МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Звіт з лабораторної роботи №3

«Музичний програвач»

з дисципліни «Програмування під Android»

Виконала:

ст. групи ПI-15-1

Михневич Т. К.

Прийняв:

ст. викл. каф. ПІ

Сокорчук І. П.

Харків 2018

**Завдання**

Розробити музичний програвач. Програвання музики виконувати за допомогою сервісу. При вхідному дзвінку та до його закінчення програвання призупиняється. В програвачі реалізувати функції перегляду списку медіафайлів пристрою, обрання будь-якого файлу і відтворення з відображенням прогресу. По закінченню програвання відбувається автоматичний перехід на наступний файл. Підтримуються режими зациклення списку і випадкового переходу. При згортанні програвача в фон програвання не зупиняється, а ознака запущеного плеєра переходить в розділ Notification з індексацією прогресу.

**Опис додатку**

У результаті виконання лабораторної роботи був розроблений додаток для програвання аудіофайлів з мобільного пристрою.

Після запуску додатку відкривається головний екран (див. рис. 1). На головному екрані відображається список наявних аудіофайлів пристрою. Для кожного з файлів відображаються назва, виконавець та альбом.

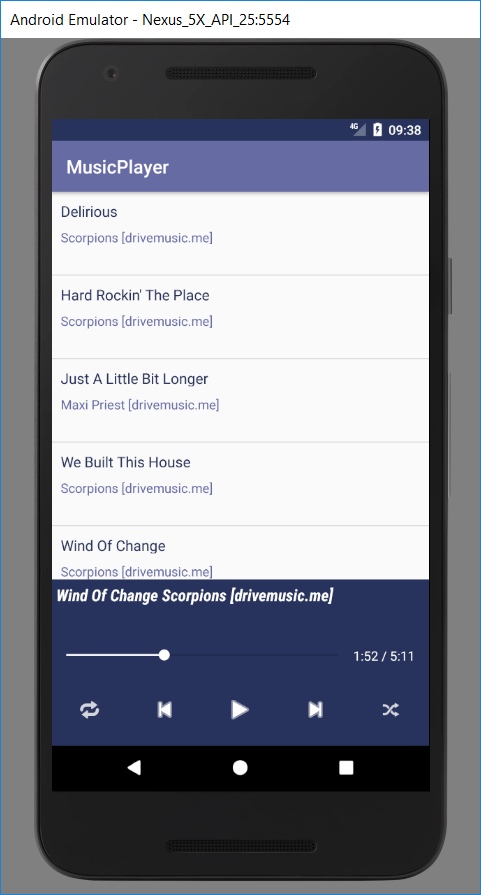


Рисунок 1 – інтерфейс головного екрану додатку

Після обрання будь-якого файлу відбувається його відтворення з відображенням прогресу. Прогрес відображається за допомогою SeekBar та відображення поточного часу відтворення аудіофайлу. По закінченню програвання відбувається автоматичний перехід на наступний файл. Підтримуються режими зациклення списку і випадкового переходу. Для зупинки, відтворення, переходу на наступний або попередній файли використовуються відповідні кнопки.

При згортанні програвача в фон програвання не зупиняється, а ознака запущеного плеєра переходить в розділ Notification з індексацією прогресу (див. рис. 2). У нотифікації відображена назва аудіофайлу, виконавець та поточний час відтворення аудіофайлу. Коли додаток стає активним знову, нотифікація зникає.

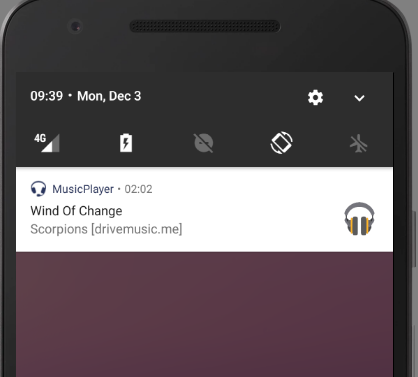


Рисунок 2 – нотифікація додатку MusicPlayer

**Опис реалізації**

Для відображення списку файлів був використаний клас, успадкований від ArrayAdapter.

Елемент списку представлений класом Audio:

**public class** Audio **implements** Serializable {  
  
 **private** String **data**;  
 **private** String **title**;  
 **private** String **album**;  
 **private** String **artist**;  
  
 **public** Audio(String data, String title, String album, String artist) {  
 **this**.**data** = data;  
 **this**.**title** = title;  
 **this**.**album** = album;  
 **this**.**artist** = artist;  
 }  
  
 **public** String getData() {  
 **return data**;  
 }  
  
 **public void** setData(String data) {  
 **this**.**data** = data;  
 }  
  
 **public** String getTitle() {  
 **return title**;  
 }  
  
 **public void** setTitle(String title) {  
 **this**.**title** = title;  
 }  
  
 **public** String getAlbum() {  
 **return album**;  
 }  
  
 **public void** setAlbum(String album) {  
 **this**.**album** = album;  
 }  
  
 **public** String getArtist() {  
 **return artist**;  
 }  
  
 **public void** setArtist(String artist) {  
 **this**.**artist** = artist;  
 }  
}

Фрагмент коду 1 – Клас Audio

Для відображення прогресу використаний SeekBar. Усі елементи головного екрану додатку розміщенні у LinearLayout.

При запуску додатку у пам’ять додатку завантажується список наявних на пристрої аудіофайлів за допомогою наступної функції:

**private void** loadAudio() {  
 Uri uri = MediaStore.Audio.Media.***EXTERNAL\_CONTENT\_URI***;  
 String selection = MediaStore.Audio.Media.***IS\_MUSIC*** + **"!= 0"**;  
 String sortOrder = MediaStore.Audio.Media.***TITLE*** + **" ASC"**;  
 Cursor cursor = getContentResolver().query(uri, **null**, selection, **null**, sortOrder);  
  
 **if** (cursor != **null** && cursor.getCount() > 0) {  
 **\_audioList** = **new** ArrayList<>();  
 **while** (cursor.moveToNext()) {  
 String data = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.***DATA***));  
 String title = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.***TITLE***));  
 String album = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.***ALBUM***));  
 String artist = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.***ARTIST***));  
  
 **\_audioList**.add(**new** Audio(data, title, album, artist));  
 }  
 cursor.close();  
 }  
}

Фрагмент коду 2 – Функція завантаження аудіофайлів

Програвання здійснюється за допомогою сервісу. У сервісі реалізовані функції програвання, зупининня, відновлення програвання аудіофайлів, обробки вхідних дзвінків, побудови та видалення нотифікації, функції, що повертають інформацію про загальний час аудіофайлу та поточний час програвання у конкретний момент та інші.

Реалізований сервіс за допомогою класу AudioPlayerService, що успадкований від класу Service. В успадкованому класі перевизначаються методи onBind, onUnbind. У сервісі використовуються такі класи, як MediaPlayer та AudioManager.

MainActivity шле сервісу броадкасти для управління аудіофайлами, що програються, наприклад:

Intent broadcastIntent = **new** Intent(***BROADCAST\_PLAY\_NEW\_AUDIO***);  
sendBroadcast(broadcastIntent);

Фрагмент коду 3 – Надсилання броадкасту

У методі onPause MainActivity визивається побудова нотифікації та її відображення. У методі onResume нотифікація прибирається.

**Висновки**

У результаті виконання лабораторної роботи був розроблений музичний програвач. Програвання аудіофайлів виконується за допомогою сервісу. У сервісі реалізовані також обробка вхідних дзвінків та побудова нотифікацій. Список аудіофайлів відображається за допомогою ArrayAdapter. Для відображення використовується LinearLayout.