



Dokumentacja
projektu na przedmiot
Inżynieria oprogramowania

Temat: MP3 player

Autorzy:
Piotr Jendrzyca
Janusz Jaśkowski

Gliwice, dnia 20.01.2016r.

Spis treści:

1. Temat projektu
2. Opis tematu
3. Motywy wyboru tematu
4. Możliwości aplikacji
5. Wykorzystane technologie
6. Wymagania
7. Podział prac
8. Opis aplikacji
9. Szczegóły techniczne
10. Struktura programu
11. Testy
12. Statystyki z programu kontroli wersji
13. Wnioski i możliwości rozwoju

1. Temat projektu

Tematem naszej pracy jest odtwarzacz muzyki "MP3player".

2. Opis tematu:

Aplikacja pozwala na odtwarzanie swojej muzycznej kolekcji na urządzeniach z systemem Android. Ponadto umożliwia wyszukiwanie i pobieranie muzyki z popularnego serwisu "YouTube" oraz pobieranie muzyki z specjalnie przez nas przygotowanych tematycznych list piosenek.

3. Motywy wyboru tematu

Wybraliśmy ten temat, ponieważ uważamy, że brakuje takiego programu, który mógłby pobierać i odtwarzać pliki muzyczne z portalu "YouTube". Nie znaleźliśmy żadnego detykowanego programu do tego celu. Teraz gdy spodoba nam się jakaś piosenka możemy szybko ją wyszukać i pobrać, wszystko w jednym programie bez potrzeby odwiedzania witryn internetowych, przeznaczonych do tego celu.

4. Możliwości aplikacji

- Odtwarzania kolekcji muzycznej zapisanej na urządzeniu
- Wyszukiwanie i pobieranie muzyki bezpośrednio z portalu "YouTube"
- Wyszukiwanie piosenek z lokalnej kolekcji
- Odtwarzanie utworów w sposób losowy bądź zapętłony
- Możliwość odkrywania nowych piosenek, dzięki "Music Genres" czyli kolekcji najnowszych piosenek z różnych rodzajów muzyki.
- Możliwość zmieniania, zatrzymywania i ponownego odtwarzania muzyki z poziomu paska notyfikacji.

5. Wykorzystane technologie

- Java SDK 23.0.2
- YouTube API
- YouTubeInMP3 API

6. Wymagania

- Do obsługi programu potrzebne jest urządzenie z systemem Android w wersji 4.0 lub wyższej

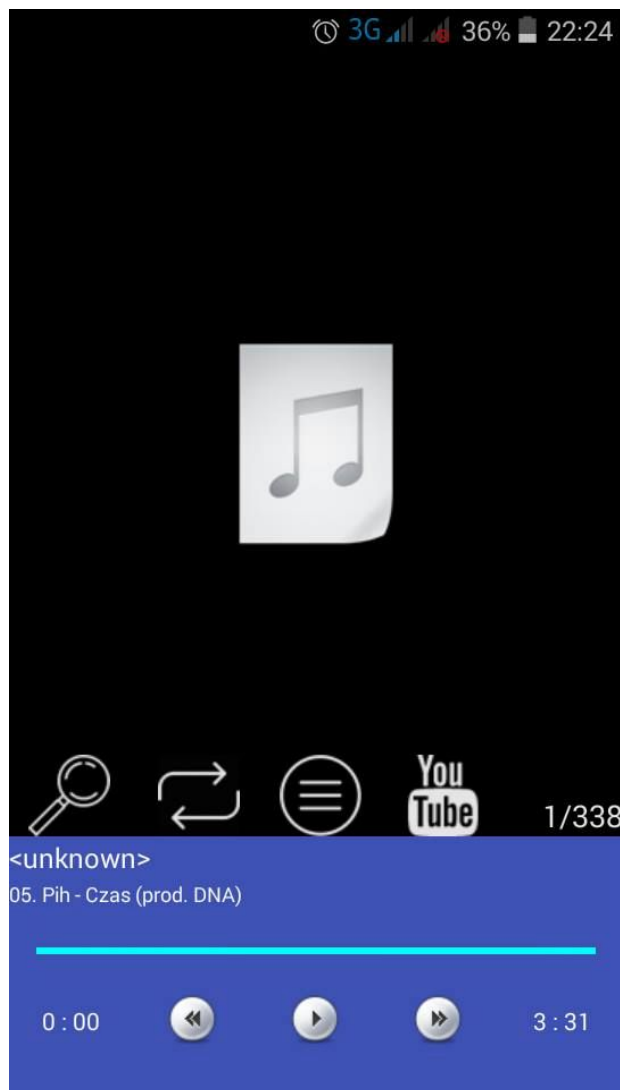
7. Podział prac

Po osiągnięciu podstawowej funkcjonalności programu (zaimplementowaniu serwisu MediaPlayer) każdy z naszej dwuosobowej sekcji odpowiedzialny był za inną część projektu. Umożliwiło to równoległą pracę. Nie było konieczności czekania, aż któryś z nas dokończy dany moduł, program funkcjonował poprawnie bez niego.

- Janusz Jaśkowski:
 - Grafika: loga, ikony, przyciski, tła
 - Notyfikacje
 - Aktywności "player" oraz "playlist"
- Piotr Jendrzyca
 - YouTube connecter, DownloadMeneger
 - Aktywności "kolekcje" oraz "wyszukiwanie z yt"
 - Implementacja serwisu: MediaPlayer

8. Opis aplikacji

- Podstawowy wygląd po uruchomieniu:



Tak wygląda nasza aplikacja po uruchomieniu. Program zawsze próbuje wydobyć okładkę albumu z piosenki, jeśli się to nie powiedzie ustawiana jest domyślna ikona, taka jaką widzimy na obrazku. Dolne niebieskie pole obejmuje autora i tytuł piosenki, ten jasny niebieski pasek wskazuje nam w którym momencie piosenki właśnie jesteśmy. Z pomocą tego paska możemy przewinąć piosenkę do dowolnego momentu. Pauzowanie oraz wznawianie odtwarzania odbywa się poprzez kliknięcie środkowego przycisku PLAY. Piosenki zmieniamy tymi dwoma przyciskami usytuowanymi obok lub też przeciągając palcem po ekranie w dowolną stronę. Po bokach widzimy również czas, ten po lewej

wskazuje aktualny czas utworu, zaś ten z prawej czas pozostały do końca. Nad niebieskim polem widzimy 4 ikony, które odpowiednio od lewej odpowiedzialne są za:

- przeszukiwanie lokalnej kolekcji utworów
- ustawianie odtwarzania w tryb losowy bądź zapętłony
- otwieranie kolekcji przygotowanych i podzielonych przez nas tematycznie utworów gotowych do pobrania
- wyszukiwanie i pobieranie piosenek z YouTube'a

Po prawej stronie usytuowany jest licznik piosenek.

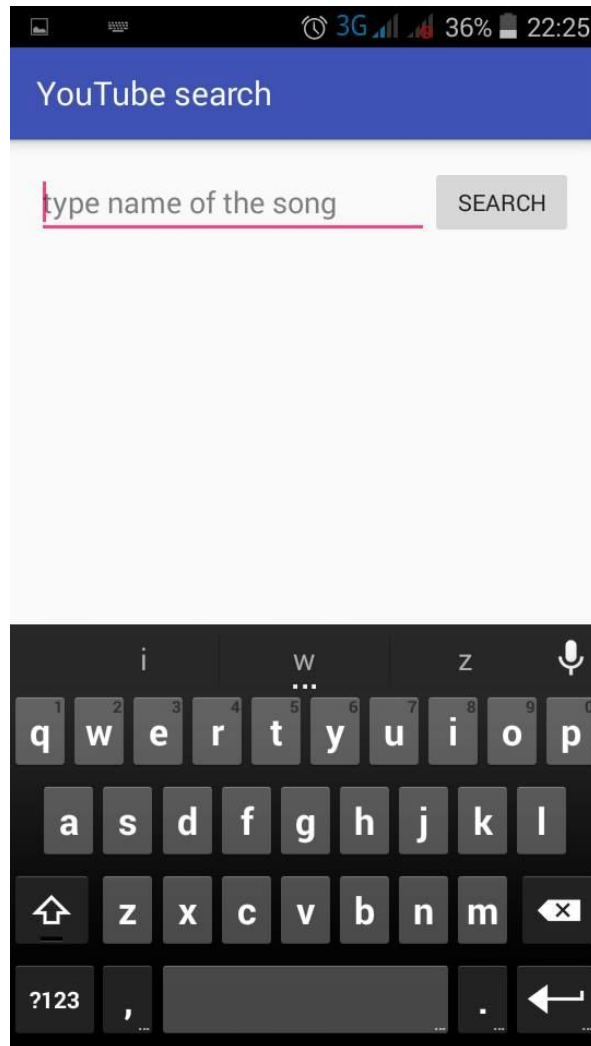
- Lista utworów zapisanych na urządzeniu



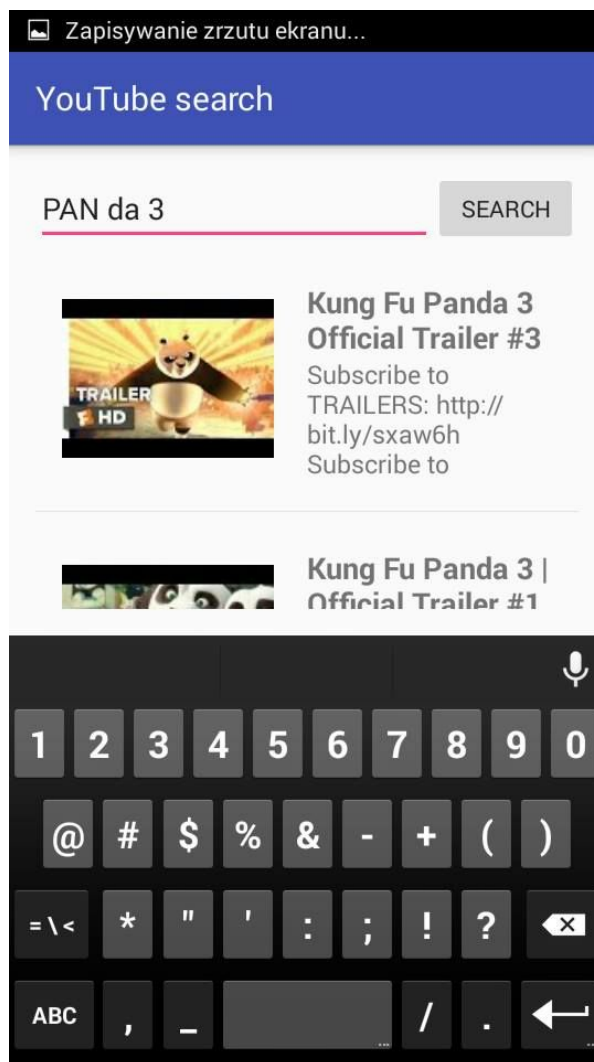
Wyświetla nam się lista utworów zapisanych na urządzeniu. Uporządkowana jest ona alfabetycznie według tytułu utworu. Pod

tytułem mamy autora utworu a po prawej stronie czas trwania. Listę możemy filtrować klikając w lupę i wpisując szukaną frazę.

- Wyszukiwanie i pobieranie z serwisu YouTube:



W pole tekstowe wpisujemy nazwę piosenki, którą chcielibyśmy wyszukać w serwisie YouTube.

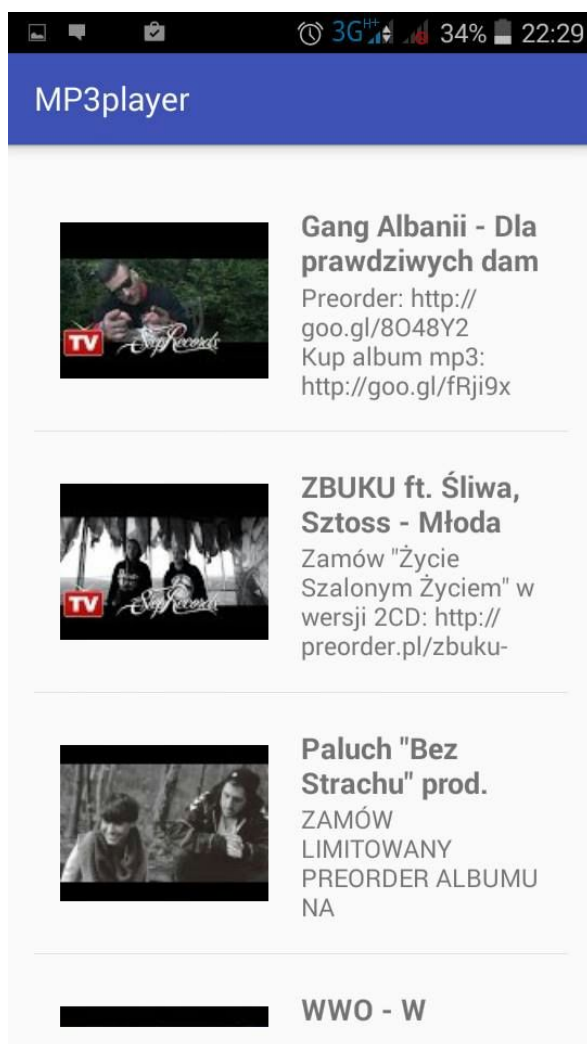


Po kliknięciu przycisku “search” ukazuje się nam lista wyników. Kliknięcie w jeden z nich spowoduje rozpoczęcie pobierania jeśli mamy dostęp do internetu.

- Kolekcja utworów do pobrania

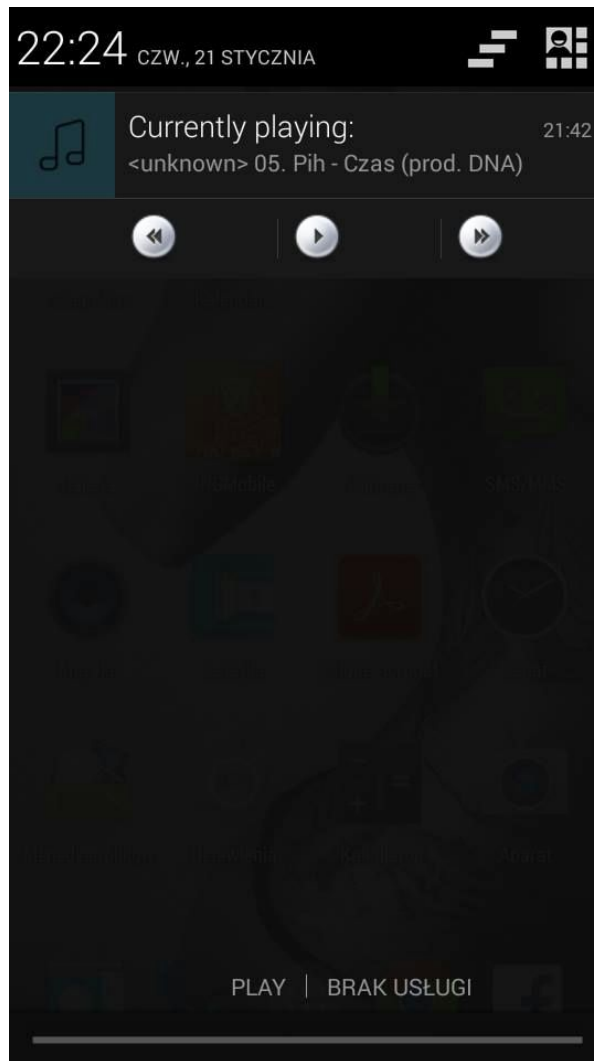


Tutaj mamy do wyboru różne rodzaje muzyki, klikając w odpowiednią kafelkę wczyta się nam lista utworów o danym gatunku muzycznym (wymagane połączenie internetowe, listy pobierane są z serwisu YouTube).



Po kliknięciu w odpowiednią kafelkę ładuję się playlista składająca się z 50 piosenek z wybranej kategorii. Listy te są dość często aktualizowane. Klikając w odpowiedni utwór rozpoczniemy jego pobieranie.

- Notyfikacje



Gdy nasz MP3player jest uruchomiony mamy możliwość zarządzania nim z poziomu paska notyfikacji. Powiadomienie wyświetla informacje na temat utworu: tytuł oraz autor. Możliwe jest zatrzymanie/wznowienie odtwarzania utworu, zmiana utworu na kolejny lub poprzedni. Klikając w notyfikację nasza aplikacja zostanie uruchomiona (jej domyślne okno), natomiast przesuwając palcem w bok (odrzucając powiadomienie) aplikacja zostanie wyłączona.

9. Szczegóły techniczne

Program został napisany w programie AndroidStudio firmy IntelliJ.

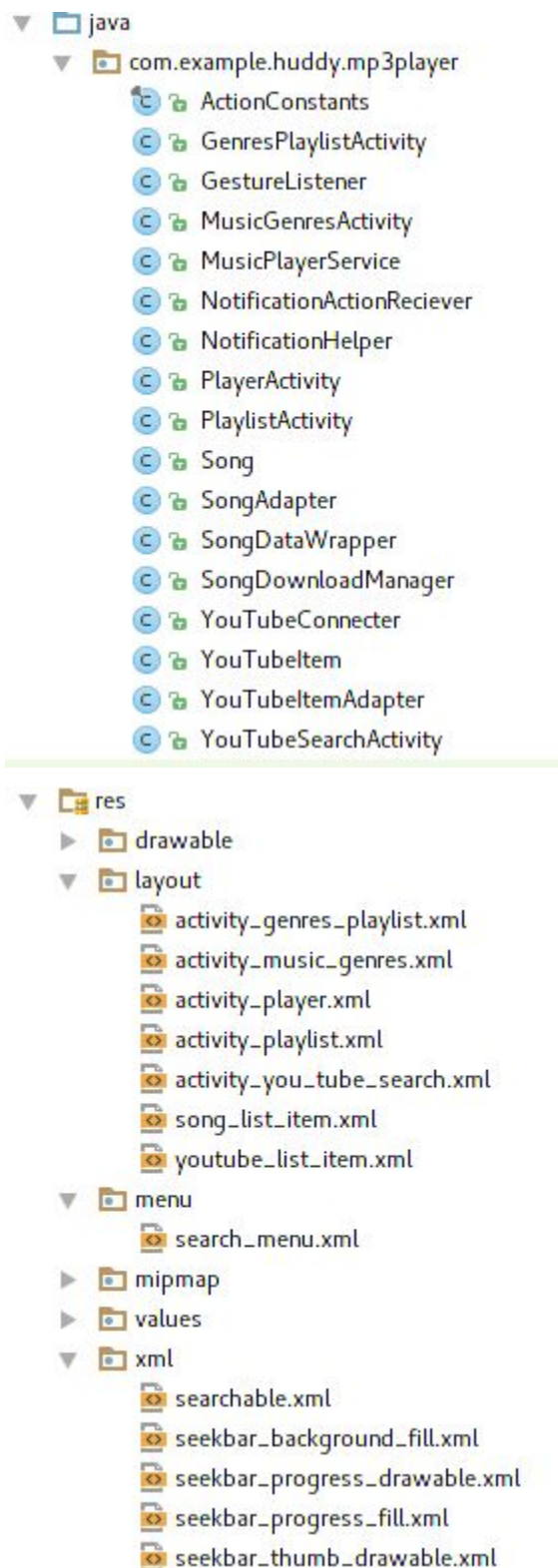
Korzystaliśmy z języka Java w wersji SDK 23.0.2, dodatkowo skompilowaliśmy biblioteki Google potrzebne do korzystania z YouTube API. Całość naszego projektu jest napisana w tzw. czystej Javie.

libs/google-api-client-1.21.0.jar	Compile	▼
libs/google-api-client-android-1.21.0.jar	Compile	▼
libs/google-api-services-youtube-v3-rev158-1.21.0.jar	Compile	▼
libs/google-http-client-1.21.0.jar	Compile	▼
libs/google-http-client-android-1.21.0.jar	Compile	▼
libs/google-http-client-jackson2-1.21.0.jar	Compile	▼
libs/google-oauth-client-1.21.0.jar	Compile	▼
libs/jackson-core-2.1.3.jar	Compile	▼
libs/jsr305-1.3.9.jar	Compile	▼
libs/picasso-2.4.0.jar	Compile	▼

Do pobierania danych z serwisu YouTube wykorzystane zostało wspomniane wyżej YouTube API, natomiast pobieranie utworów odbywa się poprzez YouTubeInMp3 API. Potrzebne jest ID piosenki, którą próbujemy pobrać. ID pobierane jest wraz informacjami o utworze z YouTube API. Kolejnym krokiem jest stworzenie odpowiedniego linku, który dzięki YouTubeInMp3 API po przekazaniu do systemowego menadżera pobierania spowoduje, iż zadany utwór zostanie ściągnięty na nasze urządzenie.

10. Struktura programu

Nasz program podzielony jest na wiele klas, plików XML oraz plików graficznych. Poniżej przedstawiamy drzewo komponentów:



11. Testy

Aplikacja była na bieżąco testowana i sprawdzana w trakcie implementacji. Powstałe błędy były natychmiastowo wykrywane i usuwane. Niektóre funkcje testowaliśmy nawzajem aby

zminimalizować ryzyko wystąpienia jakichkolwiek błędów wynikających z nieprawidłowego korzystania z aplikacji. Aby się upewnić o niezawodności naszego programu poprosiliśmy grono znajomych o przetestowanie aplikacji pod kątem odporności na “głupotę”.

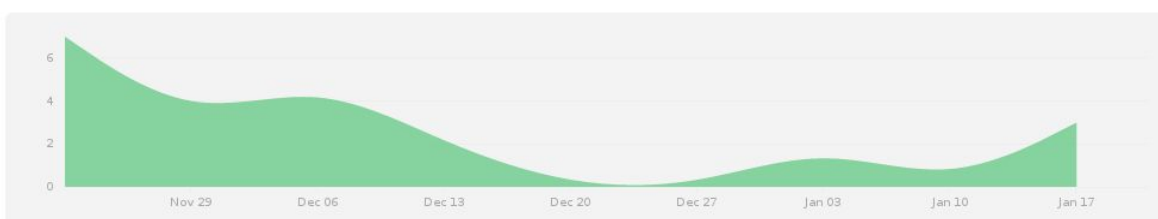
12. Statystyki z programu kontroli wersji

Do współpracy wybraliśmy repozytorium oparte na systemie kontroli wersji Git, umieszczonego w serwisie Github, natomiast do obsługi repozytorium korzystaliśmy z programu Git.

Nov 22, 2015 – Jan 21, 2016

Contributions to master, excluding merge commits

Contributions: **Commits** ▾



Contributors	Traffic	Commits	Code frequency	Punch card	Network	Members
--------------	---------	---------	----------------	------------	---------	---------

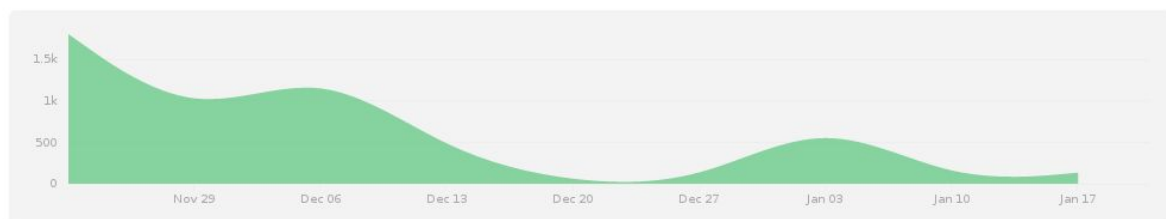
Additions and Deletions per week



Nov 22, 2015 – Jan 21, 2016

Contributions to master, excluding merge commits

Contributions: Additions ▾



13. Wnioski i możliwości rozwoju

Obsługa naszego programu jest dziecinnie prosta, jako jednym z naszych ważniejszych celów było osiągnięcie przejrzystego i łatwego w obsłudze interfejsu, aby każdy użytkownik mógł z łatwością nawigować po programie.

Dzięki zastosowaniu technologii Java SDK dydykowanej pod systemy Android, mamy pewność, iż jeszcze przez długi czas nasz program będzie działał na urządzeniach z tym systemem.

Jest jeszcze sporo miejsca na drobne ulepszenia i modyfikacje w naszym programie. Można by wprowadzić możliwość tworzenia własnych playlist opartych na bazie danych czy też zaimplementować equalizer, który pozwoliłby audiofilom zmieniać intensywność tonu o różnych częstotliwościach.

Kolejnym dobrym pomysłem na ulepszenie funkcjonalności naszego programu wydaje się możliwość odsłuchania danego utworu bez konieczności pobrania go najpierw. Myśleliśmy również nad opcją grupowania piosenek według tych najczęściej słuchanych zamiast np. alfabetycznie jak ma to miejsce teraz.