Media Player

Phase I: Proposal

Group 21
1155175490 LI Hoi Hin
1155175164 LU Man Ho
1155216903 MATTHAEI Casper Ludwig

Department of Computer Science and Engineering
The Chinese University of Hong Kong

February 20, 2024

1. Introduction
Our media player application aims to provide a comprehensive solution for playing various media files, including videos and audio, with additional features like picture-in-picture mode, subtitle support, sharing functionality, and file system management.

We have taken some real-world android application references. For example, VLC Media Player and MX Player. The former is a free and open source cross-platform multimedia player that plays most multimedia files as well as discs, devices, and network streaming protocols, while the latter is a Powerful video and music player with advanced hardware acceleration and subtitle support.

However, some potential variations may exist in our application to differentiate our application from the aforementioned out-in-the-market applications, such as playlist management, which allow users to create and manage playlists for their media files, and custom themes and UI customization, which provide options for users to personalise the application's appearance and user interface.

2. Features
Our app aims to offer a versatile media player experience with the following features:

- Support for a wide range of media formats, including video formats like .mp4, .mov, and audio formats like .mp3, .wav etc.
- Timeline scrubber: Users can navigate through the media files using a seek bar (a type of progress bar) or a timeline scrubber.
- Picture-in-picture (PiP) mode: Users can continue watching videos in a small floating window while using other applications.
- Background music playback: Audio files can be played even when the app is not active on the screen.
- Volume adjustment: The application allows the user to adjust the volume of the playing media.
- Subtitle support: The application can load and display .srt subtitle files synchronized with the media playback.
- Share functionality: Users can easily share media files through various messaging and social media applications.
- File system management: Users can designate a specific folder or storage space to store media files, manage storage, and import new media directly within the application.

3. Methodologies
The language we are using in the project is Java, and the application will mainly be implemented using AndroidX Media, which is a collection of libraries that allows developers to implement rich media features on Android applications easily and effectively [3]. It allows for easier development for the following functionalities:

3.1. Media Playback:
This section refers to the functionality related to loading and playing a media on screen.

3.1.1. Support for various media formats
The ExoPlayer, one of the libraries included in the AndroidX Media, supports the container formats specified in [4] directly. Since all media files are loaded locally on the device, the program uses Uri.fromFile(new File(path)) to retrieve and load the specified media to play [5]. If there exist media

formats that are not supported by ExoPlayer, specifically audio files, we may use the ExoPlayer FFmpeg module to allow decoding and encoding files into various media formats [6].

3.1.2. Timeline scrubber with accurate seeking and playback controls.
Library "PreviewSeekBar" will be used to implement the timeline scrubber in activity for better accuracy for navigating the media progress [7].

3.1.3. Picture-in-picture mode for videos
This can be done by switching the mode of activity to picture-in-picture mode [8] and setting "setAutoEnterEnabled" flag to true for Android 12 or above. For Android 11 and below, enterPictureInPictureMode() in the activity class is evoked when the user switches to another application.

3.1.4. Background music playback for audio files
In order to have music keep on playing by the application while the phone is in locked mode or is using other applications as foreground on screen, code components "ExoPlayer" and "Media Session" are implemented in a separate service and extends from MediaSessionService. There will also be a MediaStyle notification automatically created by MediaSessionService that displays player controls and keeps the user up to date on the progress of the media session [9].

3.1.5. Volume Adjustment
This can be implemented by creating a UI element RangeSlider and connecting the value obtained in the slider to the function that handles the volume of media. Then use e.g. MediaPlayer.setVolume in the ExoPlayer to set the volume of the media directly.

3.2. Subtitle Support
Subtitles tracking can be implemented using MediaItem.SubtitleConfiguration.Builder in the ExoPlayer library, then setup and play the subtitles through Player.setMediaItem(subtitles), Player.prepare() and Player.play() [11]. Note that Player is an ExoPlayer object.

3.3. Share functionality
3.3.1. Support for sharing files from the media player to destinations chosen by the user
This can be done by using intents to request other apps to receive the file. For android there already exists a strongly recommended method for sharing to other apps called Android Sharesheet [12]. To use Android Sharesheet Intent.createChooser() is used in combination with Intent.ACTION_SEND [12].

3.3.2. Button Placement
The button to open the Android Sharesheet also has to be well thought out and has to be placed so that it is easy to access and still be aesthetically pleasing.

3.4. File system management
3.4.1 Storage
Storage for accessing, storing, managing and renaming files within the application can easily be implemented with Android Studio via the File function by importing java.io.*.

3.4.2 UI & Design

On application storage management design has to be easy to use and aesthetically pleasing. Users will be able to create their own storage system within the application through creating folders with the File function.

3.4.3 Import media

Enabling the user to import media from internal storage to the application file system from within the application. This requires the app to gain permission to access the internal storage, if the user does not give the application permission this feature will ask for permission when tried to be used.

4. Ranking for the implementation of features

Ranking from the easiest to implement to the hardest to implement:

1. Share Function: relatively straightforward as Android Studio has built-in API.
2. UI: not hard to make UI, the hard part is how to make it appealing
3. File Management: need to handle file I/O, permission, storage access and organising files in the app, also need to consider different file formats like videos, audios and subtitles files
4. Video Playback: involve many components, some functions are challenging
5. Subtitle Support: complex when synchronizing the subtitles and displaying them correctly to the video playback, challenging when handling different subtitle format and provide user control to the subtitles

5. Workload of Groupmates

LI Hoi Hin: Methodologies: Media Playback, Subtitle Support; Ranking
LU Man Ho: Introduction; Features; Ranking
MATTHAEI Casper Ludwig: Methodologies: Share functionality, File System Management

6. Reference

[1] VLC for Android, Google Play,
https://play.google.com/store/apps/details?id=org.videolan.vlc&hl=en&gl=US&pli=1
Last access: 15 Feb, 2024

[2] MX Player, Google Play,
https://play.google.com/store/apps/details?id=com.mxtech.videoplayer.ad&hl=en&gl=US
Last access: 15 Feb, 2024

[3] Introduction to Jetpack Media3, Android Developers,
https://developer.android.com/media/media3

[4] Progressive, Android Developers,
https://developer.android.com/media/media3/exoplayer/progressive

[5] Uri, fromFile, Android Developers,
https://developer.android.com/reference/android/net/Uri#fromFile(java.io.File)

[6] FFmpeg Extension, Android Developers,
https://exoplayer.dev/supported-formats.html#ffmpeg-extension

[7] PreviewSeekBar
https://github.com/rubensousa/PreviewSeekBar

[8] Picture-in-picture, Android Developers,
https://developer.android.com/develop/ui/views/picture-in-picture

[9] Background Playback, Android Developers,
https://developer.android.com/media/media3/session/background-playback

[10] Interface Player in ExoPlayer Document
https://exoplayer.dev/doc/reference/com/google/android/exoplayer2/Player.html#setVolume(float)

[11] stackoverflow: exoplayer - Enable subtitle on video
https://stackoverflow.com/questions/70170842/exoplayer-enable-subtitle-on-video

[12] Send simple data to other apps, Android Developers,
https://developer.android.com/training/sharing/send.html