MUSIC VIDEO APPLICATION DEVELOPMENT USING ANDROID

BY

LIM YU WEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMMUNICATIONS AND NETWORKING (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2021

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Music Video Application Development Using Android

Academic Session: MAY 2021

I LIM YU WEN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ Ooi Chek Yee

(Author's signature) (Supervisor's signature)

Address:

67, Hala Pengkalan Barat 14,

Taman Pengkalan Barat,

31650 Ipoh, Perak.

Ooi Chek Yee

Supervisor's name

Date: 25th August 2021 **Date**: 25th August 2021

MUSIC VIDEO APPLICATION DEVELOPMENT USING ANDROID

BY

LIM YU WEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMMUNICATIONS AND NETWORKING ((HONOURS))

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2021

DECLARATION OF ORIGINALITY

I declare that this report entitled "**MUSIC VIDEO APPLICATION DEVELOPMENT USING ANDROID**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : LIM YU WEN

Date : 25th August 2021

ACKNOWLEDGEMENTS

First and foremost, I would like to appreciate and express my gratefulness to my supervisor, Ts. Dr Ooi Chek Yee who willing to help me regarding my proposed title for the final year project, which is Music Video Application Development Using Android. He has given me guidance and support throughout the project with his knowledge and experiences which has led me to propose several great ideas during the development of this project.

Lastly, I would like to thank my parents and friends who are supportive and encouraging during the development process. This project would not be developed successfully with their support mentally.

ABSTRACT

This project is mainly about the development of a music video application using Android. It is also known as a media player that works on the Android native operating system. In this project, it will be able to get the music or video from the local storage of smartphone devices and list down the music and video. Besides that, it also consists of three ways to control the music player which are button click, voice command, and swipe gesture. With these ways, users can control the music player easily as they just need to speak out some specific command to control the music player. For the video player, the user can easily view the video in portrait or landscape mode. The user also can enlarge the video view into a full screen to have a better view. However, there are some user interfaces of the existing mobile application that are functional but very messy at the same time. Therefore, this project will design a simple, clean and clear interface to solve this problem and let users have a better user experience. Furthermore, the methodology of system analysis and design to develop this project is Mobile Application Development Life Cycle (MADLC). Mobile Application Development Life Cycle consists of six phases which are Planning, Design, Development, Testing, Release, and Maintenance. With the help of these six phases in the Mobile Application Development Life Cycle, the development of this project will be smoother and can minimize the bugs during the developing and testing stage.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DJ	Disk Jockey
CD	Compact Disc
UML	Unified Modeling Language
RAM	Random Access Memory
GB	Gigabyte
XML	Extensible Markup Language

CHAPTER 1 INTRODUCTION

1.1 Problem Statement and Motivation

a. Difficult to search for music and video

This problem commonly occurs for users who do not have a high-level of proficiency in the language in reading, writing, and spelling. These users may have trouble when they want to play some music or video, but they do not know which playlist that they have saved their music or video in the application. Therefore, they may need to type the specific artist name or the song name in the search bar of the application to get the music or video that they want but they may have the problem of the spelling or writing the name of the music and video. This problem will cause the users to have difficulties in searching for the music or video that they wanted to play.

b. Features that rarely being use

This problem will occur because a music video application is mainly used to listen to music or watch video to enjoy the free time but some of the music video application consists of different settings and features while playing every music or video such as equalize the music into soft, dynamic, base boost versions. These features may not applicable to users that are non-professional in this field and it is applicable for users that are professional such as music composer, DJ or other related job.

c. Membership Privilege

For some of the music video applications, they require user to subscribe with the daily, weekly, monthly or annually subscription plan to unlock the all the advance features. For the subscription plan does not means that users need to buy every song or video but they need to subscribe to unlock some of the specific features, music or videos. These subscription plan normally to remove the advertisement or unlock features like sharing music or comment on the music.

1.2 Background Information

As technology gets progressively advance, there are lesser and lesser people listen to CDs and radio. The majority of them usually download the music or video to listen or watch it, enjoy the music or video from the online platform or stream it online rather than listen to CDs and radio. For playing any music or video, a mobile application is a must to execute it and media player may be one of the right choices to execute the process. Media player is an application that used to play audio and video. The purpose of this proposed application is to provide user with another way to enjoy their music and video rather than using web-based application as it may require the internet connection. By using the music video mobile application, it does not need the internet connection and it will be more user friendly to the users.

1.3 Project Objectives

a. Play music and video with able to view the music and video playlist

This proposed application is a music video application. Therefore, the main function of this application to play music and watch video and able to view the music and video playlist.

b. Include the useful and important features

For this music video application, it will integrate with some useful and important features such as different ways to control the music player, playing music in the background or others features to make it easier and faster to find out the media that user wants.

c. Clean and clear interface to make the application user friendly

This music video application will have a clean, clear and neat interface to differentiate all the categories to let the user easier to find out the features that they want. Hence, it will become user friendly to the users.

1.4 Achievement

The proposed mobile application's name is called Infinity Player, which is a media player that able to play audio and video. This mobile application was done by using Android Studio IDE.

In the proposed mobile application, the audio player has been implemented with swipe gesture control, voice command control, shuffle and repeat function. For the swipe gesture control, the

different functions will be performed by swiping in a different direction. By swiping up or down, the volume of the playing audio will be increased or decreased (Figure 1.4.1 and Figure 1.4.2). By swiping left or right, the next song or previous song will be played (Figure 1.4.3 and Figure 1.4.4).

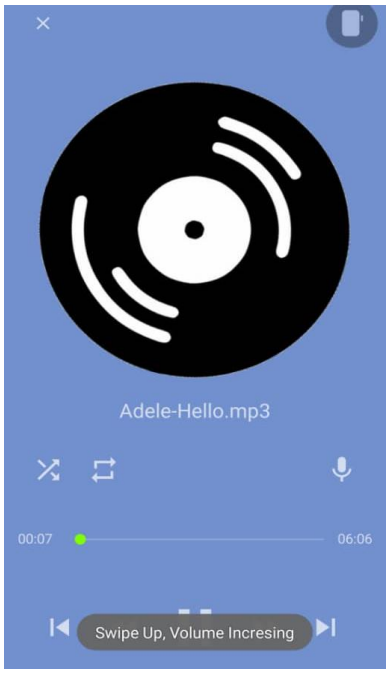


Figure 1.4.1: Swipe Gesture Up

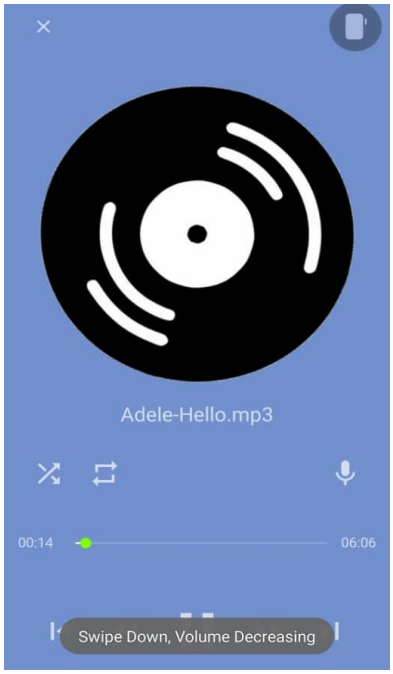


Figure 1.4.2: Swipe Gesture Down

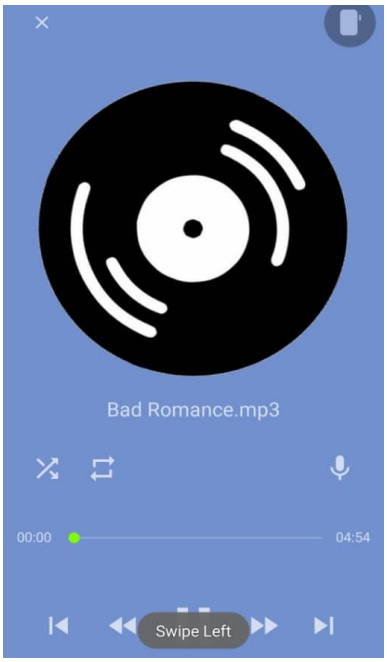


Figure 1.4.3: Swipe Gesture Left

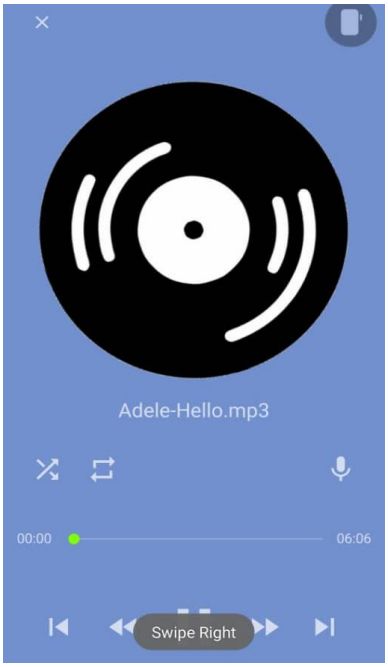


Figure 1.4.4: Swipe Gesture

For the voice command control, when the user triggers the microphone button, a pop-up google voice recognition window will be shown and the user can start to speak some specific command to perform the function such as play, pause, next, and previous (Figure 1.4.5 to Figure 1.4.8).

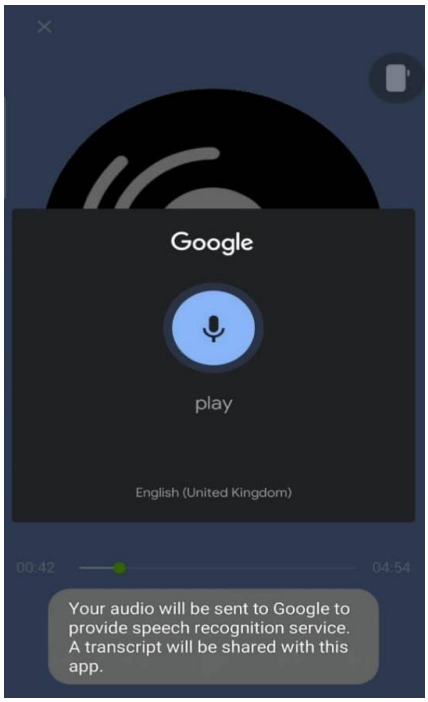


Figure 1.4.5: Voice Command "Play"

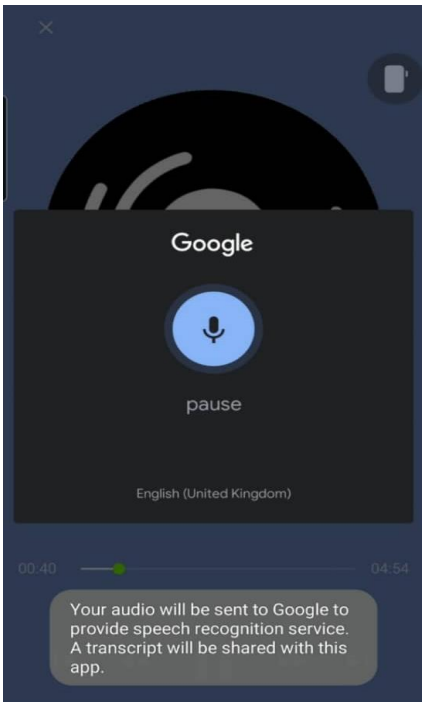


Figure 1.4.6: Voice Command "Pause"

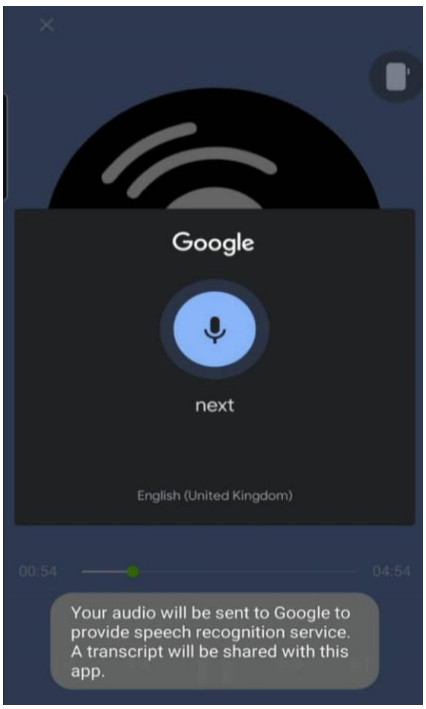


Figure 1.4.7: Voice Command "Next"

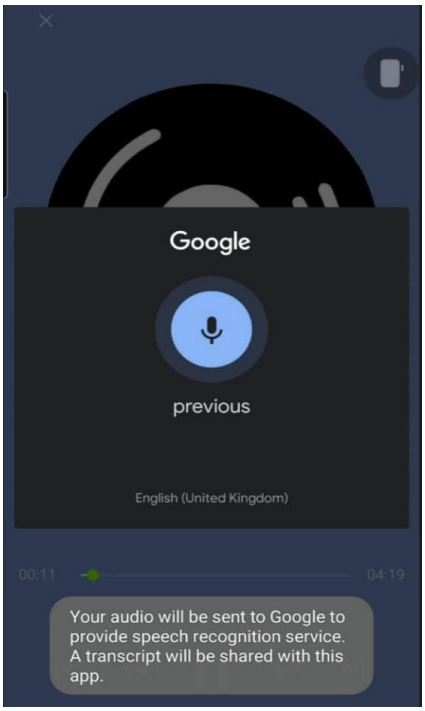


Figure 1.4.8: Voice Command "Previous"

Besides that, for the shuffle and repeat function, when the user triggers the shuffle button, the audio will be played in a random arrangement; when the user triggers the repeat button, the audio will be loop continuously (Figure 1.4.9 and Figure 1.4.10).

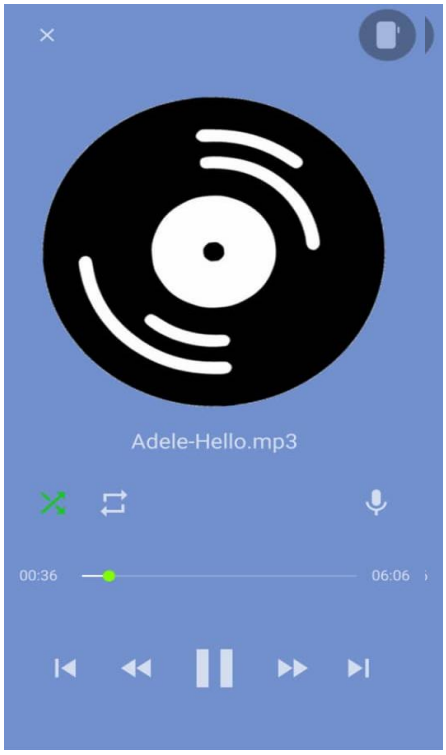


Figure 1.4.9: Shuffle Function

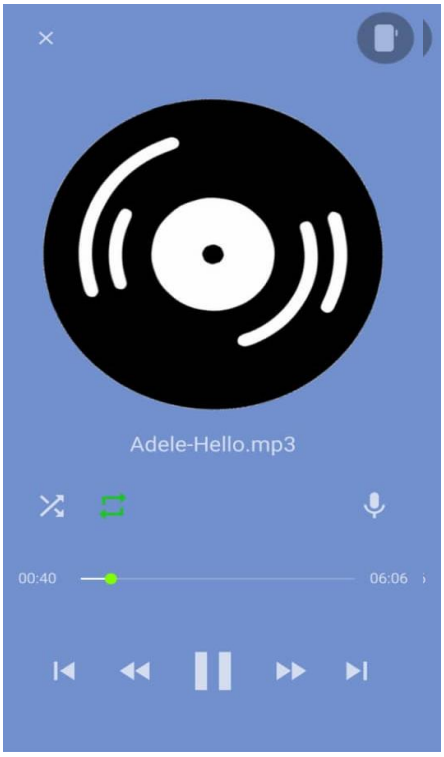


Figure 1.4.10: Repeat Function

1.5 Report Organization

This report consists of six chapters that explain the different details regarding the project. Chapter One will discuss the problem statements and motivation, background information, project objective, and the achievements of this project.

For Chapter Two, it is the literature review section. Reviews are done on three similar concepts mobile applications. The functionality of these projects is listed with their respective strength and weakness. A comparison between the proposed work and previous work is provided.

Chapter 3 is about the methodology, tools that have been used, and system design. The methodology, tools used to develop the entire project will be explained and the version of the software is stated as well. The functionalities of the project are explained in a more detailed manner. Several diagrams that will be shown such as system flowchart and use case diagram to further clarify the flow of the system. Besides that, the interface design of the project will also be included for a better understanding of the functionality provided.

As in Chapter Four, it will discuss the implementation of the system. Some of the setup and configuration is clearly stated in Chapter Five follow by some screenshots that are related to the development of the project.

Lastly, Chapter Five will be the conclusion which contains the summary of the entire project and discussion about the improvements that can be done to the system in the future.

CHAPTER 2 LITERATURE REVIEW

2.1 Review and Comparison of Previous Work

2.1.1 Review on JOOX

JOOX is a music streaming application that is available in the Google Play Store and App store which was released by Tencent. It provides up to two million songs from all around the world. To use JOOX, the user needs to register an account or owned an account. In JOOX, you can create our playlist that suit your mood and they also provide you a few of playlist such as “Top Download”, “New Releases Recommended”, “Featured Artist” that may suit your style. Besides that, JOOX will also provide different categories of the playlist as artist channels, theme playlists, new releases, and others. Other than that, another feature of JOOX is the user can import their songs that had been stored on the device.

In JOOX, it has a messy interface. It consists of a bottom navigation bar that indicates a different page in the application, which is Discover, Search, Buzz, Library, and Account (Figure 2.1.1.1). On the library page, the user can view their library such as the songs that favorited, downloaded, and stored on the device. There are also some recommended albums shown on the Library page (Figure 2.1.1.2).

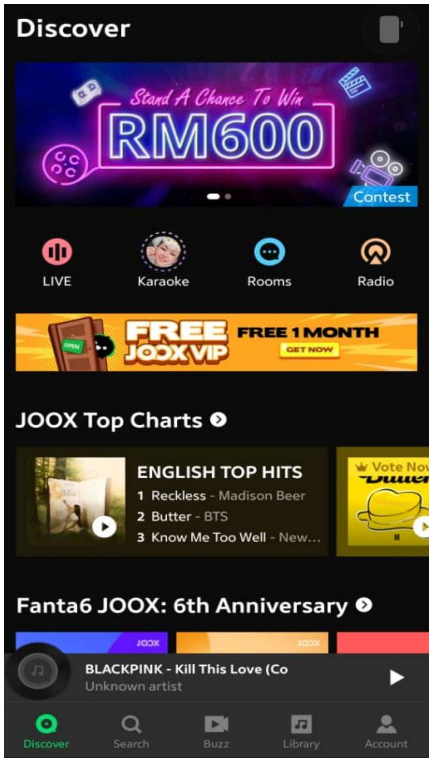


Figure 2.1.1.1: Bottom navigation menu bar in JOOX

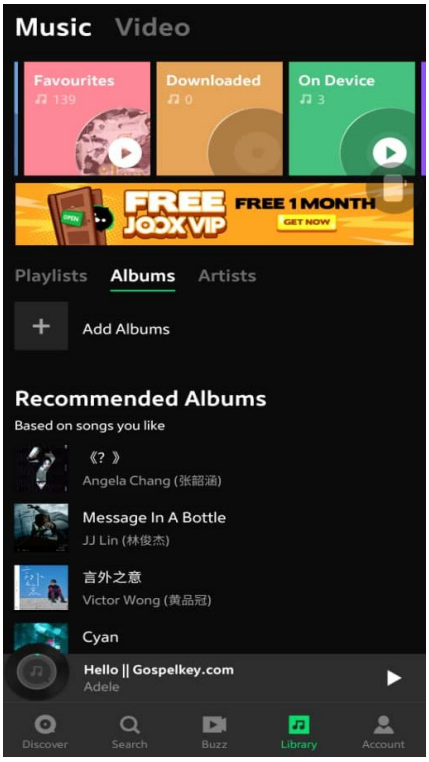


Figure 2.1.1.2: Library page in JOOX application

Furthermore, for the Search page, many kinds of playlists and categories will be shown for the user to choose from. There are also the “Most Searched” and “Recently Searched” which is the history that users searched for the songs and videos (Figure 2.1.1.3). In the media player, it provides some basic functions which are play, pause, next and previous. The user also can view the lyrics of the song, add a comment to the song, and shuffle the song (Figure 2.1.1.4).

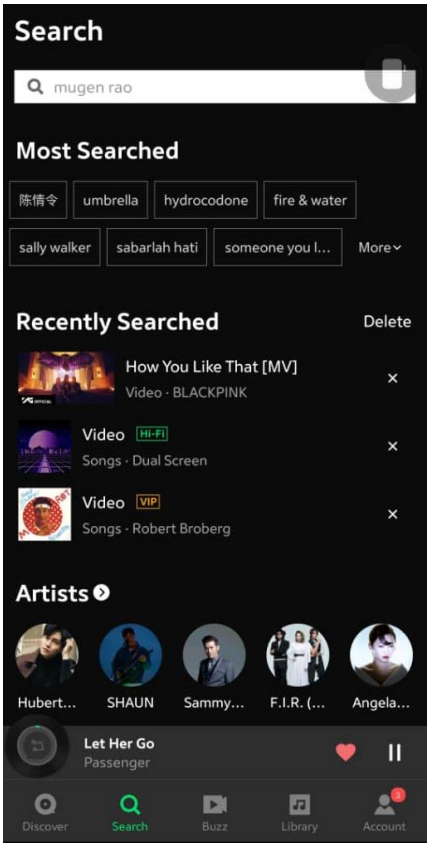


Figure 2.1.1.3: Search page in JOOX application

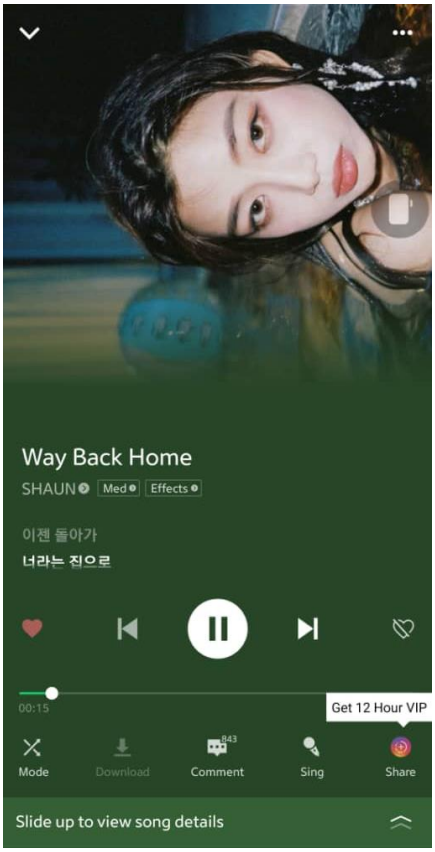


Figure 2.1.1.4: Music player in JOOX application

Moreover, in the JOOX application, the user also can play some music videos. In the music video player, the user can adjust the volume with a touch gesture by dragging up or down (Figure 2.1.1.5). In addition, the music video also can be played in full-screen mode to let the user have a bigger view of their music video (Figure 2.1.1.6).

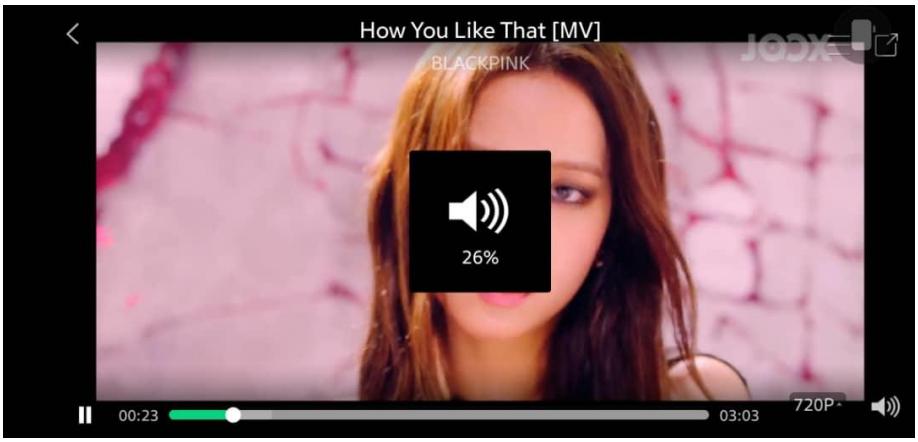


Figure 2.1.1.5: Volume control in music video player

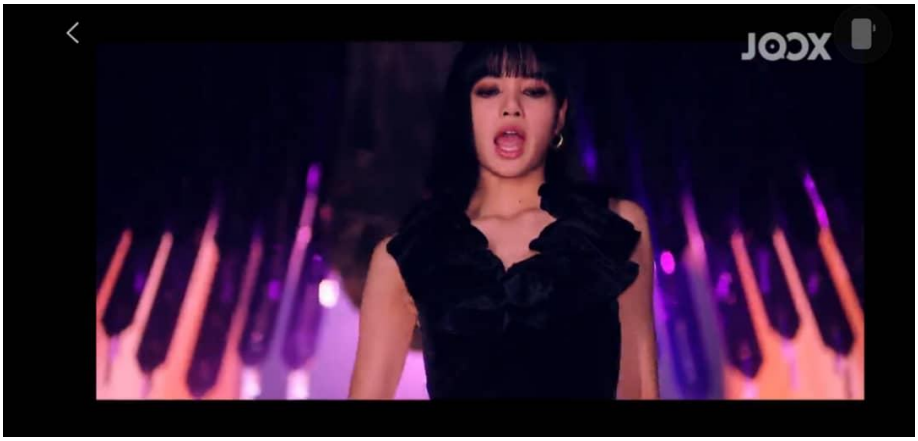


Figure 2.1.1.6: Full screen mode for music video

2.1.2 Review on Music Player & Video Player with equalizer

Music Player & Video Player with equalizer is a music video player that has an equalizer, bass booster, and 3D virtualizer. For the 3D virtualizer, there are multiple presets which are classical, flat, heavy metal, hip hop, and so on. Besides that, the music can be play in the background while using another app. The user also can browse their music in six different ways as it consists of six different categories which are tracks, artists, albums, genres, playlists, and folders. When the user newly downloads and uses the application, it will scan the music files and video files on the device automatically. Other than that, this application provides a sleep timer, can be controlled with a headset, and can shake the mobile device to switch the soundtracks.

In Music Player & Video Player with equalizer, it has a simple and with a basic color user interface. The application's structure of the interface is good as it categorizes six buttons and a music playlist (Figure 2.1.2.1). After clicking the library button, there are four tabs that display the related information of all the songs imported from the device (Figure 2.1.2.2).

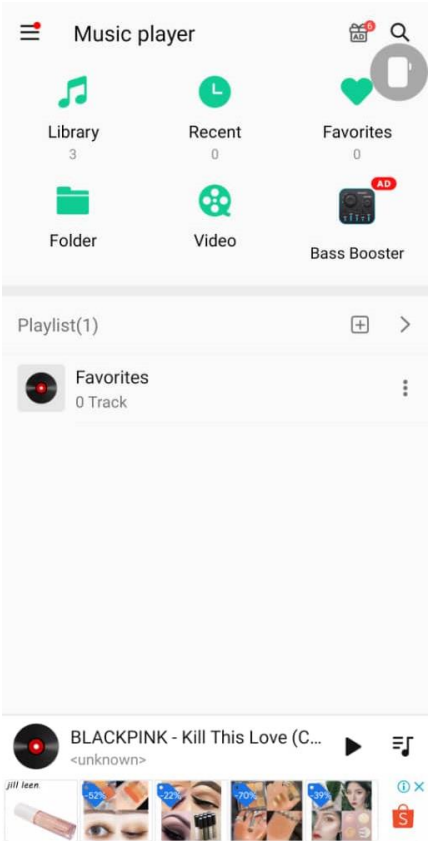


Figure 2.1.2.1: User interface in the application

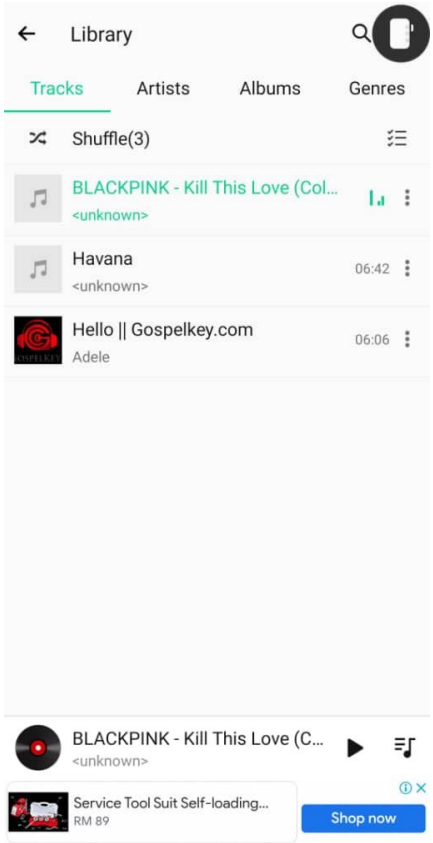


Figure 2.1.2.2: Library page in the application

Furthermore, the user also can play all their video that in the device by clicking the “Video” button on the homepage. A video playlist will be shown which is all the video saved on the mobile device (Figure 2.1.2.3). In this application, it consists of a drawer navigation menu that listed some other functions such as sleep timer, edge lightning, edge player, and others (Figure 2.1.2.4).

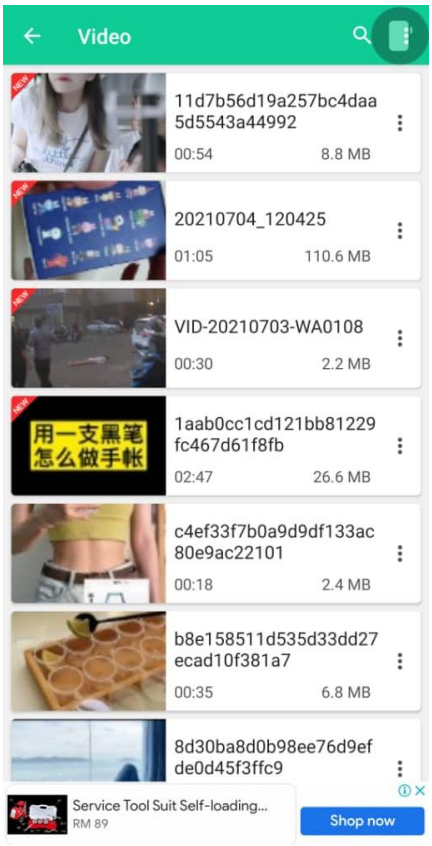


Figure 2.1.2.3: Video playlist of the videos that store in device

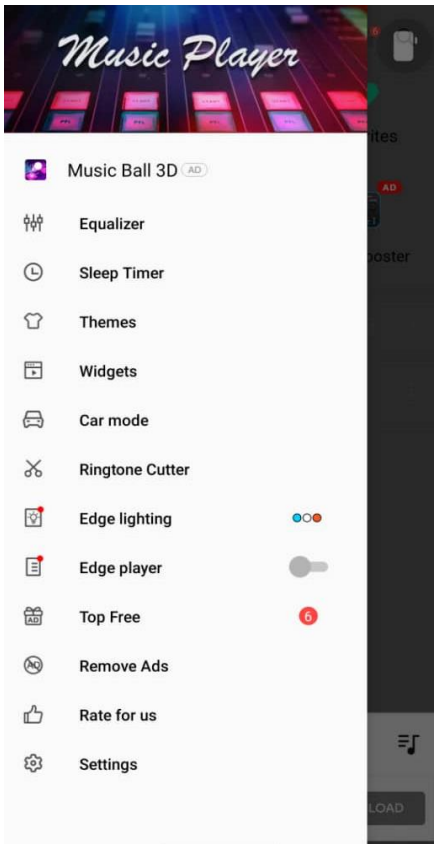
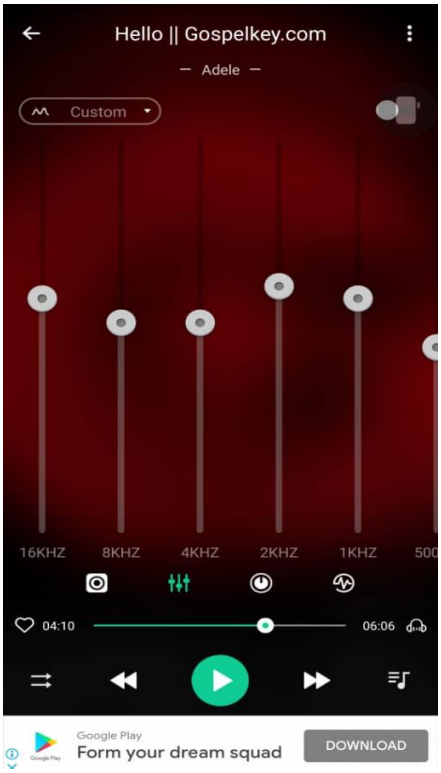


Figure 2.1.2.4: Drawer navigation menu

Moreover, the user also can use the in-app equalizer to adjust or isolates some of the frequencies may be lower them, boost them, or remain then unchanged. Besides that, the user also can on or off and adjust the ambient sound (Figure 2.1.2.5 and Figure 2.1.2.6). Other than the equalizer, the user also can choose a different sound effect for the songs that they listen to such as electronic tube, tone low, surround sound, and some other sound effects (Figure 2.1.2.7). This music video application consists of advertisement that is shown on the music player page, or it will pop up itself when you click to another page (Figure 2.1.2.8).



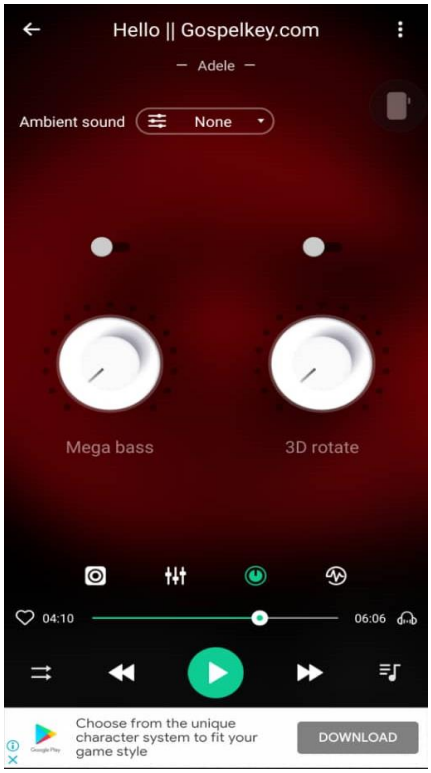


Figure 2.1.2.5 & Figure 2.1.2.6: Equalizer in the application

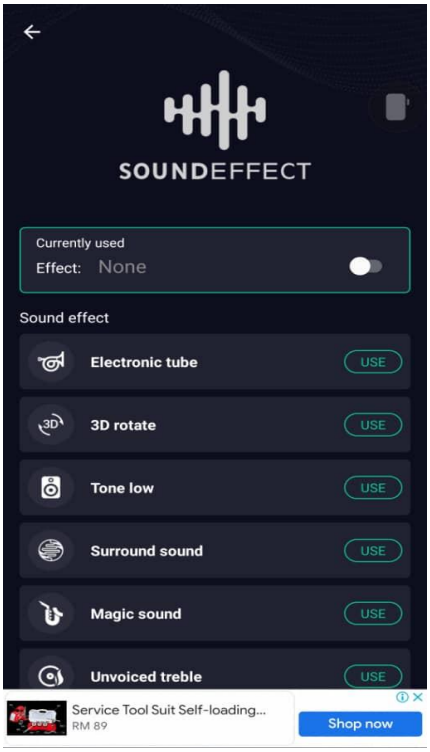


Figure 2.1.2.7: List of sound effect

Figure 2.1.2.8: Pop up advertisement

2.1.3 Review on Video Player

Video player is an application that is mainly for the user to play their video that store on their device. For video, there are three tabs on that page to indicate three different pages which are video, folders, and history. In the video page, it listed out all the video that stored in the mobile device and there are some advertisements interspersed between two videos (Figure 2.1.3.1). In folders page, it shows all the folder that contains video to let the user can find their video easily and for history page is to show the video that had been played before (Figure 2.1.3.2 and Figure 2.1.3.3). After selecting a video, a video view page will be displayed. The user can import a subtitle file, can play or pause the video, can adjust the playback speed, and also can adjust the video to full screen. Besides that, the user also can set a timer for the video to stop after some minutes, can rotate the screen into portrait or landscape and the user also can adjust the volume and pitch of the video (Figure 2.1.3.4).

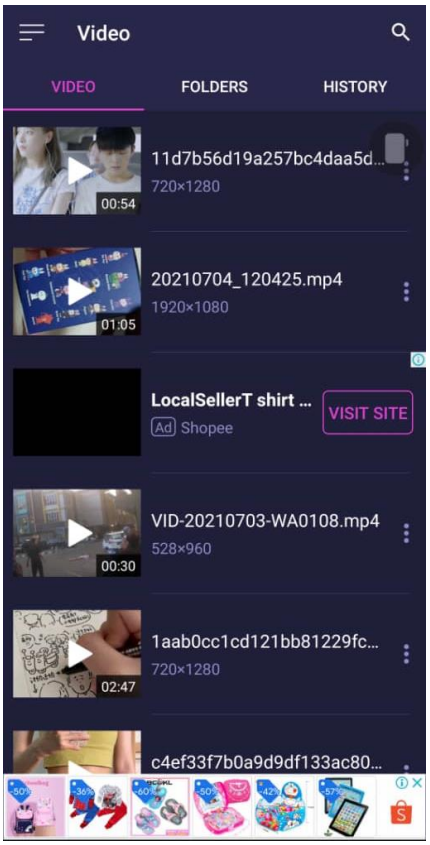


Figure 2.1.3.1: Video list in the video page

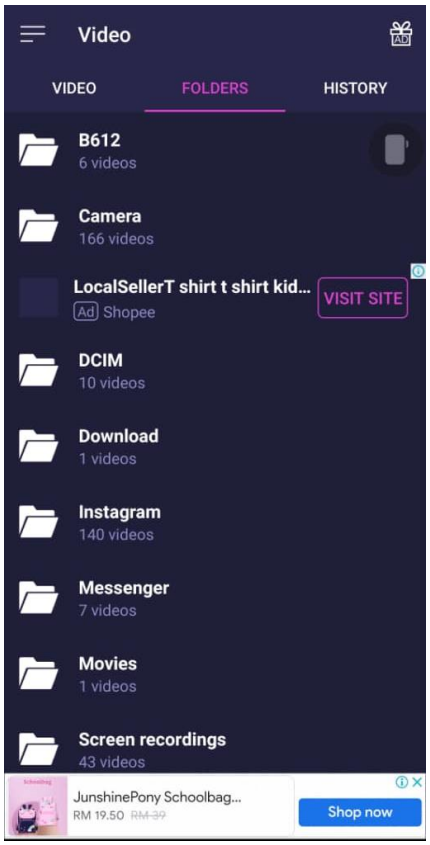


Figure 2.1.3.2: List of folders in folder page

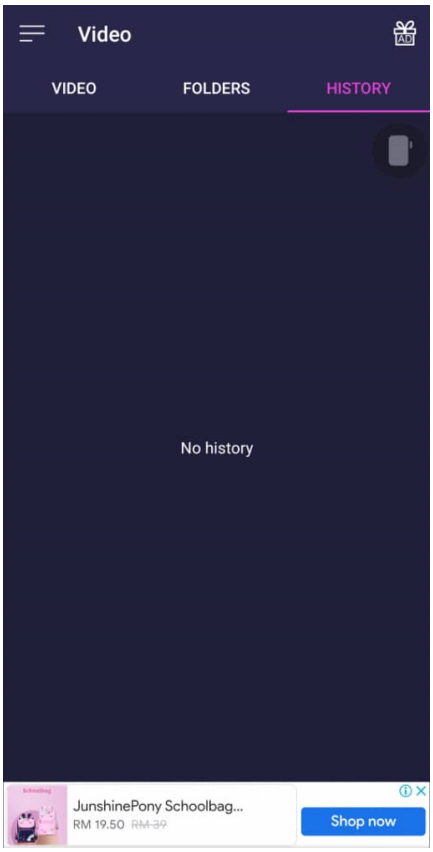


Figure 2.1.3.3: History page

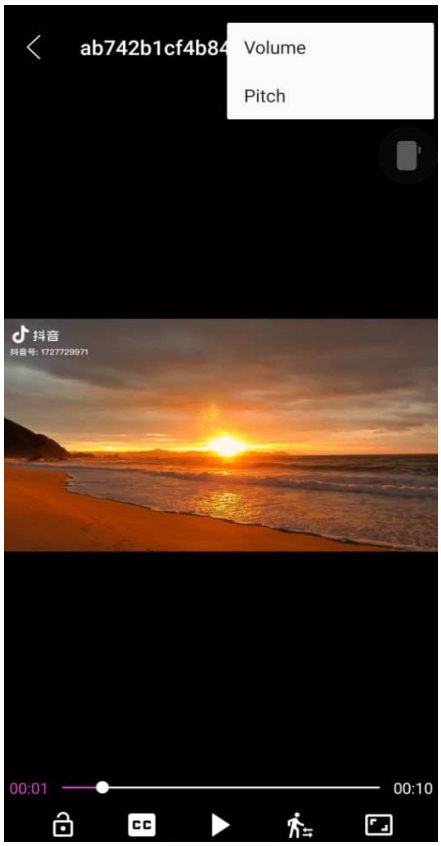


Figure 2.1.3.4: Video view and several features

Moreover, this application also can use to play the music that retrieves from the device. The user needs to expand the drawer navigation menu icon only can find out the music categories. In the drawer navigation menu, there are some different categories which are Video Player Pro and Video Maker that will link to the Google Play store to download another application, video, music, feedback, and about (Figure 2.1.3.5). After clicking the music category, six tabs that will be displayed which are all music, albums, artists, playlists, favorites, and history (Figure 2.1.3.6). In the music player, the user can use the basic activities such as play or pause the song, play the next or previous song, and shuffle the song. Besides that, the user also can add the song to a favorite playlist or create a new playlist, adjust the sound quality and effects, share the listening song with friends, adjust the volume, and set the timer for the song (Figure 2.1.3.7). Other than the advertisement shown between two videos in the video list, there are also a few advertisements will be shown such as at the bottom of every page, when clicking the song that wanted to play, click on the features in the application, and in the music player. An exit message window will pop out when the user wants to exit the application (Figure 2.1.3.8).

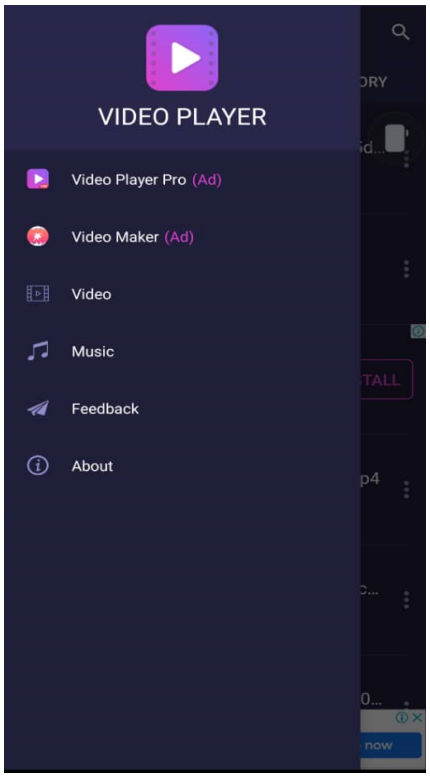


Figure 2.1.3.5: Drawer Navigation Menu

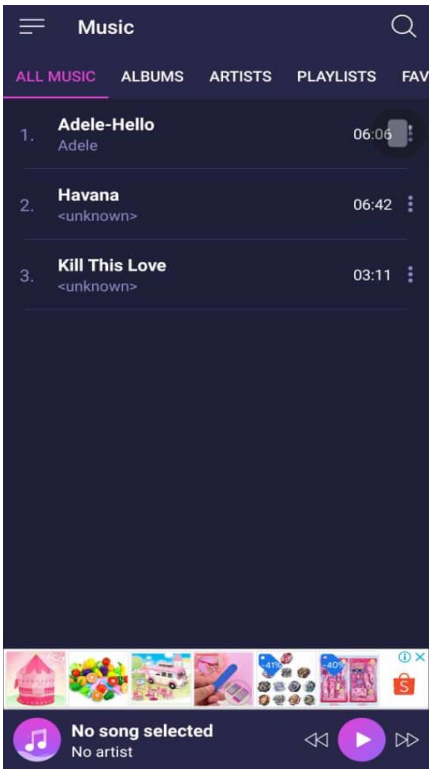


Figure 2.1.3.6: Six tabs under music category

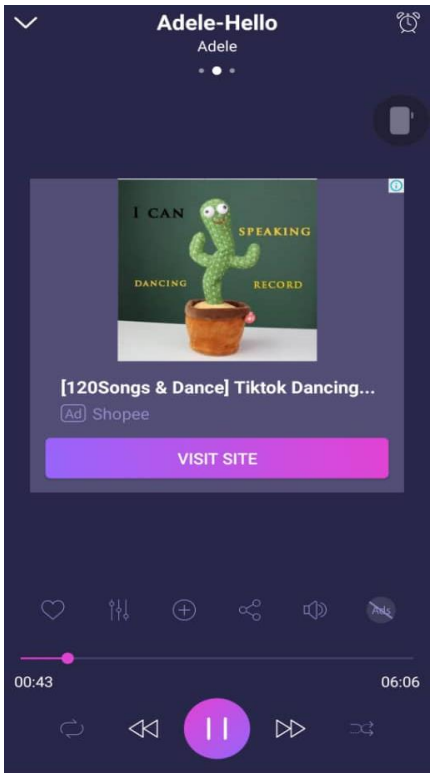


Figure 2.1.3.7: Music player in the application

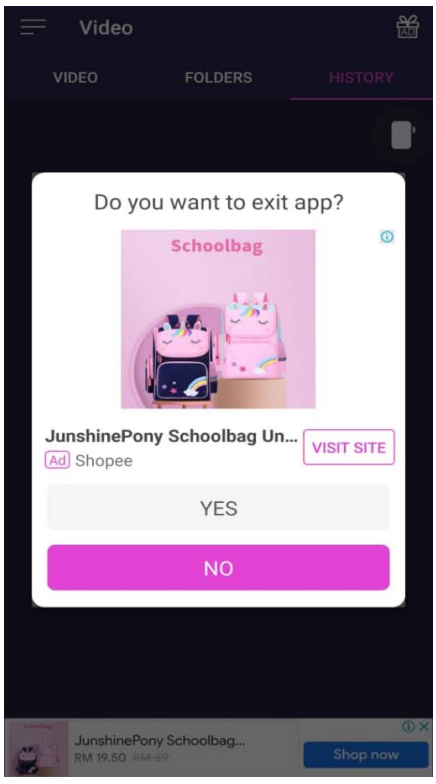


Figure 2.1.3.8: Exit message pop out window

2.2 Critical Remarks of Previous Works

2.2.1 Strength and Weakness of Reviewed Application

Table 2.2.1.1: Strength and Weakness of Reviewed Application.

Application	Strength	Weaknesses
JOOX	• The users can get VIP for free by sharing tracks on social media • Have good category filter	• Messy Interface • No family sharing plan
Music Player & Video Player with equalizer	• Clean and clear user interface • Sleep Timer provided	• Consist of some extra functions that may not be useful • Advertisements will be shown
Video Player	• Clearly classify the video and music into different page • Clear user interface	• Many advertisements will be shown • Consists of many tabs that may have duplicated media

2.2.2 Comparison of Proposed System and Previous Work Done

Table 2.2.2.1: Comparison table for the proposed system and the existing systems.

Application / Criteria	JOOX	Music Player & Video Player with equalizer	Video Player	Proposed Solution
User Interface	Messy	Normal	Normal	Simple and clear
Background Play	✔	✔	✔	✔
Support Multiple Media Format	✔	✔	✔	✔

Touch Gesture for Media Control	✔	✔	✔	✔
Swipe/Hand Gesture for Media Control	✗	✗	✗	✔
Voice Command for Media Control	✗	✗	✗	✔
Shuffle and Repeat	✔	✔	✔	✔
Advertisement	✔	✔	✔	✗

CHAPTER 3 SYSTEM DESIGN

3.1 Proposed Approach or Study

3.1.1 Methodology

The methodology for this proposed project is Mobile Application Development Life Cycle (MADLC). The figure below shown the flow of the discussed life cycle:

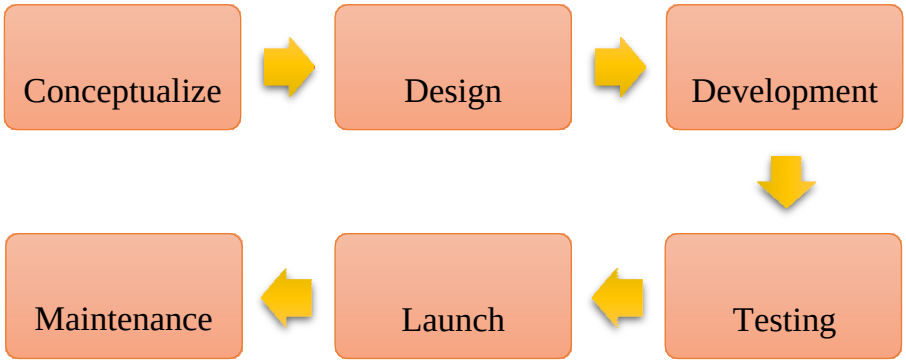


Figure 3.1.1.1: Mobile Application Development Life Cycle

Conceptualize

In Conceptualize phase, we need to understand what the purpose to build this mobile application. Besides that, we also need to identify the target audience, the requirement and some other details.

Design

In Design Phase, we need to focus on two aspects which is user interface and functionality. User interface is what the users can see in the app and they can interact with, whereas functionality is how the users can use it and how the mobile application works. We can design the mobile application using some diagram such as wireframe and UML diagram.

Development

In Development phase, there are two parts that we need take care for which is prototyping and building. Prototyping is used to test the sketched design ideas before we move forward to the actual building process for the application. For building stage, the initial ideas and sketched design from the prototyping stage need to combine and produce the final product.

Testing

In Testing phase, we need to test our application that have built in Development phase. Therefore, we can know that whether the mobile application built had meet the initial requirement.

Launch

After testing the mobile application that we built, we may release this application to some users for example students in UTAR. As this is an Android application, we may upload to Google Play store for more users to download if this application received positive evaluation from the students.

Maintenance

In Maintenance phase, we will fix the bugs of our mobile application as soon as possible that found by ourselves or the users. Therefore, users can have a better experience when using our mobile application.

3.1.2 Tool to Use

Table 3.1.2.1: Tools involved in the development of this project

Tools	Specification
Software	1. Android Studio 2. Draw.io
Hardware	1. Laptop - Processor: Intel Core i5-8265U - RAM: 12GB 2. Smartphone Device - RAM: 8GB - Phone Storage: 128GB - Operating System: Android version 10

3.2 Requirement Specifications

3.2.1 Functional Requirements

1. User should be able to import the media files from phone local storage.
2. User should be able to play audio and video.
3. User should be able to control the audio with button clicks, swipe gesture and voice command.
4. User should be able to control the video with button clicks.
5. User should be able to shuffle and repeat the music.
6. User should be able to switch between light mode and night mode.

3.2.2 Non-functional Requirements

1. User should be able to see the user interface of the mobile application easily.
2. User should be able to know the instinct uses of the mobile application without any manual or tutorial.

3.3 System Specification

The system design of this music video mobile application will be illustrated in detail in the following sections. The diagrams provided below are used to illustrate how the mobile application interact with the users.

3.3.1 Use Case Diagram

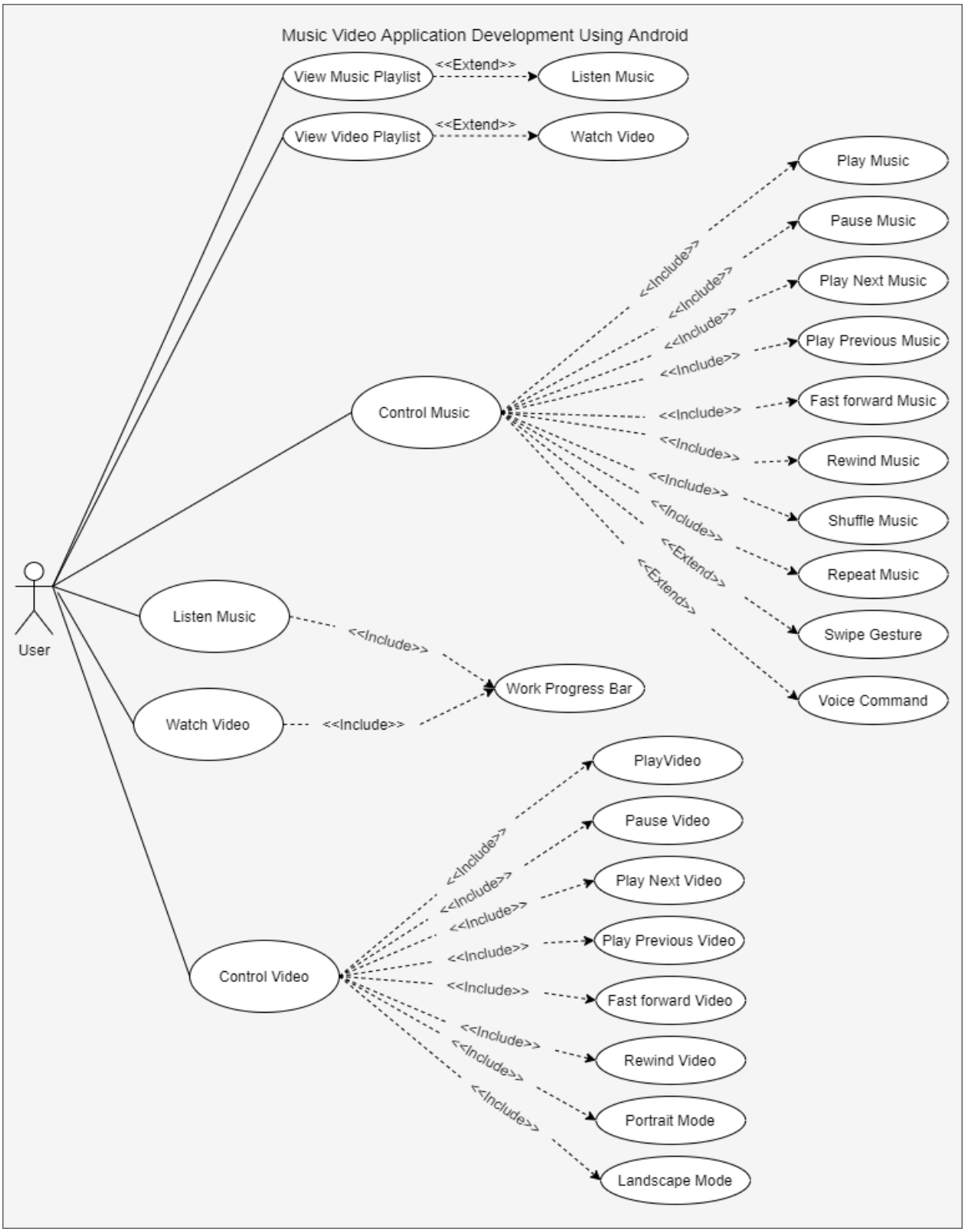


Figure 3.3.1.1: Use case diagram of Music Video Application Development using Android

3.3.2 Use Case Description

Table 3.3.2.1: Use Case Description of View Music Playlist

Use case name: View Music Playlist	ID: UC001
Actor: User	
Description: Allow user to view the list of music on the mobile application	
Trigger: 1. Click the music playlist button.	
Precondition: The user shall have the compatible music file stored in their smartphone device.	
Normal Flow of Events: 1. The user able to select a music from the playlist and play it.	
Alternate/ Exceptional Flows: -	

Table 3.3.2.2: Use Case Description of Listen Music

Use case name: Listen Music	ID: UC002
Actor: User	
Description: Allow user to listen music on the mobile application.	
Trigger: 1. Click the play button to start the music. 2. Click the next button to play the next music. 3. Click the previous button to play the previous music. 4. Click the fast forward button to fast forward the music. 5. Click the rewind button to rewind the music.	
Precondition: The user shall have the compatible music file to play in the mobile application.	
Normal Flow of Events: 1. Select a music from the music playlist. 2. Play the music until the end. 3. Play the next music by clicking the next button. 4. Play the previous music by clicking the previous button. 5. Fast forward the music by clicking the fast forward button. 6. Rewind the music by clicking rewind button.	
Alternate/ Exceptional Flows: 1. The user unable to listen the music using the mobile application if do not have the compatible music file. 2. The user can click the next or previous button to play the next or previous music; fast forward or rewind button to fast forward or rewind the music. 3. The user can exit the mobile application.	

Table 3.3.2.3: Use Case Description of Voice Command Media Control

Use case name: Voice Command Media Control	ID: UC003
Actor: User	
Description: Allow user to control the media using voice with the microphone of mobile device.	
Trigger: 1. Click the microphone button.	
Precondition: -	
Normal Flow of Events: 1. The user able to select a video from the playlist and play it.	
Alternate/ Exceptional Flows: -	

Table 3.3.2.4: Use Case Description of Shuffle and Repeat

Use case name: Shuffle and Repeat	ID: UC004
Actor: User	
Description: Allow user to shuffle or repeat the music.	
Trigger: 1. Click the shuffle button. 2. Click the repeat button.	
Precondition: -	
Normal Flow of Events: 1. The user able to shuffle the music. 2. The user able to repeat the music.	
Alternate/ Exceptional Flows: -	

Table 3.3.2.5: Use Case Description of Swipe Gesture Media Control

Use case name: Swipe Gesture Media Control	ID: UC005
Actor: User	
Description: Allow user to control media using swipe gesture.	
Trigger: 1. Swipe up to increase the volume. 2. Swipe down to decrease the volume. 3. Swipe left to play next media. 4. Swipe right to play previous media.	
Precondition: -	
Normal Flow of Events: 1. The user able to increase the volume by swiping up. 2. The user able to decrease the volume by swiping down. 3. The user able to play next media by swiping left. 4. The user able to play previous media by swiping right.	
Alternate/ Exceptional Flows: -	

Table 3.3.2.6: Use Case Description of Watch Video

Use case name: Watch Video	ID: UC006
Actor: User	
Description: Allow user to watch video on the mobile application.	
Trigger: 1. Click the play button to start the video. 2. Click the next button to play the next video. 3. Click the previous button to play the previous video. 4. Click the fast forward button to fast forward the video. 5. Click the rewind button to rewind the video.	
Precondition: The user shall have the compatible video file to play in the mobile application.	
Normal Flow of Events: 1. Select a video from the video playlist. 2. Watch the video until the end. 3. Play the next video by clicking the next button. 4. Play the previous video by clicking the previous button. 5. Fast forward the video by clicking the fast forward button. 6. Rewind the video by clicking rewind button. 7. Change to portrait mode by clicking portrait button. 8. Change to landscape mode by clicking the landscape button.	
Alternate/ Exceptional Flows: 1. The user unable to watch the video using the mobile application if do not have the compatible video file. 2. The user can click the next or previous button to play the next or previous video; fast forward or rewind button to fast forward or rewind the video. 3. The user can click the portrait or landscape button to change the view of video into portrait or landscape. 4. The user can exit the mobile application.	

Table 3.3.2.7: Use Case Description of View Video Playlist

Use case name: View Video Playlist	ID: UC007
Actor: User	
Description: Allow user to view the list of videos on the mobile application	
Trigger: 1. Click the video playlist button.	
Precondition: The user shall have the compatible video file stored in their smartphone device.	
Normal Flow of Events: 1. The user able to select a video from the playlist and play it.	
Alternate/ Exceptional Flows: -	

Table 3.3.2.8: Use Case Description of Portrait mode and Landscape mode

Use case name: Portrait mode and Landscape mode	ID: UC008
Actor: User	
Description: Allow user to switch between portrait and landscape mode.	
Trigger: 1. Click the portrait button to change the video into portrait mode. 2. Click the landscape button to change the video into landscape mode.	
Precondition: -	
Normal Flow of Events: 1. Change to portrait mode by clicking portrait button. 2. Change to landscape mode by clicking the landscape button.	
Alternate/ Exceptional Flows: -	

Table 3.3.2.9: Use Case Description of Control Audio and Video

Use case name: Control Audio and Video	ID: UC009
Actor: User	
Description: Allow user to control the audio and video on the mobile application.	
Trigger: 1. Click the icon button to control the execution of audio or video.	
Precondition: An audio or a video must be play on the mobile application.	
Normal Flow of Events: 1. The user control the execution of audio or video using icon button or media controller such as play, pause, next, previous, fast forward and rewind.	
Alternate/ Exceptional Flows: 1. The selected audio and video will play until the end if the user does not control the execution of audio or video.	

3.3.3 System Performance Definition

Table 3.3.3.1: Test actions being done on the application and the result delivered

Case	Test Action	Result	Status
1	Tap on the music fragment	Show the media player and list of music	Pass
2	Tap on the video fragment	Show the video view and list of videos	Pass
3	Tap on the more info fragment	Show the extra information and features	Pass
4	Tap on the play, pause, previous, next, fast forward, and rewind button in the music player	Able to play and pause the music; can play the next music and previous music; can fast forward and rewind the music	Pass
5	Use swipe gesture on the title can play next, play previous, increase volume, and decrease volume in the music player	Able to play next and previous music by swiping left or right; increase volume and decrease volume by swiping up or down	Pass
6	Tap on the shuffle button or repeat button	Able to shuffle the playing music or repeat the playing music	Pass
7	Tap on the play, pause, previous, next, fast forward, and rewind button in the video view	Able to play and pause the video; can play the next and previous video; can fast forward and rewind the video	Pass
8	Tap on the portrait mode and landscape mode button	Able to enlarge the screen using landscape mode button and zoom out the screen using portrait mode button	Pass
9	Tap on about us cardview	Information about the music video player will be display	Pass

10	Tap on download cardview	Will link to website that can let user to download music and video	Pass
11	Tap on the switch compact to on the nigh mode and off the light mode	Able to switch between the light and night mode by clicking the switch compact	Pass

3.4 Design

3.4.1 System Flowchart

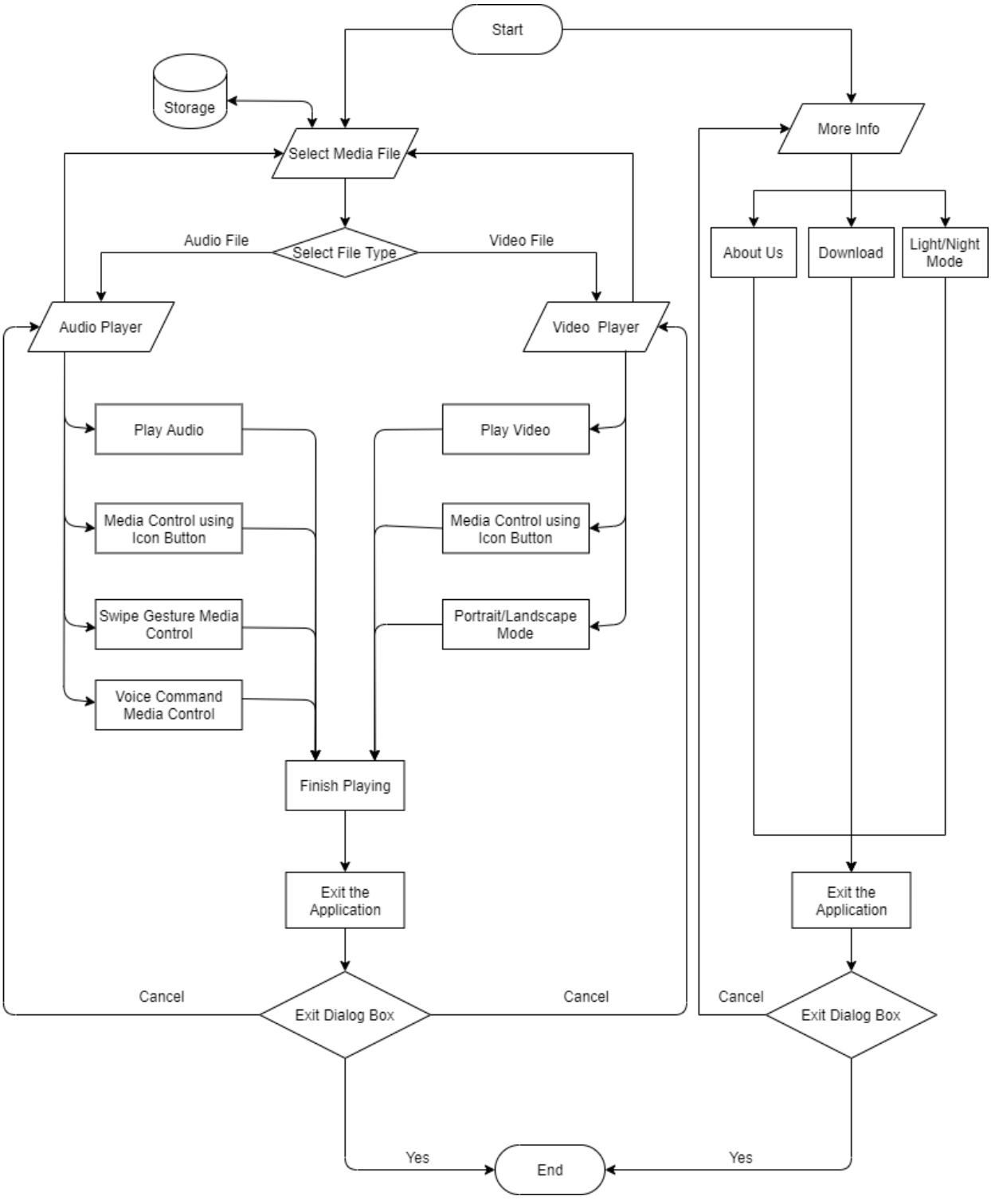


Figure 3.4.1.1: System flowchart of proposed mobile application

Table 3.3.4.1: Description of System Flowchart

Process	Functionality
Start Application	The mobile application is start and proceed to the homepage.
Audio Player	The audio player is started to play when the user selected the audio file.
Video Player	The video player is started to play when the user selected the video file.
Play Audio	The audio file started to play in the audio player.
Icon Button for Audio Player	The user can use the icon button to control the media in audio player
Swipe Gesture in Audio Player	The user can use swipe gesture to control the media and volume in audio player
Voice Command in Audio Player	The user can use voice command to control the media in audio player
Shuffle and Repeat Function	The user can shuffle the playing music or repeat the playing music by clicking the shuffle or repeat button.
Media Control with Icon Button	The user can use the built-in icon button to control the media in video player.
Portrait and Landscape button in Video Player	The user can enlarge the video into landscape mode and zoom out the video into portrait mode.
Finish Playing	The audio or video files finished playing.
View Mobile Application details	The user able to view the details about the mobile application.
Download music or video	The user able to download music or video through the link.
Switch between light mode and night mode	The user able to switch between light mode and night mode.
Exit Application	The user may exit the mobile application.

End	The mobile application is terminated when the user exits the mobile application.

3.4.2 Application User Interface Design

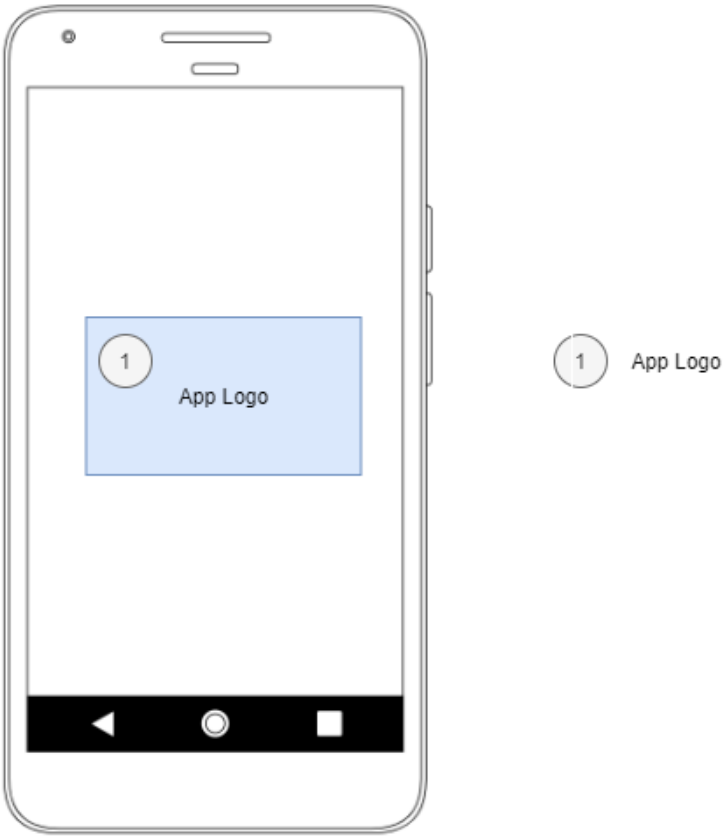


Figure 3.4.2.1: Splash Screen

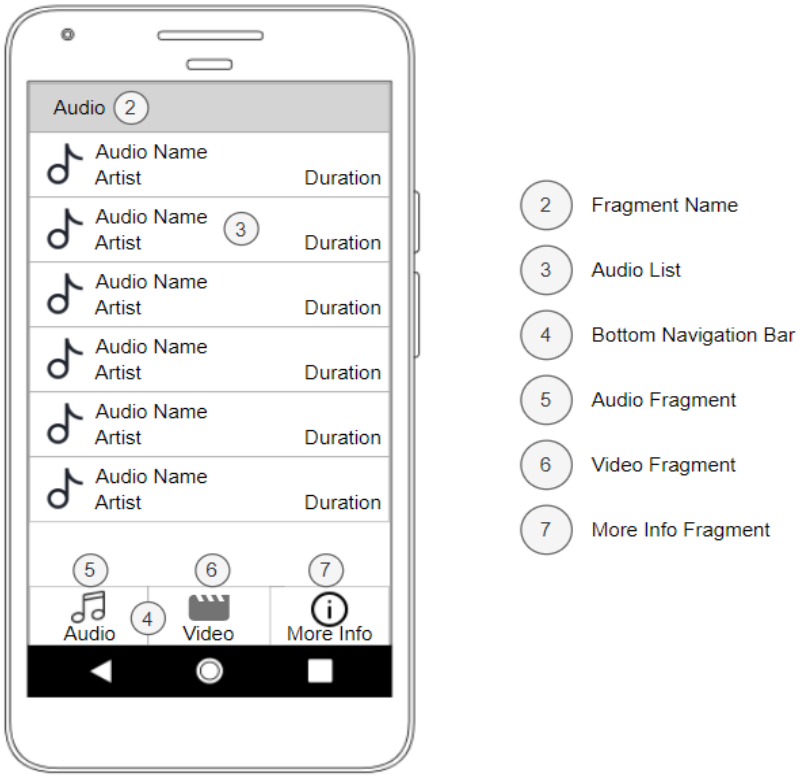


Figure 3.4.2.2: Audio Fragment

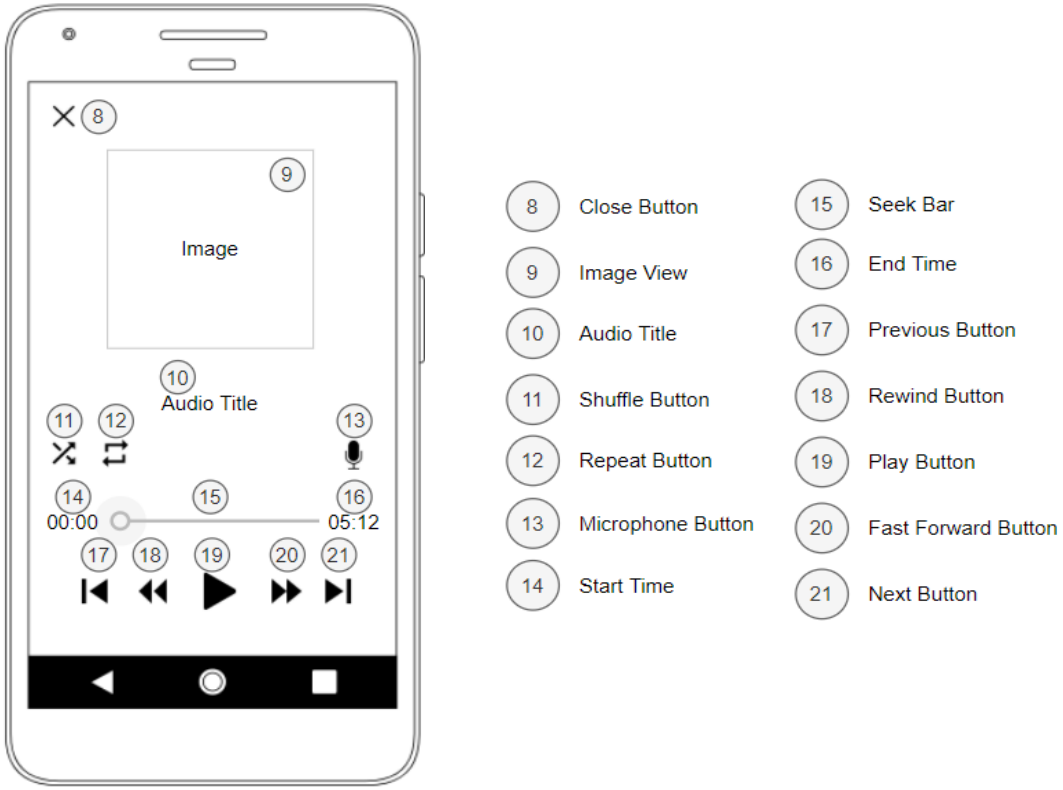


Figure 3.4.2.3: Audio Player

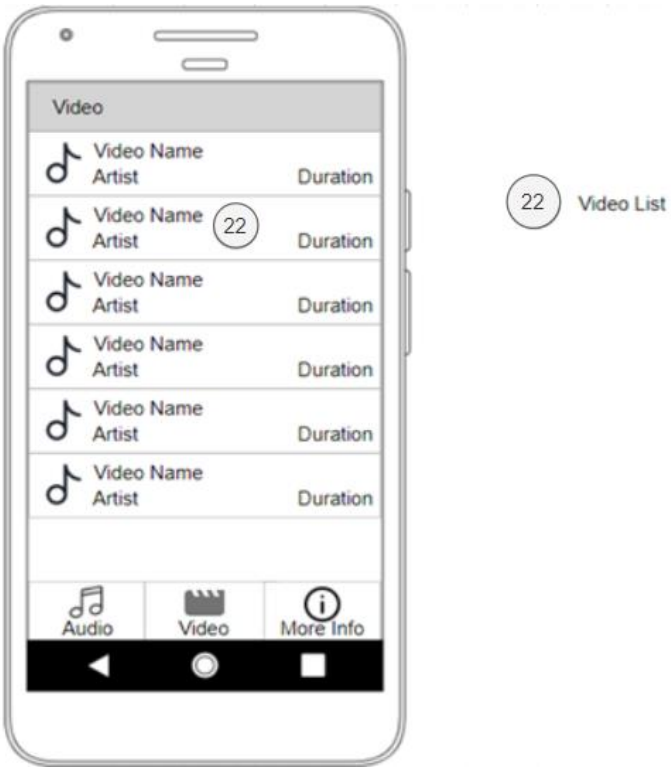


Figure 3.4.2.4: Video Fragment

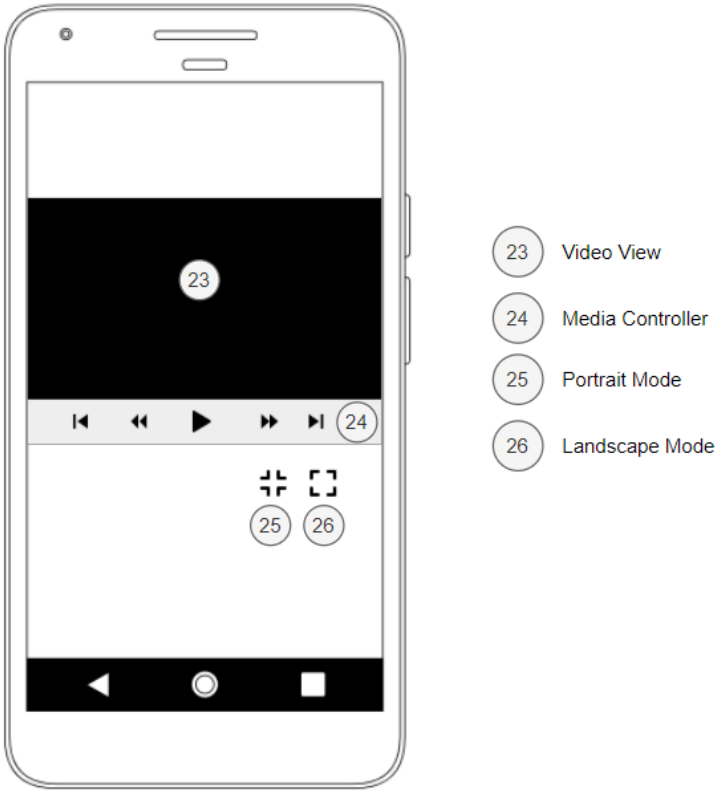


Figure 3.4.2.5: Video Player in Portrait

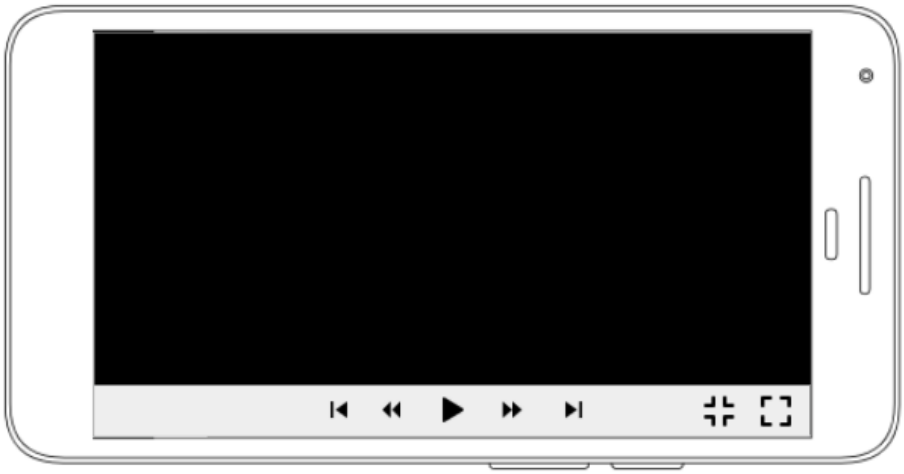


Figure 3.4.2.6: Video Player in Landscape

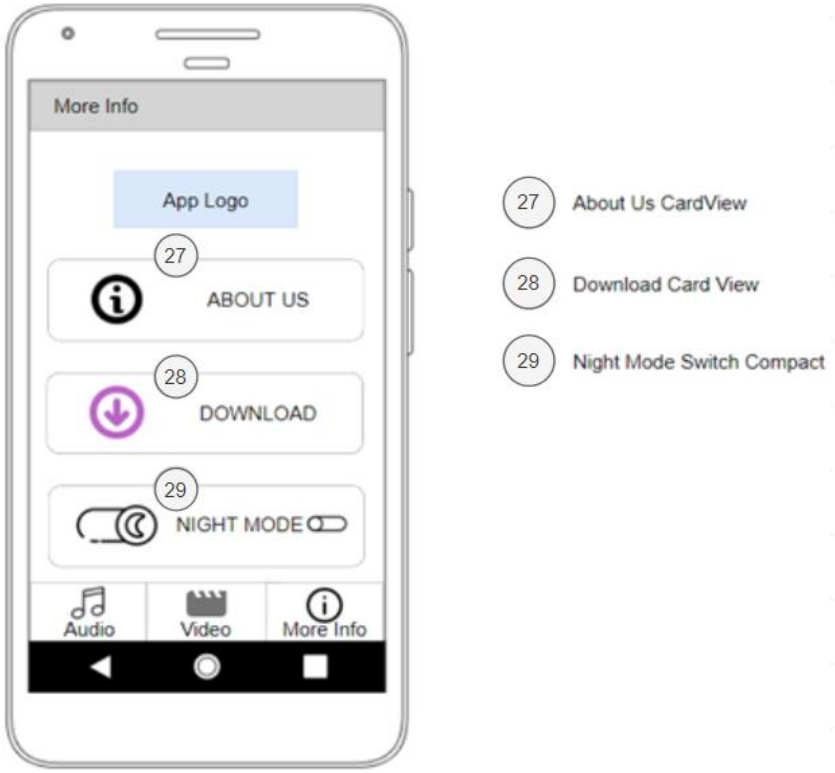


Figure 3.4.2.7: More Info Fragment

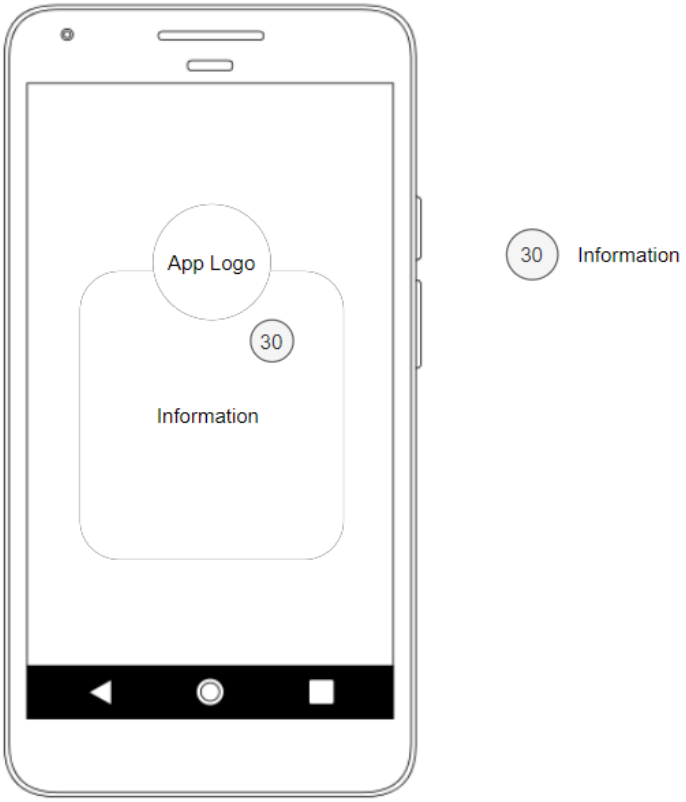


Figure 3.4.2.8: About Page

Table 3.4.2.1: Description of Application's activities

Activity	Description
Splash Screen	An initialization screen that animates when starting the mobile application.
Audio Fragment	Activity that shows the audio list from array adapter of audio.
Audio Player	Activity that able to play, pause, play next, play previous, fast forward, rewind, shuffle and repeat the audio file with the button.
Video Fragment	Activity that shows the video list from array adapter of video.
Video Player	Activity that able to play, pause, play next, play previous, fast forward and rewind the video file with the media control. It can orient into portrait or landscape.
More Info Fragment	Activity that shows the extra information and function of the mobile application.

3.5 Wireframe

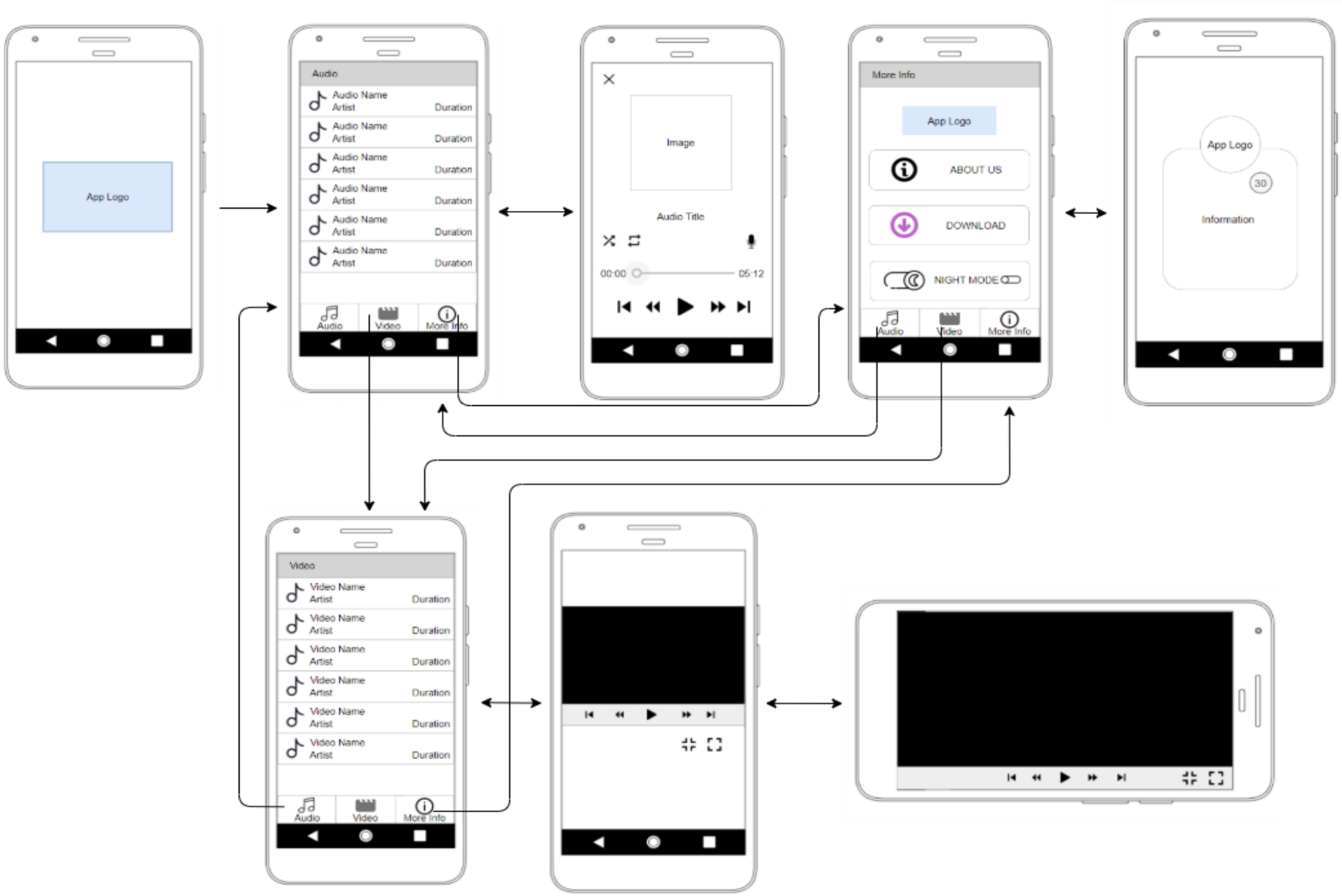


Figure 3.5.1: Wireframe of Mobile Application

3.6 System Component Diagram

3.6.1 Audio Player

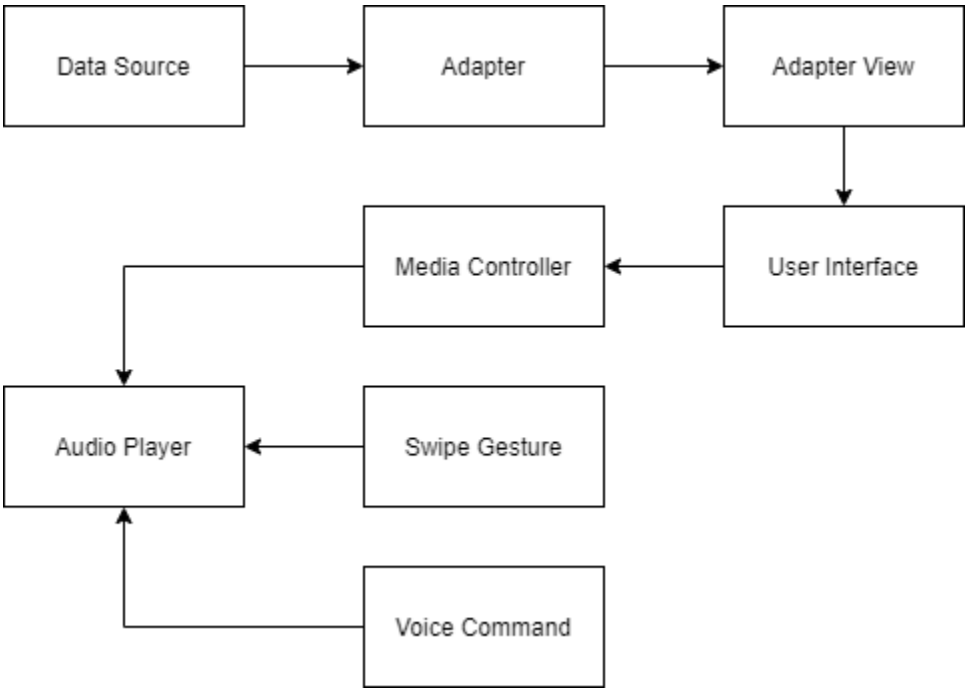


Figure 3.6.1.1: Block Diagram of Audio Player

3.6.2 Video Player

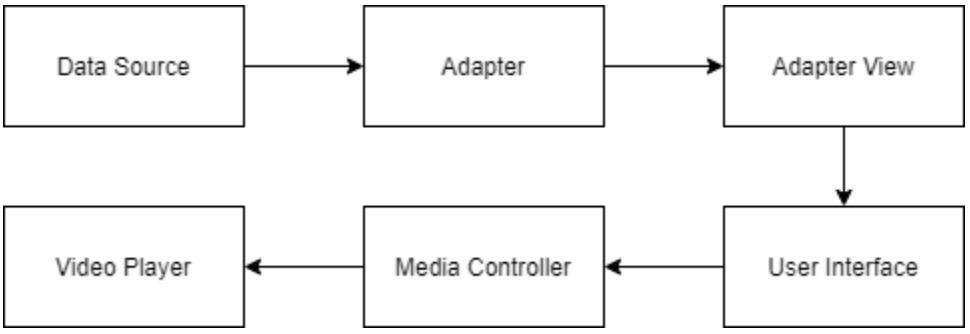


Figure 3.6.2.1: Block Diagram of Video Player

3.7 Verification Plan

Table 3.7.1: Verification Plan

Case	Test Case Action	Types of Input	Test Case Description	Expected Result
1	User Interface Design	Source Code	To match the output of the user interface design with the smartphone device	User interface works in smartphone device is same as the design in the Android Studio.
2	Audio Player	Source Code	Validate the successfulness to play music	Media player can play music successfully.
3	Video Player	Source Code	Validate the successfulness to play video	Video view able to display the video and play the video successfully.
4	Media Control	Source Code	Check the functionality of the icon in the media control pane	All the icon able to function well in media control pane.
5	Button and icon	Source Code	Check the functionality of the button in the mobile application	All the button able to perform function well when clicked.
6	Swipe Gesture	Source Code	Check the functionality of the swipe gesture in the mobile application	All the swipe gesture able to be detected when swiped.
7	Voice Command	Source Code	Check the functionality of the voice command in the mobile application	All the voice able to be recognized when speak.
8	Shuffle and Repeat Function	Source Code	Check the functionality of the button in the mobile application	All the button able to perform well when clicked.

9	Portrait and Landscape mode	Source Code	Check the functionality of the button in the mobile application	All the button able to perform well when clicked.
10	Light and Night mode switch compact	Source Code	Check the functionality of the switch compact in the mobile application	Able to switch between light mode and night when clicked on the switch compact.
11	Mobile Application	Source Code	Assure that the implemented mobile application can be install and run smoothly on the smartphone device	Implemented mobile application can be install and run smoothly on the smartphone device.

3.8 Timeline

Task Name	Duration	Start Date	End Date	Week													
				1	2	3	4	5	6	7	8	9	10	11	12	13	14
Final Year Project 2	**14 weeks**	**07/06/2021**	**10/09/2021**														
1 Analysis	**1 week**	**07/06/2021**	**11/06/2021**														
1.1 Requirements Analysis	1 week	07/06/2021	11/06/2021														
2 Design	**5 weeks**	**14/06/2021**	**16/07/2021**														
2.1 User Interface Design	2 weeks	14/06/2021	25/06/2021														
2.2 Wireframing	2 weeks	28/06/2021	09/07/2021														
2.3 Prototyping	1 week	12/07/2021	16/07/2021														
3 Implementation	**6 weeks**	**19/07/2021**	**23/08/2021**														
3.1 System Development	4 weeks	19/07/2021	13/08/2021														
3.2 System Testing	3 weeks	26/07/2021	13/08/2021														
3.3 System Refinement	1 week	16/08/2021	20/08/2021														
3.4 System Evaluation	1 week	22/08/2021	27/08/2021														
Documentation	**13 weeks**	**18/01/2021**	**03/09/2021**														
Submission for FYP2 Report	**1 Day**	**03/09/2021**	**03/09/2021**														
Presentation	**1 Day**	**06/09/2021**	**10/09/2021**														

Figure 3.8.1: Gantt Chart for the current semester

CHAPTER 4 SYSTEM APPLICATION DEVELOPMENT

4.1 Custom Creation of Music Video Player

4.1.1 Design layout, implement activity and object class

The design layout of the mobile application can be done in design mode or text mode. In the design mode, there are different attribute settings. The XML file can be created in *layout* folder. Then can start to design the layout of the mobile application. In Figure 4.1.1.1 is the design layout for audio player for this mobile application. In Figure 4.1.1.2 is the text mode of the audio player and the source code can refer to Appendices B (**audio_player.xml**).

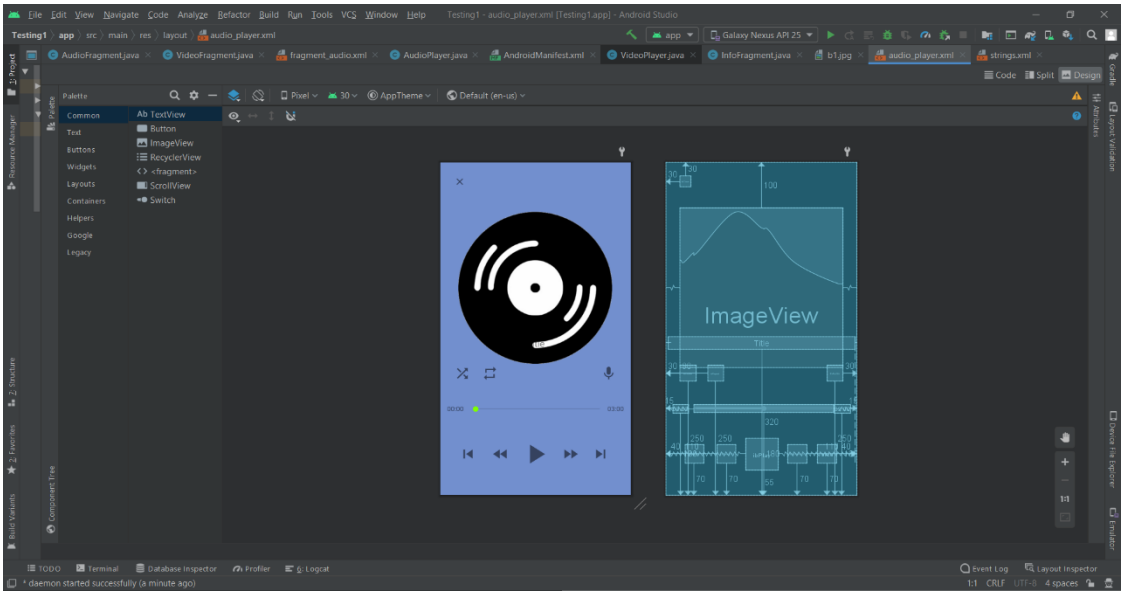


Figure 4.1.1.1: Design Mode of Audio Player's XML layout

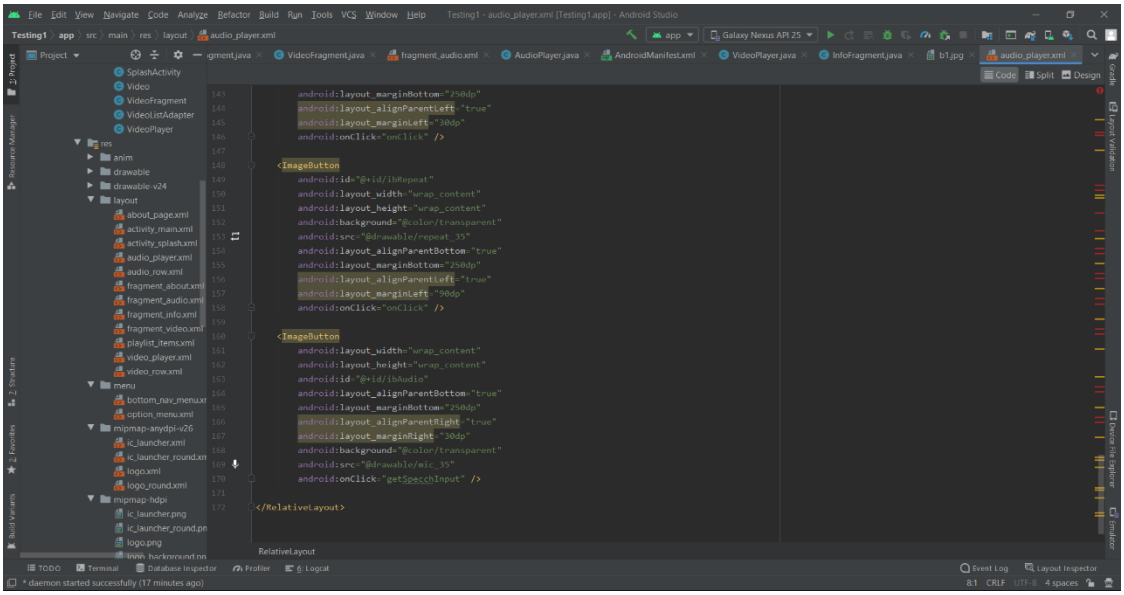


Figure 4.1.1.2: Text Mode of Audio Player's XML layout

Next, the JAVA file can be created in the *java* folder. In Figure 4.1.1.3, it is the java class for the audio player and the source code can refer to Appendices B (**AudioPlayer.java**). To retrieve the details of the song, we need to create an object class for the song which is the attributes of the music included song name, artist, duration and the path of the song, the source code refer to Appendices B (**Song.java**) (Figure 4.1.1.4).

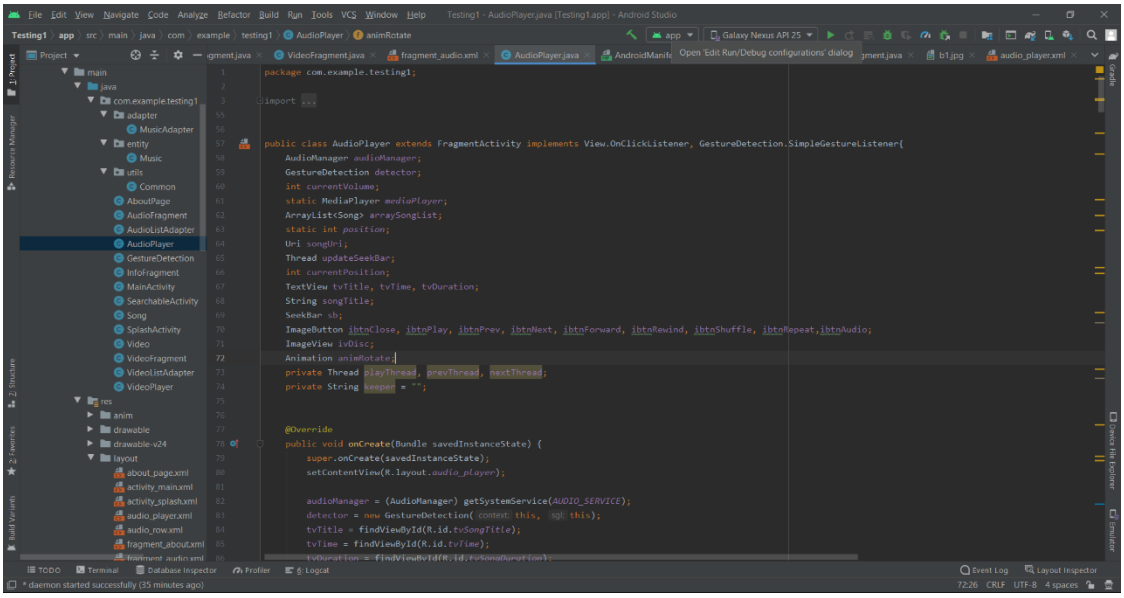


Figure 4.1.1.3: AudioPlayer's Java Class

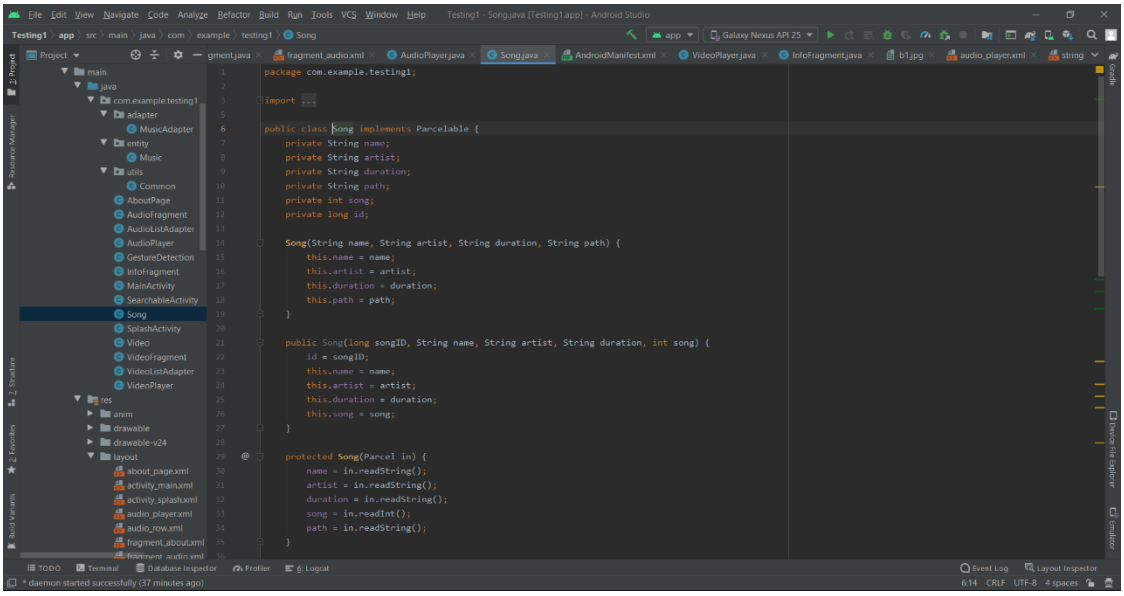


Figure 4.1.1.4: Song's Java Class

In Figure 4.1.1.5, an adapter view is needed to hold the details of the song and queue in the list view structure as the playlist in the audio fragment (**AudioListAdapter.java**). After that, the information of the playlist in the audio fragment will be extract from the mobile device (**AudioFragment.java**) (Figure 4.1.1.6). To extract the data from the mobile device, a permission needs to be added in the **AndroidMainifest.xml**. The permission will be *android.permission.READ_EXTERNAL_STORAGE*. The audio will be able to play in the audio player after successfully set up all the necessary layout and object class that included AudioPlayer.java, AudioListAdapter.java, Song.java and audio_player.xml.

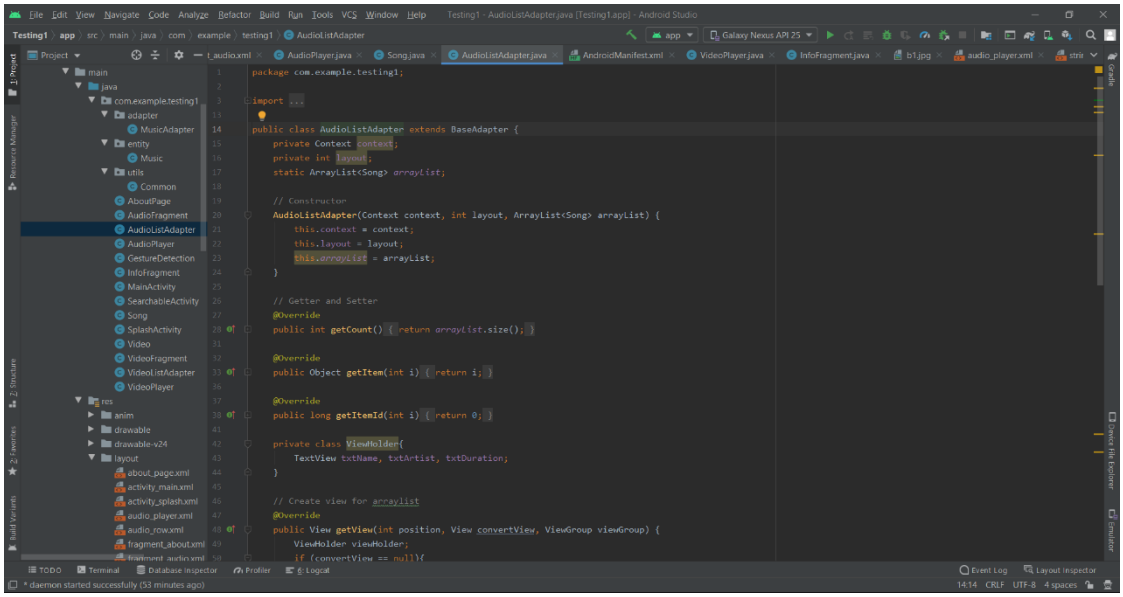


Figure 4.1.1.5: AudioListAdapter's Java Class

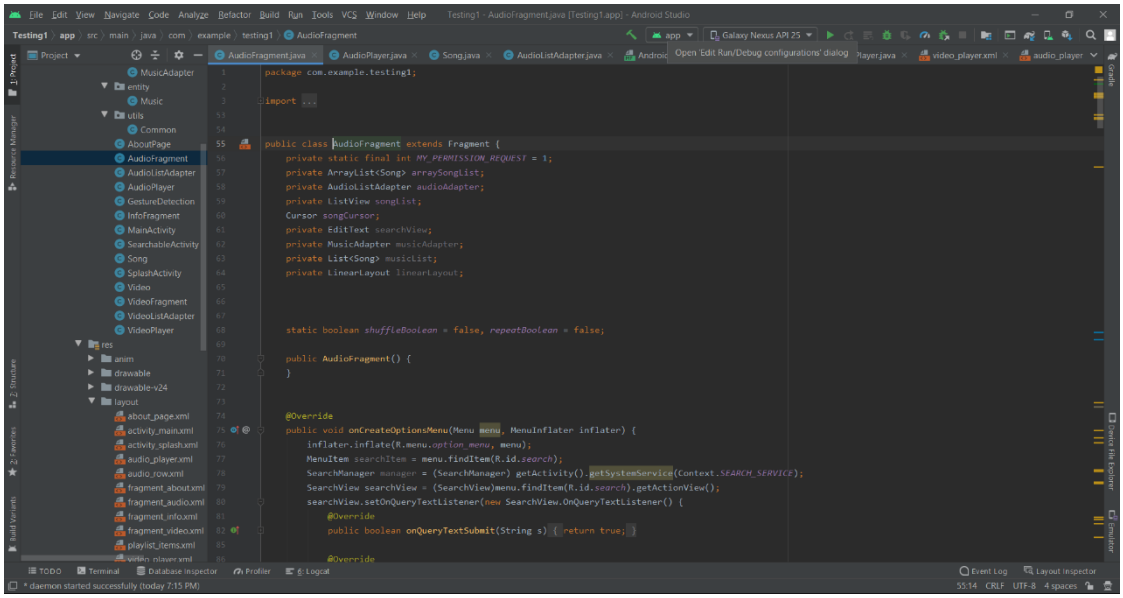


Figure 4.1.1.6: AudioFragment's Java Class

In Figure 4.1.1.7 is the design layout for video player for this mobile application. In Figure 4.1.1.8 is the text mode of the video player and the source code can refer to Appendices B (**video_player.xml**).

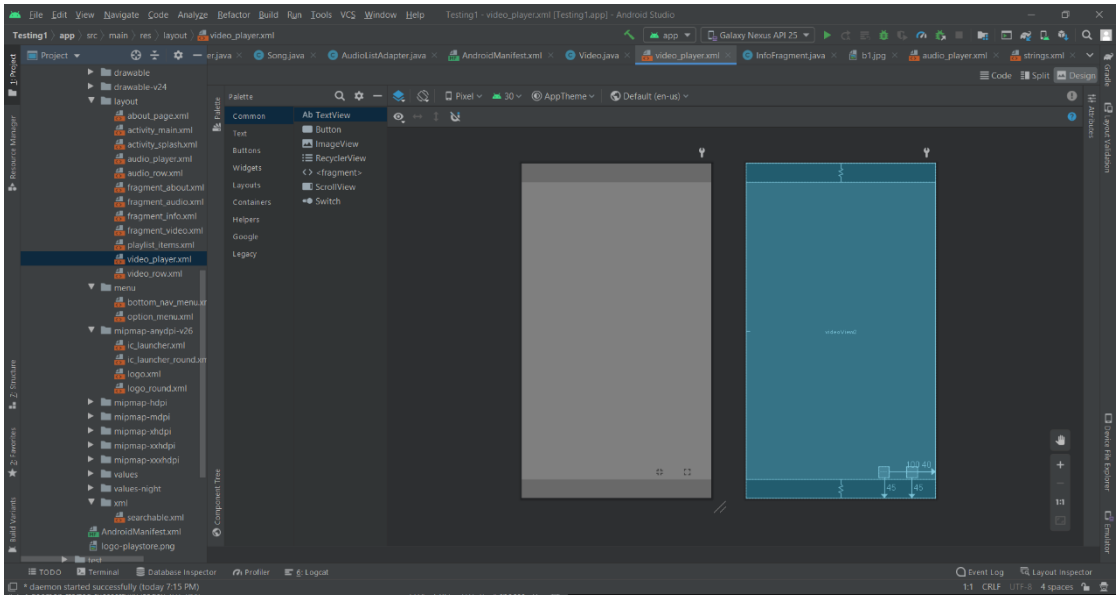


Figure 4.1.1.7: Design Mode of Video Player's XML layout

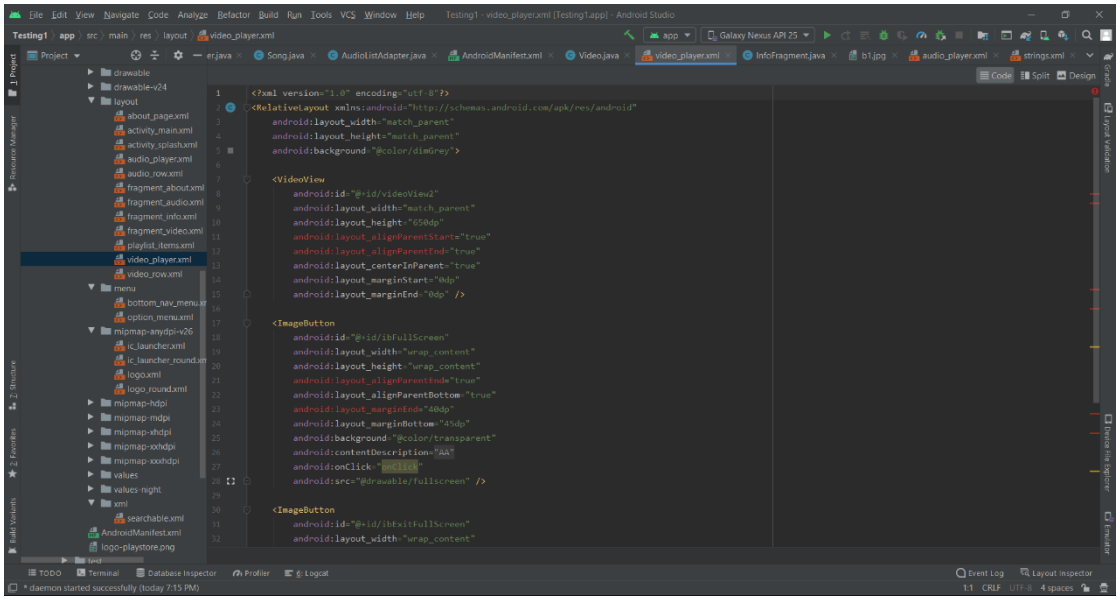


Figure 4.1.1.8: Text Mode of Video Player's XML layout

In Figure 4.1.1.9, it is the java class for the video player and the source code can refer to Appendices B (**VideoPlayer.java**). To retrieve the details of the video, we need to create an object class for the video which is the attributes of the video included video name, artist, duration and the path of the video, the source code refer to Appendices B (**Video.java**) (Figure 4.1.1.10).

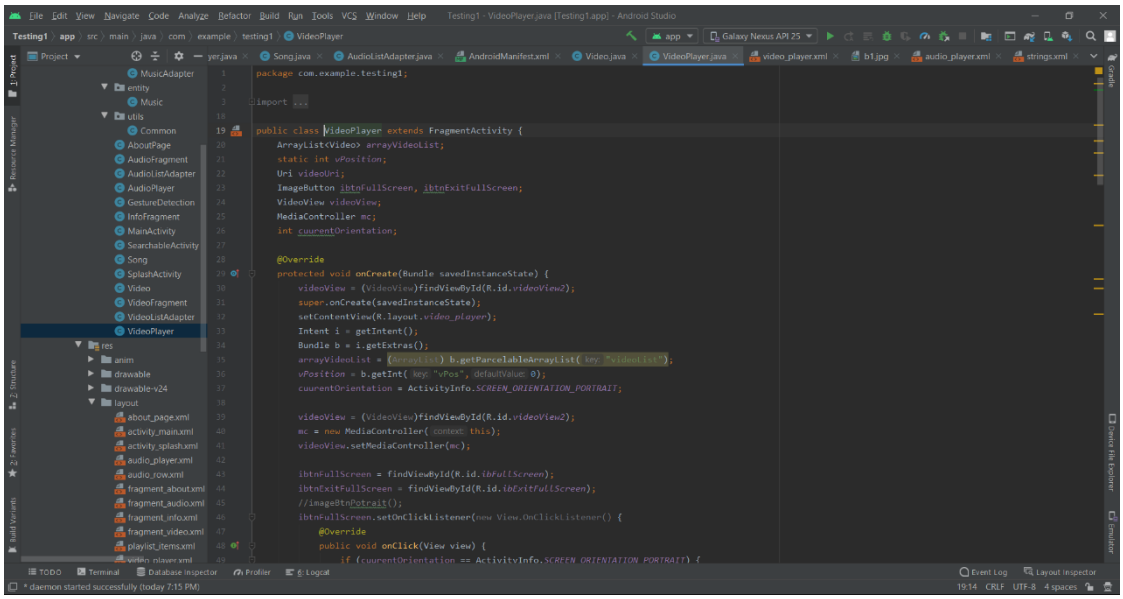


Figure 4.1.1.9: VideoPlayer's Java Class

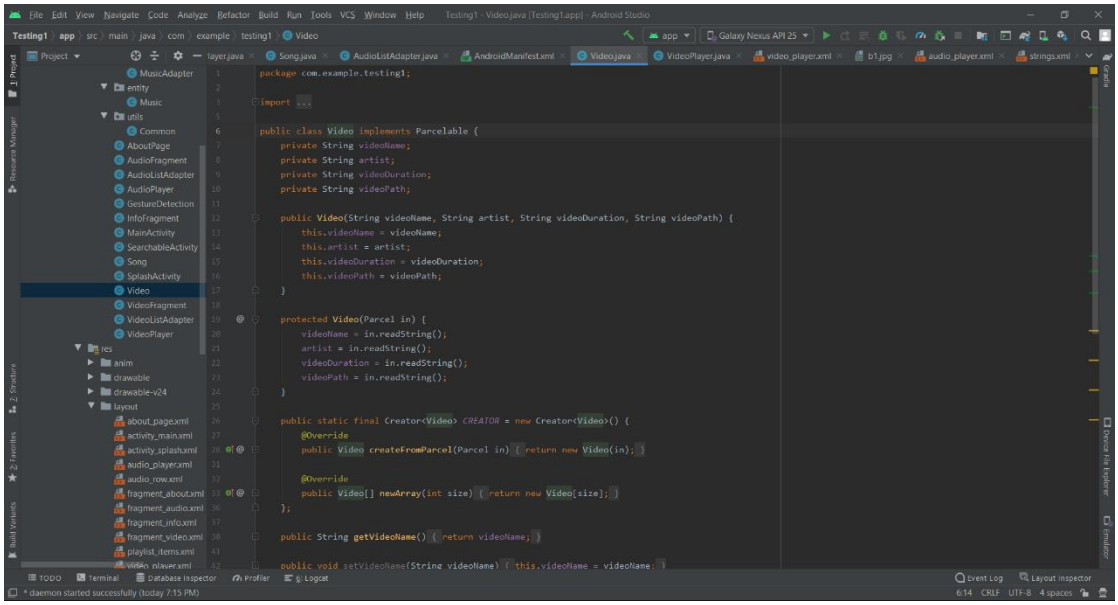


Figure 4.1.1.10: Video's Java Class

In Figure 4.1.1.11, an adapter view is needed to hold the details of the video and queue in the list view structure as the playlist in the video fragment (**VideoListAdapter.java**). After that, the information of the playlist in the video fragment will be extract from the mobile device (**VideoFragment.java**) (Figure 4.1.1.12).

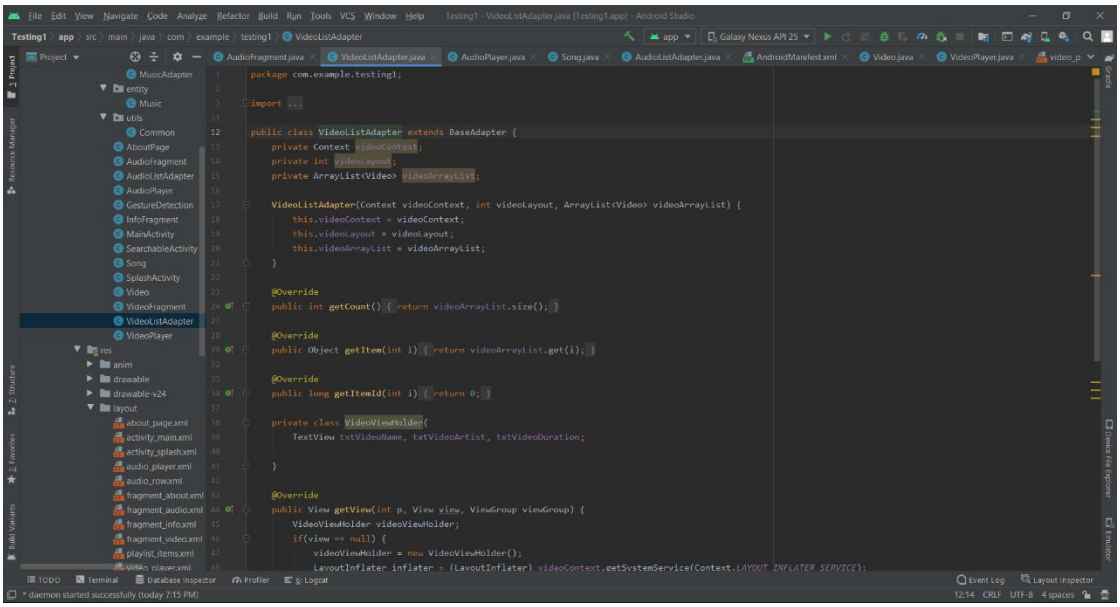


Figure 4.1.1.11: VideoListAdapter's Java Class

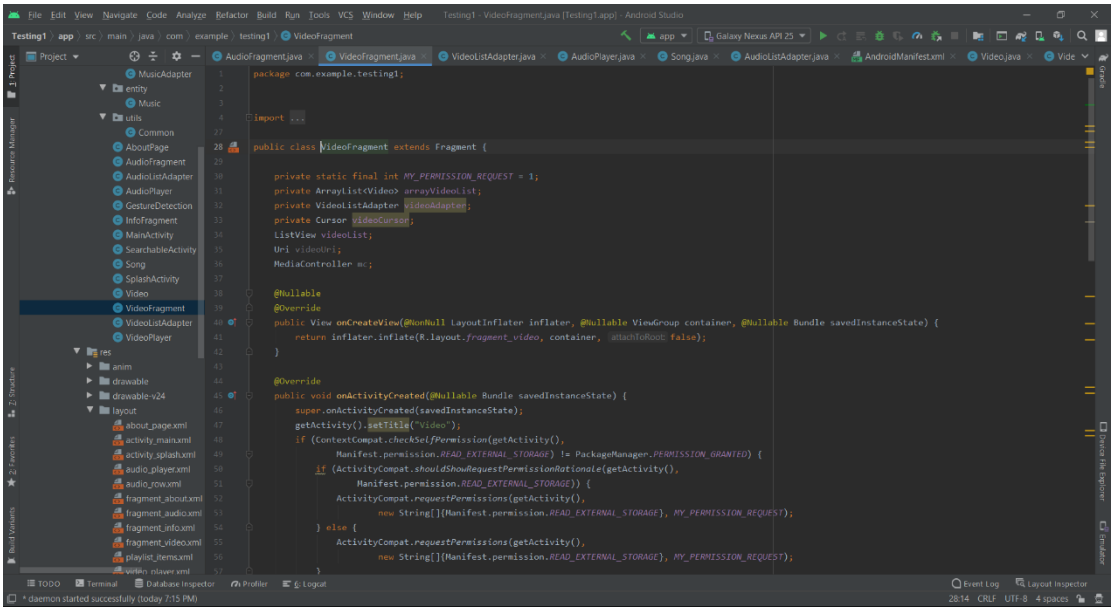


Figure 4.1.1.12: VideoFragment's Java Class

4.2 User Interface

4.2.1 Audio Fragment

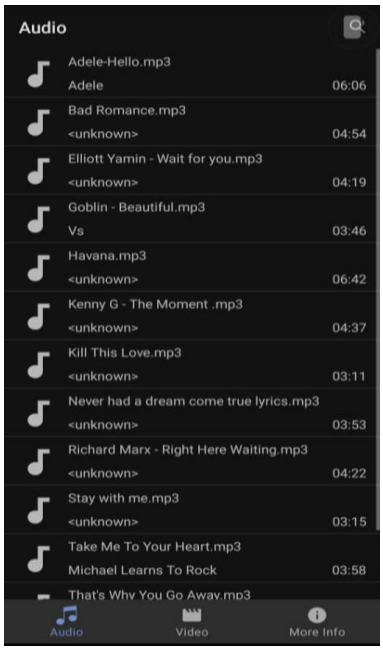


Figure 4.2.1.1: Audio Media Playlist

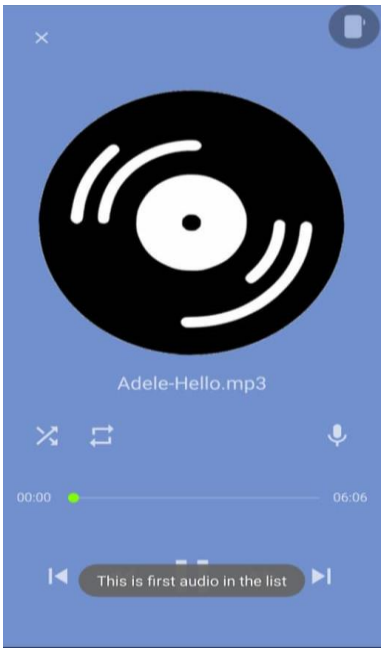


Figure 4.2.1.2: Audio Player

4.2.2 Video Fragment

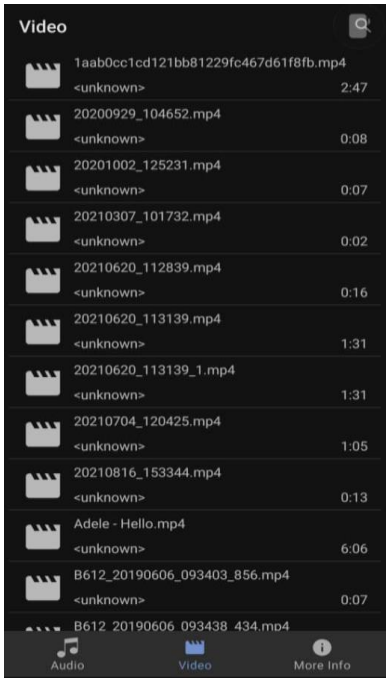


Figure 4.2.2.1: Video Media Playlist

Figure 4.2.2.2: Video Player

4.2.3 More Info Fragment

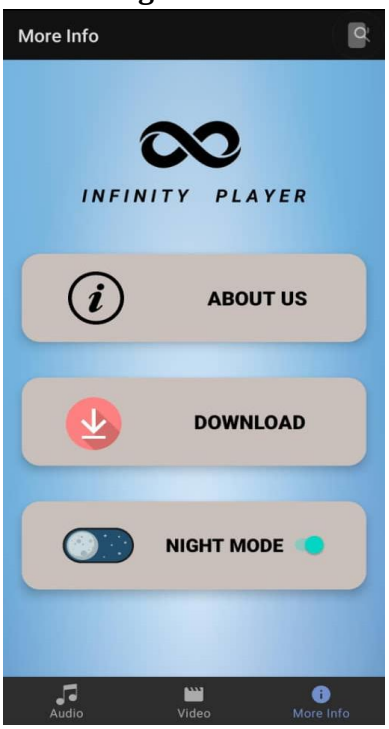


Figure 4.2.3.1: More Info Fragment

Figure 4.2.3.2: About Us Page

CHAPTER 5 CONCLUSION

5.1 Project Review

Research that is related to music video application is being done before the proposed project is developed to improve the knowledge on the concept required by this project and recognize the suitable methodology to be implemented in this project. This proposed project solved the limitation of most of the android media players. The core function of this project is to allow the user to listen to audio and video from the external storage in the mobile application easily. Also, this proposed project is developed in the Android platform to allow most of the mobile devices able to install this mobile application. Besides that, the structure view of this mobile application is neat, and it is user-friendly to the end-user. This mobile application has several ways for media control that allow the user to try different ways to control their media. Other than that, this project also provides some general information about the mobile application and links that will indicate the user to a website to download their music or video easily. Also, the user can easily switch between light mode and night mode based on their choice which is provided in this mobile application.

5.2 Future Work

In the future, this project can be enhanced by integrating with filter and search function that allows users to search the media easily. Besides that, it will be better if create playlist function can be added for future improvement as the user can easily create their playlist based on their personal preference. Moreover, the media player can also integrate with AI machine learning for scanning the music by title, artist, or album name. Furthermore, a trim video function can be integrate for the user to easily trim their video.

BIBLIOGRAPHY

Greenwich Council. (2008) Music Video A Brief History. From https://www.slideshare.net/crosswaysfederation/music-video-a-brief-history [Accessed 6 August 2020]

Laura. (2017) Music Device History. From https://www.solcatmusic.com/music-device-history/ [Accessed 6 August 2020]

Tai Campbell. (2017) Types of Music Video. From https://www.epikmusicvideos.com/blog/100- types-of-music-video-production.html [Accessed 8 August 2020]

Li Ma, Lei Gu and Jin Wang. (2014) Research and Development of Mobile Application for Android Platform. From https://pdfs.semanticscholar.org/f859/f00ed7bf29f3660c1b0474bc2d2639311150.pdf [Accessed 16 August 2020]

Zack Busch. (2019) 6 Stages of the Mobile Development Lifecycle. From https://learn.g2.com/mobile-development-lifecycle [Accessed 4 September 2020]

Nisha Gopinath Menon. (2019) What are the Various Phases of Mobile App Development?. From https://www.cognitiveclouds.com/insights/what-are-the-various-phases-of-mobile-app- development/ [Accessed 5 September 2020]

APPENDIX A – WEEKLY REPORT

FINAL YEAR PROJECT WEEKLY REPORT

~~(Project I~~ / Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.:** 3
Student Name & ID: LIM YU WEN (17ACB01113)	
Supervisor: Ts. Dr Ooi Chek Yee	
Project Title: Music Video Application Development using Android	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Literature Review

2. WORK TO BE DONE

- Layout design of the mobile application
- Wireframe of the mobile application

3. PROBLEMS ENCOUNTERED

- Consider how to design the layout of the mobile application

4. SELF EVALUATION OF THE PROGRESS

- Spend more time to review similar mobile application's user interface, arrangement and structure.

Ooi Chek Yee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

~~(Project I~~ / Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.:** 6
Student Name & ID: LIM YU WEN (17ACB01113)	
Supervisor: Ts. Dr Ooi Chek Yee	
Project Title: Music Video Application Development using Android	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Layout Design of the mobile application
- Wireframe of the mobile application

2. WORK TO BE DONE

- Implement the user interface of the mobile application
- Implement the audio player function for the mobile application

3. PROBLEMS ENCOUNTERED

- Need more time to adjust the position of the item
- Cannot fix the error

4. SELF EVALUATION OF THE PROGRESS

- Need more time to search online and figure out the solution for debugging

Ooi Chek Yee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

~~(Project I~~ / Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.:** 9
Student Name & ID: LIM YU WEN (17ACB01113)	
Supervisor: Ts. Dr Ooi Chek Yee	
Project Title: Music Video Application Development using Android	

2. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Implement the user interface of the mobile application
- Implement the audio player function for the mobile application

1. WORK TO BE DONE

- Implement the video player function for the mobile application

2. PROBLEMS ENCOUNTERED

- The application will crash when run the mobile application

3. SELF EVALUATION OF THE PROGRESS

- Need more time to search online and figure out the solution for debugging

Ooi Chek Yee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.:** 11
Student Name & ID: LIM YU WEN (17ACB01113)	
Supervisor: Ts. Dr Ooi Chek Yee	
Project Title: Music Video Application Development using Android	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Implement the video player function for the mobile application
- Done report from Chapter 1 to Chapter 3

2. WORK TO BE DONE

- Report Chapter 4 and Chapter 5

3. PROBLEMS ENCOUNTERED

- Search function does not work
- Notification does not work

4. SELF EVALUATION OF THE PROGRESS

- Does not have enough time to integrate all the suggested functionalities for the application as the implementation of notification function and search function used up too much time and the integration is not successful.

Ooi Chek Yee

Supervisor's signature

Student's signature

POSTER

Faculty of Information and Communication Technology

Music Video Application Development Using Android

Lim Yu Wen (17ACB01113)
Supervisor: Ts. Dr Ooi Chek Yee

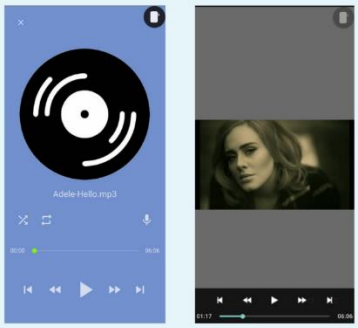
Introduction

As technology gets progressively advance, the user request for more convenience way to operate. An enchancement of the existing application by removing the unused feature and also intergating some useful and important features such as voice command and swipe gesture control.

Objectives

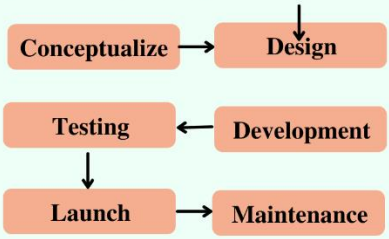
- Play music and video with able to view the music and video playlist
- Include the useful and important features
- Clean and clear interface to make the application user friendly

Infinity Player



Methodology

Mobile Application Development Life Cycle (MADLC)



Discussion

This mobile application named Infinity Player which is a simple music video player that are developed using Android language and it able to listen music and watch video. In the future, it will be improve by adding voice command and swipe gesture control.

PLAGIARISM CHECK RESULT

Music Video Application Development Using Android

ORIGINALITY REPORT

4%	3%	0%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Exeter Student Paper	1%
2	www.datasheetarchive.com Internet Source	1%
3	users.skynet.be Internet Source	<1%
4	www.slideshare.net Internet Source	<1%
5	es.scribd.com Internet Source	<1%
6	ntrs.nasa.gov Internet Source	<1%
7	www.coastalenergy.com Internet Source	<1%
8	Submitted to University of Portsmouth Student Paper	<1%
9	Submitted to Universiti Teknologi MARA Student Paper	<1%

10	Submitted to Sheffield Hallam University Student Paper	<1 %
11	apps.dtic.mil Internet Source	<1 %
12	scholarworks.uvm.edu Internet Source	<1 %
13	eprints.utar.edu.my Internet Source	<1 %
14	fr.scribd.com Internet Source	<1 %

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LIM YU WEN
ID Number(s)	17ACB01113
Programme / Course	BACHELOR OF COMMUNICATIONS AND NETWORKING ((HONOURS))
Title of Final Year Project	Music Video Application Development Using Android

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 4 % **Similarity by source** Internet Sources: 3 % Publications: 0 % Student Papers: 2 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Ooi Chek Yee

Signature of Supervisor

Name: Ooi Chek Yee

Date: 25th August 2021

Signature of Co-Supervisor

Name: _____

Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB01113
Student Name	LIM YU WEN
Supervisor Name	TS. DR OOI CHEK YEE

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. (Signature of Student) Date: 25th August 2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. Ooi Chek Yee (Signature of Supervisor) Date: 25th August 2021

APPENDIX B – SOURCE CODE

SplashActivity.java

```
package com.example.testing1;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
public class SplashActivity extends AppCompatActivity {
    private ImageView logoText;
    private static int splashTimeOut = 4000;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        logoText = findViewById(R.id.logo);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent i = new Intent(SplashActivity.this, MainActivity.class);
                startActivity(i);
                finish();
            }
        },splashTimeOut);
        Animation animSplash = AnimationUtils.loadAnimation(this, R.anim.animation_splash);
        logoText.startAnimation(animSplash);}}
```

MainActivity.java

```
package com.example.testing1;
import android.Manifest;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;
import com.google.android.material.bottomnavigation.BottomNavigationView;
public class MainActivity extends AppCompatActivity{
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
      bottomNavigationView.setOnNavigationItemSelectedListener(navListener);
      getSupportFragmentManager().beginTransaction().replace(R.id.fragment_frame, new
AudioFragment()).commit();
      if (ContextCompat.checkSelfPermission(this, Manifest.permission.RECORD_AUDIO) !=
PackageManager.PERMISSION_GRANTED){
         ActivityCompat.requestPermissions(this,
               new String[]{Manifest.permission.RECORD_AUDIO}, 1);
      }
```

```
    // Create bottom navigation view to classify fragment by clicking icon accordingly
    private BottomNavigationView.OnNavigationItemSelectedListener navListener =
            new BottomNavigationView.OnNavigationItemSelectedListener() {
                @Override
                public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
                    Fragment selectedMenuItem = null;
                    switch (menuItem.getItemId()) {
                        case R.id.audio_icon:
                            selectedMenuItem = new AudioFragment();
                            break;
                        case R.id.video_icon:
                            selectedMenuItem = new VideoFragment();
                            break;
                        case R.id.info_icon:
                            selectedMenuItem = new InfoFragment();
                            break;
                    }
                    getSupportFragmentManager().beginTransaction().replace(R.id.fragment_frame,
selectedMenuItem).commit();
                    return true;
                }
            };
    // Create option menu interface
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.option_menu, menu);
        return super.onCreateOptionsMenu(menu);
    }
    // Create AlertDialog and show up when back button of phone was pressed
    @Override
```

```
    public void onBackPressed() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle(R.string.exit);
        builder.setMessage(R.string.exit_message);
        // Action for pressing EXIT
        builder.setPositiveButton("EXIT", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                finish();
                Toast.makeText(getApplicationContext(), "Successfully Exit",
Toast.LENGTH_SHORT).show();
            }
        });
        // Action by pressing CANCEL
        builder.setNegativeButton("CANCEL", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                // Close the AlertDialog
            }
        });
        AlertDialog dialog = builder.show();
    }}
```

AudioFragment.java

```
package com.example.testing1;
import android.Manifest;
import android.app.SearchManager;
import android.content.ContentResolver;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.text.TextUtils;
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.AdapterView;
import android.widget.ListView;
import android.widget.SearchView;
import android.widget.Toast;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;
import androidx.annotation.Nullable;
import java.text.DecimalFormat;
import java.util.ArrayList;
public class AudioFragment extends Fragment {
```

```
    private static final int MY_PERMISSION_REQUEST = 1;
    private ArrayList<Song> arraySongList;
    private AudioListAdapter audioAdapter;
    private ListView songList;
    Cursor songCursor;
    static boolean shuffleBoolean = false, repeatBoolean = false;


    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.option_menu, menu);
        MenuItem searchItem = menu.findItem(R.id.search);
        SearchManager manager = (SearchManager)
getActivity().getSystemService(Context.SEARCH_SERVICE);
        SearchView searchView = (SearchView)menu.findItem(R.id.search).getActionView();
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String s) {
                return false;
            }
            @Override
            public boolean onQueryTextChange(String s) {
                if (TextUtils.isEmpty(s)) {
                    audioAdapter.filter("");
                    songList.clearTextFilter();
                } else {
                    audioAdapter.filter(s);
                }
                return true;
            }
        });
```

```
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if(id == R.id.search){
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
    @Override
    public View onCreateView( LayoutInflater inflater, ViewGroup container,  Bundle
savedInstanceState) {
       return inflater.inflate(R.layout.fragment_audio, container, false);
    }
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        getActivity().setTitle("Audio");
        if (ContextCompat.checkSelfPermission(getActivity(),
                Manifest.permission.READ_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),
                    Manifest.permission.READ_EXTERNAL_STORAGE)) {
                ActivityCompat.requestPermissions(getActivity(),
                        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
MY_PERMISSION_REQUEST);
            }
            else {
                ActivityCompat.requestPermissions(getActivity(),
                        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
MY_PERMISSION_REQUEST);
```

```
            }
        }
        else
            doThings();
    }
    private void doThings(){
        songList = getView().findViewById(R.id.listView);
        arraySongList = new ArrayList<>();
        getSong();
        audioAdapter = new AudioListAdapter(getActivity(), R.layout.audio_row, arraySongList);
        songList.setAdapter(audioAdapter);
        songList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int position, long l)
{
                Intent i = new Intent(getActivity().getApplicationContext(), AudioPlayer.class);
                i.putExtra("pos", position);
                i.putExtra("songList", arraySongList);
                startActivity(i);
            }
        });
    }
    private void getSong(){
        ContentResolver contentResolver = getActivity().getContentResolver();
        String[] proj = {MediaStore.Audio.Media.DISPLAY_NAME,
MediaStore.Audio.Media.ARTIST, MediaStore.Audio.Media.DURATION,
                MediaStore.Audio.Media.DATA};
        Uri songUri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
        songCursor = contentResolver.query(songUri, proj, null, null,
MediaStore.MediaColumns.DISPLAY_NAME);
        if(songCursor != null && songCursor.moveToFirst()){
```

```
        int songTitle =
songCursor.getColumnIndex(MediaStore.Audio.Media.DISPLAY_NAME);
        int songArtist = songCursor.getColumnIndex(MediaStore.Audio.Media.ARTIST);
        int songDuration =
songCursor.getColumnIndex(MediaStore.Audio.Media.DURATION);
        int songPath = songCursor.getColumnIndex(MediaStore.Audio.Media.DATA);
        do{
            String currentTitle = songCursor.getString(songTitle);
            String currentArtist = songCursor.getString(songArtist);
            String currentDuration = songCursor.getString(songDuration);
            String currentPath = songCursor.getString(songPath);
            int dur = Integer.parseInt(currentDuration);
            int mins =  dur / 1000 / 60;
            int secs = (dur / 1000) % 60;
            DecimalFormat df = new DecimalFormat("00");
            String m = df.format(mins);
            String s = df.format(secs);
            currentDuration = m + ":" + s;
            arraySongList.add(new Song(currentTitle, currentArtist, currentDuration,
currentPath));
        }while(songCursor.moveToNext());
      }
   }
   @Override
   public void onRequestPermissionsResult ( int requestCode, String[] permissions, int[]
grantResults){
      switch (requestCode) {
        case MY_PERMISSION_REQUEST: {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                if (ContextCompat.checkSelfPermission(getActivity(),
```

```
                    Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {

                Toast.makeText(getActivity(), "Permission granted!",
Toast.LENGTH_SHORT).show();

                doThings();

            }

            else{

                Toast.makeText(getActivity(), "No permission granted",
Toast.LENGTH_SHORT).show();

                getActivity().finish();

            }

            return;

        }

    }

  }

}
```

AudioListAdapter.java

```
package com.example.testing1;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;
import java.util.ArrayList;
import java.util.Locale;
public class AudioListAdapter extends BaseAdapter {
    private Context context;
    private int layout;
    static ArrayList<Song> arrayList;
    // Constructor
    AudioListAdapter(Context context, int layout, ArrayList<Song> arrayList) {
        this.context = context;
        this.layout = layout;
        this.arrayList = arrayList;}
    // Getter and Setter
    @Override
    public int getCount() {
        return arrayList.size();
    }
    @Override
    public Object getItem(int i) {
        return i;
    }
    @Override
    public long getItemId(int i) {
```

```
        return 0;
    }
    private class ViewHolder{
        TextView txtName, txtArtist, txtDuration;
    }
    // Create view for arraylist
    @Override
    public View getView(int position, View convertView, ViewGroup viewGroup) {
        ViewHolder viewHolder;
        if (convertView == null){
            viewHolder = new ViewHolder();
            LayoutInflater layoutInflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView = layoutInflater.inflate(layout, null);
            viewHolder.txtName = convertView.findViewById(R.id.textSongname);
            viewHolder.txtArtist = convertView.findViewById(R.id.textArtist);
            viewHolder.txtDuration = convertView.findViewById(R.id.textDuration);
            convertView.setTag(viewHolder);
        }
        else{
            viewHolder = (ViewHolder) convertView.getTag();
        }
        Song song = arrayList.get(position);
        viewHolder.txtName.setText(song.getName());
        viewHolder.txtArtist.setText(song.getArtist());
        viewHolder.txtDuration.setText(song.getDuration());
        return convertView;
    }
    public void filter(String charText){
        charText = charText.toLowerCase(Locale.getDefault());
```

```
        arrayList.clear();
        if(charText.length() == 0){
            arrayList.addAll(arrayList);
        }
        else{
            for(Song song : arrayList){
                if(song.getName().toLowerCase(Locale.getDefault())
                        .contains(charText)){
                    arrayList.add(song);
                }
            }
        }
        notifyDataSetChanged();
    }
}
```

Song.java

```
package com.example.testing1;


import android.os.Parcel;
import android.os.Parcelable;
public class Song implements Parcelable {
    private String name;
    private String artist;
    private String duration;
    private String path;
    private int song;
    private long id;
    Song(String name, String artist, String duration, String path) {
        this.name = name;
        this.artist = artist;
        this.duration = duration;
        this.path = path;
    }
    public Song(long songID, String name, String artist, String duration, int song) {
        id = songID;
        this.name = name;
        this.artist = artist;
        this.duration = duration;
        this.song = song;
    }
    protected Song(Parcel in) {
        name = in.readString();
        artist = in.readString();
        duration = in.readString();
        song = in.readInt();
```

```
        path = in.readString();
    }
    public static final Creator<Song> CREATOR = new Creator<Song>() {
        @Override
        public Song createFromParcel(Parcel in) {
            return new Song(in);
        }
        @Override
        public Song[] newArray(int size) {
            return new Song[size];
        }
    };
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getArtist() {
        return artist;
    }
    public void setArtist(String artist) {
        this.artist = artist;
    }
    public String getDuration() {
        return duration;
    }
    public void setDuration(String duration) {
        this.duration = duration;
    }
```

```
    public int getSong() {
        return song;
    }
    public void setSong(int song) {
        this.song = song;
    }
    public String getPath() {
        return path;
    }
    public void setPath(String path) {
        this.path = path;
    }
    public long getID(){ return id; }
    @Override
    public int describeContents() {
        return 0;
    }
    @Override
    public void writeToParcel(Parcel parcel, int i) {
        parcel.writeString(name);
        parcel.writeString(artist);
        parcel.writeString(duration);
        parcel.writeInt(song);
        parcel.writeString(path);
    }
}
```

AudioPlayer.java

```
package com.example.testing1;

import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentActivity;

import android.content.ActivityNotFoundException;

import android.content.Intent;

import android.media.AudioManager;

import android.media.MediaPlayer;

import android.net.Uri;

import android.os.Bundle;

import android.speech.RecognizerIntent;

import android.view.MotionEvent;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.SeekBar;

import android.widget.TextView;

import android.widget.Toast;

import java.text.DecimalFormat;

import java.util.ArrayList;

import java.util.Locale;

import java.util.Random;

import static com.example.testing1.AudioFragment.repeatBoolean;

import static com.example.testing1.AudioFragment.shuffleBoolean;

public class AudioPlayer extends FragmentActivity implements View.OnClickListener,
GestureDetection.SimpleGestureListener{

    AudioManager audioManager;

    GestureDetection detector;
```

```
    int currentVolume;
    static MediaPlayer mediaPlayer;
    ArrayList<Song> arraySongList;
    static int position;
    Uri songUri;
    Thread updateSeekBar;
    int currentPosition;
    TextView tvTitle, tvTime, tvDuration;
    String songTitle;
    SeekBar sb;
    ImageButton ibtnClose, ibtnPlay, ibtnPrev, ibtnNext, ibtnForward, ibtnRewind, ibtnShuffle,
ibtnRepeat,ibtnAudio;
    ImageView ivDisc;
    Animation animRotate;
    private Thread playThread, prevThread, nextThread;
    private String keeper = "";
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.audio_player);
        audioManager = (AudioManager) getSystemService(AUDIO_SERVICE);
        detector = new GestureDetection(this, this);
        tvTitle = findViewById(R.id.tvSongTitle);
        tvTime = findViewById(R.id.tvTime);
        tvDuration = findViewById(R.id.tvSongDuration);
        ibtnClose = findViewById(R.id.ibClose);
        ibtnPlay = findViewById(R.id.ibPlay);
        ibtnPrev = findViewById(R.id.ibPrevious);
        ibtnNext = findViewById(R.id.ibNext);
        ibtnForward = findViewById(R.id.ibForward);
```

```
ibtnRewind = findViewById(R.id.ibRewind);
ivDisc = findViewById(R.id.ivDisc);
sb = findViewById(R.id.seekBar);
animRotate = AnimationUtils.loadAnimation(this, R.anim.image_rotate);
ibtnShuffle = findViewById(R.id.ibShuffle);
ibtnRepeat = findViewById(R.id.ibRepeat);
ibtnAudio = findViewById(R.id.ibAudio);
updateSeekBar = new Thread() {
    @Override
    public void run() {
        int totalDuration = mediaPlayer.getDuration();
        currentPosition = 0;
        while (currentPosition < totalDuration) {
            try {
                sleep(500);
                currentPosition = mediaPlayer.getCurrentPosition();
                sb.setProgress(currentPosition);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
};
// Check whether the media player is playing
if (mediaPlayer != null) {
    mediaPlayer.stop(); // Stop playing
    mediaPlayer.release(); // Release media player
}
// Passing ArrayList data to new activity
Intent i = getIntent();
```

```
Bundle b = i.getExtras();
arraySongList = (ArrayList) b.getParcelableArrayList("songList");
position = b.getInt("pos", 0);
// Track the URI and then start to play
songUri = Uri.parse(arraySongList.get(position).getPath());
mediaPlayer = MediaPlayer.create(getApplicationContext(), songUri);
songTitle = arraySongList.get(position).getName();
tvTitle.setText(songTitle);
ibtnPlay.setImageResource(R.drawable.pause_70);
checkPosition();
mediaPlayer.start();
imageRotate();
tvDuration.setText(arraySongList.get(position).getDuration());
sb.setMax(mediaPlayer.getDuration());
updateSeekBar.start();
// SeekBar listener when seekbar was changed or updated
sb.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        int min = i / 1000 / 60;
        int sec = (i / 1000) % 60;
        DecimalFormat df = new DecimalFormat("00");
        String m = df.format(min);
        String s = df.format(sec);
        String progressValue = m + ":" + s;
        tvTime.setText(progressValue);
    }
    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }
```

```
        // When stop touching the seekbar
        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
            mediaPlayer.seekTo(seekBar.getProgress());
        }
    });
    ibtnShuffle.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (shuffleBoolean){
                shuffleBoolean = false;
                ibtnShuffle.setImageResource(R.drawable.shuffle_35);
            }
            else{
                shuffleBoolean = true;
                ibtnShuffle.setImageResource(R.drawable.shuffle_on_35);
            }
        }
    });
    ibtnRepeat.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (repeatBoolean){
                repeatBoolean = false;
                ibtnRepeat.setImageResource(R.drawable.repeat_35);
            }
            else{
                repeatBoolean = true;
                ibtnRepeat.setImageResource(R.drawable.repeat_on_35);
            }
```

```
        }
    });
    ibtnClose.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(mediaPlayer.isPlaying()) {
                mediaPlayer.stop();
                mediaPlayer.reset();
                mediaPlayer.release();
            }
            Fragment existing =
getSupportFragmentManager().findFragmentById(R.id.fragment_frame);
            if (existing == null) {
                getSupportFragmentManager().beginTransaction()
                        .replace(R.id.fragment_frame, new AudioFragment())
                        .commit();
            }
        }
    });
  }
  public void getSpecchInput(View view) {
    Intent speechRecognizerIntent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
Locale.getDefault());
    speechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_PROMPT,"speak to text");
    try {
        startActivityForResult(speechRecognizerIntent, 11);
    } catch (ActivityNotFoundException a){
```

```
        Toast.makeText(this, "Your device not support voice input",
Toast.LENGTH_LONG).show();
      }
   }
   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent data){
      super.onActivityResult(requestCode,resultCode,data);
      switch (requestCode){
         case 11:
         if (resultCode == RESULT_OK && data != null){
            ArrayList<String> matchesFound =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            keeper = matchesFound.get(0);
            if (keeper.equals("pause the song") || keeper.equals("pause song") ||
keeper.equals("pause the music") || keeper.equals("pause music") || keeper.equals("pause")){
               playBtnClicked();
               Toast.makeText(AudioPlayer.this,keeper,Toast.LENGTH_LONG).show();
            }
            else if (keeper.equals("play the song") || keeper.equals("play song") ||
keeper.equals("play the music") || keeper.equals("play music") || keeper.equals("play")){
               playBtnClicked();
               Toast.makeText(AudioPlayer.this,keeper,Toast.LENGTH_LONG).show();
            }
            else if (keeper.equals("play the next song") || keeper.equals("play next song") ||
keeper.equals("play the next music") || keeper.equals("play next music") ||
keeper.equals("next")){
               nextBtnClicked();
               Toast.makeText(AudioPlayer.this,keeper,Toast.LENGTH_LONG).show();
            }
            else if (keeper.equals("play the previous song") || keeper.equals("play previous song")
|| keeper.equals("play the previous music") || keeper.equals("play previous music") ||
keeper.equals("previous")){
               prevBtnClicked();
```

```
                Toast.makeText(AudioPlayer.this,keeper,Toast.LENGTH_LONG).show();
            }
        }
        else {
            Toast.makeText(getApplicationContext(), "Failed to recognize speech!",
Toast.LENGTH_LONG).show();
        }
        break;
    }
  }
  @Override
  protected void onResume() {
    playThreadBtn();
    nextThreadBtn();
    prevThreadBtn();
    super.onResume();
  }
  private void prevThreadBtn() {
    prevThread = new Thread(){
      @Override
      public void run() {
        super.run();
        ibtnPrev.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
            prevBtnClicked();
          }
        });
      }
    };
```

```
        prevThread.start();
    }
    private void prevBtnClicked() {
        if (mediaPlayer.isPlaying()){
            mediaPlayer.stop();
            mediaPlayer.release();
            if (shuffleBoolean && !repeatBoolean){
                position = getRandom(arraySongList.size()-1);
            }
            else if (!shuffleBoolean && !repeatBoolean){
                position = ((position - 1) < 0 ?  (arraySongList.size() - 1) : (position - 1));
            }
            songUri = Uri.parse(arraySongList.get(position).getPath());
            mediaPlayer= MediaPlayer.create(getApplicationContext(), songUri);
            songTitle = arraySongList.get(position).getName();
            tvTitle.setText(songTitle);
            currentPosition = mediaPlayer.getCurrentPosition();
            sb.setProgress(currentPosition);
            mediaPlayer.start();
            tvDuration.setText(arraySongList.get(position).getDuration());
            sb.setMax(mediaPlayer.getDuration());
        }
        else{
            mediaPlayer.stop();
            mediaPlayer.release();
            if (shuffleBoolean && !repeatBoolean){
                position = getRandom(arraySongList.size()-1);
            }
            else if (!shuffleBoolean && !repeatBoolean){
                position = ((position - 1) < 0 ?  (arraySongList.size() - 1) : (position - 1));
```

```
            }
            songUri = Uri.parse(arraySongList.get(position).getPath());
            mediaPlayer= MediaPlayer.create(getApplicationContext(), songUri);
            songTitle = arraySongList.get(position).getName();
            tvTitle.setText(songTitle);
            currentPosition = mediaPlayer.getCurrentPosition();
            sb.setProgress(currentPosition);
            mediaPlayer.start();
            tvDuration.setText(arraySongList.get(position).getDuration());
            sb.setMax(mediaPlayer.getDuration());
        }
    }
    private void nextThreadBtn() {
        nextThread = new Thread(){
            @Override
            public void run() {
                super.run();
                ibtnNext.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        nextBtnClicked();
                    }
                });
            }
        };
        nextThread.start();
    }
    private void nextBtnClicked() {
        if (mediaPlayer.isPlaying()){
            mediaPlayer.stop();
```

```
            mediaPlayer.release();
            if (shuffleBoolean && !repeatBoolean){
                position = getRandom(arraySongList.size()-1);
            }
            else if (!shuffleBoolean && !repeatBoolean){
                position = ((position + 1) % arraySongList.size());
            }
            songUri = Uri.parse(arraySongList.get(position).getPath());
            mediaPlayer= MediaPlayer.create(getApplicationContext(), songUri);
            songTitle = arraySongList.get(position).getName();
            tvTitle.setText(songTitle);
            currentPosition = mediaPlayer.getCurrentPosition();
            sb.setProgress(currentPosition);
            mediaPlayer.start();
            tvDuration.setText(arraySongList.get(position).getDuration());
            sb.setMax(mediaPlayer.getDuration());
        }
        else{
            mediaPlayer.stop();
            mediaPlayer.release();
            if (shuffleBoolean && !repeatBoolean){
                position = getRandom(arraySongList.size()-1);
            }
            else if (!shuffleBoolean && !repeatBoolean){
                position = ((position + 1) % arraySongList.size());
            }
            songUri = Uri.parse(arraySongList.get(position).getPath());
            mediaPlayer= MediaPlayer.create(getApplicationContext(), songUri);
            songTitle = arraySongList.get(position).getName();
            tvTitle.setText(songTitle);
```

```
            currentPosition = mediaPlayer.getCurrentPosition();
            sb.setProgress(currentPosition);
            mediaPlayer.start();
            tvDuration.setText(arraySongList.get(position).getDuration());
            sb.setMax(mediaPlayer.getDuration());
        }
    }
    private void playThreadBtn() {
        playThread = new Thread(){
            @Override
            public void run() {
                super.run();
                ibtnPlay.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        playBtnClicked();
                    }
                });
            }
        };
        playThread.start();
    }
    private void playBtnClicked() {
        if (mediaPlayer.isPlaying()){
            ibtnPlay.setImageResource(R.drawable.play_70);
            mediaPlayer.pause();
            ivDisc.animate().cancel();
            ivDisc.clearAnimation();
        }
        else {
```

```
            ibtnPlay.setImageResource(R.drawable.pause_70);
            mediaPlayer.start();
            ivDisc.setAnimation(animRotate);
            ivDisc.animate().start();
        }
    }
    // Image button click listener
    @Override
    public void onClick(View v) {
        int id = v.getId();
        switch (id) {
            case R.id.ibForward:
                mediaPlayer.seekTo(mediaPlayer.getCurrentPosition()+5000);
                break;

            case R.id.ibRewind:
                mediaPlayer.seekTo(mediaPlayer.getCurrentPosition()-5000);
                break;
        }
    }
    private int getRandom(int i) {
        Random random = new Random();
        return random.nextInt(i + 1);
    }
    // Check the position of the audio in arraylist for matching
    public void checkPosition(){
        if (position == 0) {
            ibtnPrev.setClickable(false);
            Toast.makeText(this, "This is first audio in the list", Toast.LENGTH_SHORT).show();
        }
```

```
    else if(position == arraySongList.size()-1){
        ibtnNext.setClickable(false);
        Toast.makeText(this, "This is last audio in the list", Toast.LENGTH_SHORT).show();
    }
    else {
        position = position%arraySongList.size();
        ibtnPrev.setClickable(true);
        ibtnNext.setClickable(true);
    }
}
@Override
public boolean dispatchTouchEvent(MotionEvent me) {
    this.detector.onTouchEvent(me);
    return super.dispatchTouchEvent(me);
}
@Override
public void onPointerCaptureChanged(boolean hasCapture) {
}
// Function for swiping the phone screen with different direction
@Override
public void onSwipe(int direction) {
    // TODO Auto-generated method stub
    String str = "";
    switch (direction) {
        // Swipe left to next the song
        case GestureDetection.SWIPE_LEFT:
            nextBtnClicked();
            //playNext();
            str = "Swipe Left";
            break;
```

```
            // Swipe right to previous the song
            case GestureDetection.SWIPE_RIGHT:
                prevBtnClicked();
                //playPrevious();
                str = "Swipe Right";
                break;
            // Swipe down to decrease audio volume
            case GestureDetection.SWIPE_DOWN:
                currentVolume = audioManager
                        .getStreamVolume(AudioManager.STREAM_MUSIC);
                audioManager.setStreamVolume(AudioManager.STREAM_MUSIC,
                        currentVolume - 1, 0);
                str = "Swipe Down, Volume Decreasing";
                break;
            // Swipe up tp increase audio volume
            case GestureDetection.SWIPE_UP:
                currentVolume = audioManager
                        .getStreamVolume(AudioManager.STREAM_MUSIC);
                audioManager.setStreamVolume(AudioManager.STREAM_MUSIC,
                        currentVolume + 1, 0);
                str = "Swipe Up, Volume Incresing";
                break;
        }
        Toast.makeText(this, str, Toast.LENGTH_SHORT).show();
    }
    // Image rotatation
    public void imageRotate(){
        ivDisc.startAnimation(animRotate);
    }
}
```

GestureDetection.java

```
package com.example.testing1;
import android.app.Activity;
import android.view.GestureDetector;
import android.view.MotionEvent;
public class GestureDetection extends GestureDetector.SimpleOnGestureListener {
    public final static int SWIPE_UP = 1;
    public final static int SWIPE_DOWN = 2;
    public final static int SWIPE_LEFT = 3;
    public final static int SWIPE_RIGHT = 4;
    public final static int MODE_SOLID = 1;
    public final static int MODE_DYNAMIC = 2;
    private final static int ACTION_FAKE = -13; // just an unlikely number
    private int swipe_Min_Distance = 50;
    private int swipe_Max_Distance = 350;
    private int swipe_Min_Velocity = 100;
    private int mode = MODE_DYNAMIC;
    private boolean running = true;
    private boolean tapIndicator = false
    private Activity context;
    private GestureDetector detector;
    private SimpleGestureListener listener;
    public GestureDetection(Activity context, SimpleGestureListener sgl){
        this.context = context;
        this.detector = new GestureDetector(context, this);
        this.listener = sgl;
    }
    public void onTouchEvent(MotionEvent event){
        if(!this.running)
            return;
```

```
        boolean result = this.detector.onTouchEvent(event);
        if(this.mode == MODE_SOLID)
            event.setAction(MotionEvent.ACTION_CANCEL);
        else if(this.mode == MODE_DYNAMIC){
            if(event.getAction() == ACTION_FAKE)
                event.setAction(MotionEvent.ACTION_UP);
            else if(result)
                event.setAction(MotionEvent.ACTION_CANCEL);
            else if(this.tapIndicator){
                event.setAction(MotionEvent.ACTION_DOWN);
                this.tapIndicator = false;
            }
        }
    }
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX,
                           float velocityY) {
        final float xDistance = Math.abs(e1.getX() - e2.getX());
        final float yDistance = Math.abs(e1.getY() - e2.getY());
        if (xDistance > this.swipe_Max_Distance
                || yDistance > this.swipe_Max_Distance)
            return false;
        velocityX = Math.abs(velocityX);
        velocityY = Math.abs(velocityY);
        boolean result = false;
        if (velocityX > this.swipe_Min_Velocity
                && xDistance > this.swipe_Min_Distance) {
            if (e1.getX() > e2.getX()) // right to left
                this.listener.onSwipe(SWIPE_LEFT);
            else
```

```
            this.listener.onSwipe(SWIPE_RIGHT);
        result = true;
    } else if (velocityY > this.swipe_Min_Velocity
            && yDistance > this.swipe_Min_Distance) {
        if (e1.getY() > e2.getY()) // bottom to up
            this.listener.onSwipe(SWIPE_UP);
        else
            this.listener.onSwipe(SWIPE_DOWN);
        result = true;
    }
    return result;
}
@Override
public boolean onSingleTapConfirmed(MotionEvent arg) {
    if (this.mode == MODE_DYNAMIC) {
        arg.setAction(ACTION_FAKE);
        this.context.dispatchTouchEvent(arg);
    }
    return false;
}
interface SimpleGestureListener{
    void onSwipe(int direction);
}}
```

VideoFragment.java

```
package com.example.testing1;
import android.Manifest;
import android.content.ContentResolver;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.MediaController;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;
import java.text.DecimalFormat;
import java.util.ArrayList
public class VideoFragment extends Fragment {
    private static final int MY_PERMISSION_REQUEST = 1;
    private ArrayList<Video> arrayVideoList;
    private VideoListAdapter videoAdapter;
    private Cursor videoCursor;
    ListView videoList;
    Uri videoUri;
```

```
    MediaController mc;
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_video, container, false);
    }
    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        getActivity().setTitle("Video");
        if (ContextCompat.checkSelfPermission(getActivity(),
                Manifest.permission.READ_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),
                    Manifest.permission.READ_EXTERNAL_STORAGE)) {
                ActivityCompat.requestPermissions(getActivity(),
                        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
MY_PERMISSION_REQUEST);
            } else {
                ActivityCompat.requestPermissions(getActivity(),
                        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
MY_PERMISSION_REQUEST);
            }
        }
        else
            showVideo();
    }
    private void showVideo(){
        videoList = (ListView) getView().findViewById(R.id.videoList);
        arrayVideoList = new ArrayList<>();
```

```
        getVideo();
        videoAdapter = new VideoListAdapter(getActivity(), R.layout.video_row, arrayVideoList);
        videoList.setAdapter(videoAdapter);
        videoList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int vPosition, long
l) {
                Intent i = new Intent(getActivity().getApplicationContext(), VideoPlayer.class);
                i.putExtra("vPos", vPosition);
                i.putExtra("videoList", arrayVideoList);
                startActivity(i);
            }
        });
    }
    private void getVideo(){
        ContentResolver contentResolver = getActivity().getContentResolver();
        String[] vProj = {MediaStore.Video.Media.DISPLAY_NAME,
MediaStore.Video.Media.ARTIST, MediaStore.Video.Media.DURATION,
                MediaStore.Video.Media.DATA};
        Uri songUri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
        videoCursor = contentResolver.query(songUri, vProj, null, null,
MediaStore.MediaColumns.DISPLAY_NAME);

        if(videoCursor != null && videoCursor.moveToFirst()){
            int videoTitle =
videoCursor.getColumnIndex(MediaStore.Video.Media.DISPLAY_NAME);
            int videoArtist = videoCursor.getColumnIndex(MediaStore.Video.Media.ARTIST);
            int videoDuration =
videoCursor.getColumnIndex(MediaStore.Video.Media.DURATION);
            int videoPath = videoCursor.getColumnIndex(MediaStore.Video.Media.DATA);
            do{
```

```
            String currentTitle = videoCursor.getString(videoTitle);
            String currentArtist = videoCursor.getString(videoArtist);
            String currentDuration = videoCursor.getString(videoDuration);
            String currentPath = videoCursor.getString(videoPath);

            int dur = Integer.parseInt(currentDuration);
            int mins =  dur / 1000 / 60;
            int secs = (dur / 1000) % 60;
            DecimalFormat df = new DecimalFormat("00");
            String m = Integer.toString(mins);
            String s = df.format(secs);
            currentDuration = m + ":" + s;
            arrayVideoList.add(new Video(currentTitle, currentArtist, currentDuration,
currentPath));
          }while(videoCursor.moveToNext());
       }
    }
}
```

VideoListAdapter.java

```
package com.example.testing1;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;
import java.util.ArrayList;
public class VideoListAdapter extends BaseAdapter {
    private Context videoContext;
    private int videoLayout;
    private ArrayList<Video> videoArrayList;
    VideoListAdapter(Context videoContext, int videoLayout, ArrayList<Video> videoArrayList)
{
        this.videoContext = videoContext;
        this.videoLayout = videoLayout;
        this.videoArrayList = videoArrayList;
    }
    @Override
    public int getCount() {
        return videoArrayList.size();
    }
    @Override
    public Object getItem(int i) {
        return videoArrayList.get(i);
    }
    @Override
    public long getItemId(int i) {
        return 0;
```

```
    }
    private class VideoViewHolder{
        TextView txtVideoName, txtVideoArtist, txtVideoDuration;
    }
    @Override
    public View getView(int p, View view, ViewGroup viewGroup) {
        VideoViewHolder videoViewHolder;
        if(view == null) {
            videoViewHolder = new VideoViewHolder();
            LayoutInflater inflater = (LayoutInflater)
videoContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            view = inflater.inflate(videoLayout, null);
            videoViewHolder.txtVideoName = (TextView) view.findViewById(R.id.tvVideo_title);
            videoViewHolder.txtVideoArtist = (TextView) view.findViewById(R.id.tvVideo_artist);
            videoViewHolder.txtVideoDuration = (TextView)
view.findViewById(R.id.textVideo_duration) ;
            view.setTag(videoViewHolder);
        }
        else{
            videoViewHolder = (VideoViewHolder) view.getTag();
        }
        Video video = videoArrayList.get(p);
        videoViewHolder.txtVideoName.setText(video.getVideoName());
        videoViewHolder.txtVideoArtist.setText(video.getArtist());
        videoViewHolder.txtVideoDuration.setText(video.getVideoDuration());
        return view;
    }}
```

Video.java

```
package com.example.testing1;
import android.os.Parcel;
import android.os.Parcelable;
public class Video implements Parcelable {
    private String videoName;
    private String artist;
    private String videoDuration;
    private String videoPath;
    public Video(String videoName, String artist, String videoDuration, String videoPath) {
        this.videoName = videoName;
        this.artist = artist;
        this.videoDuration = videoDuration;
        this.videoPath = videoPath;
    }
    protected Video(Parcel in) {
        videoName = in.readString();
        artist = in.readString();
        videoDuration = in.readString();
        videoPath = in.readString();
    }
    public static final Creator<Video> CREATOR = new Creator<Video>() {
        @Override
        public Video createFromParcel(Parcel in) {
            return new Video(in);
        }
        @Override
        public Video[] newArray(int size) {
            return new Video[size];
        }
```

```
    }
    public String getVideoName() {
        return videoName;
    }
    public void setVideoName(String videoName) {
        this.videoName = videoName;
    }
    public String getArtist() {
        return artist;
    }
    public void setArtist(String artist) {
        this.artist = artist;
    }
    public String getVideoDuration() {
        return videoDuration;
    }
    public void setVideoDuration(String videoDuration) {
        this.videoDuration = videoDuration;
    }
    public String getVideoPath() {
        return videoPath;
    }
    public void setVideoPath(String videoPath) {
        this.videoPath = videoPath;
    }
    @Override
    public int describeContents() {
        return 0;
    }
    @Override
```

```
    public void writeToParcel(Parcel parcel, int i) {
        parcel.writeString(videoName);
        parcel.writeString(artist);
        parcel.writeString(videoDuration);
        parcel.writeString(videoPath);
    }
}
```

VideoPlayer.java

```
package com.example.testing1;
import androidx.fragment.app.FragmentActivity;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.content.res.Configuration;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.MediaController;
import android.widget.Toast;
import android.widget.VideoView;
import java.util.ArrayList;
public class VideoPlayer extends FragmentActivity {
    ArrayList<Video> arrayVideoList;
    static int vPosition;
    Uri videoUri;
    ImageButton ibtnFullScreen, ibtnExitFullScreen;
    VideoView videoView;
    MediaController mc;
    int cuurentOrientation;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        videoView = (VideoView)findViewById(R.id.videoView2);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.video_player);
        Intent i = getIntent();
        Bundle b = i.getExtras();
        arrayVideoList = (ArrayList) b.getParcelableArrayList("videoList");
```

```
        vPosition = b.getInt("vPos",0);
        cuurentOrientation = ActivityInfo.SCREEN_ORIENTATION_PORTRAIT;
        videoView = (VideoView)findViewById(R.id.videoView2);
        mc = new MediaController(this);
        videoView.setMediaController(mc);
        ibtnFullScreen = findViewById(R.id.ibFullScreen);
        ibtnExitFullScreen = findViewById(R.id.ibExitFullScreen);
        //imageBtnPotrait();
        ibtnFullScreen.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (cuurentOrientation == ActivityInfo.SCREEN_ORIENTATION_PORTRAIT) {
                    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
                    cuurentOrientation = ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE;
                    imageBtnLandscape();
                }
                else
                {
                    // no change orientation
                    Toast.makeText(getApplicationContext(), "Already in Landscape Mode",
Toast.LENGTH_SHORT).show();
                }
            }
        });
        ibtnExitFullScreen.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (cuurentOrientation == ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE) {
                    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
                    cuurentOrientation = ActivityInfo.SCREEN_ORIENTATION_PORTRAIT;
```

```
                imageBtnPortrait();
            }
            else
            {
                // no change orientation
                Toast.makeText(getApplicationContext(), "Already in Portrait Mode",
Toast.LENGTH_SHORT).show();
            }
        }
    });
    mc.setPrevNextListeners(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            playVideoNext();
        }
    }, new View.OnClickListener(){
        @Override
        public void onClick(View v) {
            playVideoPrevious();
        }
    });
    playVideo();
}
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
}
// Play next video
private void playVideoNext(){
    if (vPosition == arrayVideoList.size()-1){
```

```
            vPosition = arrayVideoList.size()-1;
            Toast.makeText(this, "This is last video in the list", Toast.LENGTH_SHORT).show();
        }
        else {
            vPosition = vPosition+1;
        }
        videoUri = Uri.parse(arrayVideoList.get(vPosition).getVideoPath());
        videoView.setVideoPath(videoUri.toString());
        videoView.setVideoURI(videoUri);
        videoView.setMediaController(mc);
        mc.setAnchorView(videoView);
        videoView.start();
    }
    // Play previous video
    private void playVideoPrevious(){
        if(vPosition-1 <= 0){
            vPosition = 0;
            Toast.makeText(this, "This is first video in the list", Toast.LENGTH_SHORT).show();
        }
        else {
            vPosition = vPosition-1;
        }
        videoUri = Uri.parse(arrayVideoList.get(vPosition).getVideoPath());
        videoView.setVideoPath(videoUri.toString());
        videoView.setVideoURI(videoUri);
        videoView.setMediaController(mc);
        mc.setAnchorView(videoView);
        videoView.start();
    }
    // Play video
```

```
    private void playVideo(){
        if(videoView.isPlaying()){
            videoView.stopPlayback();
            mc.setEnabled(false);
        }
        if(vPosition == 0){
            Toast.makeText(this, "This is first video in the list", Toast.LENGTH_SHORT).show();
        }
        else if (vPosition == arrayVideoList.size()-1){
            Toast.makeText(this, "This is last video in the list", Toast.LENGTH_SHORT).show();
        }
        videoUri = Uri.parse(arrayVideoList.get(vPosition).getVideoPath());
        videoView.setVideoPath(videoUri.toString());
        videoView.setVideoURI(videoUri);
        videoView.setMediaController(mc);
        mc.setEnabled(true);
        mc.setAnchorView(videoView);
        videoView.start();
    }
    // Set Y-axis coordinate for portrait mode
    private void imageBtnPortrait(){
        ibtnFullScreen.setY(500);
        ibtnExitFullScreen.setY(500);
    }
    // Set Y-axis coordinate for landscape mode
    private void imageBtnLandscape(){
        ibtnFullScreen.setY(1650);
        ibtnExitFullScreen.setY(1650);
    }
}
```

InfoFragment.java

```
package com.example.testing1;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatDelegate;
import androidx.appcompat.widget.SwitchCompat;
import androidx.cardview.widget.CardView;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.CompoundButton;
import android.widget.ImageView;

public class InfoFragment extends Fragment{
    public CardView about, download, mode;
    ImageView imageView;
    SwitchCompat switchCompat;
    SharedPreferences sharedPreferences = null;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_info, container, false);
        getActivity().setTitle("More Info");

        about = (CardView) view.findViewById(R.id.about);
        about.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            Intent i;
            i = new Intent(getActivity(), AboutPage.class);
            startActivity(i);
        }
    });
    download = (CardView) view.findViewById(R.id.download);
    download.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent browseIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://musicdownload.zone/en1/"));
            startActivity(browseIntent);
        }
    });

    imageView = view.findViewById(R.id.daynight);
    mode = (CardView) view.findViewById(R.id.mode);
    switchCompat = view.findViewById(R.id.switchCompat);
    sharedPreferences = this.getActivity().getSharedPreferences("night",0);
    Boolean booleanValue = sharedPreferences.getBoolean("night_mode",true);
    if (booleanValue){
        AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_YES);
        switchCompat.setChecked(true);
        imageView.setImageResource(R.drawable.night);
    }
    switchCompat.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
```

```
            if (isChecked){
        AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_YES);
                switchCompat.setChecked(true);
                imageView.setImageResource(R.drawable.night);
                SharedPreferences.Editor editor = sharedPreferences.edit();
                editor.putBoolean("night_mode",true);
                editor.commit();
            }else {
         AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
                switchCompat.setChecked(false);
                imageView.setImageResource(R.drawable.night);
                SharedPreferences.Editor editor = sharedPreferences.edit();
                editor.putBoolean("night_mode",false);
                editor.commit();
            }
        }
      });
      return view;
   }
}
```

AboutPage.java

```
package com.example.testing1;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Build;
import android.os.Bundle;
import android.widget.ImageView;
public class AboutPage extends AppCompatActivity {
    ImageView logo, information;
    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.about_page);
        setTitle("About Us");
        logo = findViewById(R.id.logo);
        information = findViewById(R.id.information);
    }
}
```

SearchableActivity.java

```
package com.example.testing1;
import android.app.Activity;
import android.app.SearchManager;
import android.content.Intent;
import android.os.Bundle;
public class SearchableActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        handleIntent(getIntent());
    }
    @Override
    protected void onNewIntent(Intent intent) {
        handleIntent(intent);
    }
    private void handleIntent(Intent intent) {
        if (Intent.ACTION_SEARCH.equals(intent.getAction())) {
            String query = intent.getStringExtra(SearchManager.QUERY);
            //use the query to search your data somehow
        }
    }
}
```

animation_splash.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:duration="2500">
</set>
```

image_rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <rotate android:duration="120000"
      android:fromDegrees="0"
      android:pivotX="50%"
      android:pivotY="50%"
      android:repeatCount="1"
      android:startOffset="0"
      android:toDegrees="28800" />
</set>
```

option_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto">

   <item
      android:id="@+id/search"
      android:title="@string/search"
      android:orderInCategory="1"
      android:icon="@drawable/search"
      app:showAsAction="collapseActionView|ifRoom"
      app:actionViewClass="android.widget.SearchView" /></menu>
```

bottom_nav_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/audio_icon"
        android:icon="@drawable/music_player"
        android:title="@string/audio"
        android:enabled="true"
        app:showAsAction="ifRoom" />

    <item
        android:id="@+id/video_icon"
        android:icon="@drawable/movie"
        android:title="@string/video"
        android:enabled="true"
        app:showAsAction="ifRoom" />

    <item
        android:id="@+id/info_icon"
        android:icon="@drawable/info"
        android:title="@string/info"
        android:enabled="true"
        app:showAsAction="ifRoom" />
</menu>
```

activity_splash.java

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/b1"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/logo"
        android:src="@drawable/infinitylogo"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_centerHorizontal="true"
       android:layout_marginTop="200dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I N F I N I T Y      P L A Y E R"
        android:textStyle="bold|italic"
        android:textSize="28dp"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"
        android:paddingTop="100dp"
        android:textColor="@color/black"/>

</RelativeLayout>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        app:menu="@menu/bottom_nav_menu"
        app:labelVisibilityMode="labeled"/>

    <FrameLayout
        android:id="@+id/fragment_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentTop="true"
        android:layout_above="@id/bottom_navigation">
    </FrameLayout>
</RelativeLayout>
```

fragment_audio.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" />
    </ScrollView>

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</RelativeLayout>
```

audio_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="44dp"
        android:layout_height="44dp"
        android:layout_alignTop="@+id/textSongname"
        android:layout_alignParentStart="true"
        android:layout_marginStart="15dp"
        android:layout_marginTop="5dp"
        android:src="@drawable/audiotrack" />

    <TextView
        android:id="@+id/textSongname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginStart="70dp"
        android:layout_marginTop="5dp"
        android:text="Song name"
        android:textStyle="normal"/>

    <TextView
        android:id="@+id/textArtist"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="70dp"
        android:layout_marginTop="35dp"
        android:layout_marginBottom="5dp"
        android:text="Artist" />

    <TextView
        android:id="@+id/textDuration"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginTop="35dp"
        android:layout_marginEnd="15dp"
        android:layout_marginBottom="5dp"
        android:text="Duration"
       />
</RelativeLayout>
```

audio_player.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/parentRelativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary">

    <ImageButton
        android:id="@+id/ibClose"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_marginStart="30dp"
        android:layout_marginTop="30dp"
        android:background="@color/transparent"
        android:onClick="onClick"
        android:src="@drawable/close_35" />

    <ImageView
        android:id="@+id/ivDisc"
        android:layout_width="350dp"
        android:layout_height="350dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:src="@drawable/discimage"/>

    <SeekBar
```

```
    android:id="@+id/seekBar"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="60dp"
    android:layout_marginBottom="180dp"
    android:progressTint="#ffffff"
    android:thumbTint="@color/limeGreen" />

<TextView
    android:id="@+id/tvSongTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="5dp"
    android:layout_marginEnd="5dp"
    android:layout_marginBottom="320dp"
    android:text="@string/audio_title"
    android:textAlignment="center"
    android:textSize="20sp" />

<TextView
    android:id="@+id/tvTime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
```

```
    android:layout_alignParentBottom="true"
    android:layout_marginStart="15dp"
    android:layout_marginBottom="180dp"
    android:text="@string/time" />

<TextView
    android:id="@+id/tvSongDuration"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="15dp"
    android:layout_marginBottom="180dp"
    android:text="@string/audio_duration" />

<ImageButton
    android:id="@+id/ibPrevious"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="40dp"
    android:layout_marginBottom="70dp"
    android:background="@color/transparent"
    android:onClick="onClick"
    android:src="@drawable/previous_40" />

<ImageButton
    android:id="@+id/ibForward"
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="110dp"
    android:layout_marginBottom="70dp"
    android:background="@color/transparent"
    android:onClick="onClick"
    android:src="@drawable/forward_40" />

  <ImageButton
    android:id="@+id/ibPlay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="55dp"
    android:background="@color/transparent"
    android:onClick="onClick"
    android:src="@drawable/play_70" />

  <ImageButton
    android:id="@+id/ibRewind"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="110dp"
    android:layout_marginBottom="70dp"
    android:background="@color/transparent"
    android:onClick="onClick"
```

```
    android:src="@drawable/rewind_40" />

  <ImageButton
    android:id="@+id/ibNext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/tvSongTitle"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="40dp"
    android:layout_marginBottom="70dp"
    android:background="@color/transparent"
    android:onClick="onClick"
    android:src="@drawable/next_40" />

  <ImageButton
    android:id="@+id/ibShuffle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/transparent"
    android:src="@drawable/shuffle_35"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="250dp"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="30dp"
    android:onClick="onClick" />

  <ImageButton
    android:id="@+id/ibRepeat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:background="@color/transparent"
        android:src="@drawable/repeat_35"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="250dp"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="90dp"
        android:onClick="onClick" />

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/ibAudio"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="250dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="30dp"
        android:background="@color/transparent"
        android:src="@drawable/mic_35"
        android:onClick="getSpecchInput" />

</RelativeLayout>
```

fragment_video.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" />
    </ScrollView>

    <ListView
        android:id="@+id/videoList"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="5dp"
        android:layout_marginEnd="5dp" />
</RelativeLayout>
```

video_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imgIcon"
        android:layout_width="44dp"
        android:layout_height="44dp"
        android:layout_alignTop="@+id/tvVideo_title"
        android:layout_alignParentStart="true"
        android:layout_marginStart="15dp"
        android:layout_marginTop="5dp"
        android:src="@drawable/movie" />

    <TextView
        android:id="@+id/tvVideo_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_marginStart="70dp"
        android:layout_marginTop="5dp"
        android:text="@string/video_title"
        android:textStyle="normal" />

    <TextView
        android:id="@+id/tvVideo_artist"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="70dp"
        android:layout_marginTop="35dp"
        android:layout_marginBottom="5dp"
        android:text="@string/video_artist" />

    <TextView
        android:id="@+id/textVideo_duration"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginTop="35dp"
        android:layout_marginEnd="15dp"
        android:layout_marginBottom="5dp"
        android:text="@string/audio_duration" />
</RelativeLayout>
```

video_player.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/dimGrey">

    <VideoView
        android:id="@+id/videoView2"
        android:layout_width="match_parent"
        android:layout_height="650dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_centerInParent="true"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp" />

    <ImageButton
        android:id="@+id/ibFullScreen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="40dp"
        android:layout_marginBottom="45dp"
        android:background="@color/transparent"
        android:contentDescription="@string/description"
        android:onClick="onClick"
        android:src="@drawable/fullscreen" />
```

```
    <ImageButton
        android:id="@+id/ibExitFullScreen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="100dp"
        android:layout_marginBottom="45dp"
        android:background="@color/transparent"
        android:contentDescription="@string/description"
        android:onClick="onClick"
        android:src="@drawable/exit_fullscreen" />
</RelativeLayout>
```

fragment_info.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".InfoFragment"
    android:background="@drawable/b1"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ImageView
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:gravity="center_horizontal"
            android:src="@drawable/infinitylogo"
            android:layout_marginLeft="100dp"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="I N F I N I T Y      P L A Y E R"
            android:textSize="20dp"
            android:textStyle="bold|italic"
            android:layout_centerHorizontal="true"
            android:textColor="@color/black"
```

```
        android:layout_marginTop="140dp"
        android:layout_marginBottom="30dp"/>
    </RelativeLayout>

    <androidx.cardview.widget.CardView
        android:id="@+id/about"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardBackgroundColor="#00000000"
        android:layout_margin="20dp"
        app:cardCornerRadius="20dp"
        app:cardElevation="8dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:background="@color/paleSilver">

            <ImageView
                android:layout_width="60dp"
                android:layout_height="60dp"
                android:layout_marginTop="20dp"
                android:layout_marginLeft="50dp"
                android:src="@drawable/about"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:fontFamily="sans-serif-black"
                android:gravity="center_horizontal"
```

```
            android:padding="35dp"
            android:layout_marginLeft="165dp"
            android:text="@string/about"
            android:textColor="@color/black"
            android:textSize="20sp" />
    </RelativeLayout>
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
android:id="@+id/download"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:cardBackgroundColor="#00000000"
android:layout_margin="20dp"
app:cardCornerRadius="20dp"
app:cardElevation="8dp">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:background="@color/paleSilver">

    <ImageView
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:layout_marginTop="15dp"
        android:layout_marginLeft="40dp"
        android:src="@drawable/download"/>

    <TextView
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-black"
        android:gravity="center_horizontal"
        android:padding="35dp"
        android:layout_marginLeft="150dp"
        android:text="@string/download"
        android:textColor="@color/black"
        android:textSize="20sp" />
    </RelativeLayout>
</androidx.cardview.widget.CardView>

    <androidx.cardview.widget.CardView
        android:id="@+id/mode"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardBackgroundColor="#00000000"
        android:layout_margin="20dp"
        app:cardCornerRadius="20dp"
        app:cardElevation="8dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:background="@color/paleSilver">

            <ImageView
                android:id="@+id/daynight"
                android:layout_width="85dp"
                android:layout_height="85dp"
```

```
            android:layout_marginTop="9dp"
            android:layout_marginLeft="40dp"
            android:src="@drawable/day"/>

        <androidx.appcompat.widget.SwitchCompat
            android:textStyle="bold"
            android:id="@+id/switchCompat"
            android:text="@string/text"
            android:fontFamily="sans-serif-black"
            android:textSize="19sp"
            android:checked="false"
            android:padding="30dp"
            android:layout_marginTop="5dp"
            android:layout_marginLeft="125dp"
            android:textColor="@color/black"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
      </RelativeLayout>
   </androidx.cardview.widget.CardView>
</LinearLayout>
```

about_page.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".AboutPage">

    <ImageView
        android:id="@+id/information"
        android:layout_width="match_parent"
        android:layout_height="460dp"
        android:layout_marginTop="180dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_centerHorizontal="true"
        android:background="@drawable/rectangle"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="240dp"
        android:layout_marginLeft="0dp"
        android:layout_marginRight="20dp"
        android:layout_centerHorizontal="true"
        android:text="@string/information"
        android:textSize="24dp"
        android:textStyle="bold|italic"
        android:textColor="@color/black"/>
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="280dp"
        android:layout_marginLeft="80dp"
        android:layout_centerHorizontal="true"
        android:text="@string/information1"
        android:textSize="20dp"
        android:justificationMode="inter_word"
        android:fontFamily="sans-serif"
        android:textColor="@color/black"/>
    <ImageView
        android:id="@+id/logo"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:src="@drawable/infinitylogo"
        android:layout_marginTop="100dp"
        android:layout_centerHorizontal="true"
        android:background="@drawable/circle" />

</RelativeLayout>
```

searchable.xml

```
<?xml version="1.0" encoding="utf-8"?>
<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:label="@string/app_name"
    android:hint="@string/search">
</searchable>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:dist="http://schemas.android.com/apk/distribution"
    package="com.example.testing1">

    <dist:module dist:instant="true" />
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/logo"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/logo_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".AboutPage"></activity>

        <activity android:name=".MainActivity"></activity>

        <activity android:name=".SearchableActivity"
            android:launchMode="singleTop">
            <intent-filter>
                <action android:name="android.intent.action.SEARCH" />
            </intent-filter>
            <meta-data android:name="android.app.searchable"
                android:resource="@xml/searchable">
```

```
        </meta-data>
    </activity>

    <activity
        android:name=".SplashActivity"
        android:theme="@style/Theme.AppCompat.NoActionBar">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <activity
        android:name=".AudioPlayer"
        android:theme="@style/Theme.AppCompat.NoActionBar">
    </activity>

    <activity
        android:name=".VideoPlayer"
        android:configChanges="orientation|keyboardHidden|screenSize"
        android:theme="@style/Theme.AppCompat.NoActionBar" />

  </application>
</manifest>
```