

**politechnika śląska**

**wydział automatyki, elektroniki i informatyki**

**kierunek teleinformatyka**

##### Projekt inżynierski

# Odtwarzacz muzyczny wykorzystujący API SoundCloud

Autor: Krzysztof Błażełek

Kierujący pracą: dr inż. Adam Gudyś

Gliwice, grudzień 2016

Spis treści

[1. Wprowadzenie do dziedziny 5](#_Toc262680014)

[2. Podstawy teoretyczne 6](#_Toc262680015)

[2.1 Formatowanie tekstu za pomocą stylów 6](#_Toc262680016)

[2.1.1 Styl podstawowy (normalny) 7](#_Toc262680017)

[2.1.2 Tytuły rozdziałów i podrozdziałów projektu inżynierskiego 7](#_Toc262680018)

[2.1.3 Numeracja stron 7](#_Toc262680019)

[2.1.4 Przypisy 7](#_Toc262680020)

[3. Specyfikacja zewnętrzna 8](#_Toc262680021)

-wymagania sprzętowe i programowe

-interfejs użytkownika (opis każdego widoku)

[4. Specyfikacja wewnętrzna 10](#_Toc262680022)

[5. Eksperymenty 11](#_Toc262680023)

[6. Wnioski 12](#_Toc262680024)

1. Wprowadzenie do dziedziny

Wraz z gwałtownym rozwojem Internetu coraz częściej wykorzystywane są media strumieniowe. Dużą popularnością w sieci cieszą się obecnie takie serwisy jak YouTube oraz SoundCloud, które umożliwiają twórcom dostarczenie swoich dzieł do widzów/słuchaczy. SoundCloud jest platformą umożliwiającą artystom dystrybucję oraz promocję własnej muzyki. Serwis ten oprócz udostępniania muzyki pozwala też zarejestrowanym użytkownikom na komentowanie utworów, dodawanie ich do ulubionych oraz na tworzenie list odtwarzania. Oprócz tego serwis udostępnia interfejs programistyczny, który zostaw wykorzystany do napisania odtwarzacza muzycznego będącego tematem tej pracy.

* 1. Porównanie dostępnych technologii

1.1.1 Windows Forms

Jest to biblioteka przeznaczona do tworzenia desktopowych aplikacji z bogatym interfejsem graficznym użytkownika, która jest częścią platformy .NET 3.0. WinForms jest już dopracowaną technologią, która jest utrzymywana, ale nie będą dodawane do niej nowe funkcjonalności. Zastosowany w niej silnik graficzny GDI (Graphics Device Interface) charakteryzuje się prostotą, ale nie oferuje wielu funkcjonalności wspieranych przez takie silniki graficzne jak DirectX oraz OpenGL. Brak wsparcia akceleracji sprzętowej dla grafiki 3D oraz ograniczona optymalizacja sprzętowa wynikająca z zastosowanej architektury zawężają obszar zastosowania GDI. Istnieje możliwość uruchomienia aplikacji napisanej w WinForms na systemach Linux oraz Mac OS za pośrednictwem Mono – alternatywnej implementacji biblioteki .NET opartej na licencji Open Source. Przeportowanie aplikacji wiąże się z licznymi ograniczeniami wynikającymi z ścisłym powiązaniem WinForms z Windows API (WinAPI). Z tego powodu utrudnione jest projektowanie zaawansowanych interfejsów graficznych. Jeżeli istniejące kontrolki (elementy graficzne takie jak przyciski i pola tekstowe) nie spełniają wymagań projektowych, to programista jest zmuszony napisać ich własną implementację lub skorzystać z bibliotek dostarczanych przez innych wytwórców. Wspierane są wszystkie języki programowania kompatybilne z architekturą CLI (Common Language Infrastructure). Najpopularniejsze z nich to C#, VB.NET oraz C++/CLI.

1.1.2 Windows Presentation Foundation

WPF jest również częścią platformy .NET oraz następcą Windows Forms. Technologia ta utworzyła spójne środowisko dla projektowania bogatych, desktopowych aplikacji graficznych wykorzystujących grafikę 2D, 3D, dokumenty, animacje oraz media. Do definiowania interfejsu graficznego wprowadzono język XAML (Extensible Application Markup Language). Jest to deklaratywny język oparty na języku XML w którym za pomocą tagów i atrybutów opisywane są (głównie) elementy graficzne i ich właściwości, np.

***<Button x:Name="myButton" Content="Click me" Background="Aqua" />***

Parser języka XAML następnie generuje kod w którym tworzone są odpowiadające obiekty i powiązane z nimi właściwości. Hierarchiczna struktura języka XAML, style oraz szablony pozwalają na łatwiejsze projektowanie zaawansowanych interfejsów graficznych. Kontrolki nie są już powiązane z WinAPI jak to miało miejsce w WinForms. W WPF zastosowano również nowoczesny silnik graficzny – DirectX. Pozwala on na wyświetlanie bardziej zaawansowanej grafiki niż to miało miejsce w przypadku silnika GDI. Dodatkowo silnik ten oferuje większą wydajność poprzez lepsze wykorzystanie dostępnego sprzętu (głównie karty graficznej). W WPF zastosowany został mechanizm powiązania danych (data binding). Umożliwia on utworzenie połączenia pomiędzy elementem UI a obiektem znajdującym się w logice biznesowej. Połączenie takie umożliwia wzajemną aktualizację obu stron w momencie nastąpienia zmiany. Technika ta zapewnia separację logiki biznesowej od definicji interfejsu graficznego, co z kolei wpływa na czytelność i łatwość utrzymania aplikacji. WPF jest już również dojrzałą technologią i Microsoft nie planuje dalszych zmian (poza poprawkami błędów).

1.1.3 Silverlight

Silverlight jest technologią, która zadebiutowała w 2007 roku. Wykorzystuje ona wiele funkcjonalności dostępnych w WPF, ale aplikacje napisane w tej technologii uruchamiane są poprzez przeglądarkę internetową (np. Mozilla Firefox lub Internet Explorer). Technologia ta umożliwiała uruchamianie aplikacji na wielu systemach operacyjnych (m.in. macOS oraz Linux poprzez alternatywną implementację Moonlight). Silverlight wraz z kolejnymi wersjami miał problemy z kompatybilnością z przeglądarkami internetowymi oraz z wspieranymi systemami operacyjnymi. Obecnie Microsoft jak i większość przeglądarek zakończyło/planuje zakończenie wsparcia dla tej technologii.

1.1.4 Windows Runtime

WinRT jest interfejsem programistycznym, który został wprowadzony wraz z debiutem systemu Windows 8. Technologia ta nie bazuje już tak jak wcześniej wymienione na bibliotece .NET, lecz zastępuje ona wcześniej wykorzystywany interfejs WinAPI. Pomimo oderwania się od platformy .NET większość mechanizmów w niej zawartych zostało przeniesionych na WinRT. Nowy interfejs programistyczny wymusza na programiście stosowanie wywołań asynchronicznych przy operacjach wejścia/wyjścia w celu uniknięcia blokady interfejsu graficznego lub niepotrzebnego blokowania wątku. WinRT natywnie oferuje wsparcie dla procesorów x86 oraz ARM. Odłączenie się od platformy .NET wiąże się również z zmianą wspieranych języków programowania. Są to C#, VB.NET, JavaScript oraz C++/CX. Interfejsy graficzne natomiast projektuje się za pomocą wcześniej opisanego języka XAML oraz HTML/CSS. Języki te są natywnie wspierane przez system operacyjny. Aplikacje w WinRT są uruchamiane w odseparowanym środowisku (sandbox) kontrolowanym przez system operacyjny. Ma to zapewnić większe bezpieczeństwo poprzez ścisłą kontrolę dostępu aplikacji do zasobów systemu takich jak dysk, sieć oraz podłączone urządzenia. Urządzenia mobilne korzystają z odmiany WinRT o nazwie Windows Phone Runtime. Aplikacje WinRT są dystrybuowane poprzez sklep Windows (Windows Store). Podczas instalacji użytkownik proszony jest o zezwoleniu danej aplikacji na dostęp do chronionych części systemu (o ile takich aplikacja wymaga).

1.1.5 Universal Windows Platform

2. Podstawy teoretyczne.

Aplikacja została napisana za pomocą platformy udostępnionej wraz z premierą Windows’a 10 – Universal Windows Platform (UWP). Technologia ta pozwala na tworzenie aplikacji, które można uruchomić na wszystkich urządzeniach wspierających system Windows 10. Obecnie są to następujące rodziny urządzeń:

* PC
* Mobilne
* IoT (np. Raspberry PI)
* Xbox
* HoloLens

Programista jest w stanie napisać 1 aplikację, która działa na wszystkich tych urządzeniach jednocześnie. Pomimo wieloplatformowości wciąż można wykorzystać funkcjonalności specyficzne dla danej rodziny urządzeń, np. operacje na pinach w urządzeniach IoT lub zintegrowane przyciski w urządzeniach mobilnych. Możliwe jest to dzięki rozszerzeniom SDK (Extension SDK). Dostarczają one programiście API pozwalające wykorzystać te dodatkowe funkcje jeżeli będą dostępne. Inną kluczową cechą UWP jest adaptacyjny interfejs użytkownika. Zależy on od wielkości ekranu i jego rozdzielczości – DPI, sposobu komunikacji użytkownika z urządzeniem oraz typowej odległości w jakiej użytkownik korzysta z niego. W rezultacie programista podając wymiary elementów operuje na efektywnych pikselach (a nie fizycznych). Taki sposób definiowania rozmiarów powoduje, że fizycznie większy rozmiar czcionki wyświetlanej na ekranie telewizora będzie z perspektywy użytkownika odbierany tak samo jak fizycznie mniejszy rozmiar czcionki na ekranie smartfona, który znajduje się bliżej użytkownika. Dodatkowo programista ma szereg narzędzi pozwalających na uzyskanie jeszcze lepszego interfejsu użytkownika, np. dostosowujący się w zależności od dostępnego na ekranie miejsca lub ze względu na sposób komunikacji użytkownika z urządzeniem.

3. Specyfikacja zewnętrzna.  
Specyfikacja zewnętrzna to opis funkcjonalności dostępnych dla użytkownika końcowego. Opisuje ona interfejs użytkownika oraz możliwości aplikacji.

W specyfikacji zewnętrznej powinno się znaleźć wszystko to, co przeciętny użytkownik powinien wiedzieć o programie w celu jego prawidłowego użytkowania. Produkt opisywany jest z punktu widzenia użytkownika, więc należy się wystrzegać terminów z zakresu specyfikacji wewnętrznej.

Porównanie technologii:

Windows Forms -

<https://developers.soundcloud.com/docs/api/sdks> napisac, ze nie ma wrappera dla c#  
4. Specyfikacja wewnętrzna.  
5. Eksperymenty.  
6. Wnioski.