



Actividad 1 - Instalación Android Studio y Diseño de Aplicación

Desarrollo de Aplicaciones Móviles II

Ingeniería en Desarrollo de Software

Tutor: Eduardo Israel Castillo García

Alumno: Jusi Ismael Linares Gutiérrez

Fecha: 1/04/2024

Índice

Introducción	3
Descripción	3
Justificación	3
Desarrollo:	4
Interfaz	4
Codificación.....	4
Prueba de la aplicación	6
Conclusión	7
GitHub	7

Introducción

En respuesta a la inminente llegada del Buen Fin, se ha encomendado a un ingeniero en desarrollo de software la creación de una aplicación móvil para Android. Esta aplicación tiene como objetivo principal proporcionar a los usuarios una galería de ringtones, ofreciendo una experiencia personalizada y divertida para sus dispositivos móviles. En la primera actividad, se ha iniciado el desarrollo de esta aplicación, estableciendo las bases para su funcionamiento y diseño. Ahora, en esta segunda actividad, se dará continuidad al proyecto, incorporando una nueva característica que mejorará la funcionalidad y la utilidad de la aplicación. La adición de la capacidad de compartir cada uno de los audios a través de redes sociales, correo electrónico o Bluetooth ampliará las opciones y la versatilidad de la aplicación, permitiendo a los usuarios compartir sus ringtones favoritos con amigos, familiares y contactos. Esta introducción establece el contexto y el propósito de la actividad, destacando la importancia de ofrecer una experiencia completa y satisfactoria para los usuarios durante el Buen Fin y más allá.

Descripción

La aplicación debe funcionar como una galería de ringtones, lo que implica la capacidad de gestionar y reproducir una variedad de archivos de audio que pueden ser utilizados como tonos de llamada en dispositivos móviles.

En la segunda parte de la actividad, se solicita agregar una nueva característica a la aplicación existente: la capacidad de compartir los audios de la galería a otros dispositivos a través de redes sociales, correo electrónico o Bluetooth. Esto significa que el usuario podrá seleccionar un tono de llamada específico y compartirlo con amigos o familiares utilizando diversas plataformas de comunicación.

En resumen, el objetivo es mejorar la funcionalidad de la aplicación de galería de ringtones al permitir que los usuarios compartan fácilmente sus tonos de llamada favoritos con otros dispositivos y personas, lo que aumenta la utilidad y la experiencia del usuario.

Justificación

La inclusión de la función de compartir audios a través de redes sociales, correo o Bluetooth en la aplicación de galería de ringtones para el Buen Fin es fundamental para mejorar la experiencia del usuario y aumentar la utilidad de la aplicación. Esta característica permite a los usuarios compartir sus tonos de llamada favoritos con amigos y familiares de una manera conveniente y rápida, lo que fomenta la interacción social y la difusión de la aplicación.

Además, facilita la promoción de la aplicación, ya que los usuarios pueden compartir sus experiencias positivas con otros, lo que potencialmente aumentará el número de descargas y la visibilidad en las tiendas de aplicaciones. Asimismo, al permitir compartir a través de redes sociales, se abre la posibilidad de llegar a un público más amplio y diverso.

La inclusión de esta función también responde a las expectativas del mercado actual, donde la conectividad y la capacidad de compartir contenido son aspectos clave en la experiencia del usuario.

Desarrollo:

Interfaz

Codificación

Empezaremos por crear un image button con el icono de bluetooth tal y como hemos hecho con los iconos anteriores

```
<!-- Botón de compartir por Bluetooth-->
<ImageButton
    android:id="@+id/btnBlue1"
    android:layout_width="30dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:scaleType="fitCenter"
    android:src="@drawable/icon_bluetooth" />
```

El siguiente código crea una función llamada assignShareListener. Esta función toma dos parámetros: el id de un botón de imagen (ImageButton) y el nombre de un archivo de audio (fileName). Cuando se hace clic en el botón de imagen, se ejecuta una acción. Dentro de esta acción, el código obtiene el ID del recurso del archivo de audio utilizando el nombre proporcionado y la carpeta raw. Luego, crea una URI que apunta a este recurso de audio. A continuación, crea un intento para compartir (shareIntent) y lo configura para enviar un archivo de audio. Finalmente, utiliza startActivity con un selector de aplicaciones (Intent.createChooser) para permitir al usuario seleccionar la aplicación a través de la cual desea compartir el archivo de audio

```
private fun assignShareListener(button: ImageButton, fileName: String) {
    button.setOnClickListener { it: View!
        val resID = resources.getIdentifier(fileName, defType: "raw", packageName)
        val audioUri = Uri.parse(uriString: "android.resource://$packageName/$resID")

        val shareIntent = Intent().apply { this: Intent
            action = Intent.ACTION_SEND
            type = "audio/mp3"
            putExtra(Intent.EXTRA_STREAM, audioUri)
        }

        startActivity(Intent.createChooser(shareIntent, title: "Compartir audio"))
    }
}
```

Este bloque de código crea una función llamada `assignBluetoothShareListener`. Toma dos parámetros: un botón de imagen (`ImageButton`) y el nombre de un archivo de audio (`fileName`). Cuando se hace clic en el botón de imagen, se ejecuta una acción. Dentro de esta acción, el código obtiene el ID del recurso del archivo de audio y crea una URI que apunta a este recurso. Luego, crea un intento para compartir audio, especificando el tipo de archivo como audio y adjuntando la URI del audio al intento. Además, especifica que se comparta a través de Bluetooth

```
private fun assignBluetoothShareListener(button: ImageButton, fileName: String) {  
    button.setOnClickListener { it: View! ->  
        // Obtener el ID del recurso de audio  
        val resID = resources.getIdentifier(fileName, defType: "raw", packageName)  
        // Crear la URI del recurso de audio  
        val audioUri = Uri.parse(uriString: "android.resource://$packageName/$resID")  
  
        // Crear el intent para compartir audio  
        val shareIntent = Intent().apply { this: Intent ->  
            action = Intent.ACTION_SEND  
            // Especificar el tipo de archivo de audio  
            type = "audio/*"  
            // Adjuntar la URI del audio al intent  
            putExtra(Intent.EXTRA_STREAM, audioUri)  
        }  
  
        // Especificar que se comparta a través de Bluetooth  
        shareIntent.setPackage("com.android.bluetooth")  
  
        // Mostrar el selector de aplicaciones para compartir  
        startActivity(Intent.createChooser(shareIntent, title: "Compartir audio por Bluetooth"))  
    }  
}
```

Este bloque de código crea y configura botones de imagen para compartir archivos de audio mediante Bluetooth. Se crean 10 botones de imagen (btnBlue1 a btnBlue10) que se asignan a través de sus respectivos identificadores de recursos (R.id.btnBlue1 a R.id.btnBlue10). Luego, se llama a la función assignBluetoothShareListener para cada botón de imagen, pasando el botón y el nombre del archivo como parámetros. Esta función configura cada botón para que, cuando se haga clic en él, se comparta el archivo de audio asociado a través de Bluetooth

```
//Seccion de compartir por diente azul
val btnBlue1: ImageButton = findViewById(R.id.btnBlue1)
val btnBlue2: ImageButton = findViewById(R.id.btnBlue2)
val btnBlue3: ImageButton = findViewById(R.id.btnBlue3)
val btnBlue4: ImageButton = findViewById(R.id.btnBlue4)
val btnBlue5: ImageButton = findViewById(R.id.btnBlue5)
val btnBlue6: ImageButton = findViewById(R.id.btnBlue6)
val btnBlue7: ImageButton = findViewById(R.id.btnBlue7)
val btnBlue8: ImageButton = findViewById(R.id.btnBlue8)
val btnBlue9: ImageButton = findViewById(R.id.btnBlue9)
val btnBlue10: ImageButton = findViewById(R.id.btnBlue10)

assignBluetoothShareListener(btnBlue1, fileName: "arashi_no_saxophone2")
assignBluetoothShareListener(btnBlue2, fileName: "beyond_journeys_end")
assignBluetoothShareListener(btnBlue3, fileName: "chala_head_chala")
assignBluetoothShareListener(btnBlue4, fileName: "chavo_del_8")
assignBluetoothShareListener(btnBlue5, fileName: "fairy_fountain")
assignBluetoothShareListener(btnBlue6, fileName: "gusty_garden_galaxy")
assignBluetoothShareListener(btnBlue7, fileName: "jurassic_park_3_satellite_phone")
assignBluetoothShareListener(btnBlue8, fileName: "pokemon_battle_gold_and_silver")
assignBluetoothShareListener(btnBlue9, fileName: "tphone_ringtone")
assignBluetoothShareListener(btnBlue10, fileName: "zelda_theme")
```

Prueba de la aplicación

Link del video donde se prueba la funcionalidad de la aplicación

(La función bluetooth no se puso a prueba debido a que no se logro su funcionalidad, aun así se anexa el código para su funcionamiento)

<https://drive.google.com/file/d/15KvhOZmN4xAuCyRECI5b9PZsWUbedyfc/view?usp=sharing>

Conclusión

Desarrollar una aplicación móvil para gestionar una galería de tonos de llamada en Android antes del Buen Fin es una estrategia inteligente. Al añadir la función de compartir audios mediante redes sociales, correo electrónico o Bluetooth, se amplía su utilidad y alcance. Los usuarios pueden compartir fácilmente sus tonos de llamada favoritos, fomentando la interacción y promoviendo la adopción de la aplicación. Esta característica añade valor al ofrecer una forma conveniente de difundir la aplicación y sus contenidos, lo que puede aumentar la satisfacción del usuario y su fidelidad. En resumen, la implementación de esta función mejora significativamente la experiencia del usuario y fortalece la posición de la aplicación en el mercado, especialmente durante eventos comerciales clave como el Buen Fin. Además, proporciona una oportunidad valiosa para conectar con los usuarios y aumentar la visibilidad de la aplicación a través del boca a boca y las redes sociales.

GitHub

https://github.com/JusiLinGu/Practicas_UMI