

Projektowanie aplikacji mobilnych

Temat projektu:

Aplikacja do nagrywania i przechowywania plików
dźwiękowych

Paweł Chmielarski

Informatyka III rok 2018/2019, IO1

1. Wstęp

1. Treść zadania

Aplikacja mobilna do nagrywania i przechowywania plików dźwiękowych

2. Cel

Projekt i implementacja powyższego zadania na platformę systemu Android. Jego celem jest zdobycie fundamentalnej wiedzy i umiejętności z zakresu programowania na platformę Android. Projekt jest wstępnym przygotowaniem do pracy inżynierskiej, która będzie wykorzystywała niektóre schematy i mechanizmy zaimplementowane w zadaniu.

3. Wymagania funkcjonalne:

F01.01. Użytkownik musi mieć możliwość nagrania ścieżki dźwiękowej o dowolnej długości, zatrzymując nagrywanie w wybranym momencie.

F01.02. Aplikacja musi umożliwiać odtworzenie ostatnio zapisanego nagrania oraz zatrzymanie odtwarzania.

F01.03. Aplikacja musi odmierzać i wyświetlać czas dla aktualnie wykonywanego nagrania oraz dla odtwarzania nagrania wcześniej nagranego

F01.04. Użytkownik może płynnie zmieniać punkt aktualnie odtwarzanego nagrania za pomocą paska postępu.

F01.05. Aplikacja musi zapisywać w pamięci urządzenia nagrane ścieżki dźwiękowe .

F01.06. Aplikacja musi umożliwiać wyświetlenie listy sporządzonych nagrań, odtworzenie każdego z nagrań oraz usunięcie wybranego nagrania.

F01.07. Użytkownik ma możliwość nadania nazwy plikowi audio, który ma domyślnie generowaną losową nazwę.

4. Wymagania niefunkcjonalne

N01.01. Aplikacja musi oferować prosty interfejs i intuicyjną obsługę.

N01.02. Aplikacja musi posiadać interfejs typu GUI i posiadać interaktywne elementy widoku.

N01.03. Aplikacja musi działać na systemie operacyjnym Android dla minimalnego sdk w wersji 19.

N01.04. Aplikacja ma być przeznaczona dla pojedynczego użytkownika.

N01.05. Aplikacja musi zapewniać pobranie od użytkownika wymaganych do działania uprawnień.

N01.06. Aplikacja musi działać w dowolnej orientacji urządzenia (portrait i landscape) oraz zapewniać odpowiednią skalowalność elementów widoku dla każdej orientacji.

N01.07. Aplikacja musi zapewniać odpowiednią wydajność

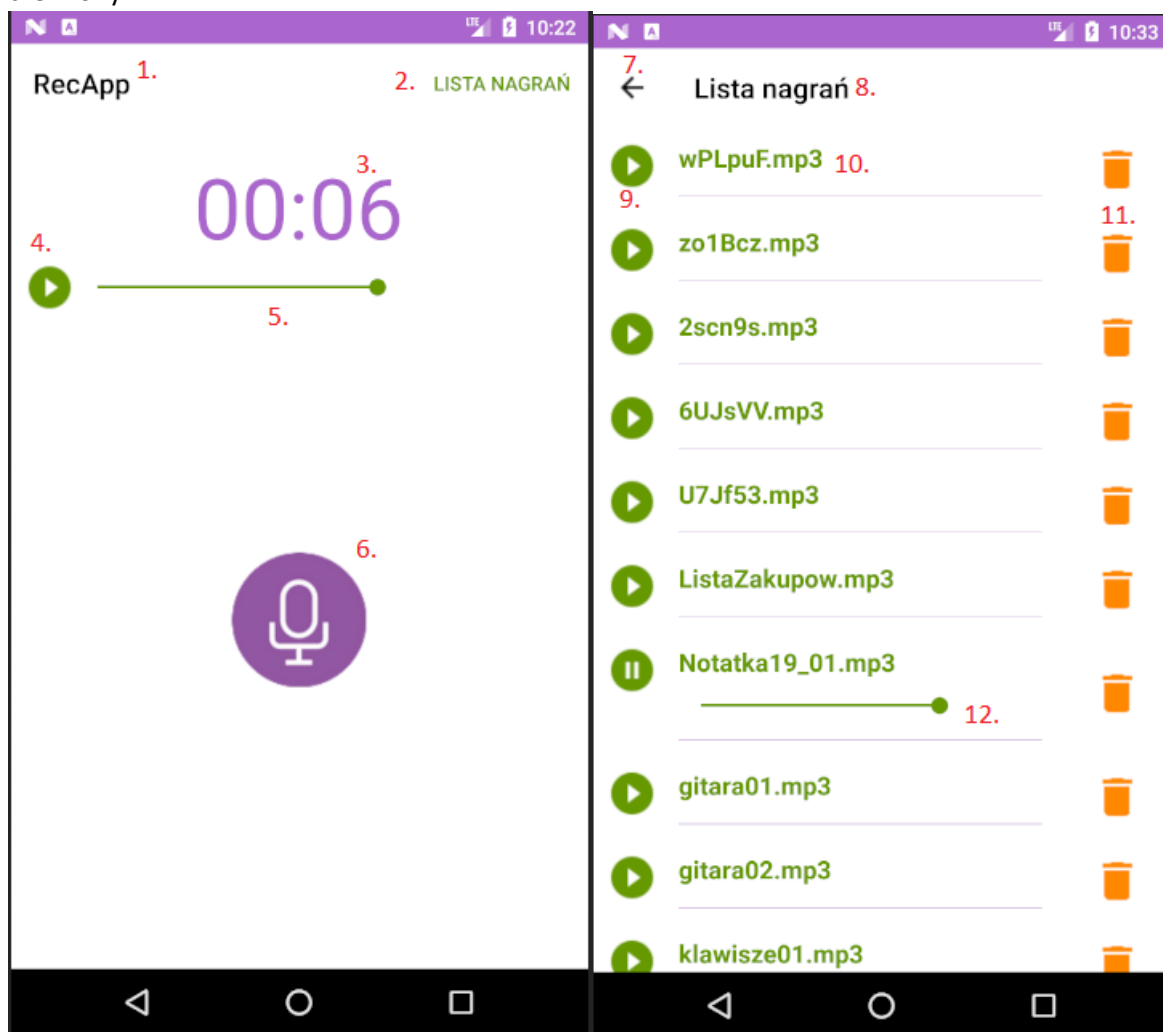
N01.08. Aplikacja musi zapewniać odporność na błędy spowodowane działaniami użytkownika oraz działaniem samej aplikacji, a także zapewniać bezpieczeństwo danych.

N01.09. Aplikacja musi zapewniać otwartość i możliwość rozbudowy w późniejszym terminie.

N01.10. Aplikacja musi umożliwiać zapisywanie plików w formacie .mp3

2. Specyfikacja zewnętrzna

Aplikacja posiada dwa widoki ekranu oraz jedno okno popup. Widoki zostały przedstawione na ilustracjach poniżej, a poszczególne ich elementy zostały opatrzone numeracją w kolorze czerwonym.



Okno główne aplikacji

Okno wyświetlające listę nagrań



Okno popup, umożliwiające nadanie plikowi nazwy

1. Nazwa Aplikacji.
2. Przycisk menu odpowiadający za otwarcie okna z listą nagrań.
3. Zegar odmierzający czas nagrania.
4. Przycisk „Play” rozpoczynający odtwarzanie nagrania. Po naciśnięciu ikona zmienia się na „Pauze” – wtedy element obiera funkcje zatrzymania odtwarzania.
5. Pasek postępu odtwarzania nagrania. Umożliwia też przewijanie nagrania.

6. Ikona rozpoczęcia nagrywania. Po naciśnięciu zmienia się na „Stop” – wtedy element obiera funkcje zatrzymania nagrywania.
7. Strzałka powrotu do ekranu głównego.
8. Nazwa bieżącego ekranu.
9. Ta sama funkcjonalność jak w punkcie 4.
10. Pole tekstowe wyświetlające nazwę nagrania.
11. Ikona umożliwiająca usunięcie nagrania.
12. Pasek postępu odtwarzania nagrania. Umożliwia też przewijanie nagrania.
13. Pole tekstowe umożliwiające wprowadzenie nowej nazwy dla pliku.
14. Przycisk zatwierdzający zmianę nazwy pliku.

Obsługa aplikacji polega na nagraniu ścieżki dźwiękowej za pomocą przycisku 6. Po ponownym dotknięciu ikony (która przyjmuje wygląd symbolu Stop) nagrywanie jest zatrzymywane. Wtedy użytkownik dostaje możliwość zmiany nazwy pliku w oknie popup. Następnie można odtworzyć przed chwilą zrobione nagranie za pomocą przycisku 4. W dowolnym momencie działania aplikacji możliwe jest otwarcie listy nagrań za pomocą przycisku 2. W oknie listy nagrań odbywa się zarządzanie wykonanymi nagraniami – każde z nich można odtworzyć lub usunąć.

3. Specyfikacja wewnętrzna

3.1. Program RecApp

Aplikacja RecApp została napisana w języku Java, w środowisku programistycznym Android Studio. Informacje dotyczące układu elementów na ekranie zapisano w plikach .xml. Program wykorzystuje natywne rozwiązania języka java i systemu Android.

3.2. Struktura aplikacji i elementy składowe

Kod programu zawiera się w czterech klasach: *MainActivity*, *RecListActivity*, *Recording* oraz *RecordingAdapter*. Dwie pierwsze klasy to klasy obsługujące aktywności aplikacji. Pozostałe klasy wykorzystywane są do wyświetlania nagrań w aktywności listy nagrań.

W katalogu *drawable* znajdują się ikony użyte w interfejsie aplikacji. Elementy te są przypisywane w kodzie do obiektów typu *imageView* i wykorzystywane do interakcji z użytkownikiem. W tym katalogu znajduje się też plik *dialog_bg.xml*, w którym opisano szczegółowy wygląd okna popup.

W katalogu *layout* znajdują się pliki .xml opisujące wygląd poszczególnych elementów widoku aplikacji. Pliki *activity_main.xml* oraz *activity_rec_list.xml* odpowiadają odpowiednio za widoki dwóch ekranów programu. *Popup_filename.xml* to wygląd okna popup, które pojawia się na ekranie w momencie nadawania plikowi nazwy. *Recording_item_layout.xml* odpowiada za wygląd pojedynczego elementu wyświetlanego w RecyclerView.

W katalogu *values*, w plikach *colours.xml* oraz *styles.xml* zapisano informacje dotyczące kolorów aplikacji.

W celu optymalizacji i zwiększenia efektywności wyświetlania elementów na ekranie listy nagrań zostało zaadaptowane rozwiązanie widoku RecyclerView, które jest obsługiwane w klasie *RecordingAdapter*.

Dokładne informacje dotyczące klas i metod znajdują się w katalogu doc.

3.3. Wykorzystane API i biblioteki

W aplikacji zastosowano szereg bibliotek umożliwiających obsługę wymaganych funkcjonalności. Są to głównie biblioteki multimedialne, obsługujące elementy widoku i przetwarzające dane. Lista wykorzystanych bibliotek:

Import android.content.Intent ; – uruchomienie nowej aktywności
import android.media.MediaPlayer; - odtwarzanie plików dźwiękowych
import android.media.MediaRecorder; - nagrywanie plików dźwiękowych
import android.os.Build; - sprawdzanie wersji sdk
import android.os.Handler; - uruchamianie nowego wątku do aktualizacji seekBar
import android.os.SystemClock; - ustawianie chronometra
import android.support.annotation.RequiresApi; - sprawdzanie wymaganego api
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.transition.TransitionManager;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Chronometer; - element wyświetlający czas
import android.widget.EditText; - wprowadzanie tekstu
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.PopupWindow; - okno popup
import android.widget.SeekBar; - pasek postępu odtwarzania
import android.widget.TextView; - wyświetlanie tekstu na ekranie
import android.widget.Toast; - do wyświetlania krótkich powiadomień na dole ekranu
import android.support.v7.widget.RecyclerView; - wyświetlanie elementów listy nagrań
import android.content.Context; - przekazywanie informacji o zasobach

import java.io.File; - obsługa plików
import java.io.IOException; - wyjątki wejścia/wyjścia
import java.security.SecureRandom; - generowanie liczb losowych
import java.util.ArrayList; - przechowywanie listy dostępnych nagrań

Aplikacja wymaga do poprawnego działania Sdk w wersji co najmniej 19.

4. Wnioski

- 4.1. W trakcie implementacji aplikacji autor natrafiał na szereg problemów, które wynikały z jego ograniczonej wiedzy na niektóre zagadnienia, gdyż jest to pierwszy tego typu program napisany przez autora. Brakująca wiedza była sukcesywnie zdobywana podczas pracy nad projektem. Główne problemy, które wystąpiły podczas pracy:
1. Zapis pliku w formacie .mp3 – w pierwszej wersji aplikacja zapisywała pliki w formacie .3gp i należało odpowiednio zostawić parametry dla klasy MediaRecorder, aby uzyskać pożądany format plików.
 2. Problem z poprawnym wyświetlaniem plików – część plików w ogóle nie była wyświetlana na liście nagrań. Po długim poszukiwaniu rozwiązania, które nie przyniosło rezultatów zdecydowano się na odmienne podejście do problemu z wykorzystaniem RecyclerView.
 3. Niepoprawne wyświetlanie czasu nagrania – element Chronometer niepoprawnie wyświetlał czas. Należało dodać zmienną *timeWhenStopped* oraz zmienić ustawienia obiektu chronometer w kilku miejscach kodu.
 4. Problemy z układem elementów na ekranie – problemy występowały głównie przy zmieniającej się orientacji urządzenia. Należało zmienić strukturę plików .xml
 5. Odpowiednie ustawienie i działanie RecyclerView – niezbędne było znalezienie informacji oraz przeanalizowanie kilku przykładowych kodów w celu zrozumienia zasady działania i sposobu implementacji elementu RecyclerView.
 6. Okno popup – problemy wynikały z odpowiedniego ustawienia elementu.
- 4.2. Niezaimplementowane funkcjonalności:
- Początkowo planowano dodanie do aplikacji funkcjonalności automatycznej synchronizacji zapisanych w katalogu aplikacji plików z dyskiem Google. Rozważano różne metody podejścia do problemu jednak ze względu na złożoność problemu oraz ograniczenia czasowe rozwiązania nie udało się zaimplementować.
- 4.3. Rozszerzenie funkcjonalności aplikacji:
- Planowane jest wykorzystanie niektórych elementów aplikacji w celu implementacji bardziej złożonego programu, który będzie umożliwiał nagrywanie i odtwarzanie w pętli wielu ścieżek dźwiękowych oraz nakładanie efektów dźwiękowych.
- Rozważana jest też możliwość dodania niezaimplementowanej aktualnie funkcjonalności, jaką jest zapis plików do chmury.