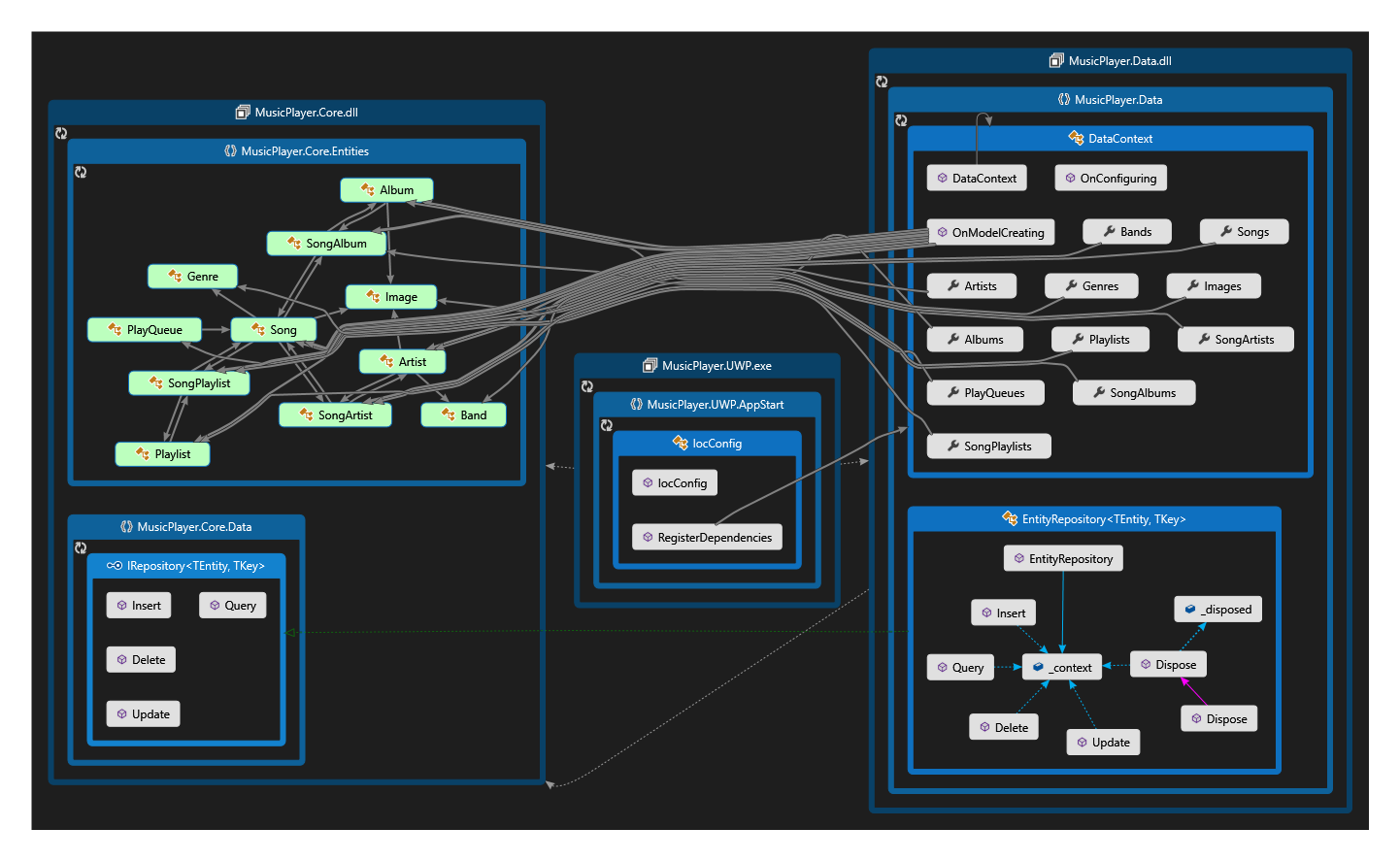
## Wzorzec #3 (kreacyjny): Factory



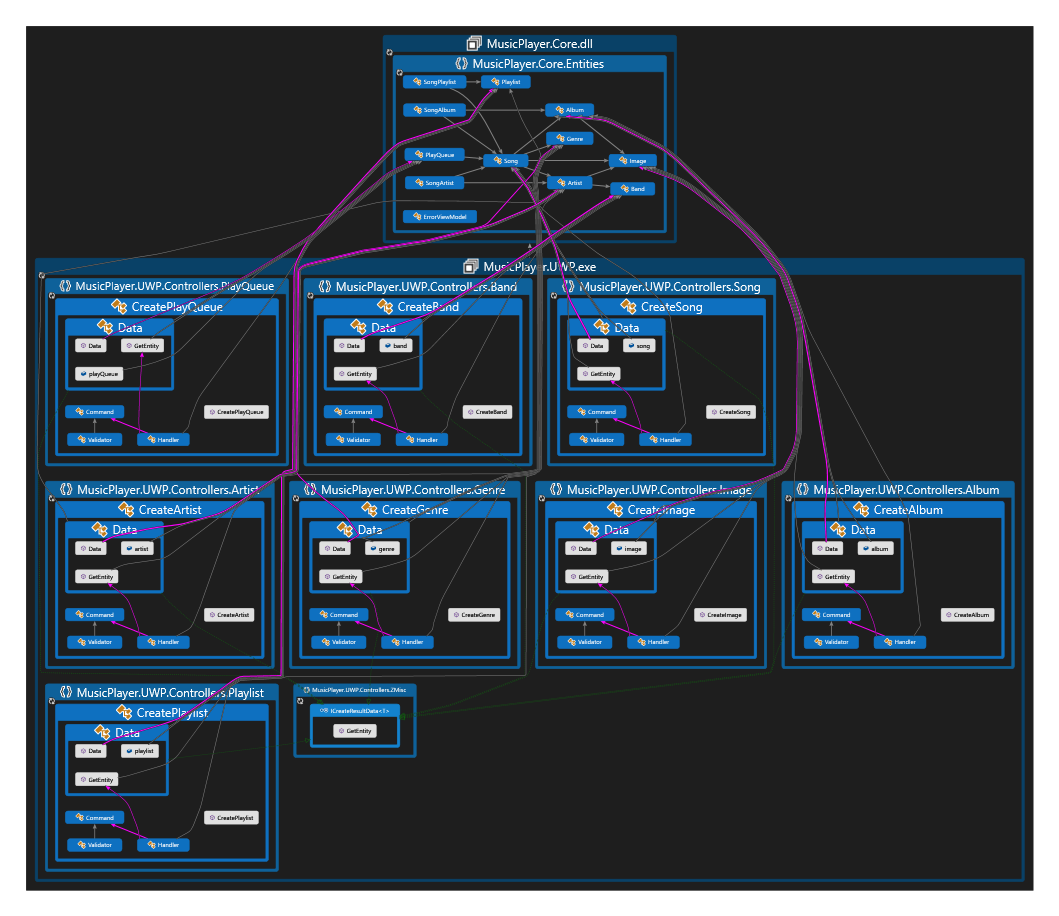
## Wzorzec #4 (strukturalny): Extension Object



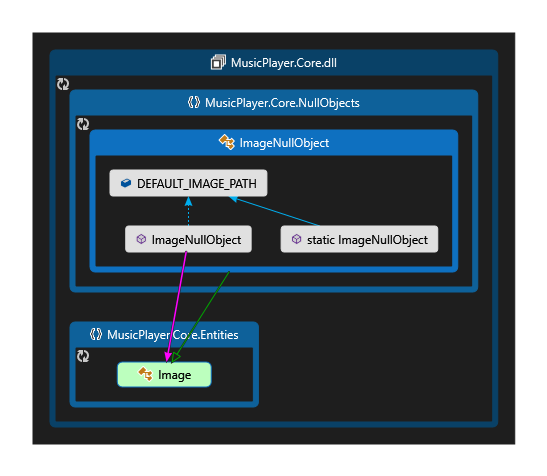
## Wzorzec #5 (strukturalny): Repository



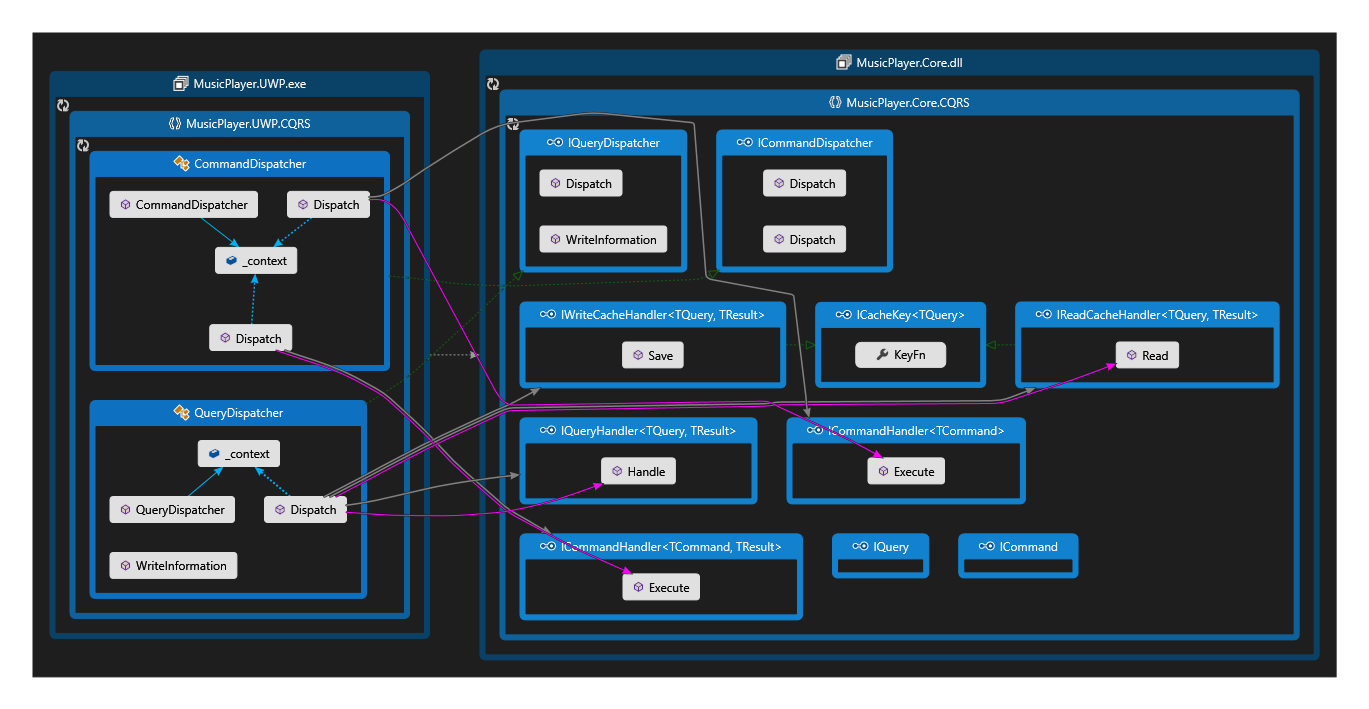
## Wzorzec #6 (strukturalny): Private class data



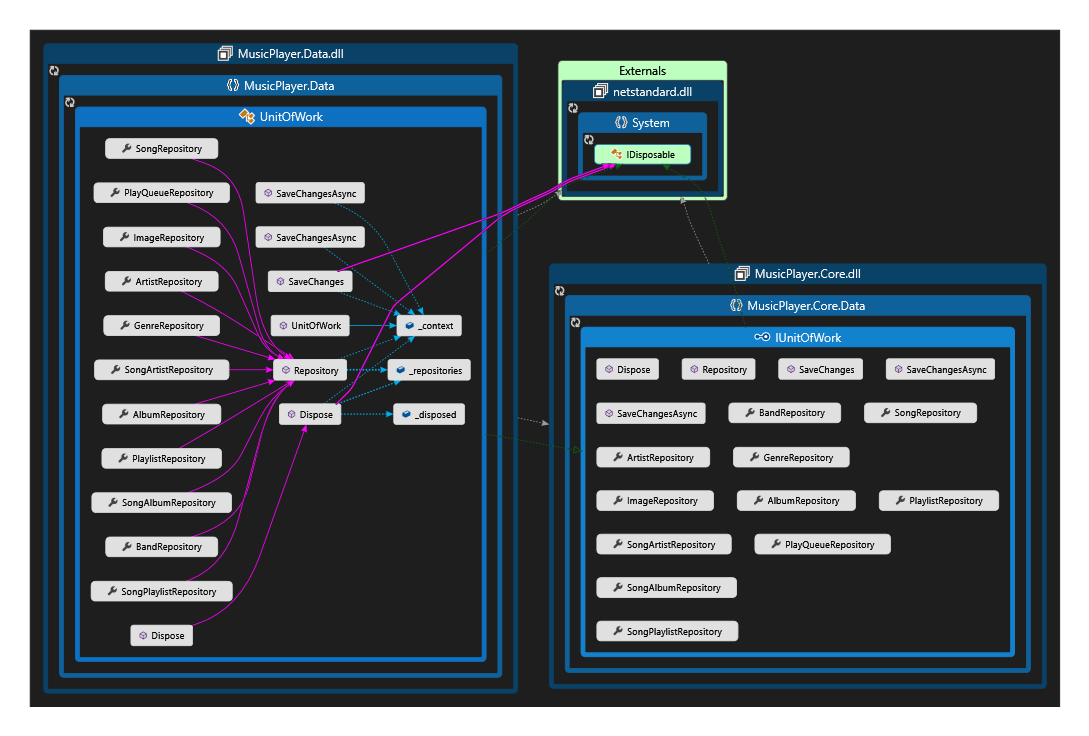
## Wzorzec #7 (czynnościowy): Null Object



## Wzorzec #8 (czynnościowy): CQRS

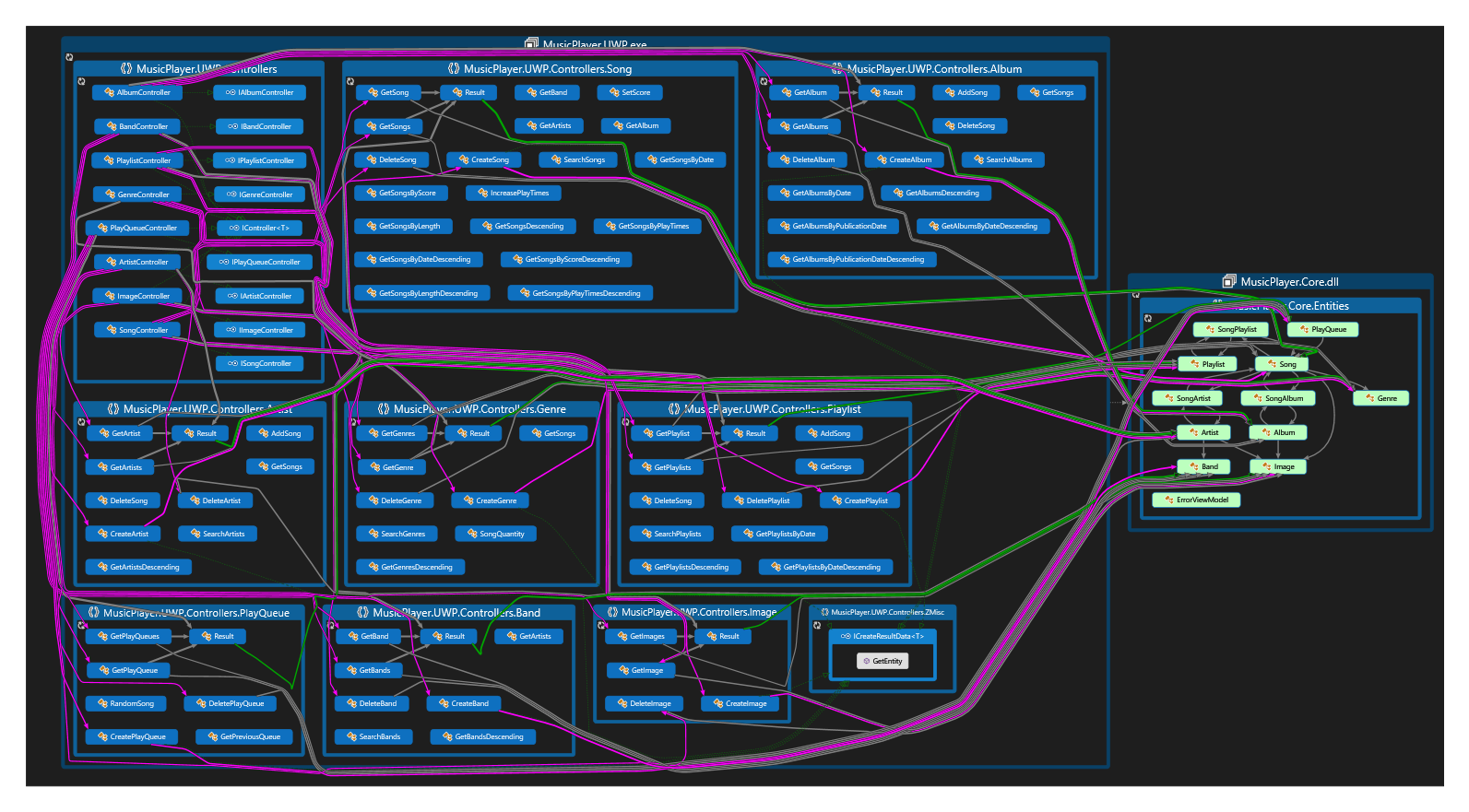


## Wzorzec #9: Unit of Work

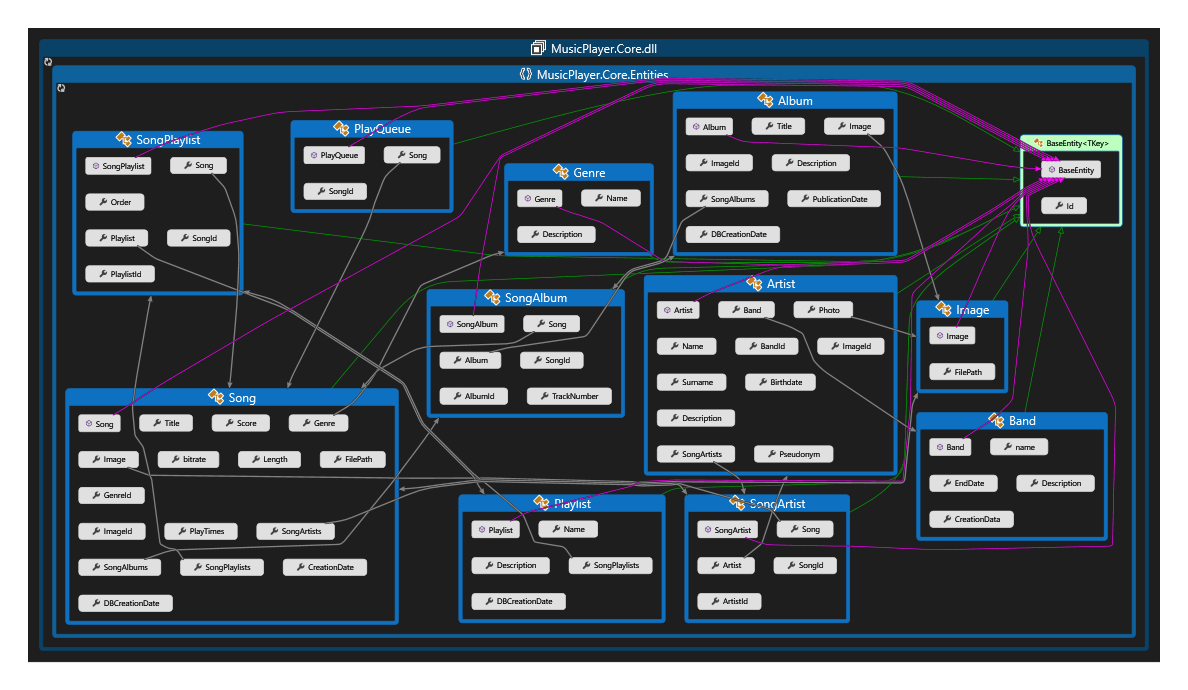


## Wzorzec #10 (architekturalny)

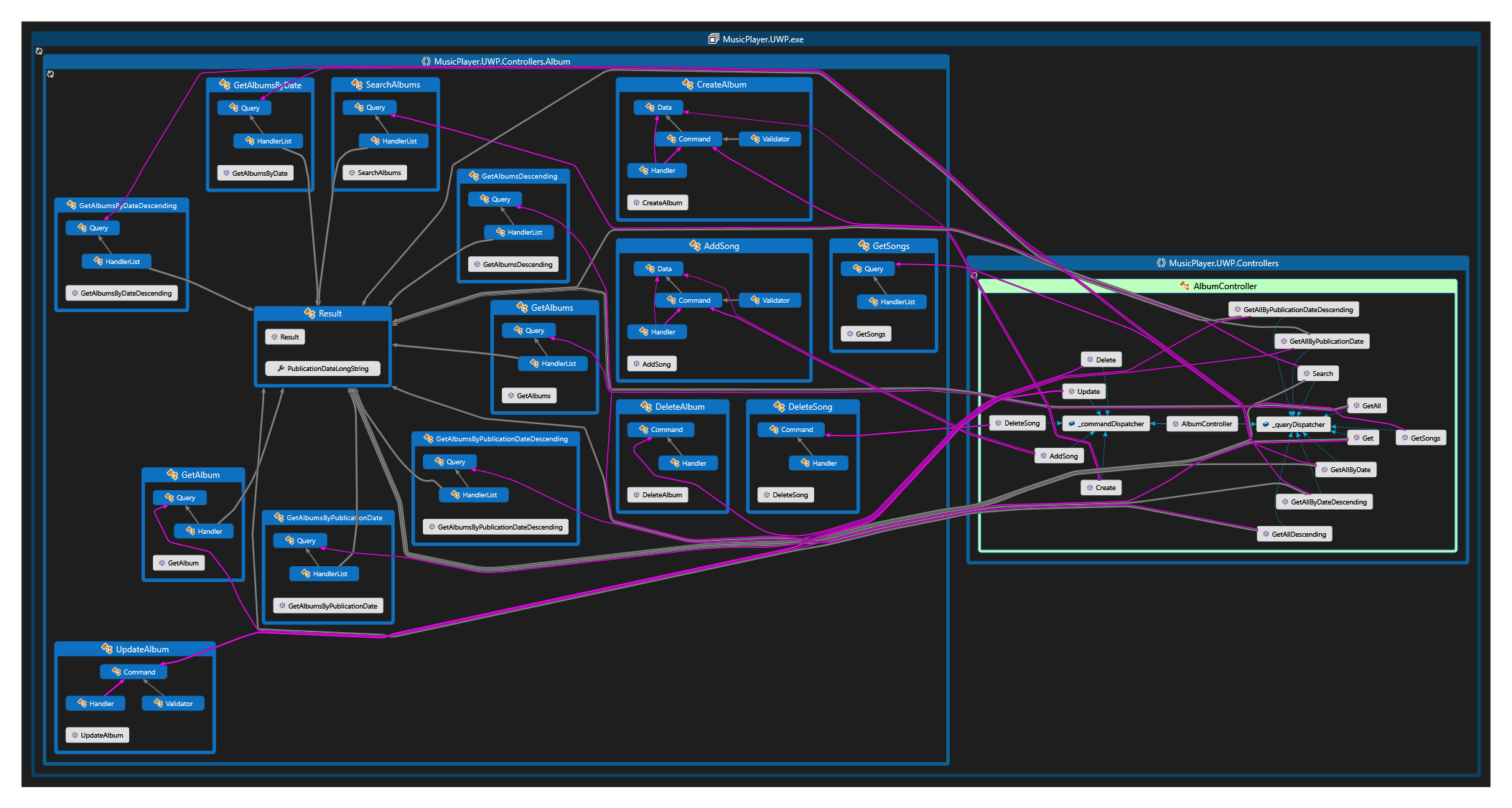
### MVP – struktura



### MVP – Modele



### MVP – Przykładowy prezenter Albumu



# Opisy wzorców

## Wzorzec #1 (kreacyjny): Singleton

## Wzorzec #2 (kreacyjny): Dependency injection

## Wzorzec #3 (kreacyjny): Factory

## Wzorzec #4 (strukturalny): Extension Object

## Wzorzec #5 (strukturalny): Repository

## Wzorzec #6 (strukturalny): Private class data

## Wzorzec #7 (czynnościowy): Null Object

## Wzorzec #8 (czynnościowy): CQRS

## Wzorzec #9: Unit of Work

## Wzorzec #10 (architekturalny)

# Podział pracy w zespole

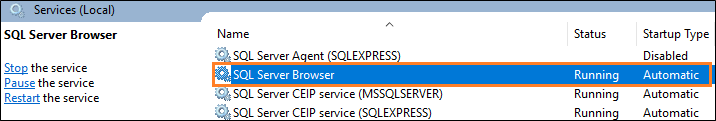
| **Czynność/Zadanie** | **Osoba** | | |
| --- | --- | --- | --- |
| **Magdalena Kalisz** | **Adam Bajguz** | **Michał Kierzkowski** |
| Zaprojektowanie bazy danych |  | + |  |
| Utworzenie modeli bazy danych | + | + |  |
| Utworzenie i skonfigurowanie rozwiązania (projektów w aplikacji) | + |  | + |
| Utworzenie prezentacji wzorców |  |  | + |
| Utworzenie końcowej dokumentacji |  | + | + |
| Utworzenie ikony aplikacji |  |  | + |
| Utworzenie map kodu |  | + |  |
| Stworzenie diagramów klas |  |  | + |
| **BACKEND** | | | |
| Utworzenie prezenterów | + |  |  |
| Pomoc przy tworzeniu prezenterów |  | + |  |
| Implementacja dostępu do bazy danych | + |  |  |
| Implementacja repozytoriów, CQRS tworzących dodatkową warstwę abstrakcji w MVP | + |  |  |
| **FRONTEND (strony w aplikacji, widok szczegółów, edycji, dodawania itp.)** | | | |
| Utworzenie nawigacji |  |  | + |
| Utworzenie stron utworu |  | + | + |
| Utworzenie stron albumu |  | + |  |
| Utworzenie stron zespołu |  | + |  |
| Utworzenie stron playlisty |  | + |  |
| Utworzenie stron gatunku |  |  | + |
| Utworzenie stron artysty |  | + |  |
| Utworzenie strony kolejki odtwarzania |  |  | + |
| Utworzenie odtwarzacza i jego integracja z bazą | + |  |  |
| **TESTOWANIE APLIKACJI** | | | |
| Testowanie backendu i bazy danych | + |  |  |
| Testowanie aplikacji (działania stron) |  |  | + |

**Podsumowanie:** podział pracy przy aplikacji był zrównoważony. W przedstawionej tabeli widać wyraźny podział zespołu na backend i frontend. Należy pamiętać że część backendowa pomimo mniejszej ilości czynności/zadań w tabeli była tak samo rozbudowana jak cześć frontendowa. Ponadto stopień skomplikowania projektu powoduje, że jeszcze bardziej szczegółowe wypisanie czynności/zadań zrealizowanych przez poszczególne osoby jest utrudnione.

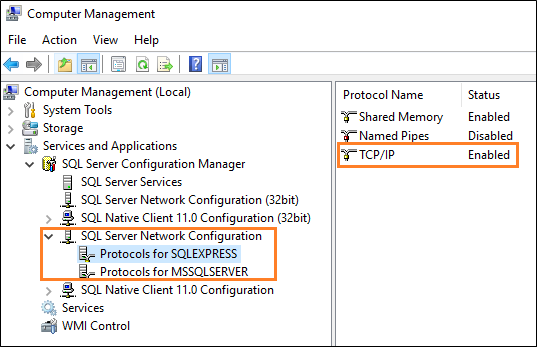
# Instrukcja obsługi aplikacji

# Instrukcja instalacji

1. Zainstalować Microsoft SQL Server 2014 zgodnie z poradnikiem: <https://www.sqlshack.com/how-to-install-sql-server-2014-management-studio/?fbclid=IwAR1Pa-VjREPi0VUOOo8WMnBHKED8d-axSk4DQnjz95rkx9w9_J0_8MzuObE>
2. Zainstalować Microsoft Visual Studio 2017 najlepiej w wersji Enterprise
3. Skonfigurować SQL Server:
   1. Uruchomić wszystkie usługi SQL Server’a



* 1. Włączyć protokół TCP/IP:



1. Utworzyć bazę danych poleceniem Update-database w konsoli NuGet w programie Microsoft Visual Studio używając projektu MusicPlayer.Migrations.Sturtup